ELSEVIER

Signal Processing 54 (1996) 161–172

SIGNAL PROCESSING

Sparse deconvolution using adaptive mixed-Gaussian models

Ignacio Santamaría-Caballero[a,*], Carlos J. Pantaleón-Prieto[a], Antonio Artés-Rodríguez[b]

[a] *DICOM, ETSII Telecomunicación-Univ. Cantabria, Av. Los Castros s/n, 39005 Santander, Spain*
[b] *DSSR, ETSI Telecomunicación-UPM, Ciudad Universitaria s/n, 28040 Madrid, Spain*

Received 20 October 1995; revised 6 May 1996

Abstract

In this paper we present a new algorithm to recover a sparse signal from a noisy register. The algorithm assumes a new prior distribution for the sparse signal that consists of 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 by means of a gradient-type algorithm. A condition is derived for the step-size parameter in order to ensure convergence. In the paper we also propose a method (based on the Expectation-Maximization algorithm) to update the mixture parameters. The estimation of the sparse signal and the optimization of the Gaussian mixture are combined in the proposed algorithm: in each iteration a new signal estimate and a new model (which approximates the distribution of the new estimate) are obtained. In this way, the proposed method can be used without any statistical knowledge about the signal. Simulation experiments show that the accuracy of the proposed method is competitive with classical statistical detectors with a lower computational load.

Zusammenfassung

In diesem Beitrag präsentieren wir einen neuen Algorithmus, um ein dünnverteiltes Signal aus dem Rauschen zurückzugewinnen. Der Algorithmus nimmt eine neue ursprüngliche Verteilung für das dünnverteilte Signal an, welches aus einer Mischung aus einem schmalen und einem breiten Gaußverteilten Signal besteht, beide ohne Gleichanteil. Ein Strafterm, welcher Lösungen begünstigt, die von diesem Modell abgeleitet werden, wird zu der üblichen Fehler-Kostenfunktion hinzuaddiert und die resultierende globale Kostenfunktion wird mittels eines Gradienten-Algorithmus minimiert. Eine Bedingung für den Schrittgrößen-Parameter wird hergeleitet, um Konvergenz sicherzustellen. Im Beitrag schlagen wir auch eine Methode vor (gegründet auf dem Erwartungs-Maximierungs-Algorithmus), um die Mischungsparameter zu aktualisieren. Die Schätzung des dünnverteilten Signals und die Optimierung der Gaußschen Mischung sind in dem vorgeschlagenen Algorithmus verbunden: in jeder Iteration wird ein neuer Signalschätzwert und ein neues Modell (welches die Verteilung des neuen Schätzwertes annähert) erhalten. Auf diesem Weg kann die vorgeschlagene Methode ohne statistisches Wissen über das Signal angewendet werden. Simulationsexperimente zeigen, daß die Genauigkeit der vorgeschlagenen Methode vergleichbar ist mit klassischen statistischen Detektoren bei einer niedrigeren Rechenlast.

Résumé

Nous présentons dans cet article un algorithme nouveau pour le recouvrement d'un signal étalé à partir d'un registre bruité. L'algorithme suppose une nouvelle distribution a priori pour le signal étalé sous la forme d'un mélange d'une gaussienne étroite et d'une gaussienne large, toutes deux de moyenne nulle. Un terme de pénalité qui favorise les solutions correspondant

*Tel.: 34-42-201552; fax: 34-42-201488; e-mail: labgts@elmhp4.dicom.unican.es.

PII S0165-1684(96)00105-3

à ce modèle est additionné à la fonction de coût d'erreur habituelle et la fonction de coût globale résultante est minimisée à l'aide d'un algorithme de type gradient. Une condition sur le paramètre de pas d'itération est donnée pour garantir la convergence. Dans cet article nous proposons également une méthode (basée sur l'algorithme d'Espérance-Maximisation) pour la mise à jour des paramètres du mélange. L'estimation du signal étalé et l'optimisation du mélange gaussien sont combinées dans l'algorithme proposé: à chaque itération une nouvelle estimée du signal et un nouveau modèle (qui approxime la distribution de la nouvelle estimée) sont obtenus. De cette manière, la méthode proposée peut être utilisée sans aucune connaissance statistique sur le signal. Des simulations montrent que la précision de la méthode proposée est compétitive vis-à-vis des détecteurs statistiques classiques tout en demandant une charge de calcul plus faible.

Keywords: Sparse deconvolution; Gaussian mixtures; Regularization; Restoration

1. 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 to as follows: given some observation sequence $\boldsymbol{z} = \{z_1, \ldots, z_M\}$, find the sparse signal $\boldsymbol{x} = \{x_1, \ldots, x_N\}$ such as

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

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

The matrix $\boldsymbol{H}$ is assumed known hereafter (nonblind deconvolution); then, solution $\boldsymbol{x}$ that minimizes the squared error $E^2 = \|\boldsymbol{z} - \boldsymbol{H}\boldsymbol{x}\|_2^2$ can be easily found. However, it is not appropriate, since the ill-conditioned character of the problem avoids the obtention of the sparse solution we are looking for. A general way to cope with this problem consists of applying regularization techniques by including some a priori knowledge on the solution, in order to select an acceptable one.

In particular, many alternatives have appeared in the literature to force a sparse solution: some of them [6] combine detection and estimation tasks using an adaptively contracted selection operator; this technique is simple and efficient, but it is very sensitive to the selection of the parameters involved in the method and, more important, sometimes misses small peaks in the first steps of the detection process. Other approaches use linear programming techniques to find a minimum L_1-norm solution; for instance, in [17] and more recently in [14], a sparse solution is found by minimizing an objective function which is a weighted sum of the L_1-norms of the estimated solution and the resultant error. It is well known that these L_1-norm minimization schemes are well suited for data driven from spiky distributions [14]. However, they have two drawbacks: the high computational cost when the data set is large, and the sparse character of the resultant error signal, which does not agree with the usual type of added noise. For cases where data are very noisy a weighted sum of the L_1-norm of the signal and the L_2-norm of the error provides better results [16].

From an alternative point of view, it is possible to consider a probabilistic model for the signal, and to regularize the problem within a Bayesian framework. Typically, the Bayesian estimator is obtained by maximizing the a posteriori probability distribution. Among the prior distributions that can be used to model sparse sequences, a popular choice is the Bernoulli–Gaussian model [10], since it permits to separate the amplitude estimation problem from the detection problem. However, this model leads to likelihood functions which are difficult to maximize; therefore, the various statistical detectors presented in [10] require a high computational effort. Considering the same prior distribution, some modifications and refinements have been proposed in [7, 8]; nevertheless, the computational load is still the main drawback.

