A Genetic Approach to Sparse Deconvolution

Armando Malanda-Trigueros, Ignacio Santamaría-Caballero*, Aníbal R. Figueiras-Vidal**
Dpto. Automática, ETSI Telecomunicación
Universidad Pública de Navarra
Campus de Arrosadía s.n., 31006 Pamplona. Spain
Tel.: +3448 16 9260
*Dpto. Electrónica, ETSI Telecomunicación
Universidad de Cantabria
Av. Los Castros s.n., 39005 Santander. Spain
**GPSS/DSSR ETSI Telecomunicación
Universidad Politécnica de Madrid
Ciudad Universitaria s.n., 28040 Madrid. Spain
Tel.: 341-336-7226, Fax.: 341-336-7350
seruaga@gtts.ssr.upm.es

Abstract [1]

Sparse deconvolution is a detection plus estimation problem: we split it accordingly and make use of the discrete character of detection to introduce Genetic Algorithms in the search of the best peaks location. We then estimate the amplitudes of these peaks via pseudoinversion and also with a gradient-type algorithm that rapidly converges to the Moore-Penrose pseudoinverse. Both ways are explored in this paper and contrasted with a more traditional sparse deconvolution approach. A discussion of final results, suggestions for improvements of the proposed methods and some open lines close this paper.

1 Introduction

Sparse deconvolution consists on the estimation of a sparse sequence s from a noisy measurements z given by

$$\mathbf{z} = \mathbf{H}\mathbf{s} + \mathbf{n} \quad (1)$$

where **H** is a wavelet matrix and **n** is the measurement noise. This problem is very frequent in signal processing applications: seismic exploration, speech modeling, synthetic aperture radar, ultrasonic exploration, etc.

A number of algorithms have been proposed to solve it: some of them are based on iterative approaches to obtain a minimum square error solution or Wiener filtering, forcing sparseness by applying an adaptive threshold [1]; however, these procedures sometimes miss small peaks in the first steps of the algorithm. There is also the possibility of using a soft threshold obtained from the statistical characteristics of the problem: this is the method proposed by Godfrey and Rocca to perform blind deconvolution [2]. Other approaches use linear programming techniques to find a minimum L_1-norm solution [3]; L_1-norm minimization is appropriate to obtain spiky solutions, but these approaches have a high computational cost. An alternative approach consists on adding a penalty term to force spiky solutions: in [4] the penalty term used is the L_1-norm of the signal and the solution is found using linear programming techniques in a way similar to [3]. More elaborated penalty terms are discussed in [5] and the solution is found using gradient-type algorithms, thus reducing the computational cost. Finally, other methods obtain a theoretical solution by using a composite model and estimation theory [6, 7]; nevertheless, they require that the peaks of s have a Gaussian distribution: a hypothesis that is not acceptable in many situations.

Sparse deconvolution is a detection plus estimation problem: first, we have to decide the elements $s(k)$ that are zero; then, to estimate the value of the nonzero components. Considering the form of the problem, it is possible to use Genetic Algorithms (GA's) to solve it: this is the possibility that we propose in this paper. This approach has the potential advantage of exploiting the parallel nature of GA's.In all this study we will assume that the number of spikes is known.

The paper is organized as follows: in Section 2 we give a general outline of GA techniques, describe our so-

[1] This work has been partly supported by CICYT grants #TIC92-6800-C05-01 and #TIC93-0277-E

ution space representation and the genetic operators used, and finally propose a fitness function adequate to the problem that is implemented in two different ways. In Section 3 we present simulation figures and results from the application of the methods described just before. In Section 4 we make a brief discussion about the use of a second fitness function and about a small modification for speeding up the algorithms. Conclusions and further research lines are commented in Section 5.

2 Genetic Algorithms

Genetic Algorithms [8] are global optimization procedures guided by similar mechanisms to those present in natural environments by which populations of different species survive and evolve. In GA's a population of individuals (tested points in a search space) is created at the initial generation in a random way. The fitnesses of these individuals are evaluated and the next generation is formed by randomly selecting individuals from the initial one, following a probability scheme proportional to the corresponding fitness of each individual: highly fitted individuals will be reproduced more probably than poorly fitted ones. The next step is to define operators for the population to go on evolving so that features present in two or more individuals may be combined in new individuals in the same fashion as a child preserve characteristics from both his parents. These two policies acting together (survival of the fittest and recombination or crossover) are the main responsible agents of the enhancing progress of the algorithm, in very much the same way as the Darwinian guidelines of natural evolution of species. For those features which have not happened to appear in the initial generation, still there is a chance to come up, with the operation generally known as mutation. This mechanism systematically introduces small changes in randomly chosen individuals of the population increasing in this way the existing set of characteristics present in the whole population; as a consequence this operator prevents the population from stagnating in suboptimal solutions before reaching the global optimum. The new population is thus far evaluated and passed to reproduction, selection, crossover and mutation, generation after generation until some acceptable solution is found. The term 'genetic' comes from the coding of the solution space: the individuals are presented as binary strings (chromosomes) for which mutation and crossover are simple bit operations. A real coding of the solution space is also possible although in that case we would no longer be talking about GA but rather about Evolution Strategies. The GAs as described here appear as powerful optimizations tools, capable of finding global optima in difficult domains (non-continuous, non-derivative, with multiple local optima, noisy, etc.). The paid price is a high computational cost but that can always be dramatically reduced using parallel implementations.

