A measure-theoretic variational Bayesian algorithm for large dimensional problems

Aurélia Fraysse, Thomas Rodet

A MEASURE-THEORETIC VARIATIONAL BAYESIAN ALGORITHM FOR LARGE DIMENSIONAL PROBLEMS.

A. FRAYSSE AND T. RODET*

Abstract. In this paper we provide an algorithm allowing to solve the variational Bayesian issue as a functional optimization problem. The main contribution of this paper is to transpose a classical iterative algorithm of optimization in the metric space of probability densities involved in the Bayesian methodology. The main advantage of this methodology is that it allows to address large dimensional inverse problems by unsupervised algorithms. The interest of our algorithm is enhanced by its application to large dimensional linear inverse problems involving sparse objects. Finally, we provide simulation results. First we show the good numerical performances of our method by comparing it with classical ones on a small tomographic problem. On a second time we treat a large dimensional dictionary learning problem and compare our method with a wavelet based one.

keywords: ill-posed inverse problems, variational bayesian methodology, sparse signal reconstruction, infinite dimensional convex optimization

1. Introduction. The recent development of information technologies has increased the expansion of inverse problems for very large dimensional datasets. Indeed whereas the 90's decade have seen the introduction of image reconstruction problems, the current main interest is on 3D sequences (3D + T), thus on large dimensional sets of data. There is therefore a significant growth in the number of measurements in the involved problems. One has frequently to treat the reconstruction of more than one million data. At the same time, the signal processing techniques have helped to overcome the limitations of measuring instruments as they supplied the design of systems involving indirect measures. These new equipments introduced in exchange novel signal processing challenges, such as super resolution deconvolution, source separation or tomographic reconstruction. All these problems are ill posed, the only information contained in the data and in the model of acquisition are not sufficient to obtain a good estimation of the unknown objects.

To solve these ill-posed problems, we introduce an *a priori* information on the data. The Bayesian approach appearing in this paper consists in a modelisation of source of information as probability density functions [7, 19, 12]. This approach allows the development of unsupervised methods, such that the parameters of probability distributions (mean, variance, etc. ..), also called hyperparameters, are adjusted automatically. These hyperparameters can tune the trade-off between information coming from data (likelihood) and *a priori* information. We call these methods "fully Bayesian" as they consist in a construction of a posterior distribution from the likelihood and from the prior information thanks to the Bayes rule. In general, this posterior distribution is known up to a constant of proportionality K. In order to determine an estimation of the unknown object, the posterior law is summed up by a point: generally, the maximum *a posteriori* (MAP) or the posterior mean (PM) are chosen. The maximization of the posterior law leads to an non convex optimization issue. In the posterior mean case, we must calculate an integral. When the constant K is unknown, we cannot determine analytically this solution. Therefore, a classical way to determine this posterior mean is to perform an empirical mean of sample under

*L2S, SUPELEC, CNRS, University Paris-Sud, 3 rue Joliot-Curie, 91190 Gif-Sur-Yvette, FRANCE. email:{firstname.lastname}@lss.supelec.fr

the posterior law thanks to stochastic Markov Chain Monte Carlos (MCMC) [30, 31]. In the MCMC principle a Markov chain is generated which converges asymptotically to a sample of the desired distribution. After a sufficient time, the so called burning time, one retains only the samples of the Markov chain close to the posterior distribution. There are two main types of MCMC methods, algorithms of Metropolis-Hastings and Gibbs sampler [31]. However for large dimensional problems involving complicated covariance matrix, MCMC methods fail to give an acceptable approximation. Furthermore these methods need a long time solely to explore the space of possibilities.

Therefore D. MacKay, inspired by statistical physics works, introduced the variational Bayesian inference as an alternative method to MCMC, see [20]. This methodology was ever since involved in computer science such as in [14] and in signal processing, for different applications such as: source separation using ICA [21, 6], Mixture Models estimation [24], hyperspectral imaging reduction [3], deconvolution [4, 2] recursive methods [35, 36]. More recently a sparse bayesian methodology using variational approach of Laplace prior was developed by Seeger [33, 34]. The main outline of this variational Bayesian methodology is to approximate the posterior distribution by a separable law. This last one is the closest to the posterior distribution in terms of Kullback-Leiber divergence. Thanks to this method, the initial inverse problem appears as a convex optimization problem in a function space. Classical variational approaches give then an analytical approximation of the posterior. However, this solution has no explicit forms. For large dimensional problems this turn out to be an important drawback.

The goal of this paper is thus to construct an iterative algorithm able to provide in a reduced computation time a close approximation of the solution of this variational problem. The main principle is to adapt a classical optimization algorithm, the gradient descent method, [27], to the space of probability distributions.

The second contribution consists in the application of the mentioned methodology to inverse problems corresponding to linear Gaussian white noise model. This model can be applied for instance for deconvolution, super-resolution, tomography, or source separation. Concerning the prior distribution, we put the emphasize on sparse information.

The sparse prior information is classically introduced by heavy-tailed distributions. These distributions can be for instance Laplace [40], Bernoulli Gaussian law [29, 10, 18], mixtures of Gaussian [37, 11], Cauchy distribution, or α-stable distributions. Recently, the Gaussian Scale Mixture class was introduced in [38] as a model of wavelet coefficients of natural images. This class of probability densities generalizes the ones previously mentioned. The main advantages of Gaussian Scale Mixture is that they can be easily written as Gaussian distribution, conditioned by an hidden variable. In our context, we modelize our sparsity information thanks to the family of Gaussian Scale Vector Mixture distribution introduced in [13]. In this case, such as Chantas *et al.* in [4], we define an extended problem where hidden variables have also to be estimated.

This article is divided as follows. In Section 2 we present the optimization algorithm in probability density space involved in the paper whereas Section 3 presents our algorithm, based on its application to the classical variational Bayesian approach. Thus, in Section 4 we apply our new method for inverse problems with a Gaussian likelihood and a prior in the Gaussian Scale Vector Mixture (GSVM) family and

present our supervised and unsupervised algorithms. In section 5, we present simulation results, first on a tomographic example where the *a priori* information promotes pulses or solutions extremely sparse and secondly on an identification problem in a very large dictionary learning context. Finally, Section 6 concludes the paper.

2. Optimization algorithm in measures space. The present section is devoted to the construction of a gradient based algorithm adapted to the probability measures space.

In this part, we assume that we stand in the measurable space $(\mathbb{R}^N, \mathcal{B}(\mathbb{R}^N))$, where $\mathcal{B}(\mathbb{R}^N)$ is the σ-field of Borel sets of $\mathbb{R}^N$. Our purpose is to construct an iterative algorithm able to provide a close approximation of the solution of the Bayesian variational method. We will see in the following that this can be seen as a maximization problem of a concave functional in a probability measures space.

Concerning the probability density functions, there are several possible representations of such objects. The first one is to consider that this space is a subset of $L^1(\mathbb{R}^N)$, that is the subset of positive integrable functions with a total mass equal to one. As $L^1(\mathbb{R}^N)$ is a Banach space, classical algorithms still holds. However, one has to pay a particular attention to the fact that the positivity of the functions together with the fixed total mass imposes additional constraints. We will see in the following why this point of view is not adapted to Bayesian variational methodology. Another point of view, see [22], is to consider this set as a subset of the space of signed Radon measures $\mathcal{M}(\mathbb{R}^N)$, that is measures that can be written $\mu = \mu^+ - \mu^-$, endowed with the norm of total variation. Once again this is a Banach space. The classical gradient descent can also be adapted in this framework, as done in [23]. However in [23], the measure obtained at each iteration is no longer a density, and cannot converge to a solution to our optimization problem.

In the following we consider as an important constraint the fact that the optimal density function is separable. Hence we rather stand in $\widetilde{\mathcal{M}}(\mathbb{R}^N) = \bigotimes_{i=1}^N \mathcal{M}(\mathbb{R})$ the cartesian product of spaces of signed measures defined on $\mathbb{R}$ endowed with norm defined by

$$\forall \mu \in \widetilde{\mathcal{M}}, \quad \|\mu\|_{TV} = \sum_{i=1}^{N} \sup_{A \in \mathcal{B}(\mathbb{R})} \int_A \mathrm{d}\mu^+(x_i) + \int_A \mathrm{d}\mu^-(x_i). \tag{2.1}$$

Note that when μ is a density measure, i.e. $\mathrm{d}\mu(\boldsymbol{x}) = q(\boldsymbol{x})\mathrm{d}\mathcal{L}(\boldsymbol{x}) = q(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$, $\mathcal{L}$ standing for the Lebesgue measure, its total variation norm coincides with the L^1 norm of its density function q.

Furthermore, a probability density is considered as an element of the closed convex set Ω, defined by

$$\Omega = \{\mu \in \widetilde{\mathcal{M}};\ \mathrm{d}\mu(\boldsymbol{x}) = \prod_{i=1}^{N} q_i(x_i)\mathrm{d}x_i,\ \text{where}\ \ q_i \in L^1(\mathbb{R})\ \text{is a function such that}\ \ q_i \geq 0\ \ \text{a.e. and}\ \int_{\mathbb{R}} \mathrm{d}\mu_i(\boldsymbol{x}) = 1\}. \tag{2.2}$$

Note that this set can also be written as the cartesian product of the Ω_i where

$$\Omega_i = \{\mu_i \in \mathcal{M}(\mathbb{R});\ \mathrm{d}\mu_i(\boldsymbol{x}) = q_i(x_i)\mathrm{d}x_i,\ \text{where}\ \ q_i \in L^1(\mathbb{R})\ \text{is a function such that}\ \ q_i \geq 0\ \ \text{a.e. and}\ \int_{\mathbb{R}} \mathrm{d}\mu_i(\boldsymbol{x}) = 1\}.$$

Our purpose is, given a concave differentiable functional $F : \widetilde{\mathcal{M}} \to \mathbb{R}$, to determine an algorithm which approximates a probability measure μ^{opt} solution of

$$\mu^{opt} = \arg\max_{\mu \in \Omega} F(\mu). \tag{2.3}$$

This problem can be seen as a constrained convex optimization problem in the infinite dimensional Banach space $(\widetilde{\mathcal{M}}, \|.\|_{TV})$.

In this framework, most results of optimization are based on the dual space, which is the space of continuous functions which go to zero at infinity, see for instance [16, 5]. In the present paper we consider a gradient-like descent algorithm on the density measures space.

Let us introduce some notations from [22]. Let $F : \widetilde{\mathcal{M}} \to \mathbb{R}$. As $\widetilde{\mathcal{M}}$ is a Banach space, one can compute the Fréchet derivative of F at $\mu \in \widetilde{\mathcal{M}}$ as the bounded linear functional $\mathrm{d}F_q(.) : \widetilde{\mathcal{M}} \to \mathbb{R}$ satisfying

$$F(\mu + \nu) - F(\mu) - \mathrm{d}F_\mu(\nu) = o(\|\nu\|), \quad \text{when } \|\nu\| \to 0.$$

In the following we will also consider the Gateaux derivative of a function:

$$\forall \nu \in \widetilde{\mathcal{M}}, \quad \partial F_q(\nu) = \lim_{t \to 0} \frac{F(q + t\nu) - F(q)}{t}.$$

In some cases, see [22], one can find a continuous bounded function $\mathrm{d}f$ such that

$$\forall \nu \in \widetilde{\mathcal{M}}, \quad \partial F_\mu(\nu) = \int_{\mathbb{R}^N} \mathrm{d}f(\mu, \boldsymbol{x}) \mathrm{d}\nu(\boldsymbol{x}). \tag{2.4}$$

Consider an auxiliary concave functional $G : \widetilde{\mathcal{M}} \to \mathbb{R}$. An important property appearing in the following is that the Fréchet differential of G is L-Lipschitz on Ω, i.e.

$$\forall (\mu_1, \mu_2) \in \Omega^2, \ \ \nu \in \widetilde{\mathcal{M}} \quad |\, dG_{\mu_1}(\nu) - \, dG_{\mu_2}(\nu)| \leq L\|\nu\|\|\mu_1 - \mu_2\|. \tag{2.5}$$

The Lipschitz differential condition of G together with its concavity implies that, see [26] for instance,

$$\forall (\mu_1, \mu_2) \in \widetilde{\mathcal{M}}, \quad 0 \geq G(\mu_1) - G(\mu_2) - \, dG_{\mu_2}(\mu_1 - \mu_2) \geq -L\|\mu_1 - \mu_2\|^2. \tag{2.6}$$

Furthermore we say that a function F is twice differentiable at $\mu \in \widetilde{\mathcal{M}}$, for every $\mu \in \widetilde{\mathcal{M}}$, if

$$\forall (\nu_1, \nu_2) \in \widetilde{\mathcal{M}}^2, \quad \mathrm{d}^2 F_\mu(\nu_1, \nu_2) = \lim_{t \to 0} \frac{\mathrm{d}F_{\mu + t\nu_1}(\nu_2) - \mathrm{d}F_\mu(\nu_2)}{t}.$$

If it exists, $\mathrm{d}^2 F$ is a bilinear application from $\widetilde{\mathcal{M}} \times \widetilde{\mathcal{M}}$ to $\mathbb{R}$. If F is a concave functional, this second-order derivative must satisfy for every $\mu \in \widetilde{\mathcal{M}}$,

$$\forall \nu \in \widetilde{\mathcal{M}}, \quad \mathrm{d}^2 F_\mu(\nu, \nu) \leq 0. \tag{2.7}$$

