Sparse Deconvolution Using Gaussian Mixtures

Ignacio Santamaría-Caballero† Aníbal R. Figueiras-Vidal‡

†Dpto. Electrónica, ETSI Telecom-Univ. Cantabria Av. Los Castros s.n., 39005 Santander, SPAIN.
‡GPSS/DSSR, ETSI Telecom - UPM Ciudad Universitaria s.n., 28040 Madrid,SPAIN

Abstract - In this paper we present a new algorithm to recover a sparse signal from a noisy register. The algorithm assumes a new model for the sparse signal that consists on a mixture of a narrow and a broad Gaussian both with zero mean. A penalty term which favors solutions driven from this model is added to the usual error cost function and the resultant global cost function is minimized with a gradient-type algorithm. In the paper we propose methods for updating the mixture parameters as well as for choosing the weighting parameter for the penalty term. Simulation experiments show that the accuracy of the proposed method is competitive with classical statistical detectors with a lower computational load. The proposed algorithm shows also a good performance when applied to a practical seismic deconvolution problem.

I. INTRODUCTION

The problem of removing the effects of noise and impulse response on a sparse signal from a data register has a wide variety of applications in digital signal processing: geophysical exploration modeling (seismic deconvolution), synthetic aperture radar design, ultrasonic analysis, speech coding (multipulse techniques),etc. The sparse deconvolution problem is usually referred as follows: given some observation sequence $\mathbf{z} = \{z_1, \cdots, z_M\}$, find the sparse (spiky) signal $\mathbf{x} = \{x_1, \cdots, x_N\}$ such as

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{n} \tag{1}$$

where $\mathbf{H}$ is an impulse response matrix, and $\mathbf{n}$ models the noise. The signal $\mathbf{x}$ is known to be sparse, i.e., only a few of its samples have nonzero values.

In L_2-norm deconvolution, solution $\mathbf{x}$ that minimizes the squared error $E^2 = \|\mathbf{z} - \mathbf{H}\mathbf{x}\|_2$ is found, but it is not appropriate, since the ill-conditioned character of the problem avoids the obtention of the sparse solution we are looking for. L_1-norm deconvolution algorithms, on the other hand, obtain $\mathbf{x}$ by minimizing the error in the L_1-norm [1] or by minimizing a weighted objective function of the error and the signal (both in the L_1-norm) [2], by means of linear programming. These approaches are well suited for data driven from a spiky distribution but the computational cost associated with linear programming techniques is high.

On the other hand, theoretical solutions to the corresponding detection plus estimation problem established in (1) are cumbersome; those available, such as [3], require a model for the signal $\mathbf{x}$ which is not acceptable in many situations. Moreover, sometimes signal statistics are not available, so a complete analytical approach is not possible.

Alternative methods have been proposed based on iterative approaches to obtain a minimum square error solution or Wiener filtering, forcing sparseness by applying an adaptive threshold [4]; these techniques are simple and efficient, but they are very sensitive to the selection of the parameters involved in the threshold procedure, and sometimes miss small peaks in the first steps of the detection process.

Finally, a simple method to obtain spiky solutions consists on adding an extra term to the error function that will penalize non sparse solutions

$$cost = square\ error + \alpha\ penalty\ term \tag{2}$$

By minimizing this cost function, the solution seeks a tradeoff between the square error and the penalty term. The relative importance of these two factors is controlled by the weighting parameter α, that must be selected to optimize the performance.

The same approach has been recently proposed to simplify neural networks architectures (pruning), so they generalize better. The application of pruning terms and algorithms to solve sparse deconvolution problems is straightforward: we only need to replace the weights of the neural network by the estimates of the signal x_i. Some pruning techniques have appeared in the literature [5] differing in the penalty term used to decide which are the negligible connections. Among them, probably the simplest consists on taking the sum of the squares of the samples $\sum_i x_i^2$. This penalty term can be viewed as the simplest way to regularize an ill-conditioned problem. Nevertheless, it is not adequate in order to obtain sparse solutions; for this reason, in the present paper we propose to use a more complex term, mainly, we will focus on a term, recently proposed by Hinton and Nowlan [6]. This penalty was originally proposed as a measure of a neural network complexity. In its simplest version this term is a mixture of a narrow and a broad Gaussian, both centered at zero. This penalty term drives small samples toward zero without forcing large peaks away from their true values, so it is well suited for our sparse deconvolution problem.