In all our GA approaches we have a population of signals **x**, tentative estimations of the original signal s. Each of these signals has an associated chromosome which represents the positions of its peaks. The chromosome length is thus the length of this signal, having ones in the non-zero positions and zeroes in the rest. We then compose the matrix **S** with all zeroes except along the main diagonal in which we place the chromosome associated to the tested signal **x**:

$$S = \left\{ \begin{array}{ll} S(i,j) = 0, & \text{i} \neq \text{j}; \\ S(i,i) = 1, & \text{if a spike is placed in i}; \\ S(i,i) = 0, & \text{in other cases.} \end{array} \right\} \quad (2)$$

We incorporate the sparse character to **x** by introducing the matrix S, i.e. **x**=S**x**. Our crossover and mutation operations are problem specific. With our crossover operator two offsprings are born from two parents preserving the number of spikes (which is considered to be known in our problem). We place in parallel the chromosomes of both parents, randomly choose a crossing site and split them by this site. Now we fix the left part of the first parent with the right part of the second and get the first offspring and fix the remaining parts of both parents and get the second offspring. We then remove or insert in random positions as many spikes as needed of the resulted offsprings so that the number of these spikes is preserved. The mutation operator simply chooses one spike at random and changes its position also in a random way.

Two different approaches are tested in this work.

2.1 Pseud_GA

Lacking any additional information, we take as the objective function the L2 norm of the error signal

$$\phi_1(x) = \|e\|_2 \quad (3)$$

being **e**=**z**-**Hx**. Given an individual **k** of the population and the matrix S_k associated to its chromosome we find the vector of amplitudes **x** applying the Moore-Penrose pseudoinverse

$$x_k = (HS_k)^{\#}\, z \quad (4)$$

2.2 Iter_GA

Our second approach also considers the L2 norm of e as the fitness function, but this time the way to calculate **x** is different as only a few steps of a gradient-type algorithm which converges to the pseudoinverse are needed:

$$x_{i,j+1} = x_{i,j} + \mu (HS_i)^T (z - HS_i x_{i,j}) \qquad (5)$$

where $x_{k,j}$ is the vector signal **x** associated to the individual **k** at iteration **j**. From each individual chromosoine, S_k is composed and processed a few times finding the vector of amplitudes. The great advantage is that the amplitudes of the parents can be taken as the starting amplitudes of the offsprings to begin the iterations of (5). Iter_GA needs only a short number of iterations because convergence of the amplitudes is achieved throughout the progress of generation and not in every evaluation, lowing down very significantly computational requirements: it is therefore a convenient way of incorporating iterative methods to GAs within the new genetic framework called hybrid methods. Again $\phi_1(x_{i,j_{max}})$ is computed as the fitness function once the iterations of a single evaluation are completed.

3 Simulations results

In this section we illustrate the performance of the proposed method by means of computer simulations. Figure 1 shows a 32 point-long sparse signal **s**, having nonzero values at points $s_3 = 8$, $s_6 = 6.8$, $s_{14} = -5$,$s_{21} = 4.5$,$s_{31} = -3.5$. The impulse response used in this example corresponds to the first 20 points of an ARMA filter having one zero at $z = 0.6$ and two poles at $z = 0.8\exp(\pm j5\pi/12)$; this wavelet is shown in Figure 2. The *SNR* used in this example is 4 dB, and is defined as the power of **Hx** with respect to the power of **n**, **n** being a zero-mean Gaussian white noise. Figure 3 shows an observation z, and Figure 4 and Figure 5, typical convergence curves of the error power associated to the best and average individuals along generations for both GA methods. We run 100 independent times those GA's described in 2.1 and 2.2.

We use a population of 50 individuals and let the algorithms run during 100 generations. The length of the chromosome is the same as that of the signal, i.e., 32. The probability of crossover and mutation are 0.9 and 0.2, respectively, and the number of times (5) is iterated in the evaluation of each individual is 5. We also repeat another 100 times the Adaptive Threshold (AT) algorithm described in [1]. In each simulation we allow 200 iterations to achieve convergence, the stepsize of the adaptation being 0.2 . We impose the constraint that the number of peaks be 5, by excluding the smallest ones if there are more than this number at the final iteration of x. Table 1 and Table 2 show the final results of all these experiments. Pse_GA and Iter_GA outperform the AT method in missdetected peaks (39 and 32 against 65) as well as in complete signal detection, i.e. experiments in which the whole 5 peaks are detected (67 and 71 against 49). Also the GA methods get better mean values for the peak amplitudes. Although we have run the GA's for 100 generations the final results are reached much earlier: Pseud_GA needs 32 generations on the average, and Iter_GA, 46.

