LEMUR: Latent EM Unsupervised Regression for Sparse Inverse Problems

Pierre Barbault, Matthieu Kowalski, Charles Soussen

LEMUR: Latent EM Unsupervised Regression for Sparse Inverse Problems

Pierre Barbault, Matthieu Kowalski and Charles Soussen, *Member, IEEE*

***Abstract*—Most methods for sparse signal recovery require setting one or several hyperparameters. We propose an unsupervised method to estimate the parameters of a Bernoulli-Gaussian (BG) model describing sparse signals. The proposed method is first derived for denoising problems using a maximum likelihood (ML) approach. Then, an extension to general inverse problems is achieved through a latent variable formulation. Two expectation-maximization (EM) algorithms are then proposed to estimate the signal together with the BG model parameters. Combining these two approaches leads to the proposed LEMUR algorithm. LEMUR is then evaluated on extensive simulations regarding the ability to recover the parameters and provide accurate sparse signal estimates.**

***Index Terms*—Bernoulli-Gaussian, Expectation-Maximization, Sparsity, Hyperparameter estimation.**

I. Introduction

Linear inverse problems are prevalent in various scientific and engineering fields, encompassing scenarios where the objective is to reconstruct an unknown signal or image from noisy measurements. Many signals of interest, such as natural images and audio signals, exhibit sparse representation, which can be exploited for reconstruction from incomplete and noisy measurements. Sparsity is typically achieved using a linear transform or dictionary that allows the signal to be represented with sparse synthesis coefficients. Thanks to the linearity of the inverse problem and the dictionary, we assume in the following, without loss of generality, that the signal of interest is sparse with respect to the canonical basis [1].

The relation between the noisy measurement vector $\mathbf{y} \in \mathbb{R}^M$ and the unknown sparse signal $\mathbf{x} \in \mathbb{R}^N$ is given by the linear operator $\mathbf{H} \in \mathbb{R}^{M \times N}$, with additive white Gaussian noise $\mathbf{e}$ having variance σ_e^2 and zero mean:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{e}. \tag{1}$$

Regularized least-squares are a popular approach for solving sparse inverse problems. The Lasso [2] or Basis Pursuit Denoising [3] is a common approach that uses an ℓ_1 constraint or penalty. While ℓ_1 regularization is prevalent, an ideal measure of sparsity is the so-called ℓ_0-"norm", which counts the number of non-zero coefficients in $\mathbf{x}$. The ℓ_0-regularized least-squares problems can then be solved using various methods, including proximal descent algorithms [4] such as the Fast Iterative Shrinkage/Thresholding algorithm (FISTA) [5]. These methods were initially developed for convex regularizers. However, they can be generalized to the non-convex case as long as the proximal operator can be efficiently computed. For the ℓ_0 regularized problem, the Iterative Hard Thresholding algorithm (IHT) [6] can be used. Alternatives to IHT are greedy algorithms such as Orthogonal Matching Pursuit [7] and Single Best Replacement [8]. While these methods are effective in finding a sub-optimal solution, exact solvers based on mixed integer programming have recently been proposed [9], [10] for moderate size problems, together with efficient branch and bound implementations.

The main shortcoming of these methods is the need to tune at least one hyperparameter, balancing the data-fidelity and sparsity-promoting terms. For the Lasso, the hyperparameter can be chosen using the Stein Unbiased Risk Estimator (SURE) in the denoising case [11], [12] by providing an unbiased estimate of the mean squared error between the estimated signal and the actual signal. This approach has been extended to inverse problems using the SUGAR [13] or Generalized-SURE (GSURE) [14]. However, SURE cannot be applied to ℓ_0 regularization as it is not weakly differentiable. In the denoising case, the SURE-based Coordinate-wise RElaxed (SCORE) [15] has been proposed, but its extension to inverse problems is not straightforward.

In the Bayesian framework, regularized least-squares can be interpreted as a Maximum *a Posteriori* (MAP) approach, where the regularization term is the negative log-likelihood of a prior distribution over the signal coefficients. Pereyra *et al.* [16] proposed to use a Gamma prior on the hyperparameter when the negative log-likelihood of the prior corresponds to a 1-homogeneous regularizer. This method can be efficiently optimized. Among Bayesian methods for sparsity, Sparse Bayesian Learning (SBL) is a popular approach that uses a Gaussian prior on the signal coefficients with a free variance for each coefficient. Variance estimation is achieved through marginalization with respect to the signal $\mathbf{x}$, leading to sparse estimation. A related model proposed by Calvetti and Somersalo [17] uses a Gamma prior on the variances.

One advantage of the Bayesian approach is the ability to use hierarchical Bayes models, where a hyperprior can be chosen for the hyperparameters. Markov Chain Monte Carlo (MCMC) methods with hierarchical Bayes models provide a fully unsupervised approach for hyperparameter selection. While these methods can be computationally expensive, they can accurately estimate the hyperparameters.

The Bernoulli-Gaussian (BG) model is a popular choice for modeling sparsity in signal processing and is strongly related to the ℓ_0 regularized problem through the MAP estimator [8]. In [18], the authors proposed a stochastic-EM procedure to

P. Barbault and C. Soussen are with L2S, Université Paris-Sacaly, CNRS, CentraleSupélec, Gif-sur-Yvette, France.

M. Kowalski is with Inria, Université Paris-Saclay, CNRS, Laboratoire Interdisciplinaire des Sciences du Numériques, Gif-sur-Yvette, France

This work was supported by the French National Agency for Research through the BMWs project (ANR-20-CE45-0018).

estimate the sparse solution of a deconvolution problem. However, a classical EM procedure becomes intractable because of the combinatorial nature of the related optimization problem. Following the EM approach developed in [19] that relies on an additional latent variable, [20] proposed an iterative procedure to estimate the parameters of a BG model when the matrix $\mathbf{H}$ is a union of unitary dictionaries. However, this approach leads to a non-convergent estimate even in the denoising case, as shown in [21], when using the chosen prior on the hyperparameters. Moreover, preliminary experiments have shown that this method is not robust for the general inverse problem. MCMC approaches for the BG sparse model do not scale well in practice [22], although recent efforts have been made to improve their efficiency [23], [24]. However, EM-GAMP [25] (Expectation-Maximization Generalized Approximate Message Passing) has been designed for sparse reconstruction, including using Bernoulli-Gaussian models [26]. It combines the efficiency of GAMP [27] for approximate inference with the flexibility of EM for hyperparameter estimation. EM-GAMP achieves excellent results when $\mathbf{H}$ is well-conditioned and resembles a random Gaussian matrix with low mean values. However, its performance degrades significantly when $\mathbf{H}$ has a high mean, resulting in instability in the reconstruction process.

Contributions and outline of the paper. This article is an extension of our conference paper [28]. The primary purpose is to estimate the BG parameters in the inverse problem framework together with the sparse signal reconstruction. On the contrary, only the parameters were estimated in [28]. Here, we aim to provide a fully unsupervised iterative/shrinkage thresholding algorithm. In Section II, we introduce the statistical models and derive the expression of likelihood functionals. Section III is dedicated to the denoising case. First, we propose a joint MAP/Maximum Likelihood method to estimate both sparse signal and hyperparameters. Additionally, we derive an EM algorithm, which is initialized using the method of moments. In Section IV, we elaborate on the extension to general inverse problems using the latent variable formulation proposed in [19]. We explore the choice of hidden variables and propose two distinct EM approaches. Finally, Section V includes a comprehensive experimental evaluation of the proposed approaches.

II. BG model

The sparse source signal $\mathbf{x}$ is modeled using the Bernoulli-Gaussian (BG) process. Assuming that the signal coefficients x_n are independent and identically distributed, x_n is modeled as a mixture of a Gaussian distribution and the Dirac δ distribution:

$$x_n \sim \mathcal{BG}(p, \sigma_x^2) = p\mathcal{N}(0, \sigma_x^2) + (1-p)\delta(x_n) \tag{2}$$

where p and σ_x^2 refer to the sparsity level and the variance of the nonzero entries, respectively. Alternatively, one may introduce the sequence of binary variables s_n such that

$$s_n = \begin{cases} 1 & \text{if } x_n \neq 0, \\ 0 & \text{if } x_n = 0. \end{cases} \tag{3}$$

s_n follows the Bernoulli distribution of parameter p, and the conditional distribution of x_n given s_n reads $(x_n|s_n) \sim \mathcal{N}(0, \sigma_x^2)$ if $s_n = 1$ and $(x_n|s_n) = 0$.

In Eq. (1), the independent noise vector $\mathbf{e}$ is assumed to be white and Gaussian with variance σ_e^2, thus we have:

$$\mathbf{y}|\mathbf{x} \sim \mathcal{N}(\mathbf{H}\mathbf{x}, \sigma_e^2 \mathbf{I}_M) \tag{4}$$

where $\mathbf{I}_M$ is the $M \times M$ identity matrix.

Due to the presence of the Dirac distribution in Eq. (2), one cannot directly define the maximum *a posteriori* (MAP) estimator for $\mathbf{x}$ using this equation. However, by introducing the indicator variable s_n in Eq. (3) and observing that $p(x_n \neq 0|s_n = 1) = 1$ and $p(x_n = 0|s_n = 0) = 1$, we can establish a joint-MAP formulation in terms of $(\mathbf{s}, \mathbf{x})$. Ultimately, we demonstrate that this joint estimator relies solely on $\mathbf{x}$, and we denote the latter estimator as the MAP estimator for $\mathbf{x}$. Denoting the set of model hyperparameters as

$$\theta = (p, \sigma_x^2, \sigma_e^2), \tag{5}$$

this estimator is expressed hereafter.

Proposition 1. *Let $\mathbf{x}$ follow the BG model Eq. (2) and $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{e}$. The MAP estimator of $\mathbf{x}$ is the solution of the following optimization problem:*

$$\underset{\mathbf{x}}{\operatorname{argmin}} \frac{1}{2\sigma_e^2}\|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \frac{\|\mathbf{x}\|_2^2}{2\sigma_x^2} + \lambda\|\mathbf{x}\|_0 + C(\sigma_e^2, p)$$

with

$$\lambda(\sigma_x^2, p) = \log\left(\sqrt{2\pi\sigma_x^2}\frac{1-p}{p}\right)$$

and

$$C(\sigma_e^2, p) = N\log\left(\frac{\sqrt{2\pi\sigma_e^2}}{1-p}\right)$$

Proof. The proof is given in Appendix A. □

The expression of $C(\sigma_e^2, p)$ is given for reasons that will become clear in the next sections devoted to estimating the parameters. This MAP estimator for the denoising case was directly stated in [21]; Proposition 1 complements this by providing rigorous proof of the result.