Finally, other approaches consist of adding a regularizing (or penalizing) term to the usual squared error cost function that will penalize nonsparse solutions. By minimizing this cost function, the solution seeks a tradeoff between both terms: their relative importance is controlled by the weighting parameter, that must be selected to optimize the performance. In fact, as it is noted in [1, 4], under the usual assumption of white Gaussian noise, these regularization methods can be

interpreted in the context of Bayesian theory. In this way, there is a close relationship between considering a prior distribution for the input signal and adding a regularizing term.

Some penalty terms, previously proposed to simplify neural network architectures (i.e., pruning), have been already applied to solve sparse deconvolution problems [3]. Nevertheless, in these methods the parameters involved in the penalty terms and even the weighting parameter are fixed in advance: note that this is equivalent to considering a fixed prior distribution for the signal. Therefore, the application of these methods requires some statistical information about the signal.

To overcome some of these limitations, in this paper we model the sparse signal as a mixture of two zero-mean Gaussians with different variances. Following a Bayesian approach, this prior information is used to derive a penalty term. To consider a Gaussian mixture presents some advantages: first, the resultant cost function is convex, thus it can be minimized by applying a gradient-type algorithm; second, a procedure for updating the parameters of the mixture (variances and mixing proportions) can be included in the algorithm. Therefore, the proposed algorithm obtains simultaneously a particular model (belonging to a parametrized family of Gaussian mixtures) and a signal driven from that model.

Section 2 introduces the Gaussian mixture model and derives an appropriate penalty term for the sparse deconvolution problem. A gradient-type algorithm (with convergence proof) and the signal model optimization procedure are presented in Section 3. Section 4 shows and discusses simulation results, and Section 5 summarizes the conclusions.

2. Gaussian mixtures for sparse deconvolution

Gaussian mixtures have been used in a wide variety of problems where data can be modeled as coming from two or more different distributions [9]. Recently, Nowlan and Hinton have discussed the application of this model to a number of problems such as neural network pruning [12], blind equalization of very distorted channels [13], or competitive learning [11]. Also, Zhao et al. consider a Gaussian mixture for modeling non-Gaussian sources for autoregressive process [18].

These previous works suggest the possibility of modeling a sparse signal by means of a Gaussian mixture. Specifically, let us assume that the prior distribution of our sparse signal can be approximated with a mixture of a narrow (subscript 1) and a broad (subscript 2) zero-mean Gaussian; the narrow Gaussian models the smaller peaks (ideally nonexistents), whereas the broad one models the true peaks:

$$p(x) = \frac{\pi_1}{\sqrt{2\pi\sigma_1}} e^{-x^2/2\sigma_1^2} + \frac{\pi_2}{\sqrt{2\pi\sigma_2}} e^{-x^2/2\sigma_2^2}, \tag{2}$$

where π_1 and π_2 are the mixing proportions and they are therefore constrained to sum 1. Varying the mixture parameters (variances and mixing proportions), different prior distributions can be approximated; in fact, decreasing the variance of the narrow Gaussian, mixture (2) can fit quite well a Bernoulli–Gaussian distribution. Moreover, this kind of distribution is adequate for this detection plus estimation problem: first, a decision must be made to assign each sample to a Gaussian (detection); second, its amplitude must be estimated (estimation).

Given a set of N observations $\boldsymbol{x} = \{x_1, \ldots, x_N\}$, and denoting each component of the mixture as G_j, the detection part of the problem is related with determining the posterior probability $p(G_j|x_i)$: the probability that a particular sample x_i was generated by a particular Gaussian G_j, which is given by

$$p(G_j|x_i) = r_j(x_i) = \frac{\pi_j p_j(x_i)}{\sum_k \pi_k p_k(x_i)}, \tag{3}$$

where $p_j(x_i)$ is the probability density of x_i under Gaussian j.

Considering now that $\{x_i\}$ are independent and identically distributed (iid) according to (2), we define a penalty term as the negative of the log-likelihood function:

$$P(\boldsymbol{x}) = -\sum_i \log \sum_j \pi_j p_j(x_i). \tag{4}$$

For small samples, which are better modeled by the narrow Gaussian, the penalty term is proportional to $x_i^2/2\sigma_1^2$; thus, as long as σ_1 is small, they are strongly pushed towards zero. Conversely, the broad Gaussian models the larger peaks, so there will be much less pressure to reduce them.

Finally, we can define the following global cost objective:

$$J(\boldsymbol{x}) = \|\boldsymbol{z} - \boldsymbol{H}\boldsymbol{x}\|_2^2 - \alpha \sum_i \log \sum_j \pi_j p_j(x_i), \qquad (5)$$

where α controls the tradeoff between the squared error and the penalty term.

From a Bayesian point of view, this cost function assumes a white Gaussian noise and prior distribution (2), while α establishes a tradeoff between our confidence in the signal model and the observations.

3. The proposed algorithm

In this section we analyze the proposed functional and present a gradient-type algorithm for its minimization. In addition, we present a method for updating the signal model as well as for selecting the weighting parameter. Finally, a complete description of the algorithm will be presented.

3.1. A descent procedure for minimizing the cost function

It is straightforward to show that, if the matrix $\boldsymbol{H}$ has full rank, then $J(\boldsymbol{x})$ is a strictly convex functional of $\boldsymbol{x}$ and therefore it has only one minimum point [15]. On the other hand, setting $\partial J(\boldsymbol{x})/\partial x_i = 0$, for $i = 0, \ldots, N-1$, it can be shown that the minimum point satisfies the following relation:

$$s_i + \alpha x_i \sum_j \frac{r_j(x_i)}{\sigma_j^2} = 0, \quad i = 0, 1, \ldots, N-1, \qquad (6)$$

where $\boldsymbol{s} = 2\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{x} - \boldsymbol{z})$ (the superscript T denoting transpose).

The above results suggest applying the following iterative procedure to minimize $J(\boldsymbol{x})$:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \boldsymbol{D}\boldsymbol{c}_k, \qquad (7)$$

where $\boldsymbol{D}$ is a diagonal matrix given by

$$\boldsymbol{D} = \mathrm{diag}(\mu_0, \mu_2, \ldots, \mu_{N-1}) \qquad (8)$$

and $\boldsymbol{c}_k$ is an $N \times 1$ column vector that can be written as

$$\boldsymbol{c}_k = \boldsymbol{s}_k + \alpha \boldsymbol{q}_k, \qquad (9)$$

where

$$\boldsymbol{s}_k = 2\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{x}_k - \boldsymbol{z}) \qquad (10)$$

is the correction due to the L_2-norm of the error, and the components of vector $\boldsymbol{q}_k$ are given by

$$q_{i,k} = x_{i,k} \sum_j \frac{r_j(x_{i,k})}{\sigma_j^2}, \qquad (11)$$

and they correspond to the correction due to the penalty term. In (11), $x_{i,k}$ is sample i at iteration k.