4 Discussion

Most of the GA errors come from solutions which yield an L_2-norm of the error signal lower than that obtained applying the pseudoinverse to the solutions with the right positions of the peaks: this only means that without any further information of the problem and with this fitness function, we reach the theoretic limit of performance; it is not the same for the Adaptive Threshold Algorithm as its performance is significantly smaller.

No better results could be obtained by including information about the noise power: a fitness function of the form

$$\phi_2(x) = |||e||_2 - N_2|^p \qquad (6)$$

increases errors and amplitude misadjustment and dispersion with respect to the previous one. In the whole search space there are many solutions of x with an error variance as large as the noise power: our GA will randomly get one of these, but the probability that this one is a 5-peak detection is small. Moreover, provided we have successfully completed the detection, there is a 5-dimensional solution space of amplitudes, a whole subset of which are roots of (6). Only one of these roots is s and if we make the GA minimize (6) it will most probably find a solution of amplitudes different from s. As we can infer from experimental results, the estimated amplitudes are on the average more distant from the sought ones than that obtained via pseudoinversion.

In a different respect, we inspected several ad-hoc refinements of our GA's in the search of solutions with slighter computational needs : one of the possible ways to accelerate the convergence of the algorithms is to take advantage from the fact that all three algorithms examined here are many times trapped (at least for a good while) in local minima, corresponding to situations in which four spikes are placed correctly and erroneously the last one. We get an important computational reduction by exhaustively changing the smallest peak of the best individual so as to find the best position for it, in the case of no improvement of performance of this best one during the few last generations.

5 Conclusions and further lines

We have presented two GA approaches to deal with sparse deconvolution using the L_2-norm of the error signal as the fitness function. The first one (Pseud_GA) makes use of the Moore-Penrose pseudoinverse to find the amplitudes of the peaks, while the second one (Iter_GA) includes a gradient-type algorithm to find those peaks. Both achieve better performance in peak detection and amplitude estimation when compared with other classical methods such as the Adaptive Threshold of [1]. The main drawback of our GA approach is computational load, which may be alleviated by means of parallel implementations. In any case, Iter_GA is much faster than Pseud_GA as it avoids pseudoinversion and only needs a few steps of the gradient-type algorithm every evaluation. Different modifications can be introduced trying to decrease computation (adaptive number of iterations per evaluation, memory tables to access to the last most repeated individuals in order of avoiding to repeat identical operations for different chromosomes, Evolution Strategies [9] to obtain the amplitudes of the sparse signal, etc.). When the register length increases, the computational cost of the proposed detectors becomes prohibitive. If we have an a priori knowledge of the spike amplitudes, we can obtain from the observation vector an estimate of the amplitudes following a convenient criterion (Minimum Variance Deconvolution, for example [6]); then, we can use these amplitudes at the positions of each inspected individual, without having to recalculate them, taking again $\phi_1(x)$ as the fitness function.

These and similar lines are promising in order to make more practical the kind of approach we are presenting here.

Detected Spikes	Pseud_GA	Iter_GA	Adaptive Threshold Algorithm
spike1	100	100	99
spike2	99	99	98
spike3	97	96	89
spike4	86	93	79
spike5	79	80	70
5-peaks detection	67	71	49

Table 1: Number of detected spikes

Amp(var)	Pseud_GA	Iter_GA	Adaptive Threshold Algorithm
spike1	8.06(1.18)	7.95(1.20)	6.83(1.43)
spike2	6.89(1.47)	6.795(1.25)	5.77(0.93)
spike3	-5.104(1.08)	-5.044(1.13)	-4.90(1.20)
spike4	4.68(1.25)	4.54(0.88)	4.33(1.11)
spike5	-3.70(0.66)	-3.83(0.94)	-3.80(0.66)

Table 2: Amplitudes and variances of the spikes

References

[1] A. R. Figueiras-Vidal, D. Docampo-Amoedo, J. R. Casar-Corredera and A. Artés-Rodríguez, Adaptive Iterative Algorithms for Spiky Deconvolution, *IEEE Trans. on Acoustics Speech and Signal Processing*, vol. 38, no. 8, pp.1462-1466, 1990.

[2] R. Godfrey and F. Rocca, Zero Memory Non-Linear Deconvolution, *GeophysicalProspecting*, vol. 29, pp. 189-228, 1981.