Our purpose is to construct an iterative algorithm providing a density at each iteration and approximating the solution of (2.3) for a certain class of functions F. The key principle of our method is given by the Radon-Nikodym theorem, see [32] for

instance. Let $k \geq 0$ be an integer and assume that $\mu^k \in \widetilde{\mathcal{M}}$ is a probability measure absolutly continuous respectively to the Lebesgue measure. We thus construct $\mu^{k+1} \in \widetilde{\mathcal{M}}$ as a measure absolutely continuous with respect to μ^k. In this case, the Radon-Nikodym theorem ensures that this measure should be written as

$$\mathrm{d}\mu^{k+1}(\boldsymbol{x}) = h_k(\boldsymbol{x})\mathrm{d}\mu^k(\boldsymbol{x}), \tag{2.8}$$

where $h_k \in L^1(\mu^k)$ is a positive function. Our aim is thus to construct a function $h_k \in L^1(\mu^k)$ such that $F(\mu^{k+1}) \geq F(\mu^k)$. Following the classical scheme given by the gradient descent method, h_k is given as a function of the Frechet derivative of F at μ^k and, according to our structure, as

$$h_k(\boldsymbol{x}) = K_k(\alpha)\exp(\alpha_k \mathrm{d}f(\mu^k, \boldsymbol{x})), \tag{2.9}$$

where $\mathrm{d}f$ is defined by (2.4) whereas $\alpha_k > 0$ is the algorithm step-size at iteration k and $K_k(\alpha)$ is the normalization constant such that $\int_{\mathbb{R}^N} \mathrm{d}\mu^{k+1}(\boldsymbol{x}) = 1$. We also impose the convention that $h_k(\boldsymbol{x}) = \infty$ when $\exp(\alpha_k \mathrm{d}f(\mu^k, \boldsymbol{x}))$ is not integrable. One can see that as soon as $\mu^0 = q^0 \mathrm{d}\boldsymbol{x}$ is a positive density, so is each μ^k. We thus can consider in the following that $\mathrm{d}\mu^k = q^k \mathrm{d}\boldsymbol{x}$. This choice of h is motivated by the positive, integrable assumption together with, as mentioned earlier, its coherence with the structure of the gradient descent method. This descent algorithm is defined as the "exponentiated gradient" descent in [17]. Since [17] it has been widely studied in the context of machine learning even in the Bayesian framework, see [9] for instance.

The optimization algorithm involved in this paper is the following:

Algorithm 1 Exponentiated Gradient algorithm

1: INITIALIZE($\mu^0 \in \Omega$)
2: **repeat**
3: Compute $\mathrm{d}f(\mu^k, \boldsymbol{x})$
4: Compute $\alpha_k = \arg\max_\alpha K_k(\alpha)\exp(\alpha \mathrm{d}f(\mu^k, .))\mu^k$
5: Compute $\mu^{k+1} = K_k(\alpha_k)\exp(\alpha_k \mathrm{d}f(\mu^k, .))\mu^k$
6: **until** Convergence

In order to determine the convergence properties of this exponentiated gradient method in our context, let us define the hypotheses that we impose on the functional F.

DEFINITION 2.1. *Let $F : \widetilde{M} \to \mathbb{R}$ be a concave functional. We say that F satisfies hypothesis (H) if:*

(i) *F can be written as $F = G + \mathcal{H}(.)$ where G is a concave L-Lipschitz Fréchet-differentiable functional whereas $\mathcal{H}$ corresponds to $-\mathcal{KL}(.||\mathcal{L})$ that is the Kullback-Leibler divergence from a measure to the Lebesgue measure.*

(ii) *F is twice differentiable in the sense of Gateaux and its first order derivative satisfies Equation (2.4).*

(iii) $\lim_{\|\mu\|\to\infty} F(\mu) = -\infty$.

REMARK 1. *The two points (ii), (iii) of the definition of hypothesis (H) just ensure that the optimal stepsize α_k defined in our algorithm indeed exists. Concerning hypothesis (i), it can be replaced by the more restrictive hypothesis that F is L-Lipschitz Fréchet differentiable. Note that our purpose is to construct density*

measures, the term $\mathcal{H}(\mu)$ also corresponds in this case to the entropy of the density function.

Let us now state the convergence result.

THEOREM 2.2. *Let F be a concave functional satisfying hypothesis (H) of Definition 2.1. Let α_k be the optimal stepsize of Algorithm 1, for every $k \geq 0$. Then the sequence $(\mu^k)_{k\geq 0}$ of elements of $\widetilde{\mathcal{M}}$ given by $\mu^{k+1} = K_k(\alpha)\exp(\alpha_k \mathrm{d}f(\mu^k, .))\mu^k$ converges to a maximizer of F on Ω.*

The proof of this theorem involves two main steps. In a first step we prove that the sequence $(F(\mu^k))_{k\in\mathbb{N}}$ is an increasing sequence. This allows in a second time to infer the convergence of the sequence $(\mu^k)_{k\geq 0}$ to a solution of (2.3).

Let $k > 0$ be fixed and μ^k be given. For every $\alpha \geq 0$ we define μ^α as the measure defined for all $\boldsymbol{x} \in \mathbb{R}^N$ by $\mathrm{d}\mu^\alpha(\boldsymbol{x}) = K_k(\alpha)\exp(\alpha \mathrm{d}f(\mu, \boldsymbol{x}))\mathrm{d}\mu^k(\boldsymbol{x}) := h_\alpha(\mu^k, \boldsymbol{x})\mathrm{d}\mu^k(\boldsymbol{x})$.

We define furthermore $g_k(\alpha) := F(\mu^\alpha)$. Thus g_k is a function from $\mathbb{R}^+$ to $\mathbb{R}$ twice differentiable and α_{opt} is an optimal step-size if $g_k(\alpha_{opt}) = \max g_k(\alpha)$, i.e.

$$\alpha_{opt} = \arg\max_\alpha g_k(\alpha). \tag{2.10}$$

The fact that $F(\mu) \to -\infty$ when $\|\mu\| \to \infty$ ensures that we can find an α_{opt}, not necessarily unique, such that

$$\forall \alpha > 0, \quad F(\mu^\alpha) \leq F(\mu^{\alpha_{opt}}). \tag{2.11}$$

Let $\alpha > 0$ be given and consider the decomposition given by Point (i) of Definition 2.1. Thanks to Equation (2.6) one has

$$G(\mu^\alpha) \geq G(\mu^k) + \mathrm{d}G_{\mu^k}(\mu^\alpha - \mu^k) - L\|\mu^\alpha - \mu^k\|^2. \tag{2.12}$$

Furthermore, as $\mu^\alpha = h_\alpha(\mu^k, .)\mu^k$ and μ^k is a probability measure one can notice that

$$-L\|\mu^\alpha - \mu^k\|_{TV}^2 = -L\|h_\alpha - 1\|_{L^1(\mu^k)}^2 \geq -L\|h_\alpha - 1\|_{L^2(\mu^k)}^2. \tag{2.13}$$

Furthermore,

$$\mathcal{H}(\mu^\alpha) = \mathcal{H}(h_\alpha\mu^k) = \mathcal{H}(\mu^k) - \int_{\mathbb{R}^N} \ln(h_\alpha(\mu^k, \boldsymbol{x}))\mathrm{d}\mu^\alpha(\boldsymbol{x}) - \int_{\mathbb{R}^N} \ln\left(\frac{\mathrm{d}\mu_k}{\mathrm{d}\mathcal{L}}\right)(h_\alpha(\mu^k, \boldsymbol{x}) - 1)\mathrm{d}\mu^k(\boldsymbol{x}).$$

Following the development made in [8] for simple functions, the Gateaux derivative at q^k of the entropy in direction $h \in L^1$ is given by

$$\partial\mathcal{H}_{q^k}(h) = -\int (\ln(q^k(\boldsymbol{x})) + 1)h(\boldsymbol{x})\mathrm{d}(\boldsymbol{x}),$$

as soon as $q^k + h$ is always positive. A similar approach in $\widetilde{\mathcal{M}}$ ensures that

$$\partial\mathcal{H}_{\mu^k}(\mu^\alpha - \mu^k) = \int_{\mathbb{R}^N} (-\ln\left(\frac{\mathrm{d}\mu_k}{\mathrm{d}\mathcal{L}}\right) - 1)(h_\alpha(\mu^k, \boldsymbol{x}) - 1)\mathrm{d}\mu^k(\boldsymbol{x}),$$

as $\mu^k + (\mu^\alpha - \mu^k)$ is obviously positive.

This entails

$$\mathcal{H}(\mu^\alpha) \geq \mathcal{H}(\mu^k) - \int_{\mathbb{R}^N} \ln(h_\alpha(\mu^k, \boldsymbol{x}))\mathrm{d}\mu^\alpha(\boldsymbol{x}) + \partial\mathcal{H}_{q^k}(q^\alpha - q^k). \tag{2.14}$$

Finally, from (2.12), (2.13) and (2.14) one has

$$F(\mu^\alpha) - F(\mu^k) \geq \mathrm{d}F_{\mu^k}(\mu^\alpha - \mu^k) - L\|h_\alpha(\mu^k, .) - 1\|^2_{L^2(\mu^k)} - \int_{\mathbb{R}^N} (\alpha \mathrm{d}f(\mu^k, \boldsymbol{x}) + \ln K_k(\alpha))\mathrm{d}\mu^\alpha(\boldsymbol{x}). \tag{2.15}$$

Finally, $F(\mu^\alpha) - F(\mu^k)$ is positive if the right side of Equation (2.15) is positive.

LEMMA 2.3. *Let F be a functional satisfying Hypothesis (H). Let also $(\mu^k)_{k\in\mathbb{N}}$ be the sequence provided by Algorithm 1. Then*

$$\exists \alpha_0 > 0,\ \forall \alpha \in (0, \alpha_0) \quad F(\mu^\alpha) - F(\mu^k) \geq 0. \tag{2.16}$$

The proof of this lemma is quite technical and reported to Appendix 7.1.

Lemma 2.3 ensures that for $\alpha > 0$ small enough, $F(\mu^\alpha) \geq F(\mu^k)$. As we choose $\mu^{k+1} = \mu^{\alpha_{opt}}$, where α_{opt} is defined by (2.10), we obviously have $F(\mu^{k+1}) \geq F(\mu^k)$.

Finally the sequence $(F(\mu^k))_{k\in\mathbb{N}}$ is increasing and upper bounded in $\mathbb{R}$, thus convergent. Moreover, it thus also satisfies that $F(\mu^{k+1}) - F(\mu^k) \to 0$.

In order to conclude we have to show that $(\mu^k)_{k\in\mathbb{N}}$ indeed converges to the maximum of F on Ω. But, for every $k \geq 0$, $\mu^k \in \Omega$, which is bounded in $\widetilde{\mathcal{M}}$ and thus in $\mathcal{M}(\mathbb{R}^N)$. Furthermore, this last one is the dual of C_0, the space of continuous functions that tend to zero at infinity, which is a separable Banach space. The Banach-Alaoglu Theorem thus holds, see [32] for instance, and ensures that there exists $\mu^{lim} \in \widetilde{\mathcal{M}}$ and a subsequence $(\mu^{k_n})_{n\in\mathbb{N}}$ such that for every continuous function that goes to zero at infinity f,

$$\int f(\boldsymbol{x})\mu^{k_n}(\mathrm{d}\boldsymbol{x}) \to \int f(\boldsymbol{x})\mu^{lim}(\mathrm{d}\boldsymbol{x}).$$

i.e. when $k \to \infty$, we have $\mu^{k_n} \rightharpoonup^* \mu^{lim} \in \Omega$.

LEMMA 2.4. *Let $(\mu^k)_{k\in\mathbb{N}}$ be the sequence of probability measures generated by Algorithm 1. There exists a subsequence $(\mu^{k_n})_{n\in\mathbb{N}}$ such that $\mu^{k_n} \rightharpoonup^* \mu^{opt}$ where μ^{opt} is given by* (2.3).

Proof.

From Lemma 2.3 we know that there exists $\alpha_0 > 0$ such that

$$F(\mu^{k+1}) = g_k(\alpha_{opt}) \geq g_k(\alpha), \quad \forall \alpha \in (0, \alpha_0).$$

However the analytic form of α_{opt} is not attainable in practice. We thus approximate it by a calculable α_{subopt}, not necessarily smaller than α_0. In order to determine this α_{subopt} we can notice that thanks to the Taylor-Young formula, for α small enough

$$g_k(\alpha) = g_k(0) + \alpha g_k'(0) + \frac{\alpha^2}{2} g_k''(0) + \alpha^2\varepsilon(\alpha) := \varphi_k(\alpha) + \alpha^2\varepsilon(\alpha), \tag{2.17}$$

where $\varepsilon(\alpha) \to 0$ when $\alpha \to 0$.