On the other hand, for $\alpha = 0$, the values of μ_i in (8) must be less than the inverse of the maximum eigenvalue of $\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}$, $\lambda_{\max}$, in order to guarantee convergence; otherwise, convergence depends as well on both α and the mixture parameters. Therefore, our objective is now to find what conditions should be imposed on the step-size parameters μ_i in order to ensure convergence. We present the following theorem.

Theorem 1. *The iterative algorithm described by (7) converges to a minimum point of $J(\boldsymbol{x})$ if, at any iteration, the step-size parameters μ_i in (8) are selected according to the following conditions:*

(1) $\mu_i = 0$ *if* $x_{i,k} = 0$,

(2) $\mu_i < \min\left\{\frac{1}{\lambda_{\max} + \alpha/2\sigma_1^2}, \frac{x_{i,k}}{c_{i,k}}\right\}$ *if* $x_{i,k}c_{i,k} > 0$,

(3) $\mu_i < \frac{1}{\lambda_{\max} + \alpha/2\sigma_1^2}$ *if* $x_{i,k}c_{i,k} < 0$.

Proof. See Appendix A.

Note that condition (1) states that no correction will be made for samples which have been already pushed to zero. On the other hand, condition (2) considers the case when $x_{i,k}c_{i,k} > 0$, which means that sample x_i will decrease due to the correction term: in this case, the possibility of a change of sign is avoided by including the condition $\mu_i < x_{i,k}/c_{i,k}$.

The behaviour of the algorithm depends on the Gaussian mixture parameters as well as on the weighting parameter α. The mixture parameters can be grouped in the vector $\theta = (\pi_1, \sigma_1, \pi_2, \sigma_2)$. This set of parameters that defines the prior distribution of the sparse signal is known as hyperparameters [7]. In the next subsection we present a procedure to get them while the algorithm proceeds. Subsequently, we will propose a method to choose the weighting parameter α.

3.2. Signal model optimization

The most obvious procedure consists of 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 $1-\lambda$. If estimates of these parameters are available, we could choose the mixture parameters in the following way: $\pi_1 = 1-\lambda$, $\pi_2 = \lambda$, $\sigma_2^2 = \sigma_x^2$, and a small value for σ_1^2. This procedure achieves good results if we dispose of an appropriate statistical knowledge of the problem.

When the set of hyperparameters θ is unknown, the problem is to find estimates for both $\boldsymbol{x}$ and θ. In this case, it is more appropriate to denote functional (5) as $J(\boldsymbol{x}, \theta)$. Specifically, to estimate θ we propose to apply the Expectation-Maximization (EM) algorithm [5]. To develop this idea, let us start by defining the observed incomplete data as the signal obtained from (7) after iteration k: $\boldsymbol{x}_k$. On the other hand, the unobserved data is given by $\chi = (\boldsymbol{d}_1, \boldsymbol{d}_2)$, where $\boldsymbol{d}_j$, $j = 1,2$, is a set of Bernoulli random variables selecting the Gaussian associated to each sample, i.e.,

$$d_{i,j} = \begin{cases} 1 & \text{if } x_i \in G_j, \\ 0 & \text{if } x_i \notin G_j. \end{cases} \tag{12}$$

Using this particular choice for the complete data, $(\boldsymbol{x}_k, \boldsymbol{d}_1, \boldsymbol{d}_2)$, and denoting the current estimate of θ after k iterations of the EM algorithm as θ_k, it is easy to see that the E-step of the next iteration is given by [9]

$$E\left[d_{i,j} | \boldsymbol{x}_k, \theta_k\right] = r_j(x_{i,k}), \tag{13}$$

where $E[\cdot]$ denotes expectation; then, the E-step is equivalent to recompute the posterior probabilities.

Once $r_j(x_{i,k})$ is known, in the M-step we reestimate the mixture parameters according to

$$\theta_{k+1} = \underset{\theta}{\operatorname{argmax}}\, J(\boldsymbol{x}_{k+1}, \theta). \tag{14}$$

Taking the derivative of $J(\boldsymbol{x}, \theta)$ with respect to θ and equating it to zero gives the following mixture parameters:

$$\sigma_{j,k+1}^2 = \frac{\sum_i x_{i,k}^2 r_j(x_{i,k})}{\sum_i r_j(x_{i,k})}, \tag{15}$$

$$\pi_{j,k+1} = \frac{1}{N}\sum_i r_j(x_{i,k}). \tag{16}$$

It is known that the convergence rate of the EM algorithm may be slow. To avoid this problem, in the present paper we propose the following modification: after each new $\boldsymbol{x}_k$ is obtained, only one iteration of the EM algorithm is carried out to obtain a new estimate of the hyperparameters.

On the other hand, note that each new estimate of θ changes the cost function; therefore, to avoid stability problems, it is important to force a slow change in the hyperparameters. For this reason, we choose to use the following updating procedure:

$$\sigma_{j,k+1}^2 = \gamma\sigma_{j,k}^2 + (1-\gamma)\frac{\sum_i x_{i,k}^2 r_j(x_{i,k})}{\sum_i r_j(x_{i,k})}, \tag{17}$$

$$\pi_{j,k+1} = \gamma\pi_{j,k} + (1-\gamma)\frac{1}{N}\sum_i r_j(x_{i,k}), \tag{18}$$

γ being a constant near to 1.

We conclude this subsection by noting that although the convergence conditions given by Theorem 1 were obtained assuming a fixed set of hyperparameters, the result is still valid if we consider in each iteration the current estimates given by the EM iteration.