II. THE PROPOSED ALGORITHM

A. *Presenting the algorithm*

Basically, the proposed algorithm assumes that the prior distribution of our sparse signal can be approximated with a mixture of a narrow (subscript n) and a broad (subscript b) Gaussian both with zero mean. The narrow Gaussian models the smaller peaks (ideally nonexistents), whereas the broad one models the true peaks:

$$p(x) = \frac{\pi_n}{\sqrt{2\pi}\sigma_n} e^{-\frac{x^2}{2\sigma_n^2}} + \frac{\pi_b}{\sqrt{2\pi}\sigma_b} e^{-\frac{x^2}{2\sigma_b^2}} \quad (3)$$

where π_n and π_b are the mixing proportions of the two Gaussians and are therefore constrained to sum 1. Considering that the samples of the sparse signal were driven from such a mixture, the probability that a particular sample x_i, was generated by a particular Gaussian j (posterior probability) is called by Hinton and Nowlan *responsibility* of that Gaussian for the sample, and is given by

$$r_j(x_i) = \frac{\pi_j p_j(x_i)}{\sum_k \pi_k p_k(x_i)} \quad (4)$$

where $p_j(x_i)$ is the probability density of x_i under Gaussian j. For a given signal $\mathbf{x}$, the narrow Gaussian gets most of the responsibility for small samples. Consequently, we can define the following global cost objective

$$\Phi(\mathbf{x}) = \|\mathbf{z} - \mathbf{H}\mathbf{x}\|_2 - \alpha \sum_{i=1}^{N} \log \sum_j \pi_j p_j(x_i) \quad (5)$$

where α controls the tradeoff between the squared error and the penalty term, and $p_j(x_i)$ is the probability density function of each Gaussian. The minimization of (5) can be accomplished by means of a gradient-type algorithm

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mu \mathbf{H}^T(\mathbf{z} - \mathbf{H}\mathbf{x}_k) - \alpha\mu\psi(\mathbf{x}_k) \quad (6)$$

where the superscript T denotes transpose, and $\psi(\mathbf{x}_k)$ is a Nx1 column vector defined by

$$\psi(\mathbf{x}_k) = col(\psi(x_{1,k}), \cdots, \psi(x_{N,k})) \quad (7)$$

where

$$\psi(x_{i,k}) = \sum_j r_j(x_{i,k}) \frac{x_{i,k}}{\sigma_j^2} \quad (8)$$

In (7) and (8), $x_{i,k}$ is sample i in iteration k.

Since we do not know the variances and mixing proportions in advance, in the next subsection we present a method to update these parameters in each iteration. In this way, it is possible to use the proposed algorithm without any statistical information about the collected data and noise. Subsequently, we will discuss alternatives to choose the weighting parameter α.

B. *A method for updating the Gaussian mixture parameters*

The most obvious procedure consists on fixing the mixture parameters according to some a priori knowledge of the problem; for instance, in a seismic deconvolution problem, we usually know in advance that the solution can be modeled by a Bernoulli-Gaussian distribution, for which the signal follows a Gaussian distribution with variance σ_x^2 with probability λ and its value is zero with probability 1-λ. If estimates of these parameters are available, we could choose the mixture parameters in the following way: $\pi_n = 1 - \lambda$, $\pi_b = \lambda$, $\sigma_b^2 = \sigma_x^2$, and a small value for σ_n^2. This procedure achieves good results if we dispose of an appropriate statistical knowledge of the problem.

A more elaborate approach is proposed in [6], where all the parameters $(\mathbf{x}, \pi_j, \sigma_j)$ are updated simultaneously using a conjugate descent procedure. In this way, the rate of change for both the mixture parameters and the signal is the same.