One can notice that the expression of the MAP estimator in Proposition 1 as a form similar to the one given in [8], but the parameter λ in [8] is defined as $\lambda = \log\left(\frac{1-p}{p}\right)$. This difference comes from the chosen model to induce the BG random variable $\mathbf{x}$. In [8], $\mathbf{x}$ is written as the product of two *independent* random variables $\mathbf{s}$ and $\mathbf{r}$, which model the support and the value of the coefficients, respectively. The derivation is then based on the factorization $p(\mathbf{s}, \mathbf{r}) = p(\mathbf{s})p(\mathbf{r})$. In Proposition 1, the variables $\mathbf{s}$ and $\mathbf{r}$ are dependents: if $s_n = 0$ then $r_n = 0$; hence the difference between the two criterions.

III. Parameter estimation in the denoising case

This section deals with parameter estimation in the *denoising* case. This case is modeled by $\mathbf{H} = \mathbf{I}_M$, hence

$$\mathbf{y} = \mathbf{x} + \mathbf{e}\ . \tag{6}$$

Here, the distribution of $\mathbf{y}$ is simply a mixture of two centered Gaussians with variances σ_e^2 and $\sigma_e^2 + \sigma_x^2$, and respective

weights $(1-p)$ and p. In other words, y_n are i.i.d. random variables such that

$$y_n \sim p\mathcal{N}(0, \sigma_x^2 + \sigma_e^2) + (1-p)\mathcal{N}(0, \sigma_e^2). \tag{7}$$

In Section III-A, we derive an EM procedure for ML estimation of parameters θ and then propose two methods of reconstruction of the sparse signal $\mathbf{x}$ knowing θ. In Section III-B, we propose a joint estimator of $\mathbf{x}$ and θ. We further derive the method of moments, which provides an initial estimate of θ (Section III-C).

A. Maximum Likelihood estimation

1) Estimation of hyperparameters: ML estimation amounts to maximizing $p(\mathbf{y}|\theta)$ with respect to θ. $\mathbf{y}$ being a mixture of two Gaussians, it is a typical case where the EM procedure can be derived [29]. In [28], we derived an EM algorithm using $\mathbf{x}$ as a hidden variable. We briefly summarize this procedure for the sake of completeness.

Denoting by $\theta^{(t)} = \{p^{(t)}, (\sigma_x^2)^{(t)}, (\sigma_e^2)^{(t)}\}$ the estimated parameters at iteration t, let

$$\begin{aligned}\phi_n^{(t)} &= p(s_n = 1|y_n, \theta^{(t)}) \\ &= \frac{\frac{p}{\sqrt{\sigma_x^{2\,(t)}+\sigma_e^{2\,(t)}}}\, e^{-\frac{y_n^2}{2(\sigma_x^{2\,(t)}+\sigma_e^{2\,(t)})}}}{\frac{p}{\sqrt{\sigma_x^{2\,(t)}+\sigma_e^{2\,(t)}}}\, e^{-\frac{y_n^2}{2(\sigma_x^{2\,(t)}+\sigma_e^{2\,(t)})}} + \frac{1-p}{\sqrt{\sigma_e^{2\,(t)}}}\, e^{-\frac{y_n^2}{2\sigma_e^{2\,(t)}}}}\end{aligned} \tag{8}$$

(where the latter equation results from Bayes' rule) and

$$\mu^{(t)} = \frac{(\sigma_x^2)^{(t)}}{(\sigma_x^2)^{(t)} + (\sigma_e^2)^{(t)}} \quad , \quad \nu^{(t)} = \frac{(\sigma_x^2)^{(t)}(\sigma_e^2)^{(t)}}{(\sigma_x^2)^{(t)} + (\sigma_e^2)^{(t)}} \, . \tag{9}$$

According to [28], the expectation $\mathbb{E}_{\mathbf{x}|\mathbf{y},\theta^{(t)}}\left[\log p(\mathbf{y}, \mathbf{x}|\theta)\right]$ reaches a maximum value when $\theta = \theta^{(t+1)}$, with

$$p^{(t+1)} = \frac{1}{N}\sum_{n=1}^{N} \phi_n^{(t)} \tag{10}$$

$$(\sigma_x^2)^{(t+1)} = \nu^{(t)} + \frac{(\mu^{(t)})^2}{N p^{(t+1)}} \sum_{n=1}^{N} y_n^2 \phi_n^{(t)} \tag{11}$$

$$(\sigma_e^2)^{(t+1)} = \frac{1}{N}\sum_{n=1}^{N} y_n^2 - \frac{2\mu^{(t)}}{N}\sum_{n=1}^{N} y_n^2 \phi_n^{(t)} + p^{(t+1)}(\sigma_x^2)^{(t+1)}. \tag{12}$$

The resulting EM algorithm is summarized in Alg. 1. Since EM algorithms are local maximization schemes, the choice of initial solution is an important issue. This point will be further discussed in Section III-C.

2) Estimation of the sparse signal: Once the hyperparameters θ have been retrieved using Algorithm 1, one can reconstruct the signal $\mathbf{x}$ using two alternative strategies.

The first strategy amounts to computing the posterior mean estimate $\hat{\mathbf{x}} = \mathbb{E}_{\mathbf{x}|\mathbf{y},\theta}\left[\mathbf{x}\right]$, which is also the Bayesian estimator minimizing the Mean Squared Error $\mathbb{E}_{\mathbf{x}|\mathbf{y},\theta}\left[\|\hat{\mathbf{x}} - \mathbf{x}\|^2\right]$ (with respect to $\mathbf{x}$). For denoising problems, the posterior mean is

Algorithm 1: Denoising problem: EM algorithm for parameter estimation [28].

Result: $p, \sigma_x^2, \sigma_e^2$
Input: $t = 0$, $p^{(0)}, (\sigma_x^2)^{(0)}, (\sigma_e^2)^{(0)}$
while *not converged* **do**
  update $\phi_n^{(t)}$ with Eq. (8);
  update $\mu^{(t)}$ and $\nu^{(t)}$ with Eq. (9);
  update $p^{(t+1)}, (\sigma_x^2)^{(t+1)}, (\sigma_e^2)^{(t+1)}$ with Eqs. (10) to (12);
  $t = t + 1$;
end

defined componentwise. Using the total probability rule, we get

$$\begin{aligned}\mathbb{E}_{x_n|y_n,\theta}\left[x_n\right] &= p(s_n = 1|y_n, \theta) \times \mathbb{E}_{x_n|y_n, s_n=1,\theta}\left[x_n\right] \\ &\quad + p(s_n = 0|y_n, \theta) \times 0 \\ &= \phi_n \, \mu \, y_n\end{aligned} \tag{13}$$

where ϕ_n and μ refer to the estimates in Eq. (8)-Eq. (9) computed with parameter θ obtained at the final iteration of Alg. 1 (for more details, see [28]).

The second strategy is based on the MAP reconstruction of the signal support, that is

$$s_n = 1 \;\Leftrightarrow\; p(s_n = 1|y_n, \theta) > p(s_n = 0|y_n, \theta). \tag{14}$$

where $p(s_n = 1|y_n, \theta)$ is given in Eq. (8) and $p(s_n = 0|y_n, \theta) = 1 - p(s_n = 1|y_n, \theta)$. Up to a few rearrangements, we find that $s_n = 1$ if and only if

$$y_n^2 > 2\sigma_e^2 \, \frac{\sigma_e^2 + \sigma_x^2}{\sigma_x^2} \log\left(\frac{1-p}{p}\sqrt{1 + \frac{\sigma_x^2}{\sigma_e^2}}\right). \tag{15}$$

Once the support is reconstructed using Eq. (15), we propose to compute the mean of the posterior distribution of $\mathbf{x}$ conditionally to $\mathbf{s}$ and θ. This estimator can be easily computed given that the latter distribution is Gaussian:

$$\mathbb{E}_{x_n|y_n,s_n,\theta}\left[x_n\right] = \begin{cases} \frac{\sigma_x^2}{\sigma_x^2+\sigma_e^2}\, y_n & \text{if } s_n = 1, \\ 0 & \text{if } s_n = 0. \end{cases} \tag{16}$$

The reader can refer to [30] for a study and comparison of both estimates in Eq. (13) and Eq. (16) in denoising problems.

B. Joint estimation of $\mathbf{x}$ and θ

As one is usually primarily interested in the estimation of signal $\mathbf{x}$, it seems natural to investigate the following joint estimation problem

$$(\hat{\mathbf{x}}, \hat{\theta}) = \underset{\mathbf{x},\theta}{\operatorname{argmin}}\, J(\mathbf{x}, \theta) \tag{17}$$

where

$$J(\mathbf{x}, \theta) = \frac{1}{2\sigma_e^2}\|\mathbf{y} - \mathbf{x}\|_2^2 + \frac{\|\mathbf{x}\|_2^2}{2\sigma_x^2} + \lambda(\sigma_e^2, p)\|\mathbf{x}\|_0 + C(\sigma_e^2, p) \tag{18}$$

with $\lambda(\sigma_e^2, p)$ and $C(\sigma_e^2, p)$ defined as in Proposition 1. This joint optimization problem can be interpreted as a joint MAP

estimator of $(\mathbf{x}, \theta)$ using a non-informative (uniform) prior on θ following the approach of Proposition 1 where the support $\mathbf{s}$ is inferred together with $\mathbf{x}$.