3.3. A method for choosing the weighting parameter

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 $\boldsymbol{H}$ and on the original signal 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 practice because the obtained solutions are not usually critical with respect to α. On the other hand, if an estimate of the noise variance $\hat{\sigma}^2$ is available, we can use this knowledge to adaptively obtain the optimum weighting parameter. In particular, the idea of the proposed method is to select a value of α which leads to a solution $\hat{\boldsymbol{x}}$ satisfying

$$\hat{\sigma}^2 - \varepsilon < \frac{1}{N}\|\boldsymbol{z} - \boldsymbol{H}\hat{\boldsymbol{x}}\|_2^2 < \hat{\sigma}^2 + \varepsilon, \tag{19}$$

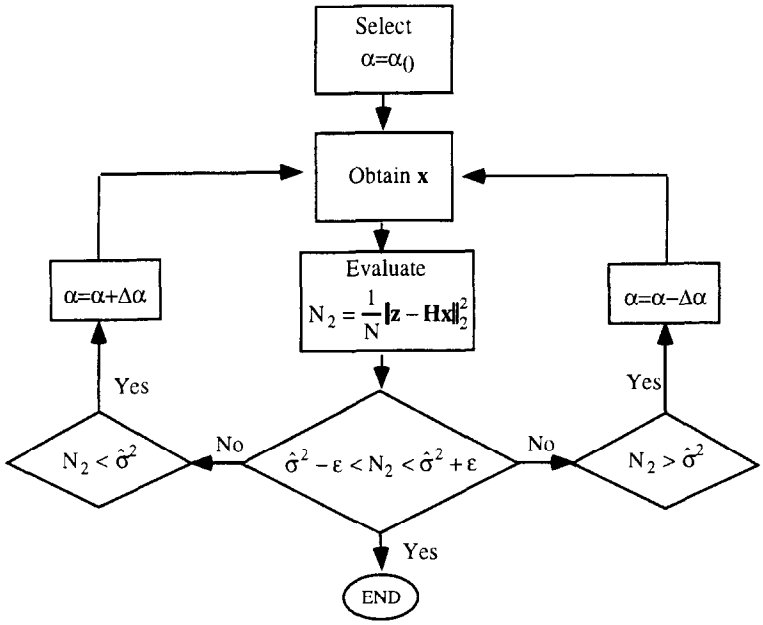

Fig. 1. Flowchart of the proposed method to select the optimum weighting parameter.

where the use of ε is prompted by statistical considerations.

The method that we propose starts selecting an initial guess for the weighting parameter, α_0; then, the solution and the variance of the associated residual (that will be denoted as $N_2 = (1/N)\|z - H\hat{x}\|_2^2$) are obtained. Next, if N_2 is higher than $\hat{\sigma}^2$ (thus meaning that a solution with few peaks has been obtained), α is reduced a fixed value $\Delta\alpha$; otherwise α is increased. This procedure continues until (19) is fulfilled. A complete description of the method is shown in Fig. 1.

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 (7) with the obtained solution for the previous α.

3.4. The overall algorithm

Finally, the proposed algorithm can be summarized in the following steps:

1. Initialize the mixture parameters $\pi_{j,0}, \sigma_{j,0}$, $j = 1, 2$; and the weighting parameter $\alpha = \alpha_0$;
2. $\boldsymbol{x}_0 = \boldsymbol{O}_{N\times 1}$.
3. For $k = 0$ to $(m_1 - 1)$,
   $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + 2\mu\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{z} - \boldsymbol{H}\boldsymbol{x}_k)$;
   end.
4. For $k = m_1$ to $(m_1 + m_2)$,
   4.1. compute $r_j(x_{i,k})$, for $i = 0, \ldots, N-1$;
   4.2. compute vector $\boldsymbol{q}_k$;
   4.3. select matrix $\boldsymbol{D} = \mathrm{diag}(\mu_0, \ldots, \mu_{N-1})$ according to Theorem 1;
   4.4. obtain new signal estimate:
   $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \boldsymbol{D}\left[2\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{x}_k - \boldsymbol{z}) + \alpha\boldsymbol{q}_k\right]$;
   4.5. recompute $r_j(x_{i,k})$, for $i = 0, \cdots, N-1$ (E-step);
   4.6. update the signal model according to (17) and (18) (modified M-step);
   end.
5. if $\hat{\sigma}^2 - \varepsilon < \frac{1}{N}\|\boldsymbol{z} - \boldsymbol{H}\boldsymbol{x}_{k+1}\|_2^2 < \hat{\sigma}^2 + \varepsilon$ then
   solution: $\hat{\boldsymbol{x}} = \boldsymbol{x}_{k+1}$;
   else
   obtain new α;
   return to 2;
   end.

Let us expose some comments about the presented algorithm. First, we have introduced a minor modification consisting of using iteration (7) without penalty term, i.e., $\alpha = 0$, for a small number of steps. This modification is necessary since the proposed method must start with a nonzero signal (recall that $\mu_i = 0$ if $x_i = 0$); in addition, a small number of iterations is enough in order to take away the signal from its starting point.

Second, we need to consider the issue of initializing the mixture parameters. A reasonable mixture initialization could be $\pi_1 = \pi_2 = 0.5$ and $\sigma_2^2 > \sigma_1^2$, with σ_1^2 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_2^2 increases and σ_1^2 becomes smaller. On the other hand, π_2 and π_1 drive toward the mean number of samples modeled by each Gaussian.

When σ_1^2 approaches zero too closely, the algorithm may become unstable. In [12] 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_1^2 is not allowed to approach zero. Nevertheless, this alternative can be avoided in our problem since we are interested in decreasing σ_1^2 as much as possible, because in this way the useless samples approach zero. For this reason, we have chosen to work directly with σ_j^2 (instead of γ_j) in the following way: iteration (7) is carried out until a maximum number of iterations m_2 is reached or a constraint of the form $\sigma_1^2 < \delta$ is satisfied, where δ is an empirical constant close to zero which prevent us from arriving to unstability.

Finally, a brief discussion about how the responsibility factors $r_j(x_i)$ evolve while the algorithm proceeds. As we pointed out previously, when the narrow Gaussian models a small sample, it is pushed towards zero, thus increasing its posterior probability $r_1(x_i)$ even more; the same happens for the broad Gaussian and the larger samples. Therefore, if over time a sample is better modeled by a particular Gaussian, its posterior probability under that Gaussian approaches 1; otherwise it approaches 0. Consequently, the posterior probabilities $r_j(x_i)$ can be viewed like soft detectors.

4. Simulation results

We have selected two computer experiments with different sparse signals: the first uses a deterministic signal and illustrates the application of the proposed method; the second uses Bernoulli–Gaussian sparse signals and it serves to compare the performance of the proposed method with a statistical detector based on that model [10].

4.1. Experiment 1

In this example, we consider a 110-point length sparse signal having nonzero values at points $x_{20} = 8$, $x_{25} = 6.845, x_{47} = -5.4, x_{71} = 4$ and $x_{95} = -3.6$. The impulse response corresponds to the first 20 points of an ARMA filter having one zero at $z = 0.6$ and two poles at $z = 0.8\ \exp(\pm j5\pi/12)$ (note that $H(m,n)$ corresponds to the values h_{m-n} of the impulse response). The SNR used in this example is 4 dB, and it is defined as the power of $\boldsymbol{Hx}$ with respect to the power of $\boldsymbol{n}$, $\boldsymbol{n}$ being a zero mean Gaussian noise.