In this paper we propose another alternative to force a slow change in σ_j and π_j which consists on updating these parameters using the following iterations

$$\sigma_{j,k+1}^2 = \gamma\sigma_{j,k}^2 + (1-\gamma)\frac{\sum_i x_{i,k}^2 r_j(x_{i,k})}{\sum_i r_j(x_{i,k})} \quad (9)$$

$$\pi_{j,k+1} = \gamma\pi_{j,k} + (1-\gamma)\frac{1}{N}\sum_i r_j(x_{i,k}) \quad (10)$$

γ being a constant near to 1. The second terms in (9) and (10) are the values that minimize the cost function (5) after recomputing the responsibility factors $r_j(x_{i,k})$ in each iteration. This last approach is simpler than a gradient descent procedure and it also achieves good results, so we choose to use this method in our simulations.

C. *A Method for Choosing the Weighting Parameter α*

A complete application of the proposed algorithm requires a method for choosing the weighting parameter α, which establishes a tradeoff between the quadratic error and the penalty term. In general, optimum α depends on the noise variance σ^2, as well as on matrix $\mathbf{H}$ and on solution $\mathbf{x}$ itself; among them, usually the most important is σ^2.

Without an estimate of σ^2, the only procedure is to fix α heuristically. This can be done in the practice because the obtained solutions are not usually critical with respect to α. On the other hand, if an estimate of σ^2 is available, we can use this knowledge to adaptively obtain the optimum weighting parameter. In particular, the method that we propose starts selecting a maximum value for the weighting parameter, α_{max}, which produces a solution with few peaks and a variance of the associated residual higher than the estimated variance of the noise; then, the weighting parameter is reduced iteratively in fixed steps $\Delta\alpha$ until a value α_{opt} is reached for which the obtained solution (after iteration k) fulfills a constraint of the form

$$\|\mathbf{z} - \mathbf{H}\mathbf{x}_k\|_2 = \sigma^2 \pm \varepsilon \quad (11)$$

The use of ε is prompted by statistical considerations.

This approach achieves better results than considering a fixed weighting parameter, but obviously the computational cost is also higher, since we must obtain a solution for each evaluated α. This last overload can be reduced if, for each new α, we initialize the iteration (6) with the obtained solution for the previous α.

D. *The Overall Algorithm*

The proposed algorithm can be summarized in the following steps:

```
1 Initialize the mixture parameters $\pi_{n,o}, \sigma_{n,o}, \pi_{b,o}, \sigma_{b,o}$,
  the weighting parameter $\alpha = \alpha_{max}$, and $\mathbf{x}_0 = \mathbf{O_{Nx1}}$.
2 for k=0 to niter1
  2.1 $\mathbf{x}_{k+1} = \mathbf{x}_k + \mu\mathbf{H}^T(\mathbf{z} - \mathbf{H}\mathbf{x}_k)$
3 for k=niter+1 to niter2
  3.1 $\mathbf{x}_{k+1} = \mathbf{x}_k + \mu\mathbf{H}^T(\mathbf{z} - \mathbf{H}\mathbf{x}_k) - \alpha\mu\psi(\mathbf{x}_k)$
  3.2 recompute $r_j(x_{i,k})$ for i=1,...,N.
  3.3 update $\sigma^2_{j,k+1}$ and $\pi_{j,k+1}$ according to (9) and (10)
4 if $\sigma^2 - \varepsilon \leq \|\mathbf{z} - \mathbf{H}\mathbf{x}_k\|_2 \leq \sigma^2 + \varepsilon$ then end
  else
  4.1 $\alpha = \alpha - \Delta\alpha$
  4.2 go to 1
```

Let us expose some comments about the presented algorithm. First, we have introduced a minor modification consisting on using iteration (6) without penalty term, i.e., $\alpha = 0$, for a small number of steps niter1.This is because to apply (6) on a nonzero signal seems to improve the results in all the simulations; in addition, the final solution is very robust with respect to this parameter.