We first notice that for a given set of parameters θ, the minimization of J with respect to $\mathbf{x}$ leads to

$$\forall n, \hat{x}_n = \begin{cases} \frac{\sigma_x^2}{\sigma_x^2+\sigma_e^2}\, y_n & \text{if } y_n^2 > T(p, \sigma_x^2, \sigma_e^2), \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

where the threshold $T(p, \sigma_x^2, \sigma_e^2)$ is given by

$$T(p, \sigma_x^2, \sigma_e^2) = 2\sigma_e^2 \frac{\sigma_x^2+\sigma_e^2}{\sigma_x^2} \log\left(\sqrt{2\pi\sigma_x^2}\,\frac{1-p}{p}\right). \tag{20}$$

By substituting $\hat{\mathbf{x}}$ into Eq. (17), it has been shown [21] that the functional to minimize can be rewritten as a function depending on the threshold T and the parameters θ. The problem reduces to minimizing with respect to θ and T the following functional

$$\tilde{J}(\theta, T) = \frac{1}{2\sigma_e^2} \sum_{\{n:y_n^2>T\}} (y_n - x_n)^2 + \frac{1}{2\sigma_x^2} \sum_{\{n:y_n^2>T\}} x_n^2 + \lambda(\sigma_e^2, p) \sum_{\{n:y_n^2>T\}} 1 + C(\sigma_e^2, p) \tag{21}$$

subject to

$$T = 2\sigma_e^2 \frac{\sigma_x^2+\sigma_e^2}{\sigma_x^2} \log\left(\sqrt{2\pi\sigma_x^2}\,\frac{1-p}{p}\right). \tag{22}$$

For a given threshold value T, the minimization with respect to the parameters θ leads to:

$$\hat{p} = \frac{1}{N} \sum_{\{n;y_n^2>T\}} 1 \tag{23}$$

$$\hat{\sigma_x^2} = \frac{\sum_{\{n;y_n^2>T\}} y_n^2}{(1+\gamma)^2 \sum_{\{n;y_n^2>T\}} 1} \tag{24}$$

$$\hat{\sigma_e^2} = \gamma \sigma_x^2 \tag{25}$$

where γ is the smallest root of

$$\gamma^2(1+\rho) + \gamma\left(2 - \rho \frac{N}{\sum_{\{n:y_n^2>T\}} 1}\right) + 1 = 0 \tag{26}$$

with

$$\rho = \frac{\sum_{\{n:y_n^2>T\}} y_n^2}{\sum_{\{n:y_n^2\le T\}} y_n^2}. \tag{27}$$

To minimize Eq. (21), one can remark that, because we are in a finite-dimensional discrete setting, the quantities $\sum_{n:y_n^2>T} 1$, $\sum_{n:y_n^2\le T} y_n^2$ and $\sum_{n;y_n^2>T} 1$ have a piecewise constant dependency with respect to T. Indeed, denoting by $\tilde{\mathbf{y}}$ the vector $\mathbf{y}$ sorted in decreasing order, the previous quantities remain constant on $[\tilde{y}_n^2, \tilde{y}_{n+1}^2[$. Overall, there are only N possible values for $\hat{p}$, $\hat{\sigma_x^2}$ and $\hat{\sigma_e^2}$. Hence, one just needs to compute these N possible parameter values, estimate the corresponding $\hat{\mathbf{x}}$, and compute the related value of J in Eq. (17). Then, the parameters yielding the minimum value of J are selected. This procedure is summarized in Alg. 2.

It is worth noticing that when N tends to infinity, one recovers the expression of the parameters given in [21]. One of the main results in [21] is that the estimation Eq. (23) is necessarily biased and nonconsistent. Moreover, it turns out that this method reduces to a Classification-EM (CEM) procedure [31] applied to a mixture of two centered Gaussians, also known to produce a biased estimator.

Algorithm 2: Joint Estimation of $\mathbf{x}$ and θ

Result: $\hat{\mathbf{x}}$, $\hat{\theta} = (\hat{p}, \hat{\sigma_x^2}, \hat{\sigma_e^2})$
Input: $\mathbf{y}$
for *all* $k \in \{1, \dots, N\}$ **do**
  Compute $\theta^k = (\hat{p}, \hat{\sigma_x^2}, \hat{\sigma_e^2})$ according to Eqs. (23) to (25) with $T = y_k^2$;
  Compute $\mathbf{x}^k$ according to Eq. (19) with $T(\hat{p}, \hat{\sigma_x^2}, \hat{\sigma_e^2})$;
  Compute $J^k = J(\mathbf{x}^k, \theta^k)$;
end
Set $\kappa = \text{argmin}_k\, J^k$;
Set $\hat{\theta} = \theta^\kappa$;
Set $\hat{\mathbf{x}} = \mathbf{x}^\kappa$;

C. Estimation by the method of moments

Algs. 1 and 2 presented above are local optimization procedures. Therefore, they are sensitive to the initial θ value. Hereafter, we propose an estimator of θ based on the method of moments that will be further used to initialize Algs. 1 and 2.

Theorem 1. *Let $\mathbf{y} = \{y_1, \dots, y_N\}$ such that, for all n, y_n are i.i.d. distributed according to Eq. (7). Define the following quantities:*

$$m_2 = \frac{1}{N}\sum_{n=1}^N y_n^2 \quad m_4 = \frac{1}{3N}\sum_{n=1}^N y_n^4 \quad m_6 = \frac{1}{15N}\sum_{n=1}^N y_n^6$$

$$A = \frac{m_6 - m_2 m_4}{2(m_4 - m_2^2)} \quad B = \frac{m_2 m_6 - m_4^2}{m_4 - m_2^2}.$$

Then, the method of moments applied to $\theta = (p, \sigma_x^2, \sigma_e^2)$ yields the following estimator:

$$\hat{\sigma_e^2} = A - \sqrt{A^2 - B}$$

$$\hat{\sigma_x^2} = \frac{m_4 - (\hat{\sigma_e^2})^2}{m_2 - \hat{\sigma_e^2}} - 2\hat{\sigma_e^2}$$

$$\hat{p} = \frac{m_2 - \hat{\sigma_e^2}}{\hat{\sigma_x^2}}$$

Moreover, this estimator is consistent.

Proof. The derivation of the method of moments is given in Appendix B. Consistency comes from the continuity of the functions used to build the estimator [32, Chp. 9]. □

Although the estimator is consistent, the method of moments is known to yield biased estimators. In practice, it could

happen that the conditions $m_2^2 < m_4$ and $m_4^2 < m_2 m_6$ (which imply that $B > 0$ and then $\hat{\sigma}_e^2$ given above is well-defined, see Appendix B) are not met for specific observations $\mathbf{y}$. Therefore, we apply an empirical procedure that iteratively removes one entry at a time in $\mathbf{y}$ to increase as much as possible the value of

$$\min\left\{0; m_4 - m_2^2\right\} + \min\left\{0; m_2 m_6 - m_4^2\right\}. \tag{28}$$

This process is stopped when $m_2^2 < m_4$ and $m_4^2 < m_2 m_6$.

IV. Extension to General Inverse problems

In this section, we extend the hyperparameter estimation procedure of Section III to general inverse problems (1). The case where $\mathbf{H}$ is orthonormal boils down to a denoising problem with vector $\mathbf{H}^T\mathbf{y}$ as input, since $\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 = \|\mathbf{H}^T\mathbf{y} - \mathbf{x}\|^2$.When $\mathbf{H}$ is not orthonormal, the distribution of $\mathbf{y}$ is no longer a mixture of two Gaussians. Then, the standard EM procedure cannot be applied anymore. Indeed, the computation and optimization of the likelihood function is known to be untractable, see [18].

We resort to the re-parametrization proposed in [19], involving a so-called latent variable $\mathbf{z}$. Eq. (1) rewrites:

$$\mathbf{y} = \mathbf{H}\mathbf{z} + \mathbf{b} \tag{29}$$
$$\mathbf{z} = \mathbf{x} + \mathbf{n}\ , \tag{30}$$

where $\mathbf{n} \sim \mathcal{N}(0, \sigma_n^2 \mathbf{I}_N)$ and $\mathbf{b} \sim \mathcal{N}(0, \Gamma_b)$ are independent random vectors. Since $\mathbf{e} = \mathbf{b} + \mathbf{H}\mathbf{n}$, the covariance of the "outer" noise $\mathbf{b}$ must satisfy

$$\Gamma_b = \sigma_e^2 \mathbf{I}_M - \sigma_n^2 \mathbf{H}\mathbf{H}^T \tag{31}$$

with $\sigma_n^2 < \frac{\sigma_e^2}{\|\mathbf{H}\mathbf{H}^T\|}$ to yield a non-degenerate distribution, where $\|\,.\,\|$ refers to the spectral norm of a matrix.

One can notice from (29) that the distribution of $\mathbf{y}$ given $(\mathbf{z}, \mathbf{x})$ is independent of $\mathbf{x}$. Moreover, from Bayes' rule, we have $p(\mathbf{x}|\mathbf{y}, \mathbf{z}) \propto p(\mathbf{y}|\mathbf{x}, \mathbf{z})\, p(\mathbf{x}|\mathbf{z})$ (where $\propto$ denotes proportionality). It follows that

$$p(\mathbf{y}|\mathbf{z}, \mathbf{x}) = p(\mathbf{y}|\mathbf{z}) \quad \text{and} \quad p(\mathbf{x}|\mathbf{y}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z}). \tag{32}$$

since $p(\mathbf{y}|\mathbf{x}, \mathbf{z})$ does not depends on $\mathbf{x}$. With the reparametrization (30), the parameters to be estimated are now $(p, \sigma_x^2, \sigma_e^2, \sigma_n^2)$. Hereafter, we will set

$$\sigma_n^2 = \alpha \sigma_e^2 \tag{33}$$

where α is a predefined parameter satisfying $\alpha < \frac{1}{\|\mathbf{H}\mathbf{H}^T\|}$. Therefore, the remaining free parameters to be estimated are $\theta = (p, \sigma_x^2, \sigma_n^2)$.

Because

$$p(\mathbf{z}|\mathbf{y}, \mathbf{x}, \theta) \propto p(\mathbf{y}|\mathbf{z}, \theta)\, p(\mathbf{z}|\mathbf{x}, \theta) \tag{34}$$

and both probability density functions appearing in the latter equation are Gaussian, the posterior distribution of $\mathbf{z}$ given $(\mathbf{y}, \mathbf{x})$ is a multivariate Gaussian distribution written (see *e.g.*, [32, Chap. 10])

$$\mathbf{z}|\mathbf{y}, \mathbf{x}, \theta \sim \mathcal{N}(\mu_z, \Gamma_z) \tag{35}$$

with

$$\mu_z = \mathbf{x} + \alpha \mathbf{H}^T(\mathbf{y} - \mathbf{H}\mathbf{x}) \tag{36}$$
$$\Gamma_z = \sigma_n^2(\mathbf{I} - \alpha \mathbf{H}^T\mathbf{H})\ . \tag{37}$$

Notice that the computation of μ_z reads as a gradient descent iteration on the ℓ_2-loss $\frac{1}{2}\|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2$ with step α.