To apply the proposed method we initialize the mixture parameters with the following values: $\pi_1 = \pi_2 = 0.5$, $\sigma_2^2 = \sigma_z^2$ and $\sigma_1^2 = \sigma_z^2/2$, where σ_z^2 is the variance of the observations. We use iteration (7) without penalty term until $\boldsymbol{x}_{20}$ and then we apply the proposed method with a fixed value $\alpha = 2$.

Figs. 2(a)–(d) show the initial noisy register, and the obtained signal using the proposed method after iteration 10, 30 and 50 (final result), respectively. It is interesting to note that we obtain an accurate solution very quickly. Also, we can remark once again how the narrow Gaussian pulls the false peaks to zero (see Fig. 2(d)), thus avoiding the need of applying a final threshold.

Figs. 3(a)–(c) illustrate the evolution of the mixture parameters versus the number of iterations k: the decrease in σ_1^2 and the increase in σ_2^2 indicate a good behavior of the algorithm.

To complete the summary of results for this example, in Figs. 4(a) and (b) we represent the posterior probability for the broad Gaussian after 10 iterations and its final value, respectively. Considering our detection plus estimation problem, $r_2(x_i)$ solves the detection part: if a sample x_i is captured by the broad Gaussian then $r_2(x_i) = 1$, and this implies that at position i there is a peak. As we can see in Fig. 4(a),

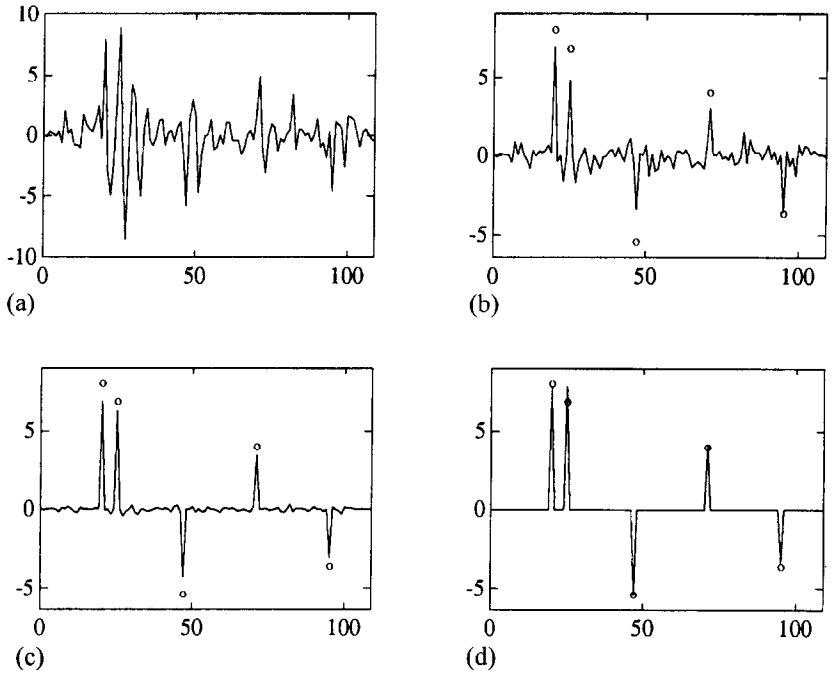

Fig. 2. (a) Convolved signal plus noise (SNR = 4 dB); (b) solution after 10 iterations with penalty term (circles depict true event amplitudes); (c) after 30 iterations; (d) after 50 iterations (final solution).

during the first iterations $r_2(x_i)$ has a value between 0 and 1 for the doubtful peaks, thus establishing a soft-detection procedure.

4.2. Experiment 2

In this example we evaluate the performance of our algorithm using synthetic signals generated according to the Bernoulli–Gaussian model (this distribution is often used for seismic deconvolution cases). Also, we present a comparison with the Single Most Likely Replacement (SMLR) detector [10]: a statistical detector based on the Bernoulli–Gaussian model.

We generated a Bernoulli-Gaussian sequence of 500 points with $\lambda = 0.05$ (percentage of nonzero samples) and $\sigma_x^2 = 10$ (variance of the Gaussian distribution), and then we convolved this sequence with the impulse response described in Example 1. Finally, a zero-mean Gaussian noise was added to the result to produce a SNR=10 dB.

To apply the proposed method we initialize the mixture parameters with the same values used in Example 1: $\pi_1 = \pi_2 = 0.5$, $\sigma_2^2 = \sigma_z^2$ and $\sigma_1^2 = \sigma_z^2/2$, where σ_z^2 is the variance of the observations. We have used a value $\alpha = 0.15$, which has been selected to fulfill (19).

Fig. 5(a) shows the result obtained with the proposed method (circles depict true event amplitudes): there are six missed detections and no false alarms; while Fig. 5(b) shows the result obtained with the SMLR detector: in this case, there are five missed detections and one false alarm. These results are confirmed in Table 1, which shows the averaged results of 50 simulations for different SNRs. In particular, Table 1 shows the average detection percentage, and (in parentheses) the false alarm percentage. The 95% confidence interval is about $\pm 2\%$ for the detection percentage and $\pm 0.1\%$ for the false alarm percentage.

From these results it can be concluded that the performance of both methods is fairly equivalent: the SMLR detector achieves a higher detection probability than the proposed method, but the false alarm percentage is also higher. On the other hand, the computational cost of the proposed method is very low: between 25 and 75 iterations were enough for all the