Let us determine the derivatives of g. For this purpose, we have to determine the derivative of the function $\widetilde{g}' : \alpha \mapsto h_\alpha(\mu^k, .)$. As $h_\alpha(\mu^k, \boldsymbol{x}) = K_k(\alpha)e^{\alpha \mathrm{d}f(q^k, \boldsymbol{x})}$, its derivative is given by

$$\forall \boldsymbol{x} \in \mathbb{R}^N, \quad \widetilde{g}'(\alpha)(\mathrm{d}\boldsymbol{x}) = K_k'(\alpha)e^{\alpha \mathrm{d}f(q^k, \boldsymbol{x})} + \mathrm{d}f(q^k, \boldsymbol{x})K_k(\alpha)e^{\alpha \mathrm{d}f(q^k, \boldsymbol{x})}.$$

As α is supposed to be close to zero, one can assume that it is not greater than one which allows to use the dominated convergence Theorem and gives

$$K_k'(\alpha) = -\frac{\int \mathrm{d}f(q^k, \boldsymbol{x})K_k(\alpha)e^{\alpha \mathrm{d}f(q^k, \boldsymbol{x})}\mathrm{d}\mu^k(\boldsymbol{x})}{\int e^{\alpha \mathrm{d}f(q^k, \boldsymbol{x})}\mathrm{d}\mu^k(\boldsymbol{x})^2} = -K_k(\alpha)\int \mathrm{d}f(q^k, \boldsymbol{x})\mathrm{d}\mu^\alpha(\boldsymbol{x}).$$

Finally one obtains that

$$\forall \boldsymbol{x} \in \mathbb{R}^N, \quad \widetilde{g}'(\alpha)(\mathrm{d}\boldsymbol{x}) = \mathrm{d}\mu^\alpha(\boldsymbol{x})\left(\mathrm{d}f(\mu^k, \boldsymbol{x}) - \int_{\mathbb{R}^N} \mathrm{d}f(\mu^k, \boldsymbol{y})\mathrm{d}\mu^\alpha(\boldsymbol{y})\right). \tag{2.18}$$

Thus

$$g_k'(\alpha) = \mathrm{d}F_{\mu^\alpha}(\mathrm{d}\mu^\alpha) = \int_{\mathbb{R}^N} \mathrm{d}f(\mu^\alpha, \boldsymbol{x})\left(\mathrm{d}f(\mu^k, \boldsymbol{x}) - \int_{\mathbb{R}^N} \mathrm{d}f(\mu^k, \boldsymbol{y})\mu^\alpha(\mathrm{d}\boldsymbol{y})\right)\mathrm{d}\mu^\alpha(\boldsymbol{x}),$$

and

$$\begin{aligned} g_k'(0) = \mathrm{d}F_{\mu^k}(\widetilde{g}'(0)) &= \int_{\mathbb{R}^N} \mathrm{d}f(\mu^k, \boldsymbol{x})\left(\mathrm{d}f(\mu^k, \boldsymbol{x}) - \int_{\mathbb{R}^N} \mathrm{d}f(\mu^k, \boldsymbol{y})\mu^k(\mathrm{d}\boldsymbol{y})\right)\mathrm{d}\mu^k(\boldsymbol{x}) \\ &= \int_{\mathbb{R}^N} \mathrm{d}f(\mu^k, \boldsymbol{x})^2\mu^k(\mathrm{d}\boldsymbol{x}) - \left(\int_{\mathbb{R}^N} \mathrm{d}f(\mu^k, \boldsymbol{y})\mathrm{d}\mu^k(\boldsymbol{y})\right)^2 \end{aligned} \tag{2.19}$$

The critical point of $\varphi_k(\alpha)$ is $\alpha_{subopt} = -\frac{g_k'(0)}{g_k''(\theta)}$, as soon as $g_k''(0) \neq 0$, which gives in (2.17):

$$g_k(\alpha_{subopt}) = g_k(0) - \frac{g_k'(0)^2}{2g_k''(0)} + \alpha^2\varepsilon(\alpha), \tag{2.20}$$

and by construction of $F(\mu^{k+1})$,

$$F(\mu^{k+1}) \geq g_k(\alpha_{subopt}) = g_k(0) - \frac{g_k'(0)^2}{2g_k''(0)} + \alpha^2\varepsilon(\alpha) = F(\mu^k) - \frac{g_k'(0)^2}{2g_k''(0)} + \alpha^2\varepsilon(\alpha).$$

As $F(\mu^{k+1}) - F(\mu^k) \to 0$, obviously $\lim_{k\to\infty} \frac{g_k'(0)^2}{2g_k''(0)} = 0$. Let us consider the converging subsequence $(k_n)_{n\in\mathbb{N}}$ and denote by $(\alpha_{k_n})_{n\in\mathbb{N}}$ the sequence defined by $\forall n \in \mathbb{N}$, $\alpha_{k_n} = \alpha_{subopt}$. Hence, $-\frac{g_{k_n}'(0)^2}{g_{k_n}''(0)} = g_{k_n}'(0)\alpha_{k_n} \to 0$. As $\mathrm{d}F$ is continuous, the sequence $(g_{k_n}(0))_{n\in\mathbb{N}}$ is convergent and either $\alpha_{k_n} \to 0$ or $g_{k_n}'(0) \to 0$. Let us assume that $\alpha_{k_n} \to 0$ and that $g_{k_n}'(0) \to l \neq 0$. Let $\beta > 0$ be given. As $\alpha_{k_n} \to 0$ we have, for n large enough that

$$\begin{aligned} g_{k_n}(\frac{\alpha_{k_n}}{\beta}) - g_{k_n}(0) &= \frac{\alpha_{k_n}}{\beta}g_{k_n}'(0) + \frac{\alpha_{k_n}^2}{2\beta^2}g_{k_n}''(0) + \alpha_{k_n}^2\varepsilon(\alpha_{k_n}) \\ &= \frac{\alpha_{k_n}}{\beta}g_{k_n}'(0)\left(1 - \frac{1}{2\beta}\right) + \alpha_{k_n}^2\varepsilon(\alpha_{k_n}). \end{aligned}$$

Hence,

$$\frac{g_{k_n}(\frac{\alpha_{k_n}}{\beta}) - g_{k_n}(0)}{\alpha_{k_n}/\beta} = g'_{k_n}(0)\left(1 - \frac{1}{2\beta}\right) + \alpha_{k_n}\varepsilon(\alpha_{k_n}), \tag{2.21}$$

and when n tends to infinity, α_{k_n} tends to zero, and taking limits in (2.21) one obtains

$$l = l\left(1 - \frac{1}{2\beta}\right),$$

which is impossible as soon as $\beta \neq \frac{1}{2}$. Hence, $g'_{k_n}(0) \to 0$ when $n \to \infty$.

Furthermore,

$$g'_k(0) = \int_{\mathbb{R}^N} \mathrm{d}f(\mu^k, \boldsymbol{x})^2 \mu^k(\mathrm{d}\boldsymbol{x}) - \left(\int_{\mathbb{R}^N} \mathrm{d}f(\mu^k, \boldsymbol{y})\mu^k(\mathrm{d}\boldsymbol{y})\right)^2.$$

Hence, for n large enough,

$$\|\mathrm{d}f(\mu^{k_n}, .)\|_{L^2(\mu^{k_n})} - \|\mathrm{d}f(\mu^{k_n}, .)\|_{L^1(\mu^{k_n})} \to 0,$$

and thus $\mathrm{d}f(\mu^{k_n}, .)$ converges to a constant λ, independent of $\boldsymbol{x} \in \mathbb{R}^N$.

Let ν be any element of Ω. Applying the Main Value Theorem to the real valued function $f(\theta) = F(\theta\mu^{lim} + (1-\theta)\nu)$ we see that

$$F(\nu) = F(\mu^{lim}) + \mathrm{d}F_{\mu^{lim}}(\nu - \mu^{lim}) + \frac{1}{2}\mathrm{d}^2 F_{\tilde{\mu}}(\nu - \mu^{lim}, \nu - \mu^{lim}),$$

where $\tilde{\mu} = \theta_0\nu + (1-\theta_0)\mu^{lim}$ and $\theta_0 \in (0,1)$ is fixed.

As $\mathrm{d}F_{\mu^{lim}}(\nu - \mu^k) = 0$, and from concavity of F we have

$$F(\nu) \leq F(\mu^{lim}) \qquad \forall \nu \in \Omega,$$

which means that $F(\mu^{lim})$ is a maximum of F over Ω.

□

In the present part we have presented an algorithm adapted to the space of probability measures. Our main interest is its application in the context of variational Bayesian methodology. For the sake of completeness, let us remind this methodology introduced in [20].

3. Our Variational Bayesian Algorithm. For the sake of simplicity, in the following, we will only work with density functions q instead of measures μ. We also denote by $\boldsymbol{y} \in \mathbb{R}^M$ the M dimensional vector containing the data information whereas $\boldsymbol{w} \in \mathbb{R}^N$ represents the vector to be estimated, which is considered as a realization of a random vector $\boldsymbol{W}$. We also denote by p the prior probability density function (p.d.f.) of $\boldsymbol{W}$. The Bayes rule entails that this prior distribution is closely related to the posterior one, $p(\boldsymbol{w}|\boldsymbol{y})$, up to a normalization constant. However, even in simple cases this posterior may not be separable. Hence, in the variational Bayesian framework, we approximate this posterior distribution by a separable probability density

$$q(\boldsymbol{w}) = \prod_i q_i(w_i). \tag{3.1}$$

Taking separable laws obviously simplify the problem even if it introduces some approximation errors.

Therefore, we want to determine a separable probability density function q close of the true posterior, in the sense defined by the Kullback-Leibler divergence, see [35] for instance. The optimal approximating density q is given by

$$\forall i \in \{1, \dots, N\}, \quad q_i(w_i) = K_k^i \exp\left(< \ln p(\boldsymbol{y}, \boldsymbol{w}) >_{\prod_{j \neq i} q_j} \right), \tag{3.2}$$

where K_k^i is the normalization constant and $< \ln p(\boldsymbol{y}, \boldsymbol{w}) >_{\prod_{j \neq i} q_j} = \int_{\mathbb{R}^{N-1}} \ln p(\boldsymbol{y}, \boldsymbol{w}) \prod_{j \neq i} q_j(w_j)$ comes from Eq. (3.1).

Although this solution is obtained analytically, Equation (3.2) clearly does not have an explicit form. In order to have implementable methods a first step is to consider conjugate priors. Hence the optimization turns out to an update of the distribution parameters. However even for conjugate priors, this solution is hardly tractable in practice, and is thus approximated thanks to iterative fixed points methods. In Equation (3.2), the calculus of q_i imposes the knowledge of all q_j for j different from i, this optimization is either performed alternatively or by group of coordinate. In both cases, the computation complexity can be important.

For large dimensional problems these methods are not tractable in practice. Our purpose is thus to solve the functional optimization problem given by the Bayesian variational method in an efficient way thanks to the algorithm defined in Section 2.

3.1. Variational Bayesian Exponentiated Gradient Algorithm. In this section we define an iterative method which allows to compute efficiently at each iteration each q_i independently of the others in order to decrease the computational cost of one iteration.

A first step in this part is to rewrite the minimization problem as a convex optimization problem independent of the posterior distribution to be approximated. Instead of minimizing the Kullback-Leibler divergence, we thus remark, as in [6], that

$$\ln p(\boldsymbol{y}) = \ln \frac{p(\boldsymbol{y}, \boldsymbol{w})}{p(\boldsymbol{w}|\boldsymbol{y})}, \tag{3.3}$$

where $\boldsymbol{w}$ is the vector of hidden variables and parameters.

As the log-likelihood $\ln p(\boldsymbol{y})$ in (3.3) does not depends on $\boldsymbol{w}$ one can write

$$\ln p(\boldsymbol{y}) = \mathcal{F}(q(\boldsymbol{w})) + \mathcal{KL}[q(\boldsymbol{w})||p(\boldsymbol{w}|\boldsymbol{y}))].$$

In this case,

$$\mathcal{F}(q(\boldsymbol{w})) = \int_{\mathbb{R}^N} q(\boldsymbol{w}) \ln(\frac{p(\boldsymbol{y}, \boldsymbol{w})}{q(\boldsymbol{w})}) \mathrm{d}\boldsymbol{w}, \tag{3.4}$$

is the negative free energy. Thus minimizing the Kullback-Leibler divergence is obviously equivalent to maximize this negative free entropy.

Therefore in the following we will consider the problem of maximizing

$$\begin{aligned}\mathcal{F}(q(\boldsymbol{w})) &= \int_{\mathbb{R}^N} \ln p(\boldsymbol{y},\boldsymbol{w}) q(\boldsymbol{w}) \mathrm{d}\boldsymbol{w} - \int_{\mathbb{R}^N} \ln(q(\boldsymbol{w})) q(\boldsymbol{w}) \mathrm{d}\boldsymbol{w} \\ &= \langle \ln p(\boldsymbol{y},\boldsymbol{w}) \rangle_{q(\boldsymbol{w})} + \mathcal{H}(q(\boldsymbol{w})),\end{aligned} \tag{3.5}$$

where

$$\langle \ln p(\boldsymbol{y},\boldsymbol{w}) \rangle_{q(\boldsymbol{w})} = \int \ln(p(\boldsymbol{y},\boldsymbol{w})) q(\boldsymbol{w}) \mathrm{d}\boldsymbol{w}$$

represents the mean of $\ln p(\boldsymbol{y},\boldsymbol{w})$ under the distribution $q(\boldsymbol{w})$ whereas

$$\mathcal{H}(q(\boldsymbol{w})) = -\int_{\mathbb{R}^N} \ln(q(\boldsymbol{w})) q(\boldsymbol{w}) \mathrm{d}\boldsymbol{w},$$

is the entropy of q. The main advantage of this approach is that the objective functional does not depend on the true posterior anymore but only on the joint distribution $p(\boldsymbol{y},\boldsymbol{w})$, which is more easily tractable.