We provide two Bayesian estimators. In Section IV-A, using a noninformative uniform prior on the parameters, we derive the joint MAP estimator of $(\mathbf{x}, \theta)$:

$$(\hat{\mathbf{x}}, \hat{\theta}) = \underset{\mathbf{x},\theta}{\operatorname{argmax}}\, p(\mathbf{x}, \theta|\mathbf{y})\ . \tag{38}$$

In Section IV-B, we exhibit the joint MAP estimator of $(\mathbf{z}, \theta)$:

$$(\hat{\mathbf{z}}, \hat{\theta}) = \underset{\mathbf{z},\theta}{\operatorname{argmax}}\, p(\mathbf{z}, \theta|\mathbf{y})\ . \tag{39}$$

already proposed in our preliminary conference publication [28]. In the denoising case (and by extension, when $\mathbf{H}$ is orthogonal), the former yields a biased estimator of θ (see Section III-B). The latter reduces to the ML estimator described in Section III-A. It yields a consistent estimator of θ [28].

A. Joint estimation of $(\mathbf{x}, \theta)$ using $\mathbf{z}$ as hidden variable

Adapting [19] to the BG model, we can derive the EM algorithm using $\mathbf{z}$ as a hidden variable. In this context, $\mathbf{x}$ is treated as a random variable within the probabilistic framework of the model. The EM algorithm is applied to the extended model, where $\mathbf{z}$ plays the role of the hidden variable and $(\mathbf{x}, \theta)$ are regarded as unknown parameters.

By definition, the EM algorithm aims to maximize the likelihood $p(\mathbf{y}|\mathbf{x}, \theta)p(\mathbf{x}|\theta)p(\theta)$, which implicitly relies on the distribution of $\mathbf{z}$ conditioned on $(\mathbf{y}, \mathbf{x}, \theta)$. In practice, this involves identifying the realization of $\mathbf{x}$ that maximizes the likelihood for a given realization of $\mathbf{y}$. This method aligns with the empirical Bayes framework, where point estimates are derived for certain variables while preserving a probabilistic interpretation for others.

a) E-Step: Exploiting the decoupling property in (32), the E-step of (38) reads:

$$\begin{aligned} Q(\mathbf{x}, \theta|\mathbf{x}^{(t)}, \theta^{(t)}) &= \mathbb{E}_{\mathbf{z}|\mathbf{y},\mathbf{x}^{(t)},\theta^{(t)}}\left[-\log p(\mathbf{x}, \theta, \mathbf{z}|\mathbf{y})\right] \\ &= \mathbb{E}_{\mathbf{z}|\mathbf{y},\mathbf{x}^{(t)},\theta^{(t)}}\left[-\log p(\mathbf{y}|\mathbf{z}, \theta) - \log p(\mathbf{z}, \mathbf{x}, \theta)\right] + C_1 \\ &= \mathbb{E}_{\mathbf{z}|\mathbf{y},\mathbf{x}^{(t)},\theta^{(t)}}\left[-\log p(\mathbf{y}|\mathbf{z}, \theta) - \log p(\mathbf{x}, \theta|\mathbf{z})\right] + C_2 \end{aligned} \tag{40}$$

where C_1 and C_2 do not depend on $\mathbf{x}$ nor θ.

Let us now express both terms within (40). According to Eq. (29), we have $(\mathbf{y}|\mathbf{z}, \theta) \sim \mathcal{N}(\mathbf{H}\mathbf{z}, \Gamma_b)$. Using elementary identities on the expectation of quadratic forms, we get

$$\begin{aligned} \mathbb{E}_{\mathbf{z}|\mathbf{y},\mathbf{x}^{(t)},\theta^{(t)}}\left[-\log p(\mathbf{y}|\mathbf{z}, \theta)\right] &= \frac{1}{2}\mathrm{Trace}[\Gamma_z^{(t)}\mathbf{H}^T\Gamma_b^{-1}\mathbf{H}] \\ &+ \frac{1}{2}(\mathbf{y} - \mathbf{H}\mathbf{z}^{(t)})^T\Gamma_b^{-1}(\mathbf{y} - \mathbf{H}\mathbf{z}^{(t)}) + \frac{1}{2}\log|2\pi\Gamma_b| \end{aligned} \tag{41}$$

where $\mathbf{z}^{(t)} = \mathbf{x}^{(t)} + \alpha\mathbf{H}^T(\mathbf{y} - \mathbf{H}\mathbf{x}^{(t)})$ and $\Gamma_z^{(t)} = (\sigma_n^2)^{(t)}(\mathbf{I} - \alpha\mathbf{H}^T\mathbf{H})$ are defined from Eqs. (36) and (37).

In order to derive the second term of (40), we need to express $-\log p(\mathbf{x}, \theta|\mathbf{z})$. Since $\mathbf{z}$ reads as a noisy version of

$\mathbf{x}$ (see (30)), one can use Proposition 1 in the denoising case. $-\log p(\mathbf{x},\theta|\mathbf{z})$ thus reads

$$\frac{1}{2\sigma_n^2}\|\mathbf{z}-\mathbf{x}\|_2^2+\frac{1}{2\sigma_x^2}\|\mathbf{x}\|_2^2+\lambda(\sigma_x^2,p)\|\mathbf{x}\|_0+C(\sigma_n^2,p). \quad (42)$$

It follows from Eq. (35) that

$$\mathbb{E}_{\mathbf{z}|\mathbf{y},\mathbf{x}^{(t)},\theta^{(t)}}\left[-\log p(\mathbf{x},\theta|\mathbf{z})\right]=\frac{||\mathbf{z}^{(t)}-\mathbf{x}||_2^2+\text{Trace}[\Gamma_z^{(t)}]}{2\sigma_n^2}+\frac{1}{2\sigma_x^2}\|\mathbf{x}\|_2^2+\lambda(\sigma_x^2,p)\|\mathbf{x}\|_0+C(\sigma_n^2,p). \quad (43)$$

b) M-Step: Summarizing, the criterion $Q(\mathbf{x},\theta|\mathbf{x}^{(t)},\theta^{(t)})$ reads as the sum of (41) and (43), plus a constant independent of $\mathbf{x}$ and θ. One can notice that (43) depends on both $\mathbf{x}$ and θ, whereas (41) solely depends on σ_e^2. Indeed, all vectors and matrices within (41) do not depend on θ, except for $\Gamma_b=\sigma_e^2\left(\mathbf{I}-\alpha\mathbf{H}\mathbf{H}^T\right)$, see Eqs. (31) and (33).

Following these dependencies and taking into account (33), the minimization scheme is as follows:

1) $(\mathbf{x}^{(t+1)},\theta^{(t+1)})=\underset{\mathbf{x},\theta}{\text{argmin}}\,\mathbb{E}_{\mathbf{z}|\mathbf{y},\mathbf{x}^{(t)},\theta^{(t)}}\left[-\log p(\mathbf{z},\mathbf{x},\theta)\right]$
2) $\left(\sigma_e^2\right)^{(t+1)}=\frac{(\sigma_n^2)^{(t+1)}}{\alpha}$

The first step is very similar to the joint denoising problem of Section III-B applied to $\mathbf{z}^{(t)}$, with the extra term $\frac{\text{Trace}[\Gamma_z^{(t)}]}{2\sigma_n^2}$. Hence, one can derive an algorithm very similar to Alg. 2 to estimate jointly $\mathbf{x}^{(t+1)}$ and $\theta^{(t+1)}$. The only difference is related to the constant ρ in Eq. (27), which must be set as follows:

$$\rho=\frac{\sum\limits_{\{n:(z_n^{(t)})^2>T\}}\left(z_n^{(t)}\right)^2}{\sum\limits_{\{n:(z_n^{(t)})^2\leq T\}}\left(z_n^{(t)}\right)^2+\text{Trace}[\Gamma_z^{(t)}]}. \quad (44)$$

The whole EM procedure is summarized in Alg. 3, where parameter α is set to the limit value $\frac{1}{\|\mathbf{H}\mathbf{H}^T\|}$.

Algorithm 3: Joint estimation of $(\mathbf{x},\theta)$

Result: $\hat{\mathbf{x}}$, $\hat{\theta}$
Input: $t=0$, $\mathbf{z}^{(t)}=\mathbf{0}$, $\mathbf{x}^{(t)}=\mathbf{0}$, $\alpha=\frac{1}{\|\mathbf{H}\mathbf{H}^T\|}$
while *Not converged* **do**
  $\mathbf{z}^{(t)}=\mathbf{x}^{(t)}+\alpha\mathbf{H}^T(\mathbf{y}-\mathbf{H}\mathbf{x}^{(t)})$;
  $\Gamma_z^{(t)}=\sigma_n^{2\,(t)}(\mathbf{I}-\alpha\mathbf{H}^T\mathbf{H})$;
  Estimate $\mathbf{x}^{(t)}$ and $\theta^{(t)}$ by applying Alg. 2 with $\mathbf{y}=\mathbf{z}^{(t)}$ and the modified expression (44) for ρ;
  $\left(\sigma_e^2\right)^{(t+1)}=\frac{(\sigma_n^2)^{(t+1)}}{\alpha}$;
  $t=t+1$;
end

B. Joint estimation of $\mathbf{z},\theta$ using $\mathbf{x}$ as hidden variable

We first briefly summarize the procedure described in our conference paper [28]. We use $\mathbf{z}$ as the variable to be estimated and consider $\mathbf{x}$ as the hidden variable that is marginalized out. Using a uniform prior on θ and exploiting the property (32), the E-step reads

$$\mathbb{E}_{\mathbf{x}|\mathbf{y},\mathbf{z}^{(t)},\theta^{(t)}}\left[-\log p(\mathbf{x},\mathbf{z},\theta|\mathbf{y})\right]=-\log p(\mathbf{y}|\mathbf{z})-\mathbb{E}_{\mathbf{x}|\mathbf{z}^{(t)},\theta^{(t)}}\left[\log p(\mathbf{z},\mathbf{x}|\theta)\right]+C \quad (45)$$

where C does not depend on $\mathbf{z}$ nor θ. Similar to Section IV-A, we set $\sigma_n^2=\alpha\sigma_e^2$. The resulting EM algorithm alternates between updates of θ and $\mathbf{z}$, as detailed in [28]. At iteration t, $\mathbf{z}$ is estimated according to

$$\mathbf{z}^{(t+1)}=\mathbf{x}^{(t)}+\alpha\mathbf{H}^T(\mathbf{y}-\mathbf{H}\mathbf{x}^{(t)}). \quad (46)$$

Then, the update of θ reduces to the ML estimate of the denoising problem in $\mathbf{z}^{(t)}$ discussed in Section III-A.