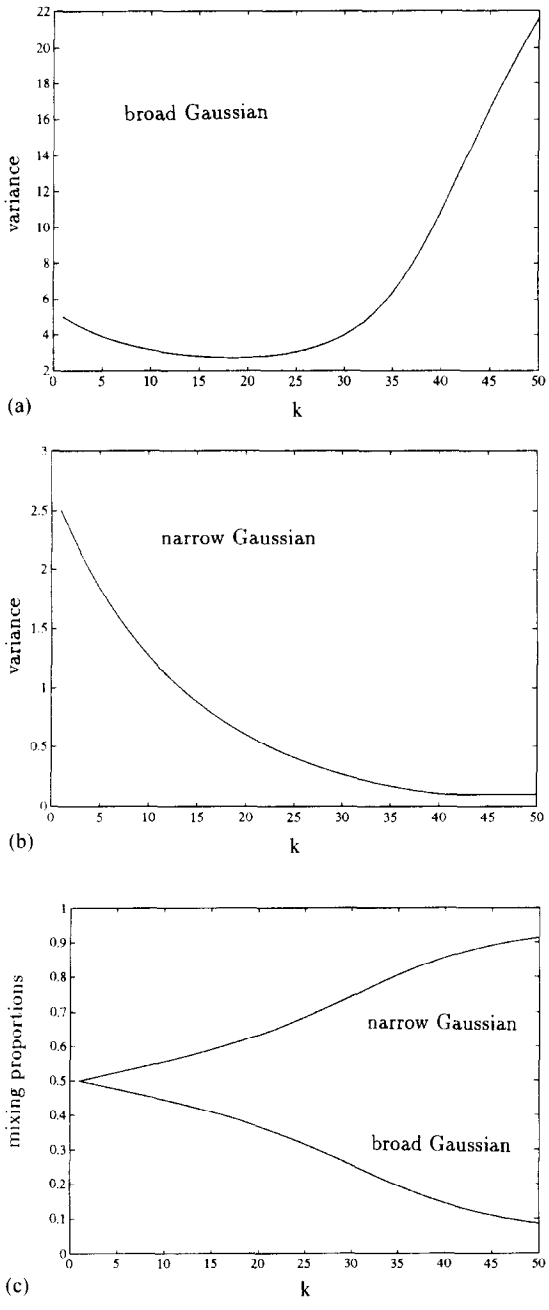

Fig. 3. Evolution of the Gaussian mixture parameters versus number of iterations (k): (a) variance of the broad Gaussian; (b) variance of the narrow Gaussian; (c) mixing proportions.

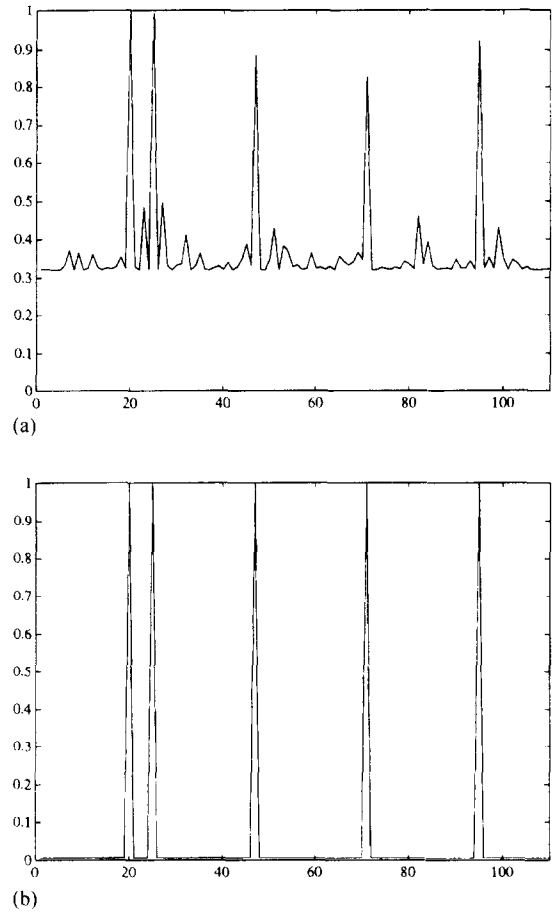

Fig. 4. Posterior probability for the samples under the broad Gaussian: (a) after iteration 10; (b) after convergence.

performed simulations; while the SMLR detector requires at least the inversion of an $N \times N$ matrix, where N is the register length. In order to give an approximate idea about the gain in speed achieved using the proposed method we have compared the overall runtime of both approaches: using programs written in MATLAB and running on a PC (100 MHz), the SMLR detector takes several minutes in order to deconvolve a register of 500 points, while the proposed method takes only a few seconds. Furthermore, the SMLR detector only solves the detection part of the sparse deconvolution problem; usually, the amplitude estimates are obtained by using a minimum variance deconvolution filter, and the model parameters are estimated in a previous step. The proposed method, however, is able to solve the whole problem: it detects the nonzero positions and estimates their amplitudes; besides, it also estimates the model parameters.

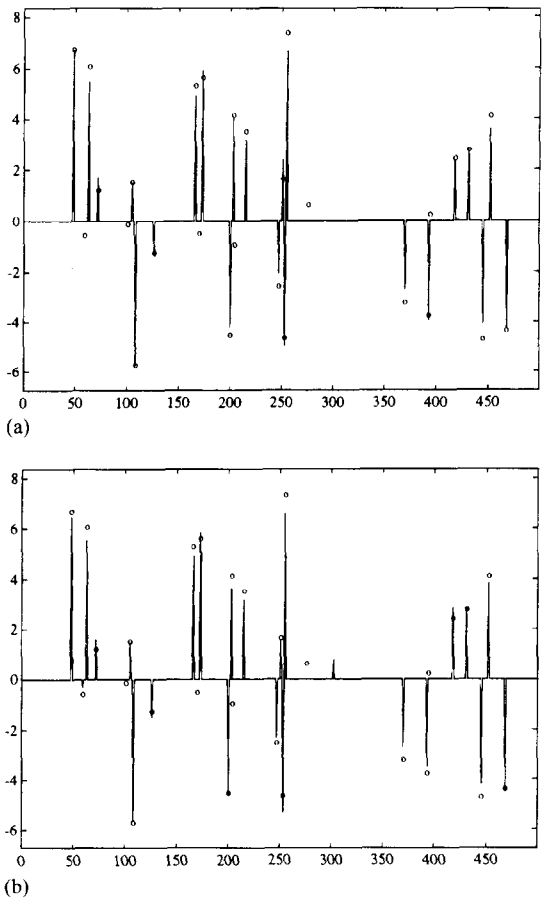

Fig. 5. (a) Proposed algorithm estimates (circles depict true event amplitudes); (b) SMLR detector estimates.

Table 1
Comparison of the results from the algorithms: (A.1) proposed method; (A.2) SMLR detector. The table shows the average detection percentage and (in parentheses) the false alarm percentage

	4 dB	6 dB	8 dB	10 dB
A.1	64.8 (0.7)	70.9 (0.6)	73.6 (0.2)	79.4 (0.1)
A.2	74.4 (1.2)	80.3 (1.2)	84.2 (1.0)	87.5 (1.0)

5. Conclusions

We have presented a new algorithm to recover a sparse signal from a noisy register. Considering a Gaussian mixture for modeling the sparse signal, we have derived a penalty term that can be used to regularize the usual quadratic error cost function. The convex character of the resultant functional allows to find the minimum by using simple gradient-techniques. The proposed algorithm includes a method (based on the EM algorithm) for optimizing the mixture parameters over time; therefore, our procedure can be used without any statistical knowledge about the signal: this establishes one of the main differences with other related approaches. A procedure for selecting the regularization or weighting parameter completes the description of our algorithm.