One can also notice that the problem of finding

$$q^{opt} = \underset{q\ p.d.f.}{\arg\max}\, \mathcal{F}(q) \tag{3.6}$$

is equivalent to the problem of finding

$$\mu^{opt} = \underset{\mu \in \Omega}{\arg\max}\, F(\mu). \tag{3.7}$$

Where the functional F is defined by $\forall \mu \in \Omega,\ F(\mu) = \mathcal{F}(q)$. Furthermore the corresponding function F satisfies hypothesis (H). Hence we can apply Theorem 2.2 in this context.

Remark 2. *As mentioned earlier, a classical method in our context is to consider each density function q as a $L^1(\mathbb{R}^N)$ function and to apply classical algorithms. In the present framework, taking the non-negativity and the total mass assumptions into account, the algorithm involved is given by the projected gradient method which gives:*

$$\forall \boldsymbol{w} \in \mathbb{R}^N \qquad q^{k+1}(\boldsymbol{w}) = P_\Theta(q^k(\boldsymbol{w}) + \rho^k \mathrm{d}f(q^k,\boldsymbol{w})), \tag{3.8}$$

where P_Θ is the projector operator on the subspace $\Theta = \{f \in L^1(\mathbb{R}^N);\ f(\boldsymbol{w}) \geq 0$ and $\|f\|_{L^1} = 1\}$.

However, this algorithm requires that $\mathrm{d}f(q^k,\boldsymbol{w}) \in L^1(\mathbb{R}^N)$ which is not the case in general.

Hence, we apply the algorithm introduced in Section 2 to the Variational Bayesian framework of Section 3. We consider

$$\mathcal{F}(q(\boldsymbol{w})) = \langle \ln p(\boldsymbol{y},\boldsymbol{w}) \rangle_{q(\boldsymbol{w})} + \mathcal{H}(\boldsymbol{w}).$$

In this case, the Fréchet differential of $F(\mu) = \mathcal{F}(q)$ at $\mu \in \Omega$ separable is given by $\mathrm{d}F_\mu(\nu) = \sum_i \int_{\mathbb{R}^N} \mathrm{d}_i f(q_i, x_i) \nu_i(\mathrm{d}x)$ where

$$\forall i \in \{1,\ldots,N\}, \quad \forall \boldsymbol{w} \in \mathbb{R}^N, \quad \mathrm{d}_i f(q, w_i) = \langle \ln p(\boldsymbol{y},\boldsymbol{w}) \rangle_{\prod_{j\neq i} q_j} - \ln q_i(w_i) - 1.$$

Let $k \geq 0$ be given and q^k be constructed. Following the scheme defined by Algorithm 1 and Equation (2.9), at the following iteration we consider q^α given, for $\alpha > 0$, by

$$\forall \boldsymbol{w} \in \mathbb{R}^N, \quad q^\alpha(\boldsymbol{w}) = K_k q^k \exp\left[\alpha_k \mathrm{d}f(q^k, \boldsymbol{w})\right] \tag{3.9}$$

$$= \tilde{K}_k q^k(\boldsymbol{w}) \left(\prod_i \frac{\exp\left(< \ln p(\boldsymbol{y}, \boldsymbol{w}) >_{\prod_{j\neq i} q_j^k(w_j)} \right)}{q_i^k(w_i)} \right)^{\alpha_k}$$

$$= \tilde{K}_k q^k(\boldsymbol{w}) \left(\prod_i \frac{q_i^r(w_i)}{q_i^k(w_i)} \right)^{\alpha_k} \tag{3.10}$$

where $\tilde{K}_k$ is the normalization constant and q^r is an intermediate measure given by

$$\forall i \in \{1, \ldots, N\}, \quad q_i^r(w_i) = \exp\left(< \ln p(\boldsymbol{y}, \boldsymbol{w}) >_{\prod_{j\neq i} q_j^k(w_j)} \right)$$

The main challenge is to determine the optimal value of $\alpha > 0$. This optimal value should satisfy $g_k'(\alpha_{opt}) = 0$. However, this quantity is hardly tractable in practice. Therefore, we consider instead the suboptimal value given by

$$\alpha_{subopt} = -\frac{g_k'(0)}{g_k''(0)}, \tag{3.11}$$

when $g_k''(0) \neq 0$. This leads to the main algorithm of this paper.

Algorithm 2 Variational Bayesian Exponentiated Gradient Like Algorithm

1: INITIALIZE($q^0 \in \Omega$)
2: **repeat**
3: **function** ITERATION(Compute $q^{k+1} = K_k q^k \exp\left[\alpha_k \mathrm{d}f(q^k, \boldsymbol{w})\right]$)
4: Compute $q_i^r(w_i) = \exp\left(< \ln p(\boldsymbol{y}, \boldsymbol{w}) >_{\prod_{j\neq i} q_j^k(w_j)} \right)$ for every $i = 1, \ldots, N$
5: Compute $\alpha_{subopt} = -\frac{g'(0)}{g''(0)}$
6: Compute $q^{\alpha_{subopt}}(\boldsymbol{w}) = q^k(\boldsymbol{w}) \left(\frac{q^r(\boldsymbol{w})}{q^k(\boldsymbol{w})} \right)^{\alpha_{subopt}}$.
7: Take $q^{k+1} = q^{\alpha_{subopt}}$.
8: **end function**
9: **until** Convergence

4. Application to linear inverse problems.

4.1. Statement of the problem. The following of this paper presents the application of Algorithm 2 to linear inverse ill-posed problems. More precisely we consider its implementation for Bayesian study of heavy-tailed information. The model of observations chosen in the following is given by

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{b}, \tag{4.1}$$

where $\boldsymbol{y} \in \mathbb{R}^M$ is the vector of observations given as a linear function of the unknowns vector $\boldsymbol{x} \in \mathbb{R}^N$ to be estimated. Here, $\boldsymbol{b} \in \mathbb{R}^M$ is the noise vector whereas $\boldsymbol{H}$ is a matrix in $M_{N\times M}$. We also suppose that $\boldsymbol{x}$ is a realization of a random vector $\mathbf{X}$.

In the following we stand in a white noise model which induces that the noise is supposed to be an iid Gaussian vector $\mathcal{N}(0, \sigma_b^2 \mathbf{I})$. We can deduct easily the likelihood

$$p(\boldsymbol{y}|\boldsymbol{x}) = (2\pi\sigma_b^2)^{-M/2} \exp\left[-\frac{\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|^2}{2\sigma_b^2}\right]. \tag{4.2}$$

Concerning the prior distribution we choose to take sparsity into account by considering $\mathbf{X}$ distributed following a separable heavy tailed distribution. The most general case is given by Gaussian Vector Scale Mixture (GVSM) defined in [13]. In this case, for $i = 1, \ldots, N$, we suppose that $X_i \sim U_i/\sqrt{Z_i}$ where $\mathbf{U} \sim \mathcal{N}(0, \sigma_s^2 \mathbf{I})$, $\mathbf{Z} = \prod Z_i$ is a positive random vector of independent positive coordinates and $\mathbf{U}$ and $\mathbf{Z}$ are independents. As a consequence the density of $\mathbf{X}$ is given in an integral form as

$$\forall i \in \{1, \ldots, N\}, \qquad p(x_i) = \int_{\mathbb{R}} \frac{\sqrt{z_i}}{(2\pi)^{1/2}\sigma_s} e^{-\frac{z_i x_i^2}{2\sigma_s^2}} \phi_{z_i}(z_i) dz_i.$$

Note that in the definition, for the sake of simplicity, we consider $\mathbf{Z}$ as a precision parameter and not as a dilatation one. Gaussian Vector Scale Mixture form a large class of nongaussian random variables recently developed as a model of wavelet coefficients of natural images, see [38]. The main interest of this model is, by solving an extended problem due to the presence of a hidden random vector $\mathbf{Z}$, to allow the use of Bayesian hierarchic approaches.

The Gaussian Scale Mixture family offers a large class of random variables including Gaussian mixing, when $\mathbf{Z} \sim Z\mathbf{I}$ a discrete random vector or Student laws if all Z_i are Gamma random variables. With different hypothesis on the distribution of $\mathbf{Z}$ one can also define Generalized Gaussian distributions or α-stable ones, see [38]. Indeed GSM offer a simple representation of a large class of nongaussian probability distributions, which justify the increasing interest on this model.

In our context, we choose to consider $\mathbf{Z}$ as a independent Gamma random vector, i.e. for $i = 1, \ldots, N$, we have $Z_i \sim \mathcal{G}(\tilde{a}_i, \tilde{b}_i)$ and

$$\forall i \in \{1, \ldots, N\}, \qquad p(x_i) = \frac{\tilde{b}_i^{\tilde{a}_i}}{\Gamma(\tilde{a}_i)} \int_{\mathbb{R}} \frac{\sqrt{z_i}}{(2\pi)^{1/2}\sigma_s} e^{-\frac{z_i x_i^2}{2\sigma_s^2}} z_i^{\tilde{a}_i - 1} e^{-z_i \tilde{b}_i} dz_i. \tag{4.3}$$

For $\tilde{a}_i = \tilde{b}_i = \frac{\nu}{2}$, the p.d.f. of $\mathbf{X}$ corresponds to a Student-t distribution, as in the model used in [4]. This model of $\boldsymbol{Z}$ ensures that $\boldsymbol{X}$ satisfies the conjugate priors condition.

One can easily check that when the prior information is given by (4.3), Equation (4.2) gives the following posterior distribution

$$p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{y}) \propto \sigma_b^{-M} \exp\left[-\frac{\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|^2}{2\sigma_b^2}\right] \prod_{i=1}^{N} \frac{\sqrt{z_i}}{\sigma_s} \exp\left[-\frac{z_i x_i^2}{2\sigma_s^2}\right] \frac{\tilde{b}_i^{\tilde{a}_i} z_i^{\tilde{a}_i - 1} e^{-z_i \tilde{b}_i}}{\Gamma(\tilde{a}_i)}. \tag{4.4}$$

Considering that we do not know the involved constants and that the mixing matrix $\boldsymbol{H}$ is high dimensional, this posterior distribution cannot be evaluated directly.

4.2. Numerical implementation. The aim of variational Bayesian methodology and therefore of our method in the context established in part 4 is the approximation of the posterior p.d.f given by (4.4) by a separable one $q(\boldsymbol{x}, \boldsymbol{z}) = \prod_i q_i(x_i) \prod_j \tilde{q}_j(z_j)$.

As we have chosen conjugate prior for $\boldsymbol{X}$ and $\boldsymbol{Z}$, the optimum approximating distribution of $\boldsymbol{X}$ is known to belong to a Gaussian family, whereas the p.d.f. of $\boldsymbol{Z}$ belongs to a Gamma one.

$$q^k(\boldsymbol{x}) = \prod_i \mathcal{N}(\boldsymbol{m}_k(i), \boldsymbol{\sigma}_k^2(i))$$
$$\tilde{q}^k(\boldsymbol{z}) = \prod_j \mathcal{G}(a_k(j), b_k(j))$$

Hence at the initialization stage, we consider

$$q^0(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{m}_0, \mathrm{Diag}(\boldsymbol{\sigma}_0^2))$$
$$\tilde{q}^0(\boldsymbol{z}) = \prod_j \mathcal{G}(a_0(j), b_0(j))$$

where $\mathrm{Diag}(\boldsymbol{v})$ is a diagonal matrix with $\boldsymbol{v}$ on its diagonal, and $\boldsymbol{\sigma}_0^2 \in \mathbb{R}^N$ is the vector of initial variances.

Our minimization problem can be analyzed following the alternate iterative scheme:

$$\tilde{q}^{k+1}(\boldsymbol{z}) = \arg\max_{\tilde{q}(\boldsymbol{z})} F\big(q^k(\boldsymbol{x})\tilde{q}(\boldsymbol{z})\big)$$
$$q^{k+1}(\boldsymbol{x}) = \arg\max_{q(\boldsymbol{x})} F\big(q(\boldsymbol{x})\tilde{q}^{k+1}(\boldsymbol{z})\big)$$

4.2.1. Approximation of $\tilde{q}$. One can see in Equation (4.4) that the conditional posterior $p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y})$ is separable. In this case the classical Bayesian variational approach is efficient enough to be implemented directly. Actually for $i = 1, \ldots, N$, $\tilde{q}_i^{k+1}(z_i)$ does not depend on other coordinates. Hence all the z_i can be computed simultaneously, knowing only $q(\boldsymbol{x})$. Thanks to the classical Bayesian variational approach [20] described in Section 3, we deduce $\tilde{q}^{k+1}$ thanks to Equation (3.2) and Equation (4.4), for every $i = 1, \ldots, N$

$$\tilde{q}_i^{k+1}(z_i) = \exp\Big(< \ln p(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{z}) >_{\prod_{j\neq i} \tilde{q}_j^k(z_j) q^k(\boldsymbol{x})}\Big) \tag{4.5}$$
$$\propto \exp\left((\tilde{a}_i - \frac{1}{2})\ln(z_i) - \int (\frac{x_i^2 z_i}{2\sigma_s^2} + z_i \tilde{b}_i) \prod_l q_l^k(x_l) \prod_{j\neq i} \tilde{q}_j^k(z_j) \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{z}\right)$$
$$\propto \exp\left((\tilde{a}_i - \frac{1}{2})\ln(z_i) - z_i\tilde{b}_i - \int \frac{x_i^2 z_i}{2\sigma_s^2} q_i^k(x_i)\mathrm{d}x_i\right)$$
$$\propto \exp\left((\tilde{a}_i - \frac{1}{2})\ln(z_i) - z_i\tilde{b}_i - \frac{(\boldsymbol{\sigma}_k^2(i) + \boldsymbol{m}_k^2(i))z_i}{2\sigma_s^2}\right)$$
$$\propto z_i^{\tilde{a}_i - \frac{1}{2}} \exp\left(-z_i\Big[\tilde{b}_i + \frac{(\boldsymbol{\sigma}_k^2(i) + \boldsymbol{m}_k^2(i))}{2\sigma_s^2}\Big]\right) \tag{4.6}$$

This entails that $\tilde{q}_i^{k+1}(z_i)$ corresponds to a Gamma p.d.f. of parameters:

$$\forall i \in \{1, \ldots, N\}, \qquad \boldsymbol{a}_{k+1}(i) = \tilde{a}_i + \frac{1}{2}, \tag{4.7}$$
$$\boldsymbol{b}_{k+1}(i) = \frac{\boldsymbol{m}_k^2(i) + \boldsymbol{\sigma}_k^2(i)}{2\sigma_s^2} + \tilde{b}_i. \tag{4.8}$$

4.2.2. Approximation of q by Algorithm 2. Let us assume that we start with a Gaussian p.d.f. $q^0(\boldsymbol{x})$ with mean $\boldsymbol{m}_0$ and covariance matrix $\mathrm{Diag}(\boldsymbol{\sigma}_0^2)$. At each iteration we determine the approximation of $q^k(\boldsymbol{x})$ thanks to our method. At each iteration $k+1$, we define an auxiliary measure $q^r(\boldsymbol{x})$ by $q_i^r(x_i) = \exp\left(\langle \ln p(\boldsymbol{y},\boldsymbol{x},\boldsymbol{z})\rangle_{\prod_{j\neq i} q_j^k(x_j) q^{k+1}(\boldsymbol{z})}\right)$.