The EM procedure is given in Alg. 4, which provides estimates of the model parameters p and σ_x^2 as well as the noise variance $\sigma_n^2=\alpha\sigma_e^2$. By construction, the algorithm also provides an estimate for $\mathbf{x}$, which is the posterior mean of the denoising problem for $\mathbf{z}$.

Algorithm 4: Joint estimation of $(\mathbf{z},\theta)$

Result: $\hat{\mathbf{z}}$, $\hat{\theta}$
Input: $t=0$, $\mathbf{z}^{(t)}=\mathbf{0}$, $\mathbf{x}^{(t)}=\mathbf{0}$, $\alpha=\frac{1}{\|\mathbf{H}\mathbf{H}^T\|}$
while *not converged* **do**
  $\mathbf{z}^{(t+1)}=\mathbf{x}^{(t)}+\alpha\mathbf{H}^T\left(\mathbf{y}-\mathbf{H}\mathbf{x}^{(t)}\right)$;
  Estimate $\theta^{(t+1)}$ using Alg. 1 with $\mathbf{y}=\mathbf{z}^{(t+1)}$;
  Estimate $\mathbf{x}^{(t+1)}$ using Eq. (13) with $\mathbf{y}=\mathbf{z}^{(t+1)}$;
  $t=t+1$;
end

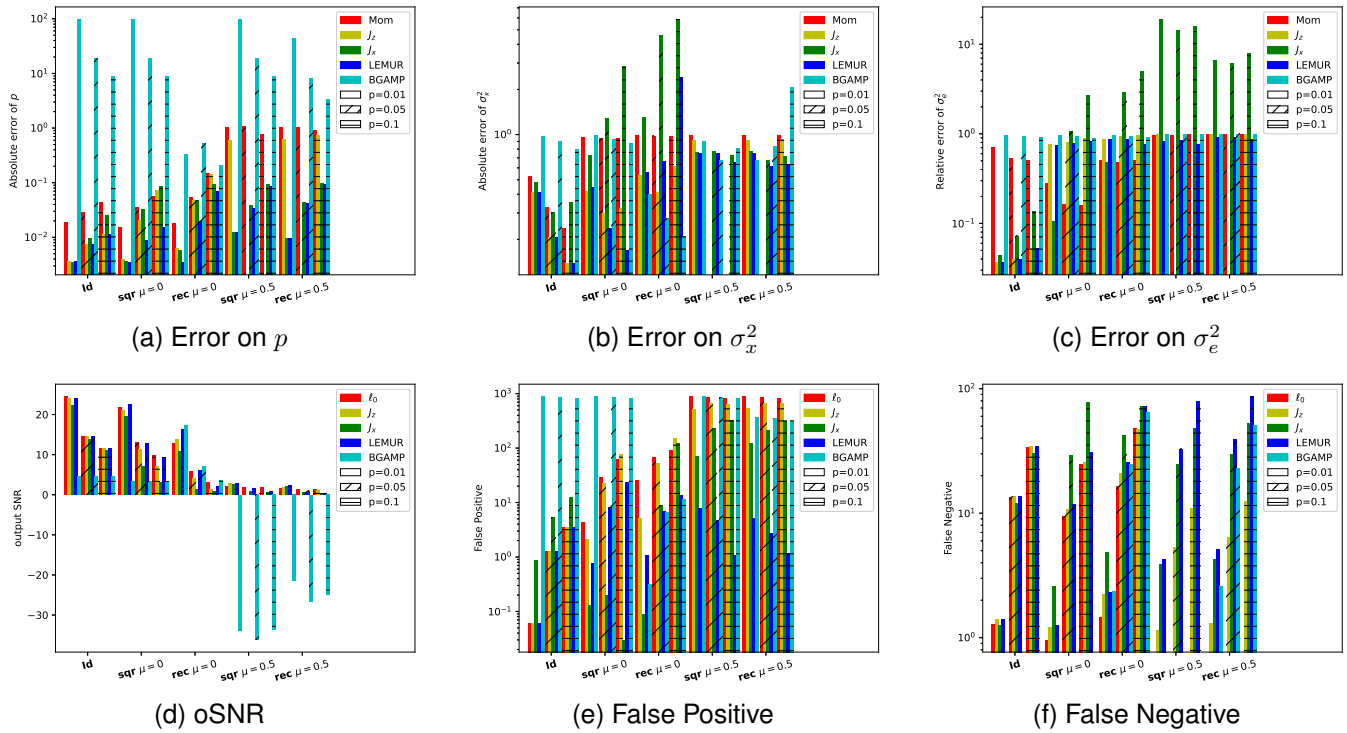
The EM that uses $\mathbf{z}$ as a hidden variable is proved to be asymptotically biased [33] when it comes to parameter estimation but appears to be stable in practice, that is, not very sensitive to initial settings. Conversely, the EM algorithm that uses $\mathbf{x}$ as a hidden variable is unbiased at the optimum but sensitive to initialization. Therefore, we propose the complete **Latent EM Unsupervised Regression (LEMUR)** procedure, where Alg. 4 is initialized by Alg. 3.

V. Experimental Study

This section assesses the proposed methods for parameter estimation and sparse signal recovery. To do so, we generate synthetic BG signals $\mathbf{x}$ following model Eq. (2) for varying degrees of sparsity controlled by the Bernoulli parameter $p\in\{0.01,0.05,0.1\}$. The variance of the non-zero coefficients is set to $\sigma_x^2=1$. We then generate several observations $\mathbf{y}$ following the direct model (1) using various matrix operators $\mathbf{H}$. In addition to the simple denoising model, where $\mathbf{H}=\mathbf{I}_M$ with $M=900$, we generate correlated Gaussian random matrices of size 900×900 and 600×900. The covariance matrix used for generating the columns of $\mathbf{H}$ is the identity, and the mean vector is constant such that

$$h_m\sim\mathcal{N}(\mu\,\mathbf{1}_M,\mathbf{I}) \quad (47)$$

where $\mathbf{1}_M$ denotes the M dimensional vector of 1 and $\mu\geq0$. We also performed some experiments with $\mu=0$ and using a covariance matrix $\mathbf{C}$ such that $C_{ij}=w^{|i-j|}$ with $w\in(0,1)$.



(a) Error on p (b) Error on σ_x^2 (c) Error on σ_e^2

(d) oSNR (e) False Positive (f) False Negative

Fig. 1. iSNR = 5dB. Top: Parameter estimation ($\text{error}_p, \text{error}_{\sigma_x^2}, \text{error}_{\sigma_e^2}$ as defined in Eq. (49) and Eq. (50)), Bottom: output SNR as defined in Eq. (51), False Negative and False positive, Eq. (52)

However, it turned out that the parameter μ has much more influence on the results than the parameter w. Hence, we display the results with a fixed covariance matrix $\mathbf{C} = \mathbf{I}$. When $\mu = 0$ $\mathbf{H}$ reduces to an i.i.d. random matrix as found in compressive sensing. The matrices $\mathbf{H}$ are then normalized to have unit norm columns. For each $\mathbf{y}$, we generate a white Gaussian noise vector with variance

$$\sigma_e^2 = ||\mathbf{Hx}|| \times 10^{-\text{iSNR}/10} \tag{48}$$

where iSNR stands for the input Signal-to-Noise Ratio expressed in dB.

We monitor the estimation of both parameters $(p, \sigma_e^2, \sigma_x^2)$ and sources $\mathbf{x}$. Denoting by $(\hat{p}, \hat{\sigma}_e^2, \hat{\sigma}_x^2)$ the estimated hyperparameters, the metric used to quantify the quality of parameter estimation is the absolute error for $p \in \{0.01, 0.05, 0.1\}$ and $\sigma_x^2 = 1$:

$$\text{error}_p = |p - \hat{p}| \qquad \text{error}_{\sigma_x^2} = |\sigma_x^2 - \hat{\sigma}_x^2| \ . \tag{49}$$

The value of σ_e^2 being fixed to reach the target iSNR, as it depends directly on all the parameters as well as the matrix $\mathbf{H}$, we define the relative error:

$$\text{error}_{\sigma_e^2} = \frac{|\sigma_e^2 - \hat{\sigma}_e^2|}{\sigma_e^2} \ . \tag{50}$$

The results are averaged over 100 realizations of $(\mathbf{x}, \mathbf{e})$ for each setup $(\mathbf{H}, p, \text{iSNR})$.

Let us denote by $\hat{\mathbf{x}}$ an estimate of the original signal $\mathbf{x}$. To quantify the quality of source recovery, we define the output Signal-to-Noise Ratio (oSNR):

$$\text{oSNR} = 10 \log_{10} \left(\frac{||\mathbf{x}||_2^2}{||\mathbf{x} - \hat{\mathbf{x}}||_2^2} \right) \tag{51}$$

We also count the number of False positives (Fp) and False negatives (Fn) in the support of $\hat{\mathbf{x}}$, that is

$$\text{Fp} = \#\{\hat{x}_n \neq 0 | x_n = 0\}, \quad \text{Fn} = \#\{\hat{x}_n = 0 | x_n \neq 0\}. \tag{52}$$

The algorithms are all initialized, when needed, using the method of moments described in Section III-C, with $\mathbf{H}^T\mathbf{y}$ as input. The initial choice of the estimated sources is the zero vector. The output of Alg. 4 being a posterior mean estimate, the recovered signal $\mathbf{z}$ is not sparse. Hence, we choose to estimate the support of the signal by considering the coordinates such that $p(x_n \neq 0 | z_n)$ (given by Eq. (8) with $y_n \leftarrow z_n$) is greater than $\frac{1}{2}$.