[3] H. Taylor, S. Banks and F. McCoy, Deconvolution with the L_1-norm c. *Geophysics*, vol. 44, no. 1, pp.39-52, 1979.

[4] M. S. O'Brien, A. N. Sinclair and S. Kramer, Recovery of a Spike Time Series by L_1-Norm Deconvolution, submitted to *IEEE Trans. on Signal Proc.*, 1992.

[5] J. Cid-Sueiro and A. R. Figueiras-Vidal, Pruning Techniques for Spiky Deconvolution, *Proc. of the 2nd. Cost 229 Workshop on Adaptive Algorithms in Communications*, Bordeaux (France), 1992.

[6] J. M. Mendel, *Maximum-Likelihood Deconvolution*. New-York,Springer-Verlag, 1990.

[7] J. J. Kormylo and J. M. Mendel, Maximum Likelihood Detection and Estimation of Bernoulli-Gaussian Processes. *IEEE Trans. on Inf. Theory*, vol. 28, no.3, pp.482-488, 1982.

[8] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. New York, Addison-Wesley, 1988.

[9] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, New York, Springer-Verlag, 1992.

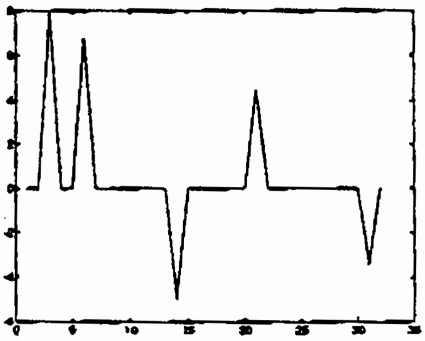

Figure 1: Signal s

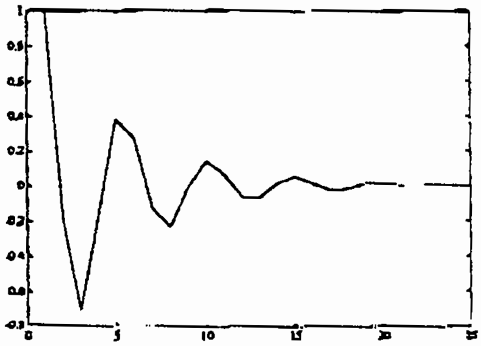

Figure 2: Wavelet h

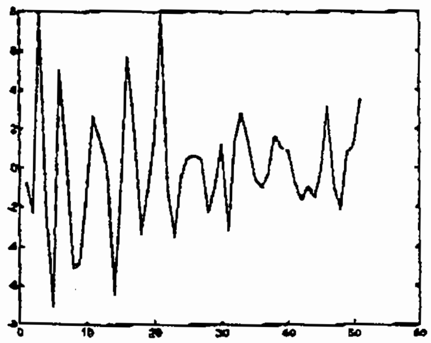

Figure 3: Observation signal z

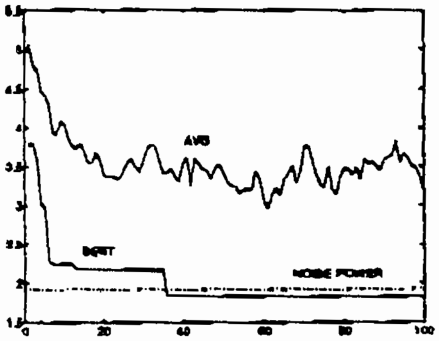

Figure 4: An example of the convergence of $||e||_2$ for the best and the average individuals along generations (Iter_GA)

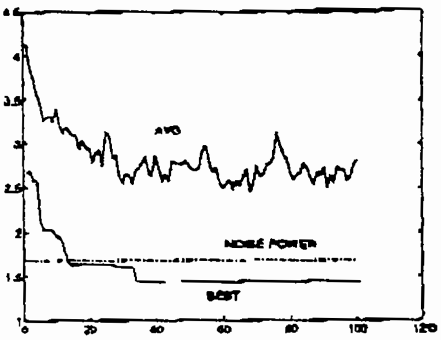

Figure 5: An example of the convergence of $||e||_2$ for the best and average individuals along generations (Pseud_GA)

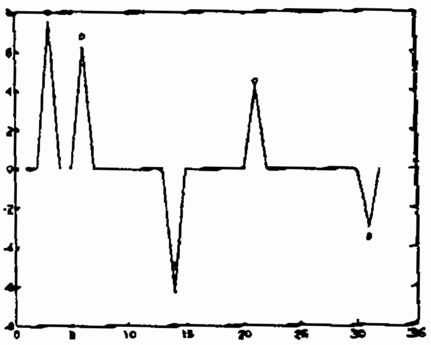

Figure 6: An example of the final estimated signal x after 100 iterations of (Iter_GA)