Simulations show that the proposed method obtains results competitive with other statistical detectors, but with a much lower computational cost.

Notation

$\boldsymbol{z}$	observations vector
$\boldsymbol{x}$	sparse signal vector
$\boldsymbol{n}$	noise vector
$\boldsymbol{H}$	impulse response matrix
$\boldsymbol{x}_k$	estimate of $\boldsymbol{x}$ at iteration k
$\boldsymbol{c}_k$	correction vector at iteration k
$\boldsymbol{s}_k$	correction due to the L_2-norm of the error (iteration k)
$\boldsymbol{q}_k$	correction due to the penalty term (iteration k)
x_i	ith component of vector $\boldsymbol{x}$
$x_{i,k}$	estimate of x_i at iteration k
μ_i	step-size parameter corresponding to the ith component
σ_1^2, π_1	variance and proportion of the narrow Gaussian
σ_2^2, π_2	variance and proportion of the broad Gaussian
θ	set of hyperparameters
G_j	mixture components
$r_j(x_i)$	posterior probability of component x_i under Gaussian G_j
$\lambda_{\max}$	maximum eigenvalue of matrix $\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}$
α	regularization (weighting) parameter
λ	sparseness parameter for the Bernoulli–Gaussian model

Appendix A

Proof of Theorem 1. The proof of Theorem 1 is based on a well-known result from optimization theory, which can be written as follows.

Theorem 2. *Let $J(\boldsymbol{x})$ be a convex functional in R^N; if $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ are generated by iteration (7), and $J(\boldsymbol{x}_{k+1}) < J(\boldsymbol{x}_k)$ for any k, then the sequence has a limit point $\boldsymbol{x}$, which is a minimum point for $J(\boldsymbol{x})$.*

Proof. See [2].

Then, to ensure convergence, we must impose the condition

$$J(\boldsymbol{x}_{k+1}) - J(\boldsymbol{x}_k) < 0, \quad \forall k. \tag{20}$$

Before doing that, note that $J(\boldsymbol{x})$ can be rewritten as

$$J(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} - \boldsymbol{x}^{\mathrm{T}}\boldsymbol{b} - \alpha \sum_i \log \sum_j \pi_j p_j(x_i), \tag{21}$$

where we have discarded the constant term $\|\boldsymbol{z}\|_2^2$, $\boldsymbol{A} = \boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}$ and $\boldsymbol{b} = 2\boldsymbol{H}^{\mathrm{T}}\boldsymbol{z}$.

Now, evaluating the difference between $J(\boldsymbol{x}_{k+1})$ and $J(\boldsymbol{x}_k)$, we obtain

$$\begin{aligned} &J(\boldsymbol{x}_{k+1}) - J(\boldsymbol{x}_k) \\ &\quad = \boldsymbol{c}_k^{\mathrm{T}}\boldsymbol{D}\boldsymbol{A}\boldsymbol{D}\boldsymbol{c}_k - \boldsymbol{c}_k^{\mathrm{T}}\boldsymbol{D}\boldsymbol{s}_k \\ &\qquad -\alpha \sum_i \log \left\{ \frac{\sum_j \pi_j p_j(x_{i,k} - \mu_i c_{i,k})}{\sum_j \pi_j p_j(x_{i,k})} \right\}. \end{aligned} \tag{22}$$

Using the definition of the posterior probabilities, it is not difficult to see that the last term in (22) can be rewritten as

$$\begin{aligned} &\sum_i \log \left\{ \frac{\sum_j \pi_j p_j(x_{i,k} - \mu_i c_{i,k})}{\sum_j \pi_j p_j(x_{i,k})} \right\} \\ &= \sum_i \log \left\{ \sum_j r_j(x_{i,k}) \exp \frac{(-\mu_i^2 c_{i,k}^2 + 2\mu_i c_{i,k} x_{i,k})}{2\sigma_j^2} \right\}. \end{aligned} \tag{23}$$

Additionally, we impose the condition

$$\mu_i < \frac{x_{i,k}}{c_{i,k}} \quad \text{if} \quad x_{i,k} c_{i,k} > 0. \tag{24}$$

This condition only avoids a change of sign in sample i due to the correction term.

Taking into account (24) and since $0 \leqslant r_j(x_{i,k}) \leqslant 1$, the following bound holds:

$$\begin{aligned} &\sum_i \log \left\{ \sum_j r_j(x_{i,k}) \exp \frac{(-\mu_i^2 c_{i,k}^2 + 2\mu_i c_{i,k} x_{i,k})}{2\sigma_j^2} \right\} \\ &\quad \geqslant \sum_i \sum_j \frac{r_j(x_{i,k})}{2\sigma_j^2} \left(-\mu_i^2 c_{i,k}^2 + 2\mu_i x_{i,k} c_{i,k}\right). \end{aligned} \tag{25}$$

The proof of (25) is based on the following inequalities:

$$\log(A + B) \geqslant \log(A) + \log(B) \quad \text{for } 0 < A, B < 1; \tag{26}$$

$$\log(AB) \leqslant A \log(B) \quad \text{for} \quad 0 < A, B < 1. \tag{27}$$

Now, let us define the following diagonal matrix:

$$\boldsymbol{R} = \operatorname{diag}\left(\sum_j \frac{r_j(x_{0,k})}{2\sigma_j^2}, \ldots, \sum_j \frac{r_j(x_{N-1,k})}{2\sigma_j^2} \right). \tag{28}$$

Using (28) and the definition of vector $\boldsymbol{q}_k$ given by (11), the right term of (25) can be written as

$$\begin{aligned} &\sum_i \sum_j \frac{r_j(x_{i,k})}{2\sigma_j^2} \left(-\mu_i^2 c_{i,k}^2 + 2\mu_i x_{i,k} c_{i,k}\right) \\ &\quad = -\boldsymbol{c}_k^{\mathrm{T}}\boldsymbol{D}\boldsymbol{R}\boldsymbol{D}\boldsymbol{c}_k + \boldsymbol{c}_k^{\mathrm{T}}\boldsymbol{D}\boldsymbol{q}_k. \end{aligned} \tag{29}$$