The algorithms are designated with shorthand notations. Alg. 3 will be referred to as J_x for joint estimation of $\mathbf{x}$ and θ. Similarly, Alg. 4 is referred to as J_z. We consider the IHT algorithm [6] dedicated to the minimization of the cost function $\frac{1}{2}\|\mathbf{y} - \mathbf{Hx}\|_2^2 + \lambda \|\mathbf{x}\|_0$ which can be thought of as a limit form of the cost function appearing in Proposition 1 when σ_x^2 tends to infinity. The latter is a supervised algorithm where λ is set empirically. IHT is run repeatedly for 100 decreasing λ-values defined on a logarithmic grid with warm-restart. We keep the IHT output yielding the best SNR score, as often done in practice. The results are referred to as ℓ_0 in the Figures. The Bernoulli-Gaussian Approximate Massage Passing (BGAMP) algorithm [26] is also used for comparison as a state-of-the-art unsupervised estimation algorithm of BG signals. In addition, we analyze the Sparse Bayesian Learning (SBL) model, often used for sparse signals. While it differs from a BG model by not explicitly employing a Bernoulli-Gaussian prior, sparsity is achieved through the iterative adjustment of hyperparameters

representing the variance of each signal component. The corresponding algorithm used here is EM CHAMPAGNE [34], with an estimated σ_e^2 initialized by the true value. The last considered algorithm is LEMUR, in which Alg. 4 is initialized using the output of Alg. 3, as described in Section IV-B.

The computational complexity of LEMUR is similar to that of iterative shrinkage/thresholding methods, such as IHT, with an additional cost from the EM algorithm. This cost corresponds to the well-established and efficient case of estimating the parameters of a two-Gaussian mixture. The experiments are run with Python on a laptop with an Intel Core i7 CPU at 1.8 GHz with 16 GB of RAM[1]. BGGAMP results have been obtained using the MATLAB implementation available at https://sourceforge.net/projects/gampmatlab/.

Figs. 1 and 2 display the performance of algorithms in terms of parameter and signal recovery for specific input signal-to-noise ratios (iSNR) of 5 and 10 dB, respectively. The top part of each figure illustrates the absolute error on parameters p and σ_x^2, and the relative error on σ_e^2. The bottom part of the figures showcases metrics related to signal recovery, that is, the output SNR and the number of false negatives and false positives. This evaluation is repeated for different values of p and matrix operators $\mathbf{H}$, including the identity matrix and random matrices with varying coefficients μ (see Eq. (47)). For parameter estimation, the proposed EM algorithms are compared against the methods of moments (in red). Regarding signal recovery, the three algorithms J_x, J_z, and LEMUR are compared against the outcomes of the supervised IHT algorithm (in red). The behavior remains consistent across varying iSNRs. For clarity, we will focus our analysis primarily on Fig. 2 corresponding to an iSNR of 10dB.

In addition, Fig. 3 focuses on the quality of signal and its support estimation, using the (SNR, Fp, Fn) metrics, with the same modalities as in previous figures, but with a thinner variation on μ. It also contains the results of the EM CHAMPAGNE for the SBL model.

A. Parameter estimation

We first evaluate the quality of BG parameter estimation, as displayed in Figs. 2a to 2c.

Regarding the recovery of parameter p, in the denoising scenario, J_x, J_z, and LEMUR give the lowest errors. However, as soon as μ increases, J_z produces more errors while J_x and LEMUR remain consistently better. It has been observed that LEMUR performs better in estimating σ_x^2 for low values of p and large μ. On the other hand, in the compressed sensing scenario with $\mu = 0$, J_z is found to be competitive. It is worth noting that J_z outperforms other methods in the denoising case and for low (p, μ). When focusing on σ_e^2 estimation, the method of moments exhibits strong performance. It can be explained by remembering that it is a statistical approach valid when the number of noise realisations is large. This is true here because the signals are highly sparse. On the other hand, LEMUR consistently outperforms J_z and J_x for all cases.

It is essential to highlight that the methods of moments and the J_x method are prone to generating biased estimators, as discussed in Section III, which leads to lower accuracy in estimating parameters, especially in denoising scenarios. For most scenarios and iSNR levels, LEMUR is the most robust and competitive approach for parameter estimation, even with varying sparsity levels of the actual solution.

B. Signal and support estimation

We now focus on evaluating the quality of signal and support recovery, see Figs. 2d to 2f for a variety of matrices and Fig. 3 for the scenarios with $\mathbf{H} \in \mathbb{R}^{600\times 900}$ defined in Eq. (47) with varying μ.

The oSNR metric is used to evaluate the recovery of a signal. Among all approaches, EMGAMP consistently gives the weakest performance in denoising situations. It can be explained by the necessity for this method to work with $\mathbf{H}$ with zero-mean columns. For operators that match the former condition, we can see that the quality of reconstruction by EMGAMP is not stable regarding μ, confirming its sensitivity to operators with higher mean values as stressed in [26], [35]. On the other hand, LEMUR is more robust towards this parameter and shrinks at worst to the result of classical IHT methods. Finally, the EM CHAMPAGNE (i.e., SBL) dominates all other methods regarding oSNR while also being robust to μ. However, the σ_e^2 parameter being initialized using the true value, these results must be used as an oracle. Moreover, identifying the support of a sparse signal also appears to be an important aspect that complements the signal estimation process.

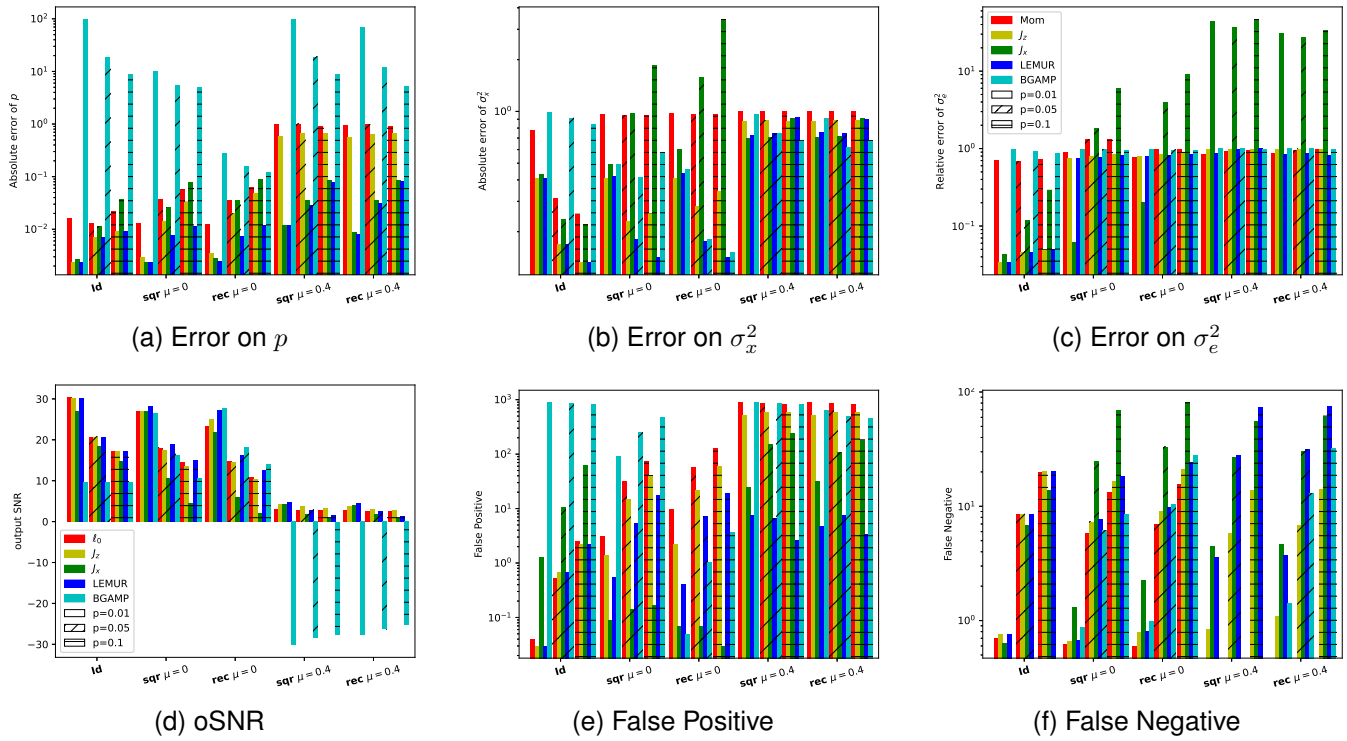
The estimation quality of the support is measured by the False Positive (where a coefficient is mistakenly considered present in the signal) and False Negative (where a true coefficient is missed) scores. Among various scenarios, LEMUR is the most effective method for False Positive rates. Indeed, just like for the output SNR, the EMGAMP method gives better results for uncorrelated dictionary matrices but yields an instability regarding μ. The same behavior is witnessed with the IHT, mainly because we choose the hyperparameter that gives the best output SNR and not necessarily the best support estimation. It can also be seen that in those scenarios, the SBL produces the highest rate of False Positive, which makes it the less sparse reconstruction. The absence of strong sparsity constraints in the SBL model explains this.

LEMUR gives the best compromise between quality of reconstruction, sparsity, and stability towards dictionaries. As for the numerical cost this method mainly contains matrix and vector products, which prevents the computational cost from scaling too much regarding the dimensions of the problem. For instance, one run of LEMUR in the previous figure took around 3 seconds, while one SBL run (with matrix inversions) took approximately 260 seconds.

VI. Conclusion

The LEMUR algorithm effectively addresses the challenges of sparse signal recovery, particularly balancing data fidelity and sparsity promotion through hyperparameter tuning. This approach extends the well-established Iterative Shrinkage/Thresholding algorithm, initially designed for

[1]Code is available a https://github.com/PBrblt/BGBox



(a) Error on p (b) Error on σ_x^2 (c) Error on σ_e^2

(d) oSNR (e) False Positive (f) False Negative

Fig. 2. iSNR = 10dB. Top: Parameter estimation (error_p, $\text{error}_{\sigma_x^2}$ and $\text{error}_{\sigma_e^2}$), Bottom: output SNR, False Negatives and False positives.

Lasso/BPDN and related variants, to an unsupervised setting. Focusing on parameter estimation within the Bernoulli-Gaussian model, the methodology leverages maximum likelihood estimation for denoising tasks, further expanding these techniques to general inverse problems through latent variable formulations.

We proposed two distinct expectation-maximization approaches to address signal and BG model parameter estimation. While one approach displays robustness to initialization but shows bias even in simple denoising scenarios, the second approach suffers from sensitivity to initialization. LEMUR leverages both EM strategies to yield stable estimates of both hyperparameters and signals, showcasing superior performance in output SNR and False Positive Rate recovery across various scenarios.

Notably, LEMUR operates in a fully unsupervised manner, positioning it favorably compared to the optimally (empirically) tuned IHT algorithm. Moreover, LEMUR relies on the iterative shrinkage/thresholding algorithm, ensuring computational efficiency without imposing the prohibitive demands of MCMC methods.