Hence $\forall i \in \{1,\dots,N\}$,

$$
\begin{aligned}
q_i^r(x_i) &= \exp\left(< \ln p(\boldsymbol{y},\boldsymbol{x},\boldsymbol{z}) >_{\prod_{j\neq i} q_j^k(x_j)\tilde{q}^{k+1}(\boldsymbol{z})}\right) \qquad (4.9)\\
&\propto \exp\left(-\int\left(\frac{\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x}\|^2}{2\sigma_b^2}+\frac{x_i^2 z_i}{2\sigma_s^2}\right)\prod_{j\neq i} q_j^k(x_j)\tilde{q}^{k+1}(\boldsymbol{z})\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{z}\right)\\
&\propto \exp\left[-\int\left(\frac{\boldsymbol{x}^T\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{x}-2\boldsymbol{x}^T\boldsymbol{H}^T\boldsymbol{y}}{2\sigma_b^2}+\frac{x_i^2 z_i}{2\sigma_s^2}\right)\prod_{j\neq i} q_j^k(x_j)\tilde{q}^{k+1}(z_i)\mathrm{d}\boldsymbol{x}\mathrm{d}z_i\right]\\
&\propto \exp\left[-\frac{1}{2\sigma_b^2}\Big(x_i^2\mathrm{diag}(\boldsymbol{H}^T\boldsymbol{H})_i - 2x_i(\boldsymbol{H}^T\boldsymbol{y})_i + 2x_i(\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{m}_k)_i\right.\\
&\quad \left.-2x_i\mathrm{diag}(\boldsymbol{H}^T\boldsymbol{H}))_i\boldsymbol{m}_k(i)\Big)+\frac{x_i^2\boldsymbol{a}_{k+1}(i)}{2\sigma_s^2\boldsymbol{b}_{k+1}(i)}\right] \qquad (4.10)
\end{aligned}
$$

where $\mathrm{diag}(A)$ is the vector composed by the diagonal entries of A. Note that $q^r(\boldsymbol{x})$ corresponds, up to the normalization term, to a Gaussian distribution with mean $\boldsymbol{m}_r$ and variance $\boldsymbol{\sigma}_r^2$, where, for every $i = 1,\dots,N$,

$$
\boldsymbol{\sigma}_r^2(i) = \left(\frac{\mathrm{diag}(\boldsymbol{H}^T\boldsymbol{H})_i}{\sigma_b^2}+\frac{\boldsymbol{a}_{k+1}(i)}{\boldsymbol{b}_{k+1}(i)\sigma_s^2}\right)^{-1} \qquad (4.11)
$$

and

$$
\boldsymbol{m}_r(i) = \boldsymbol{\sigma}_r^2(i)\times\left(\frac{\boldsymbol{H}^T\boldsymbol{y}-(\boldsymbol{H}^T\boldsymbol{H}-\mathrm{diag}(\boldsymbol{H}^T\boldsymbol{H}))\boldsymbol{m}_k}{\sigma_b^2}\right)_i \qquad (4.12)
$$

Therefore, by Equation (3.10), we have for every $i = 1,\dots,N$,

$$
\begin{aligned}
q_i^\alpha(x_i) &= K_k q_i^k(x_i)\left(\frac{q_i^r(x_i)}{q_i^k(x_i)}\right)^\alpha\\
&= \sqrt{\frac{\boldsymbol{\sigma}_k^2(i)}{\boldsymbol{\sigma}_r^2(i)}}^{\alpha} K_k\exp\left[-\frac{(x_i-\boldsymbol{m}_k(i))^2}{2\boldsymbol{\sigma}_k^2(i)}\right]\exp\left[-\alpha\frac{x_i^2(\boldsymbol{\sigma}_k^2(i)-\boldsymbol{\sigma}_r^2(i))}{2\boldsymbol{\sigma}_r^2(i)\boldsymbol{\sigma}_k^2(i)}\right]\\
&\quad\times\exp\left[-\alpha\frac{-2x_i(\boldsymbol{m}_r(i)\boldsymbol{\sigma}_k^2(i)-\boldsymbol{m}_k(i)\boldsymbol{\sigma}_r^2(i))+\boldsymbol{m}_r(i)^2\boldsymbol{\sigma}_k^2(i)-\boldsymbol{m}_k(i)^2\boldsymbol{\sigma}_r^2(i)}{2\boldsymbol{\sigma}_r^2(i)\boldsymbol{\sigma}_k^2(i)}\right]\\
&= \sqrt{\frac{\boldsymbol{\sigma}_k^2(i)}{\boldsymbol{\sigma}_r^2(i)}}^{\alpha} K_k\exp\left[-\frac{1}{2}\left(x_i^2\frac{\boldsymbol{\sigma}_r^2(i)+\alpha(\boldsymbol{\sigma}_k^2(i)-\boldsymbol{\sigma}_r^2(i))}{\boldsymbol{\sigma}_r^2(i)\boldsymbol{\sigma}_k^2(i)}\right)\right]\\
&\quad\times\exp\left[-\frac{1}{2}\left(-2x_i\frac{\boldsymbol{m}_k(i)\boldsymbol{\sigma}_r^2(i)+\alpha(\boldsymbol{m}_r(i)\boldsymbol{\sigma}_k^2(i)-\boldsymbol{m}_k(i)\boldsymbol{\sigma}_r^2(i))}{\boldsymbol{\sigma}_r^2(i)\boldsymbol{\sigma}_k^2(i)}+t(\alpha)\right)\right]
\end{aligned}
$$

where q^α is defined in Section 5.1.1, and $t(\alpha) = \alpha\frac{\boldsymbol{m}_r(i)^2\boldsymbol{\sigma}_k^2(i)-\boldsymbol{m}_k(i)^2\boldsymbol{\sigma}_r^2(i)}{2\boldsymbol{\sigma}_r^2(i)\boldsymbol{\sigma}_k^2(i)}$ is a constant. Finally, $q_i^\alpha(\boldsymbol{x})$ is still a Gaussian p.d.f. with parameters $\boldsymbol{m}_\alpha$ and $\mathrm{Diag}(\boldsymbol{\sigma}_\alpha^2)$

satisfying:

$$\boldsymbol{\sigma}_{\alpha}^2(i) = \frac{\boldsymbol{\sigma}_r^2(i)\boldsymbol{\sigma}_k^2(i)}{\boldsymbol{\sigma}_r^2(i) + \alpha(\boldsymbol{\sigma}_k^2(i) - \boldsymbol{\sigma}_r^2(i))} \tag{4.13}$$

$$\boldsymbol{m}_{\alpha}(i) = \frac{\boldsymbol{m}_k(i)\boldsymbol{\sigma}_r^2(i) + \alpha(\boldsymbol{m}_r(i)\boldsymbol{\sigma}_k^2(i) - \boldsymbol{m}_k(i)\boldsymbol{\sigma}_r^2(i))}{\boldsymbol{\sigma}_r^2(i) + \alpha(\boldsymbol{\sigma}_k^2(i) - \boldsymbol{\sigma}_r^2(i))}. \tag{4.14}$$

In order to construct q^{k+1} we choose in the previous equation $\alpha = \alpha_{subopt}$ defined in Equation (3.11).

Finally, we obtain the following algorithm.

Algorithm 3 Supervised Sparse Reconstruction algorithm (SSR)

```
INITIALIZE(q^0, q̃^0)
repeat
    function ESTIMATE q̃^{k+1}(z)(q^k(x))
        update a_{k+1} by Equation (4.7)
        update b_{k+1} by Equation (4.8)
    end function
    function ESTIMATE q^{k+1}(x)(q̃^{k+1}(z))
        compute q^r(x) ← (m_r, σ_r^2) by Equation (4.12) and Equation (4.11)
        compute α_subopt
        compute q^α(x) ← (m_α, σ_α^2) by Equation (4.14) and Equation (4.13)
    end function
until Convergence
```

4.3. Unsupervised algorithm. The algorithm described in the previous part is not a fully Bayesian one as it still depends on some parameters, namely the parameters induced by the model (4.1) and (4.3). We see in the following how this method can be extended to an unsupervised one by estimating these parameters. The parameters of the underlying Gamma random variable are not estimated in the following as they define the sharpness of the prior distribution. We thus only estimate the variance parameter of this prior together with the trade off between the prior and the noise.

The main interest of the variational Bayesian approach introduced by D. MacKay [20] and involved in this paper is its flexibility. That is an unsupervised algorithm can be easily deduced from the method proposed in Section 4.2, by considering that the parameters are realizations of a random variable with a given Jeffrey's prior distribution.

In order to simplify the different expressions, we introduce in the following the notations $\gamma_b = 1/\sigma_b^2$ and $\gamma_s = 1/\sigma_s^2$. Hence, γ_b and γ_s are the precision parameters of the noise and of the prior distribution. From now on they are also assumed to be random variable with Gamma prior of parameters $(\tilde{a}_b, \tilde{b}_b)$ resp. $(\tilde{a}_s, \tilde{b}_s)$. As we do not have information on these precision parameters γ_b and γ_s, this prior is obtained by fixing $(\tilde{a}_b = 0, \tilde{b}_b = 0)$ resp. $(\tilde{a}_s = 0, \tilde{b}_s = 0)$.

With these assumptions, the posterior distribution from Equation (4.4) can be

written as

$$p(\boldsymbol{x},\boldsymbol{z},\gamma_b,\gamma_s|\boldsymbol{y}) \propto \gamma_b^{\frac{M}{2}} \exp\left[-\frac{\gamma_b\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x}\|^2}{2}\right] \gamma_s^{\frac{N}{2}} \prod_i \sqrt{z_i}\exp\left[-\frac{\gamma_s z_i x_i^2}{2}\right] \frac{\tilde{b}_i^{\tilde{a}_i} z_i^{\tilde{a}_i-1} e^{-z_i\tilde{b}_i}}{\Gamma(\tilde{a}_i)}$$
$$\times \frac{\tilde{b}_b^{\tilde{a}_b}\gamma_b^{\tilde{a}_b-1}e^{-\gamma_b\tilde{b}_b}}{\Gamma(\tilde{a}_b)} \frac{\tilde{b}_s^{\tilde{a}_s}\gamma_s^{\tilde{a}_s-1}e^{-\gamma_s\tilde{b}_s}}{\Gamma(\tilde{a}_s)}. \tag{4.15}$$

As in the previous section, the conditional posterior $p(\boldsymbol{z},\gamma_b,\gamma_s|\boldsymbol{x},\boldsymbol{y})$ is separable and can be approximated thanks to the classical variational Bayesian approach. Once again only the distribution of $\boldsymbol{X}$ needs the use of Algorithm 2. Here the alternate optimization scheme to carry out is:

$$\begin{aligned}
\tilde{q}^{k+1}(\boldsymbol{z}) &= \arg\max_{\tilde{q}(\boldsymbol{z})} F\big(q^k(\boldsymbol{x})\tilde{q}(\boldsymbol{z})q_b^k(\gamma_b)q_s^k(\gamma_s)\big)\\
q^{k+1}(\boldsymbol{x}) &= \arg\max_{q(\boldsymbol{x})} F\big(q(\boldsymbol{x})\tilde{q}^{k+1}(\boldsymbol{z})q_b^k(\gamma_b)q_s^k(\gamma_s)\big)\\
q_b^{k+1}(\gamma_b) &= \arg\max_{q(\gamma_b)} F\big(q^{k+1}(\boldsymbol{x})\tilde{q}^{k+1}(\boldsymbol{z})q_b(\gamma_b)q_s^k(\gamma_s)\big)\\
q_s^{k+1}(\gamma_s) &= \arg\max_{q(\gamma_s)} F\big(q^{k+1}(\boldsymbol{x})\tilde{q}^{k+1}(\boldsymbol{z})q_b^{k+1}(\gamma_b)q_s(\gamma_s)\big)
\end{aligned}$$

4.3.1. Optimization of the approximate p.d.f. q_b. Concerning the random vectors $\boldsymbol{Z}$ and $\boldsymbol{X}$, the updating process follows the same scheme than the supervised case, see Section 4.2, and are not recalled here. The main differences reside in the update of the parameters distributions.