Second, we need to consider the issue of initializing the mixture parameters. A reasonable mixture initialization could be $\pi_n = \pi_b = 0.5$ and $\sigma_b^2 > \sigma_n^2$ with σ_n^2 being a small fraction of the observations variance σ_z^2. As long as the algorithm proceeds, the broad Gaussian becomes even broader, i.e., σ_b^2 increases, and σ_n^2 becomes smaller. On the other hand, π_b and π_n drive toward the mean number of samples modeled by each Gaussian.

When σ_n^2 approaches 0 too closely, the algorithm may become unstable. In [6] this problem is solved working with a set of auxiliary variables of the form: $\sigma_j^2 = e^{\gamma_j}$, where the value of γ_j is unrestricted. In this way, σ_n^2 is not allowed to approach 0. Nevertheless, our sparse deconvolution problem has some particularities that make it different from a neural network pruning problem: here, we are interested in decreasing σ_n^2 as much as possible, because in this way the useless samples approach zero. For this reason, we have chosen to work directly with σ_n^2 (instead of γ_j)in the following way: iteration (6) is carried out until a maximum number of iterations niter2 is reached or a constraint of the form $\sigma_n^2 < \delta$ is satisfied, where δ is an empirical constant close to zero which prevent us from arriving to unstability. It may seem that the proposed algorithm involves a wide number of parameters, but indeed none of them is critical or hard to adjust for a particular problem.

III. SIMULATION EXPERIMENTS

We have selected two computer experiments with different sparse signals: the first uses randomly generated sparse signals according to a preestablished model: specifically, we generate sparse signals with Gaussian or uniform amplitude distributions. The second experiment consists on an application to real seismic data. For the two examples we have used a mixture of a narrow and a broad Gaussian, both centered at zero.

A. *Experiment 1*

In this example we evaluate the performance of our algorithm using synthetic signals according to the following model: x(k)=r(k)q(k), where q(k) is a Bernoulli process for which q(k)=1 with probability λ and q(k)=0 with probability 1-λ; r(k) is a white random process with zero mean, variance σ_r^2 and whose amplitudes fit a Gaussian or uniform distribution (in particular, the Gaussian distribution is often used fo seismic deconvolution cases). Registers of five hundred samples were generated according to the above models (with $\lambda = 0.05$ and $\sigma_r^2 = 10$), and then convolved with the first 20 points of an ARMA filter having a zero at z=0.6 and two poles at z=0.8exp(±j5π/12). Finally, a zero mean Gaussian noise was added to the result to produce a SNR=4 dB.

For this example the simulations compare the performance of the algorithms corresponding to:

A1) a mixture of two Gaussians with an optimum weighting parameter chosen to fulfill (11).

A2) a one-shot threshold detector.

A3) a Single Most Likely Replacement detector.

The last two algorithms are classical statistical detectors based on the Bernoulli-Gaussian model assumption. For a complete description of these two algorithms see [3].

For the first algorithm we initialize the mixture parameters with the following values: $\pi_{n,0}$=$\pi_{b,0}$=0.5, $\sigma^2_{b,0}$=$2\sigma^2_z$ and $\sigma^2_{n,0}$=$\sigma^2_z/2$, where σ^2_z is the observations variance. We use iteration (6) without penalty term until $\mathbf{x}_{10}$, and we ensure convergence selecting μ=0.1. The parameters of the mixture are updated according to (9) and (10).

Table 1(a) shows the averaged results of 25 simulations when there is a Gaussian amplitude distribution for the three detectors. The first column shows the average detection percentage, and the second the percentage of false peaks detected. Table 1(b) shows the same kind of results for a uniform amplitude distribution of the spiky signal.

Somehow surprisingly, the three algorithms give better results for a uniform amplitude distribution of the sparse signal, however, this can be easily explained since for a fixed variance, data driven from a Gaussian distribution are near zero (and, therefore, are more difficult to detect) with higher probability than if they were driven from a uniform distribution. The proposed method gives intermediate results between the one-shot detector and the more elaborated SMLR detector, but with a smaller computational load: between 25 and 75 iterations of (6) were enough for all the simulations performed, while the statistical detectors require at least the inversion of an N by N matrix, where N is the register length.