Appendix A
Proof of Proposition 1

Using the indicator variable $\mathbf{s}$, and after observing that $p(\mathbf{y}|\mathbf{x},\mathbf{s},\theta) = p(\mathbf{y}|\mathbf{x},\theta)$, we can define the joint-MAP estimator of $(\mathbf{x},\mathbf{s})$ as

$$\begin{aligned}(\hat{\mathbf{x}},\hat{\mathbf{s}}) &= \underset{\mathbf{x},\mathbf{s}}{\operatorname{argmax}}\, p(\mathbf{x},\mathbf{s}|\mathbf{y},\theta) \\ &= \underset{\mathbf{x},\mathbf{s}}{\operatorname{argmax}}\, p(\mathbf{y}|\mathbf{x},\theta)\, p(\mathbf{x},\mathbf{s}|\theta). \end{aligned} \tag{A.1}$$

With the fact that $p(x_n \neq 0|s_n = 1) = 1$ and $p(x_n = 0|s_n = 0) = 1$, we can write

$$p(\mathbf{x},\mathbf{s}|\theta) = \prod_{n=1}^{N} p(x_n, s_n|\theta) \tag{A.2}$$

$$= \prod_{\{n:s_n=0\}} p(x_n = 0|s_n = 0,\theta)\, p(s_n = 0|\theta) \times \prod_{\{n:s_n=1\}} p(x_n|s_n = 1,\theta)\, p(s_n = 1|\theta) \tag{A.3}$$

$$= \prod_{\{n:s_n=0\}} (1-p) \prod_{\{n:s_n=1\}} \left(\frac{p}{\sqrt{2\pi\sigma_x^2}}\, e^{-\frac{x_n^2}{2\sigma_x^2}} \right) \tag{A.4}$$

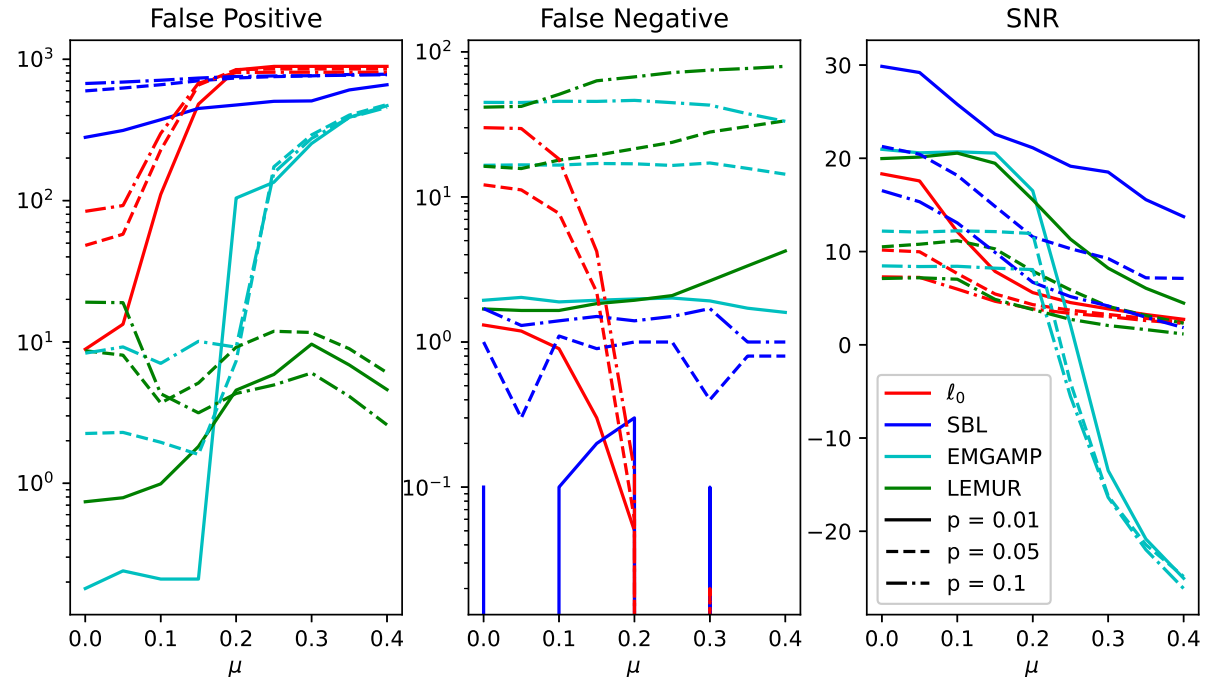
$$= \prod_{n=1}^{N} \left(\frac{1}{\sqrt{2\pi\sigma_x^2}} e^{-\frac{x_n^2}{2\sigma_x^2}} \right)^{s_n} p^{s_n}(1-p)^{1-s_n} \tag{A.5}$$

Then, using the fact that for all n, $s_n x_n = x_n$ and $\sum_n s_n = \|\mathbf{x}\|_0$, we can write

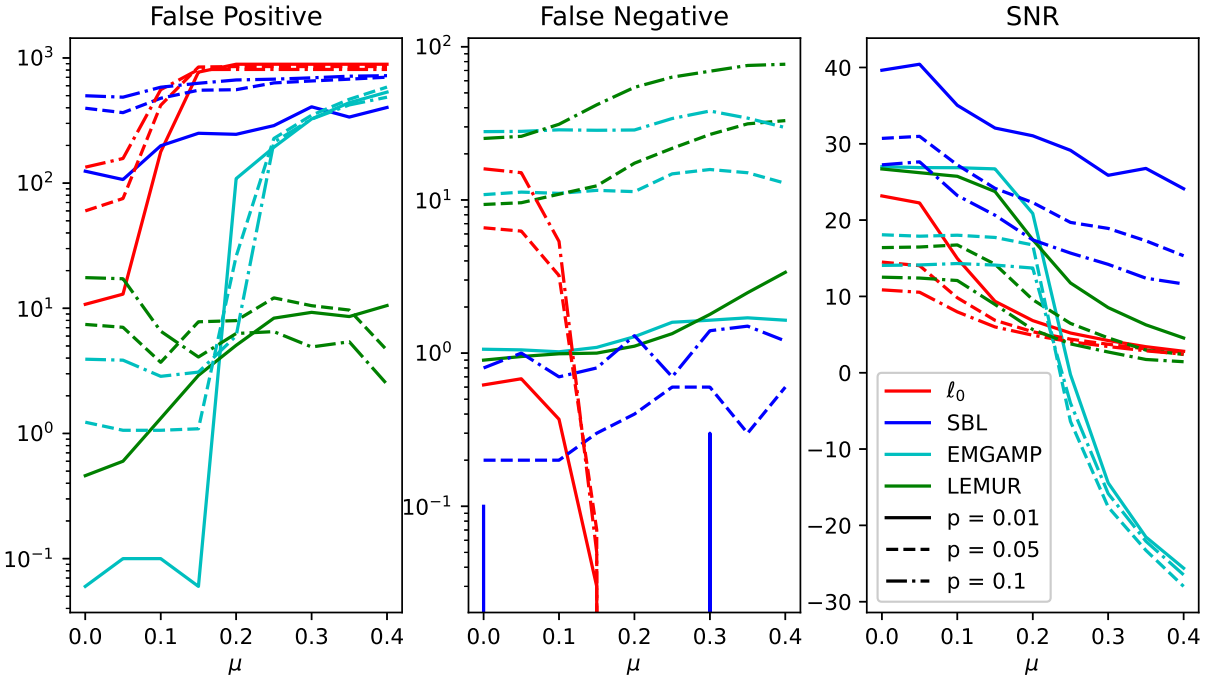
$$-\log p(\mathbf{x},\mathbf{s},\theta) = \frac{1}{2\sigma_x^2}\|\mathbf{x}\|^2 + \log\left(\sqrt{2\pi\sigma_x^2}\, \frac{1-p}{p} \right) \|\mathbf{x}\|_0 - N\log(1-p) \tag{A.6}$$

which does not depend on $\mathbf{s}$ anymore. The conclusion follows from Eq. (A.1) using

$$-\log p(\mathbf{y}|\mathbf{x},\theta) = \frac{1}{2\sigma_e^2}\|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + N\log(\sqrt{2\pi\sigma_e^2})\ .$$



(a) Input SNR 5dB



(b) Input SNR 10dB

Fig. 3. Evolution of False Positive / False Negative estimation of the support and the SNR with respect to the mean parameter μ of the matrix operator $\mathbf{H}$ for two levels of iSNR.

APPENDIX B
PROOF OF THEOREM 1

The first three non-zero moments of the mixture of two centered Gaussians, see (7), are given by

$$\mathbb{E}\{y_n^2\} = p\sigma_x^2 + \sigma_e^2 \tag{B.7}$$
$$\mathbb{E}\{y_n^4\} = 3p(\sigma_x^2 + \sigma_e^2)^2 + 3(1-p)(\sigma_e^2)^2 \tag{B.8}$$
$$\mathbb{E}\{y_n^6\} = 15p(\sigma_x^2 + \sigma_e^2)^3 + 15(1-p)(\sigma_e^2)^3. \tag{B.9}$$

Let us define the empirial moment estimators as follows:

$$m_2 = \frac{1}{N}\sum_{n=1}^{N} y_n^2 \ , \ m_4 = \frac{1}{3N}\sum_{n=1}^{N} y_n^4 \ , \ m_6 = \frac{1}{15N}\sum_{n=1}^{N} y_n^6$$

Denoting by $\hat{p}, \hat{\sigma_e^2}, \hat{\sigma_x^2}$ the estimates of $p, \sigma_e^2, \sigma_x^2$ given by the method of moments, we have:

$$m_2 = \hat{p}(\hat{\sigma_x^2} + \hat{\sigma_e^2}) + (1-p)\sigma_e^2 \tag{B.10a}$$
$$= \hat{p}\hat{\sigma_x^2} + \hat{\sigma_e^2} \tag{B.10b}$$

$$m_4 = \hat{p}(\hat{\sigma_x^2} + \hat{\sigma_e^2})^2 + (1-\hat{p})(\hat{\sigma_e^2})^2 \tag{B.11a}$$
$$= \hat{p}\hat{\sigma_x^2}(\hat{\sigma_x^2} + 2\hat{\sigma_e^2}) + (\hat{\sigma_e^2})^2 \tag{B.11b}$$

$$m_6 = \hat{p}(\hat{\sigma_x^2} + \hat{\sigma_e^2})^3 + (1-\hat{p})(\hat{\sigma_e^2})^3 \tag{B.12}$$