As the distributions of γ_b and γ_s are supposed to be Gamma, which is coherent with the conjugate priors hypothesis, at each iteration we just adapt the parameters. Hence we initialize our algorithm by considering that

$$q_b^0(\gamma_b) = \mathcal{G}(a_b^0, b_b^0)$$

At iteration $k+1$ we consider the maximum of the free energy from Equation (3.2) which gives

$$\begin{aligned}
q_b^{k+1}(\gamma_b) &\propto \exp\Big[< \ln \Pr(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\gamma_b,\gamma_s) >_{\prod_j q_j^{k+1}(x_j)\prod_j \tilde{q}_j^{k+1}(z_j)q_s^k(\gamma_s)}\Big]\\
&\propto \exp\left[< (\frac{M}{2}+\tilde{a}_b-1)\ln(\gamma_b) - \gamma_b(\frac{\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x}\|^2}{2}+\tilde{b}_b) >_{\prod_j q_j^{k+1}(x_j)}\right]\\
&\propto \exp\Bigg[(\frac{M}{2}+\tilde{a}_b-1)\ln(\gamma_b)\\
&\quad -\gamma_b\Big(\frac{1}{2}\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{m}_{k+1}\|^2 + \frac{1}{2}\sum_{i=1}^N \mathrm{diag}\left(\boldsymbol{H}^{\mathrm{t}}\boldsymbol{H}\right)_i \boldsymbol{\sigma}_{k+1}^2(i) + \tilde{b}_b\Big)\Bigg]
\end{aligned}$$

So $q_b^{k+1}(\gamma_b)$ is a Gamma p.d.f. of parameters:

$$a_b^{k+1} = \frac{M}{2} + \tilde{a}_b \tag{4.16}$$

$$b_b^{k+1} = \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{m}_{k+1}\|^2 + \frac{1}{2}\sum_{i=1}^N \mathrm{diag}\left(\boldsymbol{H}^{\mathrm{t}}\boldsymbol{H}\right)_i \boldsymbol{\sigma}_{k+1}^2(i) + \tilde{b}_b \tag{4.17}$$

4.3.2. Optimization of the approximate p.d.f. q_s. As for γ_b, the approximation of q_s is performed in the family of Gamma distributions. Hence at the initialization step we assume that

$$q_s^0(\gamma_s) = \mathcal{G}(a_s^0, b_s^0)$$

and at iteration $k+1$, thanks again to Equation (3.2) we obtain

$$\begin{aligned} q_s^{k+1}(\gamma_s) &\propto \exp\left[< \ln \Pr(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \gamma_b, \gamma_s | I) >_{\prod_j q_j^{k+1}(x_j) \prod_j \tilde{q}_j^{k+1}(z_j) q_b^{k+1}(\gamma_b)}\right] \\ &\propto \exp\left[< (\frac{N}{2} + \tilde{a}_s - 1)\ln(\gamma_s) - \gamma_s(\frac{1}{2}\sum_{i=1}^{N} z_i x_i^2 + \tilde{b}_s) >_{\prod_j q_j^{k+1}(x_j)\tilde{q}_j^{k+1}(z_j)}\right] \\ &\propto \exp\left[< (\frac{N}{2} + \tilde{a}_s - 1)\ln(\gamma_s) - \gamma_s(\frac{1}{2}\sum_{i=1}^{N} \frac{\boldsymbol{a}_{k+1}(i)}{\boldsymbol{b}_{k+1}(i)}(\boldsymbol{m}_{k+1}^2(i) + \boldsymbol{\sigma}_{k+1}^2(i)) + \tilde{b}_s)\right] \end{aligned}$$

So $q_s^{k+1}(\gamma_s)$ is a Gamma p.d.f. and its parameters are deduced by identification.

$$a_s^{k+1} = \frac{N}{2} + \tilde{a}_s \tag{4.18}$$

$$b_s^{k+1} = \frac{1}{2}\sum_{i=1}^{N} \frac{\boldsymbol{a}_{k+1}(i)}{\boldsymbol{b}_{k+1}(i)}(\boldsymbol{m}_{k+1}^2(i) + \boldsymbol{\sigma}_{k+1}^2(i)) + \tilde{b}_s \tag{4.19}$$

Finally the algorithm performed can be summed up as follows.

Algorithm 4 UnSupervised Sparse Reconstruction algorithm (USSR)

1: INITIALIZE$(q^0, \tilde{q}^0, q_b^0, q_s^0)$
2: **repeat**
3: **function** ESTIMATE $\tilde{q}^{k+1}(\boldsymbol{z})(q^k(\boldsymbol{x}), q_b^k(\gamma_b), q_s^k(\gamma_s))$
4: update $\boldsymbol{a}_{k+1}$ using Equation (4.7)
5: update $\boldsymbol{b}_{k+1}$ using Equation (4.8)
6: **end function**
7: **function** ESTIMATE $q^{k+1}(\boldsymbol{x})(\tilde{q}^{k+1}(\boldsymbol{z}), q_b^k(\gamma_b), q_s^k(\gamma_s))$
8: compute $q^r(\boldsymbol{x}) \leftarrow (\boldsymbol{m}_r, \boldsymbol{\sigma}_r^2)$ using Equation (4.12) and Equation (4.11)
9: compute α_{subopt}
10: compute $q^\alpha(\boldsymbol{x}) \leftarrow (\boldsymbol{m}_\alpha, \boldsymbol{\sigma}_\alpha^2)$ using Equation (4.14) and Equation (4.13)
11: **end function**
12: **function** ESTIMATE $q_b^{k+1}(\gamma_b)(\tilde{q}^{k+1}(\boldsymbol{z}), q^{k+1}(\boldsymbol{x}))$
13: update a_b^{k+1} using Equation (4.16)
14: update b_b^{k+1} using Equation (4.17)
15: **end function**
16: **function** ESTIMATE $q_s^{k+1}(\gamma_s)(\tilde{q}^{k+1}(\boldsymbol{z}), q^{k+1}(\boldsymbol{x}))$
17: update a_s^{k+1} using Equation (4.18)
18: update b_s^{k+1} using Equation (4.19)
19: **end function**
20: **until** Convergence

5. Simulations. This section is devoted to numerical validations of the method proposed in this paper. For the sake of completeness we will treat two cases. The

first one is given by a noisy tomographic problem in a small dimensional case. This example allows a comparison of our method with classical reconstruction ones. In a second example, we will see a component identification problem in a large dimensional dataset. This second case ensures that the method proposed in this paper is valid for large dimensional cases.

The first inverse problem considered is given by a tomographic example. The goal is to enhance the accuracy and the effectiveness of our approach, by comparison with classical ones, such as classical Variational Bayesian methods or Monte Carlos Markov Chain (MCMC) methods. From the limitations of these concurrent approaches, we choose to consider only a small dimensional inverse problem (4096 unknowns), and thus to invert the Radon transform of a small sparse image (64×64 pixels).

The second experimental result is devoted to a relatively large inverse problem (≈ 300000 unknowns). In this case, the problem is to identify different components in a dictionary learning process. This problem is performed in a very noisy environment, such as the signal to noise ratio can take negative values. This signal processing problem can appear for instance in astrophysical context (detection of gravitational waves [28]) or in radar imaging [39, 1].

In both cases the sparsity information is introduced by an iid Student't prior. This prior is a particular case of GSM. In the following we fix $\tilde{a}_i = \frac{\nu}{2}$ and $\tilde{b}_i = \frac{\nu}{2}$ in Equation (4.4).

5.1. Tomographic example. For the sake of completeness, a short description of the comparative approaches is given, enhancing the main differences between them. In a second part, we describe the phantom together with the simulation parameters and the results.

5.1.1. Algorithms descriptions.

Filtered Back Projection (FBP). Filtered Back Projection is the classical approach to invert the Radon transform [25, 15]. This algorithm is obtained by sampling the continuous inversion formula. Each line of the sinogram (see Fig. 5.1) is filtered with a ramp filter. The filtered data are backprojected. The discrete version of the backprojection operator is given by $\boldsymbol{H}^{\mathrm{t}}$.

Monte Carlos Markov Chain. The MCMC method contains a large class of Bayesian algorithms [31]. In the following we consider the Gibbs algorithm for its efficiency when the size of the problem increases. The principle is to obtain samples of the posterior law given by Equation (4.4) by an alternate sampling with conditional laws. The algorithm is as follows:

(i) $\boldsymbol{z}^k$ sampled with $p(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{x}^{k-1})$
(ii) $\boldsymbol{x}^k$ sampled with $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{z}^k)$
(iii) go to i) until convergence of the Markov chain.

As the conditional law $p(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{x}^{k-1})$ is a separable Gamma distribution, the computation of the sample $\boldsymbol{z}^k$ is easy. Furthermore $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{z}^k)$ is a correlated Gaussian distribution with a covariance matrix $\boldsymbol{R}_k = \boldsymbol{M}_k^{\mathrm{t}}\boldsymbol{M}_k = [\frac{1}{\sigma_b^2}\boldsymbol{H}^{\mathrm{t}}\boldsymbol{H} + \frac{1}{\sigma_s^2}\mathrm{Diag}(\boldsymbol{z}^k)]^{-1}$ and a mean $\boldsymbol{m}_k = \frac{1}{\sigma_b^2}\boldsymbol{R}_k\boldsymbol{H}^{\mathrm{t}}\boldsymbol{y}$. The sampling under this correlated distribution is performed by sampling a vector of centered iid Gaussian random variables with variance 1. Afterward this vector is multiplied by the correlation matrix $\boldsymbol{M}_k$ and added to $\boldsymbol{m}_k$.

REMARK 3. *At each sampling iteration the covariance matrix of size $N \times N$ have to be inverted.*

Classical Bayesian Variational approach. This method was already described in Section 3. In the alternate gradient descent algorithm, one can chose the degree of separability of the approximative distribution. In the following we consider two cases. In the first case, the so called VBBloc, we consider that the separation of the approximating law is only between $\boldsymbol{x}$ and $\boldsymbol{z}$. This leads to consider the approximating distribution as:

$$q(\boldsymbol{x}, \boldsymbol{z}) = q(\boldsymbol{x})\tilde{q}(\boldsymbol{z})$$

and

$$\begin{aligned} q(\boldsymbol{x}) &= \mathcal{N}(\boldsymbol{m}, \boldsymbol{R}) \\ \tilde{q}(\boldsymbol{z}) &= \mathcal{G}(\boldsymbol{a}, \boldsymbol{b}) \end{aligned}$$

Thus, with Equation (3.2), we obtain $\forall i \in \{1, \dots, N\}$ the following updating equations:

$$\begin{aligned} \boldsymbol{a}(i) &= \frac{\nu}{2} + \frac{1}{2}, \\ \boldsymbol{b}_{k+1}(i) &= \frac{\nu}{2} + \frac{\boldsymbol{m}_k^2(i) + \mathrm{diag}(\boldsymbol{R}_k)(i)}{2\sigma_s^2} \\ \boldsymbol{R}_{k+1} &= \left(\frac{1}{\sigma_s^2} \mathrm{Diag}(\boldsymbol{a}/\boldsymbol{b}_{k+1}) + \frac{1}{\sigma_b^2} \boldsymbol{H}^{\mathrm{t}} \boldsymbol{H} \right)^{-1} \\ \boldsymbol{m}_{k+1} &= \frac{1}{\sigma_b^2} \boldsymbol{R}_{k+1} \boldsymbol{H}^{\mathrm{t}} \boldsymbol{y}. \end{aligned}$$

REMARK 4. *At each step, the updating of the covariance matrix requires the inversion of a $N \times N$ matrix, but the convergence rate is better than for the MCMC approach.* To overcome the limit given by a matrix inversion in the classical variational Bayesian framework, we can construct an approximative distribution separable on $\boldsymbol{x}$. Hence, we estimate a vector of variance instead of the matrix of covariance. This approach is called VBComp in the following.

$$q(\boldsymbol{x}, \boldsymbol{z}) = \Pi_i q_i(x_i)\tilde{q}(\boldsymbol{z})$$

In this case Equation (3.2) give the following updating equations, $\forall i \in \{1, \dots, N\}$:

$$\begin{aligned} \boldsymbol{a}(i) &= \frac{\nu}{2} + \frac{1}{2}, \\ \boldsymbol{b}_{k+1}(i) &= \frac{\nu}{2} + \frac{\boldsymbol{m}_k^2(i) + \boldsymbol{\sigma}_k^2(i)}{2\sigma_s^2} \end{aligned}$$

And, for every $i \in \{1, \dots, N\}$:

$$\begin{aligned} \boldsymbol{\sigma}_{k+1}^2(i) &= \left(\frac{1}{\sigma_s^2} \boldsymbol{a}(i)/\boldsymbol{b}_{k+1}(i) + \frac{1}{\sigma_b^2} (\boldsymbol{H}^{\mathrm{t}} \boldsymbol{H})_{(i,i)} \right)^{-1} \\ \boldsymbol{m}_{k+1}(i) &= \frac{\boldsymbol{\sigma}_{k+1}^2(i)}{\sigma_b^2} (\boldsymbol{H}^{\mathrm{t}} \boldsymbol{y}(i) - (\boldsymbol{d}(i) - (\boldsymbol{H}^{\mathrm{t}} \boldsymbol{H})_{(i,i)} \boldsymbol{m}_k(i))) \\ \boldsymbol{d} &= \boldsymbol{H}^{\mathrm{t}} \boldsymbol{H} \boldsymbol{m}_k \end{aligned}$$

REMARK 5. *For each pixel x_i, the corresponding value of $\boldsymbol{d} = \boldsymbol{H}^{\mathrm{t}} \boldsymbol{H} \boldsymbol{m}_k$ must be determined.*

Coordinate	(28,28)	(25,28)	(28,25)	(40,28)	(32,38)	(48,48)	(8,52)
Value	1	1	1	0.5	0.7	0.8	0.6

TABLE 5.1
Peaks definition in the phantom

5.1.2. Simulation configuration. The test image is given by a sparse phantom, composed of 7 peaks on a grid 64×64 (see Tab. 5.1 and Fig. 5.2(a)). We have simulated data in a parallel beam geometry. These projections are collected from 32 angles θ, uniformly spaced over $[0, 180[$. Each projection is composed of 95 detector cells. We add a white Gaussian noise (iid) with standard deviation equal to 0.3 (see Fig. 5.1). Data have thus a relatively bad signal to noise ratio and the number of unknowns is larger than the number of data, which leads to an ill-posed inverse problem.