After substitution into (22), the following inequality is obtained:

$$\begin{aligned} &J(\boldsymbol{x}_{k+1}) - J(\boldsymbol{x}_k) \\ &\quad \leqslant \boldsymbol{c}_k^{\mathrm{T}}\boldsymbol{D}(\boldsymbol{A} + \alpha\boldsymbol{R})\boldsymbol{D}\boldsymbol{c}_k - \boldsymbol{c}_k^{\mathrm{T}}\boldsymbol{D}(\boldsymbol{s}_k + \alpha\boldsymbol{q}_k); \end{aligned} \tag{30}$$

recalling that $\boldsymbol{c}_k = \boldsymbol{s}_k + \alpha\boldsymbol{q}_k$, we obtain the convergence condition

$$\boldsymbol{c}_k^{\mathrm{T}}\boldsymbol{D}(\boldsymbol{A} + \alpha\boldsymbol{R})\boldsymbol{D}\boldsymbol{c}_k - \boldsymbol{c}_k^{\mathrm{T}}\boldsymbol{D}\boldsymbol{c}_k < 0. \tag{31}$$

A sufficient condition to fulfill (31) is obtained if, for each μ_i, we impose

$$\mu_i^2 \boldsymbol{c}_k^{\mathrm{T}}(\boldsymbol{A} + \alpha\boldsymbol{R})\boldsymbol{c}_k - \mu_i \boldsymbol{c}_k^{\mathrm{T}}\boldsymbol{c}_k < 0. \tag{32}$$

Denoting $\boldsymbol{B} = \boldsymbol{A} + \alpha\boldsymbol{R}$, this last inequality reduces to

$$\boldsymbol{c}_k^{\mathrm{T}}(\boldsymbol{I} - \mu_i\boldsymbol{B})\boldsymbol{c}_k > 0, \tag{33}$$

where $\boldsymbol{I}$ is the identity matrix. This condition implies that $\boldsymbol{I} - \mu_i\boldsymbol{B}$ must be positive definite; therefore, since

all eigenvalues of matrix $\boldsymbol{B}$ are positive, the iterative algorithm converges if

$$0 < \mu_i < \frac{1}{b_{\max}}, \tag{34}$$

where $b_{\max}$ is the maximum eigenvalue of $\boldsymbol{B}$.

Now, recalling that $0 \leqslant r_j(x_{i,k}) \leqslant 1$ and that $\sigma_1^2 < \sigma_2^2$, the following bound holds:

$$b_{\max} < \lambda_{\max} + \frac{\alpha}{2\sigma_1^2}, \tag{35}$$

where $\lambda_{\max}$ is the maximum eigenvalue of $\boldsymbol{A} = \boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}$, α is the weighting parameter and σ_1^2 is the variance of the narrow Gaussian. Hence, we get the final result

$$0 < \mu_i < \frac{1}{\lambda_{\max} + \alpha/2\sigma_1^2}. \tag{36}$$

This completes the proof of Theorem 1. □

References

[1] S. Alliney and S.A. Ruzinsky, "An algorithm for the minimization of mixed L_1 and L_2 norms with application to Bayesian estimation", *IEEE Trans. Signal Process.*, Vol. SP-42, No. 3, March 1994, pp. 618–627.

[2] M. Aoki, *Introduction to Optimization Techniques*, Macmillan, New York, 1971.

[3] J. Cid-Sueiro and A.R. Figueiras-Vidal, "Pruning techniques for spiky deconvolution", *Proc. 2nd. Cost 229 on Adaptive Algorithms in Communications*, Bordeaux, France, October 1992, pp. 161–168.

[4] G. Demoment, "Image reconstruction and restoration: Overview of common estimation structures and problems", *IEEE Trans. Acoust. Speech Signal Process.*, Vol. ASSP-37, No. 12, December 1989, pp. 2024–2036.

[5] A.P. Dempster, N.M. Laird and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm", *J. Roy. Statist. Soc.*, Vol. B-39, No. 1, January 1977, pp. 1–37.

[6] A.R. Figueiras-Vidal, D. Docampo-Amoedo, J.R. Casar-Corredera and A. Artés-Rodríguez, "Adaptive iterative algorithms for spiky deconvolution", *IEEE Trans. Acoust. Speech Signal Process.*, Vol. ASSP-38, No. 8, August 1990, pp. 1462–1466.

[7] Y. Goussard, G. Demoment and F. Monfront, "Maximum a posteriori detection-estimation of Bernoulli–Gaussian processes", *Proc. 2nd. IMA Conf. on Mathematics in Signal Processing*, Univ. of Warwick, UK, 1988, pp. 121–138.

[8] M. Lavielle, "Bayesian deconvolution of Bernoulli–Gaussian processes", *Signal Processing*, Vol. 33, No. 1, July 1993, pp. 67–79.

[9] G.J. McLachlan and K.E. Basford, *Mixture Models: Inference and Applications to Clustering*, Marcel Dekker, New York, 1988.

[10] J.M. Mendel, *Maximum Likelihood Deconvolution*, Springer, New York, 1990.

[11] S.J. Nowlan, "Maximum likelihood competitive learning", in: D.S. Touretzky, ed., *Advances in Neural Information Processing Systems II*, Morgan Kaufmann, San Mateo, CA, 1990, pp. 574–582.

[12] S.J. Nowlan and G.E. Hinton, "Simplifying neural networks by soft weight-sharing", *Neural Comput.*, Vol. 4, No. 4, 1992, pp. 473–493.

[13] S.J. Nowlan and G.E. Hinton, "A soft decision-directed LMS algorithm for blind equalization", *IEEE Trans. Commun.*, Vol. COM-41, No. 2, February 1993, pp. 275–279.

[14] M.S. O'Brien, A.N. Sinclair and S.M. Kramer, "Recovery of a sparse spike time series by L_1-norm deconvolution", *IEEE Trans. Signal Process.*, Vol. SP-42, No. 12, December 1994, pp. 3353–3365.

[15] I. Santamaría-Caballero, Sparse deconvolution by means of penalty methods (in Spanish), Ph.D. Dissertation, Polytechnic University of Madrid, July 1995.

[16] I. Santamaría-Caballero and A.R. Figueiras-Vidal, "Some weighted objective approaches to sparse deconvolution", *Proc. European Signal Processing Conf.*, Edinburgh, Scotland, UK, 13–16 September 1994, pp. 1094–1097.

[17] H.L. Taylor, S.C. Banks and J.F. McCoy, "Deconvolution with the L_1-norm", *Geophysics*, Vol. 44, No. 1, January 1979, pp. 39–52.

[18] Y. Zhao, X. Zhuang and S.J. Ting, "Gaussian mixture density modeling of non-Gaussian source for autoregressive process", *IEEE Trans. Signal Process.*, Vol. SP-43, No. 4, April 1995, pp. 894–903.