B. *Experiment 2*

In the last example we apply the proposed algorithm to a section of real seismic data. The source wavelet used in this example is shown in Fig.1. The initial mixture parameters

a)	**A1**	**A2**	**A3**
correct detections (%)	64.7	56.6	73.1
false detections (%)	0.8	0.7	1.1

b)	**A1**	**A2**	**A3**
correct detections (%)	70.3	64.0	75.32
false detections (%)	0.8	0.4	1.1

Table 1. Averaged results of 25 simulations for the three detectors. The first column shows the average detection percentage, and the second the percentage of false peaks detected. (a) Gaussian amplitude distribution and (b) Uniform amplitude distribution.

are π_b=π_n = 0.5, $\sigma_b = 5$ and $\sigma_n = 0.2$, and we use a fixed $\alpha = 0.3$ and $\mu = 0.01$ for all the traces. Fig. 2 shows the real data section, which consists on 25 traces, and Fig. 3 the corresponding estimated reflectivity sequences using the proposed algorithm. These reflectivity sequences, when convolved with the wavelet shown in Fig. 2, fit quite well the original seismic data, thus indicating a reasonable behavior of the algorithm.

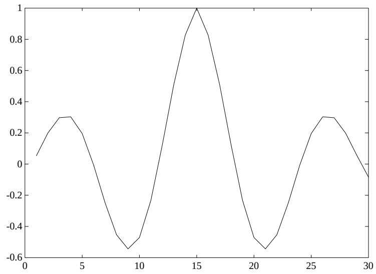

Fig. 1. Wavelet used in example 2

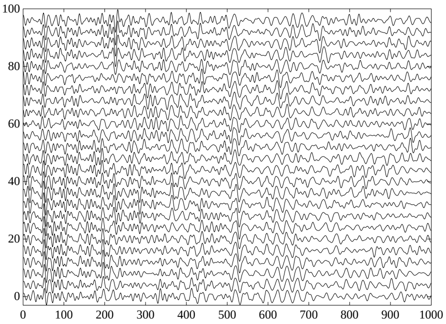

Fig. 2. A section of 25 traces of real seismic data

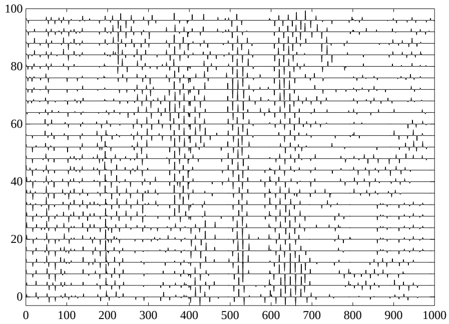

Fig. 3. The corresponding estimated reflectivity sequences using the proposed algorithm

IV. CONCLUSIONS

A penalty term that favors solutions driven from a mixture of a narrow and a broad Gaussian has been proposed to recover a sparse signal from a noisy register. The narrow gaussian drives small samples toward zero, while the broad one models the true spikes. Also, we present a method to adaptively obtain the optimum weighting parameter between the quadratic error and the penalty term. Simulations show a good performance of the proposed method when applied to a wide variety of examples, as well as for a real seismic deconvolution case; besides, the speed of the proposed method is much faster than that of existing statistical detectors.

ACKNOWLEDGMENT

The authors would like to thank Marc Lavielle from Laboratoire de Statistiques, (France) for providing us with the seismic data.

REFERENCES

[1] J. Claerbout and F. Muir, "Robust modeling with erratic data",*Geophysics*, vol. 38, no. 3, pp. 826-844, 1973.

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

[3] J. M. Mendel,*Maximum-Likelihood deconvolution.* New-York: Springer-Verlag, 1990.

[4] 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 Proc.*, vol. 38, no. 8, pp. 1462-1466, August 1990.

[5] S. J. Hanson and L. Y. Pratt, "Comparing biases for minimal network Construction with BackPropagation", in D. S. Touretzky, Ed., *Advances in neural information processing systems 1*; San Mateo,CA:Morgan Kaufman Publishers, 1990.

[6] S. J. Nowlan and G. E. Hinton, "Simplifying neural networks by soft weight-sharing", *Neural Computation*, vol. 4, pp. 473-493, 1992.