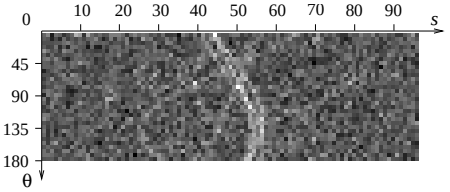


FIG. 5.1. *Data collected : sinogram composed of 32 angles and 95 detector cells.*

5.1.3. Results and discussion. In this section, we expose the inversion of a tomographic problem with the approaches described earlier. All the iterative approaches are initialized with a zero mean and a variance equal to one, and the hyperparameters σ_b^2, σ_s^2 and ν are respectively fixed to 1, 0.05 and 0.1. The original image and its different reconstructions are summed up on Fig. 5.2. A comparison of Fig. 5.2 (b) with 5.2 (c), 5.2 (d) and 5.2 (e) clearly shows that the analytical inversion of the Radon transform performed by Filtered Back Projection (FBP) is less robust to noise than Bayesian approaches. Asymptotically, in Bayesian cases, theoretical results are favorable to the MCMC approach, as it does not need any approximation. In practice, the number of samples is too small to fit with the asymptotic results of MCMC methods, which explains the bad reconstruction observed in Fig. 5.2(c). Finally, the Supervised Sparse Reconstruction (SSR) (see Fig. 5.2(f)) has the same reconstruction quality than the classical variational Bayesian approaches (see VBBloc Fig. 5.2(d) and VBComp Fig. 5.2(e)). However when we compare the execution time (see Tab. 5.2), we see that our approach is 10 time faster than the VBBloc approach, 40 time faster than the VBComp approach and 370 faster than the MCMC approach for this small inverse problem. Moreover this ratio increases with the size of the problem as both MCMC and classical variational Bayesian need the inversion of a covariance matrix at each iteration, which is not the case for our algorithm.

5.1.4. Hyperparameters estimation. As seen in the section 4.2, our approach is defined in a fully Bayesian framework. We thus estimate the values of hyperparameters in introducing a non informative Jeffrey's prior, as described in Part 4.3. We estimate thus the trade off between the likelihood and the prior through the estimation of σ_b^2 and σ_s^2. Hence, we apply the algorithm UnSupervised Sparse Reconstruction (USSR) (see Algorithm 4) in our tomographic dataset. As for the previous simulation,

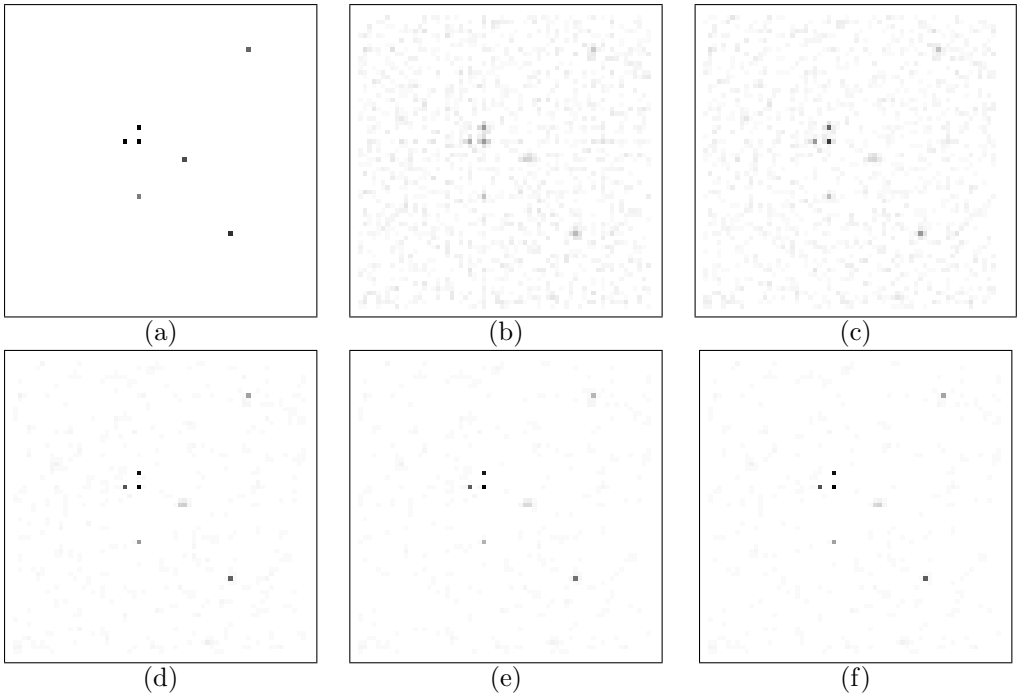


FIG. 5.2. *Images are presented with the same inversed grayscale: (a) true image of 7 peaks, (b) FBP with ramp filter, (c) MCMC Gibbs approach, (d) classical variational Bayesian (VBBloc) with bloc optimization, (e) classical variational Bayesian (VBComp) with component optimization, (f) SSR approach.*

TABLE 5.2
Comparison of the different approaches: computing time (second) and quality of estimation (SNR).

Method	FBP	VBBloc	VBComp	VBGrad (SSR)	MCMC Gibbs
CPU time (s)	0.05	586.20	1759.1	44.41	37079.50
Nb of iterate	1	15	8(×4096)	500	1000
SNR	-2.04	5.87	5.76	6.00	-0.60

the initial values of the mean are fixed to zero and the variance are fixed to one. For the hyperparameters σ_b^2 and σ_s^2 the initial values are respectively fixed to 1 and 0.05 to begin with a prior information more important than the likelihood.

The results are summed up in Fig. 5.3. We observe that the hyperparameters estimation intensifies the sparsity of the reconstructed image together with the SNR, as it goes from 6.00 db in the previous case to 10.06 db. Estimating the true hyperparameters is in this case more relevant than arbitrarily chosen parameters. We observe on Fig. 5.3 (c) that the background is equal to zero even if some additional peaks appear in the reconstructed image

Finally we see on Fig. 5.3, (d) and (e), the behavior of the estimation of the hyperparameters respectively to the number of iterations. This plot is in a logarithm scale due to the dynamic of the fluctuations. We observe that for σ_b^2 this estimation converges to the true value (dashed line). For σ_s^2 we do not known the true value, but one can notice that this estimation converges also to some given value.

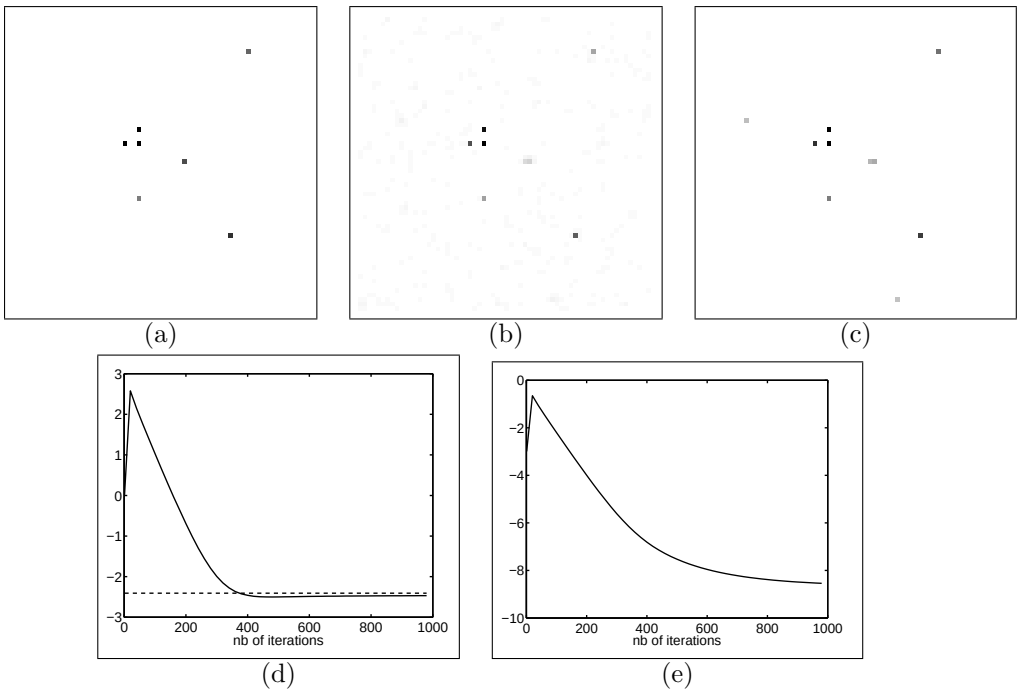


FIG. 5.3. *Results with hyperparameters estimation: (a) True image, (b) reconstructed image with SSR algorithm (hyperparameters are fixed), (c) reconstructed image with USSR (image and hyperparameters are estimated jointly), (d) logarithm of σ_b^2 during the iterations the dashed line correspond to the true value, (e) logarithm of σ_s^2 during the iterations*

5.2. Component identification (denoising). As enhanced by the previous simulation, our method can be more efficient than classical ones on relatively small problem (4096 unknowns). But its main advantage is that it allows to address larger problems (≈ 300000 unknowns in the present case). In the present part, our purpose is to treat an identification problem on a dictionary decomposition. More precisely, we identify many chirps function $\psi_k(t)$ in a linear mixture. This identification issue appears for instance in astrophysics, in order to detect gravitational waves [28] or in radar imaging [39, 1]. Unfortunately, this mixture is corrupted by noise and spurious signal. To identify each chirps in the mixture and to remove the effect of the spurious signal, we develop a dictionary learning approach. To build our dictionary we make the following assumptions : all chirps have the same duration T_{Chirp}, the chirps rate ζ is digitalized on very few values (eight in the present case), the spurious signal can be represented on very few coefficients of Discrete Fourier Transform and the noise is white and Gaussian. However, we do not make any assumption on the variance of the noise or on the number of chirp functions in the mixture.

In the following section, we present the formalism of the dictionary learning approach. After that, we expose the simulation condition. Then, we illustrate the efficiency of our approach with three numerical experiments. In the first one, we consider a nominal case to study the quality of estimation. After what, we study a limit case with a very high level of noise (SNR at -5 db), in order to illustrate the robustness of our approach with regard to noise. In the last experiment, we study the behavior

of the reconstruction quality when the number of chirp functions or the noise level increase.

5.2.1. Dictionary decomposition. In this section the signal considered, denoted by $s(t)$, is obtained by the following dictionary decomposition.

$$s(t) = \sum_{i=1}^{N_{freq}} (u_i + jv_i)\phi_i(t) + \sum_{l=1}^{N_{trans}} \sum_{k=1}^{N_{rate}} c_{l,k}\psi_k(t - l\Delta_t), \tag{5.1}$$

where

$$\phi_i(t) = \exp\left[2j\pi f_i t\right],$$

corresponds to a pure frequency f_i with $j^2 = -1$ whereas

$$\psi_k(t) = cos(2\pi(f_0 t + \frac{1}{2}t^2\zeta_k))\Pi_{0,T_{Chirp}}(t)$$

corresponds to the chirps components. Here, f_0 is the starting frequency (at time $t = 0$), ζ_k is the chirp rate, that is the increasing rate of the frequency, $\Pi(t)$ is a gate function, T_{Chirp} is the duration of the chirp, Δ_t is the shift between two chirps functions and $t_l = l\Delta_t$ is the first time where the $\psi_k(t - l\Delta_t)$ is not null. We merge all the dictionary's coefficients in a single vector $\boldsymbol{x} = (u_1, \ldots, u_{Nfreq}, v_1, \ldots, v_{Nfreq}, c_{1,1}, \ldots, c_{Ntrans,Nrate})^{\mathrm{t}}$. Where $Nfreq$ is the number of pure frequency functions contained on the dictionary whereas $Ntrans$ is the number of chirp shifts and $Nrate$ is the number of chirp rate. Moreover, we store sampled version of functions ϕ_i and $\psi_{l,k}$ into a matrix $\boldsymbol{H}$. Hence, the sampled version of Equation (5.1) is given by

$$\boldsymbol{s} = \boldsymbol{H}\boldsymbol{x}, \tag{5.2}$$

where $\boldsymbol{s} = (s(t_0), \ldots, s(t_N))^{\mathrm{t}}$.