Hereafter, we will use the simplified notations $p, \sigma_e^2, \sigma_x^2$ instead of $\hat{p}$, $\hat{\sigma_e^2}$, $\hat{\sigma_x^2}$ for improved readability. Eq. (B.10b) yields:

$$p = \frac{m_2 - \sigma_e^2}{\sigma_x^2} \ . \tag{B.13}$$

Since $0 < p < 1$, one can check that the following inequalities are satisfied:

$$\sigma_e^2 < m_2 \tag{B.14}$$
$$m_2 - \sigma_e^2 < \sigma_x^2 \tag{B.15}$$
$$m_4^2 < m_2 m_6. \tag{B.16}$$

The latter is obtained from Eq. (B.10a), Eq. (B.11a) and Eq. (B.12):

$$m_4^2 - m_2 m_6 = p\,(p-1)\,\sigma_e^2\,(\sigma_x^2 + \sigma_e^2)\,(\sigma_x^2)^2 < 0. \tag{B.17}$$

Plugging Eq. (B.13) in Eq. (B.11b), one has

$$m_4 = (m_2 - \sigma_e^2)(\sigma_x^2 + 2\sigma_e^2) + (\sigma_e^2)^2 \tag{B.18}$$

hence

$$\sigma_x^2 = \frac{m_4 - (\sigma_e^2)^2}{m_2 - \sigma_e^2} - 2\sigma_e^2 \ . \tag{B.19}$$

Also, combining Eq. (B.10b) with Eq. (B.11b), we get $m_2^2 - m_4 = \sigma_x^2\,(p^2 - p) < 0$, so

$$m_2^2 < m_4. \tag{B.20}$$

Using Eqs. (B.13) and (B.19) within Eq. (B.12), and after some tiedous algebra, we get

$$(m_4 - m_2^2)(\sigma_e^2)^2 + (m_2 m_4 - m_6)\sigma_e^2 + (m_2 m_6 - {m_4}^2) = 0 \tag{B.21}$$

Hence, we obtain the following quadratic equation in σ_e^2:

$$\varphi(\sigma_e^2) := (\sigma_e^2)^2 + \frac{m_2 m_4 - m_6}{m_4 - {m_2}^2}\sigma_e^2 + \frac{m_2 m_6 - {m_4}^2}{m_4 - {m_2}^2} = 0. \tag{B.22}$$

According to (B.14), we need to have one root of φ in between 0 and m_2. On the one hand,

$$\varphi(\sigma_e^2 = m_2) = {m_2}^2 + \frac{{m_2}^2 m_4 - m_2 m_6}{m_4 - {m_2}^2} + \frac{m_2 m_6 - {m_4}^2}{m_4 - {m_2}^2}$$
$$= {m_2}^2 - m_4 < 0 \tag{B.23}$$

using Eq. (B.20). On the other hand,

$$\varphi(0) = \frac{m_2 m_6 - {m_4}^2}{m_4 - {m_2}^2} > 0 \tag{B.24}$$

using Eq. (B.16). It follows that $\varphi(\sigma_e^2) = 0$ admits one root in $(0, m_2)$.

Since $m_4 - {m_2}^2 > 0$, Descartes's rule of signs ensures that the weight of σ_e^2 within Eq. (B.22) is negative. Using Eq. (B.16), we get $m_2 m_4 - m_6 < 0$. Then, the root in $(0, m_2)$ is given by

$$\sigma_e^2 = A - \sqrt{A^2 - B} \tag{B.25}$$

with

$$A = \frac{m_6 - m_2 m_4}{2(m_4 - {m_2}^2)} \text{ and } B = \frac{m_2 m_6 - {m_4}^2}{m_4 - {m_2}^2} \ . \tag{B.26}$$

REFERENCES

[1] M. Elad, P. Milanfar, and R. Rubinstein, "Analysis versus synthesis in signal priors," *Inverse problems*, vol. 23, no. 3, p. 947, 2007.

[2] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B*, vol. 58, no. 1, pp. 267–288, 1996.

[3] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM review*, vol. 43, no. 1, pp. 129–159, 2001.

[4] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *SIAM Journal on Multiscale Modeling and Simulation*, vol. 4, no. 4, pp. 1168–1200, 2005.

[5] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Science*, vol. 2, no. 1, pp. 183–202, 2009.

[6] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Appl. Comp. Harmonic Anal.*, vol. 27, no. 3, pp. 265–274, 2009.

[7] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. 27th Asilomar Conf. on Signals, Systems and Computers*, vol. 1, Nov. 1993, pp. 40–44.

[8] C. Soussen, J. Idier, D. Brie, and J. Duan, "From Bernoulli-Gaussian deconvolution to sparse signal restoration," *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 4572–4584, 2011.

[9] S. Bourguignon, J. Ninin, H. Carfantan, and M. Mongeau, "Exact sparse approximation problems via mixed-integer programming: Formulations and computational performance," *IEEE Transactions on Signal Processing*, vol. 64, no. 6, pp. 1405–1419, 2015.

[10] H. Hazimeh, R. Mazumder, and A. Saab, "Sparse regression at scale: Branch-and-bound rooted in first-order optimization," *Mathematical Programming*, vol. 196, no. 1-2, pp. 347–388, 2022.

[11] D. L. Donoho and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *Journal of the american statistical association*, vol. 90, no. 432, pp. 1200–1224, 1995.

[12] T. Blu and F. Luisier, "The SURE-LET approach to image denoising," *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2778–2786, 2007.

[13] C.-A. Deledalle, S. Vaiter, J. Fadili, and G. Peyré, "Stein unbiased gradient estimator of the risk (sugar) for multiple parameter selection," *SIAM Journal on Imaging Sciences*, vol. 7, no. 4, pp. 2448–2487, 2014.

[14] R. Giryes, M. Elad, and Y. C. Eldar, "The projected GSURE for automatic parameter tuning in iterative shrinkage methods," *Applied and Computational Harmonic Analysis*, vol. 30, no. 3, pp. 407–422, 2011.

[15] C.-A. Deledalle, G. Peyré, and J. Fadili, "Stein consistent risk estimator (score) for hard thresholding," *arXiv preprint arXiv:1301.5874*, 2013.

[16] M. Pereyra, J. M. Bioucas-Dias, and M. A. Figueiredo, "Maximum-a-posteriori estimation with unknown regularisation parameters," in *23rd European Signal Processing Conference (EUSIPCO)*, 2015, pp. 230–234.

[17] D. Calvetti, E. Somersalo, and A. Strang, "Hierachical Bayesian models and sparsity: ℓ_2-magic," *Inverse Problems*, vol. 35, no. 3, p. 035003, 2019.

[18] F. Champagnat, Y. Goussard, and J. Idier, "Unsupervised deconvolution of sparse spike trains using stochastic approximation," *IEEE Transactions on Signal Processing*, vol. 44, no. 12, pp. 2988–2998, Dec. 1996.

[19] M. A. Figueiredo and R. D. Nowak, "An EM algorithm for wavelet-based image restoration," *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 906–916, 2003.

[20] M. Kowalski and T. Rodet, "An unsupervised algorithm for hybrid/morphological signal decomposition," in *Proc. IEEE ICASSP*, 2011, pp. 4112–4115.

[21] E. Gassiat, F. Monfront, and Y. Goussard, "On simultaneous signal estimation and parameter identification using a generalized likelihood approach," *IEEE transactions on information theory*, vol. 38, no. 1, pp. 157–162, 1992.

[22] D. Ge, J. Idier, and E. Le Carpentier, "A new MCMC algorithm for blind Bernoulli-Gaussian deconvolution," in *Proc. 16th Eur. Sig. Proc. Conf.*, 2008, pp. 1–5.

[23] M. Amrouche, H. Carfantan, and J. Idier, "Efficient sampling of bernoulli-gaussian-mixtures for sparse signal restoration," *IEEE Transactions on Signal Processing*, vol. 70, pp. 5578–5591, 2022.

[24] L. Chaari, J.-Y. Tourneret, and C. Chaux, "Sparse signal recovery using a Bernoulli generalized Gaussian prior," in *23rd European Signal Processing Conference (EUSIPCO)*. IEEE, 2015, pp. 1711–1715.

[25] J. P. Vila and P. Schniter, "Expectation-maximization gaussian-mixture approximate message passing," *IEEE Transactions on Signal Processing*, vol. 61, no. 19, pp. 4658–4672, 2013.

[26] J. Vila and P. Schniter, "Expectation-maximization bernoulli-gaussian approximate message passing," in *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*. IEEE, 2011, pp. 799–803.

[27] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," in *2011 IEEE International Symposium on Information Theory Proceedings*. IEEE, 2011, pp. 2168–2172.

[28] P. Barbault, M. Kowalski, and C. Soussen, "Parameter estimation in sparse inverse problems using Bernoulli-Gaussian prior," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 5413–5417.

[29] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society: Series B*, vol. 39, no. 1, pp. 1–22, 1977.

[30] M. Protter, I. Yavneh, and M. Elad, "Closed-form MMSE estimation for signal denoising under sparse representation modeling over a unitary dictionary," *IEEE Transaction on Signal Processing*, vol. 58, no. 7, pp. 3471–3484, 2010.

[31] G. Celeux and G. Govaert, "A classification EM algorithm for clustering and two stochastic versions," *Computational statistics & Data analysis*, vol. 14, no. 3, pp. 315–332, 1992.

[32] S. M. Kay, *Fundamentals of statistical signal processing: estimation theory*. Prentice-Hall, Inc., 1993.

[33] Y. Goussard, G. Demoment, and J. Idier, "A new algorithm for iterative deconvolution of sparse spike trains," in *Proc. IEEE ICASSP*, 1990, pp. 1547–1550.

[34] A. Hashemi, C. Cai, G. Kutyniok, K.-R. Müller, S. S. Nagarajan, and S. Haufe, "Unification of sparse bayesian learning algorithms for electromagnetic brain imaging with the majorization minimization framework," *NeuroImage*, vol. 239, p. 118309, 2021.

[35] M. Pereyra, P. Schniter, E. Chouzenoux, J.-C. Pesquet, J.-Y. Tourneret, A. O. Hero, and S. McLaughlin, "A survey of stochastic simulation and optimization methods in signal processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 2, pp. 224–241, 2015.