The measurement of the signal $\boldsymbol{s}$ is also corrupted by an iid Gaussian noise $\boldsymbol{b}$ with a variance σ_b^2, such that the observations are given by

$$\boldsymbol{y} = \boldsymbol{s} + \boldsymbol{b} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{b}. \tag{5.3}$$

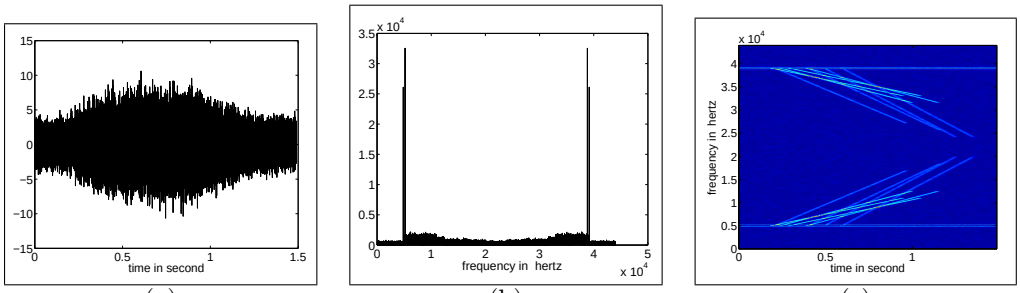
In the following, the dictionary is composed of chirp functions with only 8 different chirp rates ($N_{rate} = 8$). The frequency f_0 is equal to 5 000 hz, the chirp rates ζ_k are uniformly spaced between 6 000 and 20 000 hz, the shift parameters Δ_t is fixed up to a sampling period ($T_e = 1/F_e$). Finally, the duration of the chirp (T_{Chirp}) is equal to the half time of the measurements (T_{mes}).

REMARK 6. *Our dictionary is redundant as the number of coefficients is 4.5 times greater than the number of observations. Thus, the component identification is an ill-posed problem.*

5.2.2. Common simulation conditions. Simulated data are composed by the sum of N_{Cos} cosine functions with different frequencies and N_{Chirp} chirp functions taken in our dictionary. The simulated data are sampled at a frequency $F_e = 44$khz, and they are composed of $N = 2^{16}$ points, thus the duration of measurement T_{mes} is equal to 1.5 second.

TABLE 5.3
Parameters of the different components

type of function	Amplitude	frequency (hz)	rate (hz)	first time (s)
cosine	1	5169.7	-	-
cosine	0.8	4834	-	-
Chirp	1.4	-	8000	0.2
Chirp	1.4	-	10000	0.25
Chirp	1.0	-	16000	0.22
Chirp	1.0	-	20000	0.5
Chirp	1.2	-	10000	0.4
Chirp	1.0	-	18000	0.41
Chirp	1.0	-	20000	0.6
Chirp	1.4	-	8000	0.3



(a) (b) (c)

FIG. 5.4. *Data in different fields: (a) time, (b) frequency, (c) time-frequency*

5.2.3. First result. The simulated data are composed of two cosine functions and eight chirp functions, the parameters of all this functions are given in Tab. 5.3. Once again simulated data have a relatively bad SNR, at 5.68 db in this case. They are plotted in Fig. 5.4 (a), whereas their Fourier transform are given in Fig. 5.4 (b) and their time-frequency transform, computed with a STFT (Short Time Fourier Transform) are on Fig. 5.4(c). We can see in the time frequency representation that there are overlaps between the different components making the identification harder.

Moreover this inverse problem is treated with the unsupervised approach given by Algorithm 4. Our algorithm was launched with the shape parameter of the Student't equals to $\nu = 0.01$ in order to introduce a very sparse prior. The initialization parameters are:

- The mean of $q^0(\boldsymbol{x})$, $\boldsymbol{m}_0 = 0$,
- the variance of $q^0(\boldsymbol{x})$, $\boldsymbol{\sigma}_0^2 = 1$,
- the mean of $q_b^0(\gamma_b)$, is equal to 10^{-5},
- the mean of $q_s^0(\gamma_s)$, is equal to 10^5.

After 400 iterations (316 s), the algorithm (USSR) converges to a stable solution. It gives parameters of different approximated laws. We consider here that the estimation $\hat{\boldsymbol{x}}$ is obtained by taking the posterior mean for the coefficients (see Fig. 5.5).

Fig. 5.5 (a) represents the real part of the Fourier coefficients (the vector $\boldsymbol{u}$ is the Equation (5.1)). We recognize the Fourier transform of the two cosine functions. Moreover, when we compare Fig. 5.5 (a) with the Fourier transform of the data, Fig.

5.4 (b), we observe that the algorithm USSR selected only the sparse components of the Fourier transform. We plot on Figs 5.5 (b-i) the chirp coefficients $c_{l,k}$ for eight chirp rates (6000 until 20000) with respect to the first time t_l where the chirp does not vanishes. In these figures the estimated values are plotted with a line and the true value of non null coefficients are marked by a cross. All chirp coefficients have been reconstructed in the right place, and their amplitudes are very close to the real ones.

Estimation $\hat{\boldsymbol{s}}$ of s is performed thanks to the estimation of the coefficients $\hat{\boldsymbol{x}}$ and Equation (5.2). The SNR of $\hat{\boldsymbol{s}}$ is equal to 22.6 db. It is a very good result knowing that the SNR of data $\boldsymbol{y}$ is equal to 5.68 db.

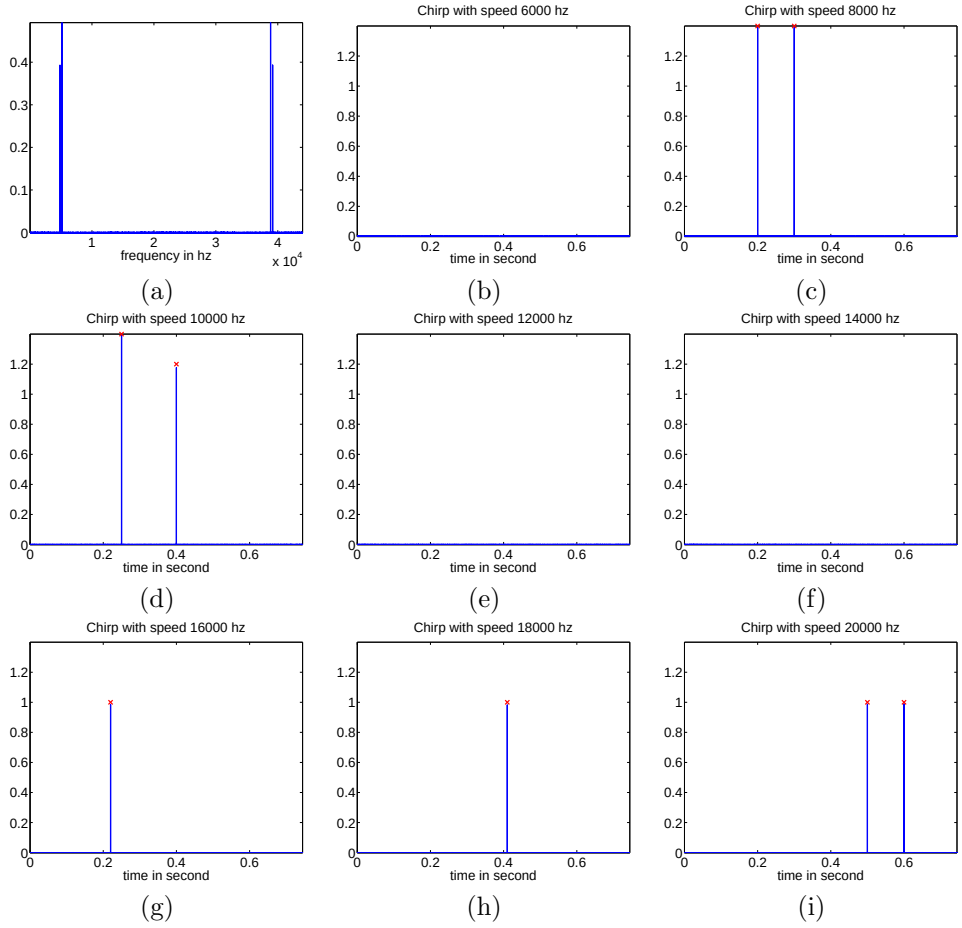


FIG. 5.5. *Dictionary decomposition: true values of coefficients are marked by a cross*

5.2.4. Limit case. This section illustrates the performance of the algorithm USSR when the signal s is hidden by the noise, the SNR of data being equal to -5 db (see Fig. 5.6 (a)). We generate a signal composed of four chirps Equation (5.1) which parameters are summed up in Tab. 5.4.

The estimation of coefficients is performed using our USSR algorithm with the same initialization as in the previous case. After 400 iterations we obtain the coefficients illustrated on Fig. 5.6, (c) and (d). The reconstructed coefficients are

TABLE 5.4
Parameters of the different components

type of function	Amplitude	Chirp rate (hz)	first time (s)
Chirp	0.9526	16000	0.1530
Chirp	1.1606	16000	0.1945
Chirp	0.7369	18000	0.2000
Chirp	1.1724	18000	0.1865

represented by a line and the true values of the coefficients are marked by a cross. We observe that all coefficients are in the right place and that the peaks amplitudes are systematically underestimated, but the estimated values are relatively close to the true ones. Fig. 5.6 (b) points out the estimator $\hat{\boldsymbol{s}}$. We see that the shape of the signal is close to the true one, and that when the signal is missing, between 1 and 1,45 s, the level of the residual noise is relatively low.

TABLE 5.5
Signal to Noise Ratio (SNR) in db

data	USSR approach	best Wiener filter	best wavelet soft threshold
-5.0	15.05	1.1941	1.8677

In Tab. 5.5 we compare the reconstructed signal $\hat{\boldsymbol{s}}$ with the reconstruction obtained with two classical denoising approaches, namely the Wiener filter and the soft wavelet shrinkage, with the four vanishing moments symmlet. In these two methods, we furthermore have to tune a parameter which is the correlation power for the Wiener filter and the threshold for the soft wavelet shrinkage. Hence we choose the value of this hyperparameter which minimizes the Signal to Noise Ratio (SNR). Unlike in the USSR approach, we thus have to we know the true signal in order to tune this parameter. Furthermore, our approach increases hugely the SNR (20 db), thus the noise level is divided by 100 whereas the classical methods reduce the noise only by a factor 4 or 5. This example enhances the efficiency of the dictionary decomposition in the denoising context.

5.2.5. Behavior of our method versus level of noise and number of components. In this part, we perform the study of the robustness of the USSR algorithm regarding Signal to Noise Ratio. This simulation allows a better understanding of the reconstruction properties of our method.

Simulations. In the following we consider simulated data with 6 different SNR $(-5, -2, 1, 2, 5, 10)$ and with low (4) and high (16) number of components. For each SNR and each number of components, we simulate 30 set of data, the components of the signal being randomly chosen between 1 and 9. If the number is equal to 1, we consider a cosine function with a frequency f_i is uniformly taken between 0 and the sampling frequency (F_e= 44 khz), and an amplitude randomly chosen between 0.6 and 1.4. If the number is equal to 2 resp. $3, \ldots, 9$, we simulate a chirp function with a chirp rate ζ_k equal to 6000 resp. $8000, \ldots, 20000$. The first time of the chirp t_l is uniformly taken between 0.1 and 0.6 second, and as previously, the amplitude $c_{l,k}$ is randomly chosen between 0.6 and 1.4.

Reconstruction. We reconstruct this 360 sets of data by the algorithm USSR taking the same configuration and the same initialization as in previous cases.

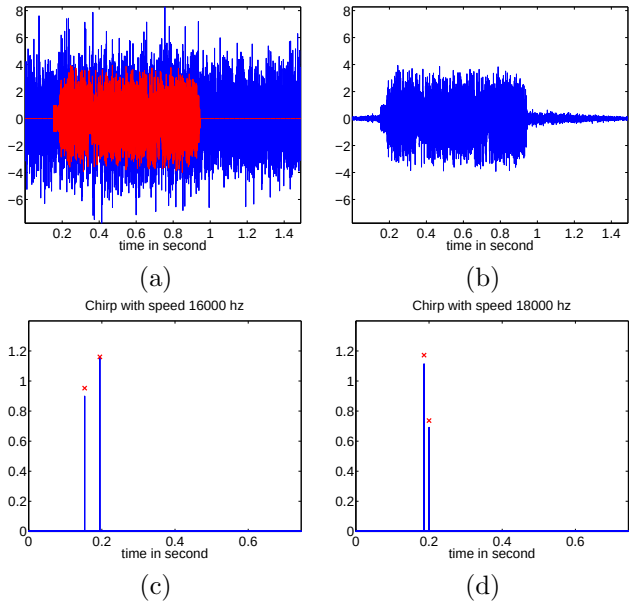


FIG. 5.6. *Limit case: (a) data and true signal, (b) reconstructed signal*

The results are summed up on Fig 5.7 (a-d). Each point of the plot is computed by averaging the result of 30 reconstructions. In Fig 5.7 (a), resp. (b), we plot the true positive proportion, resp. false positive proportion, of the significant reconstructed coefficients[1]. At first sight we see that there are no false positive with our approach. Indeed, as the approach is unsupervised with a sparsity prior, the coefficients with low energy are considered as noise. Moreover we can reconstruct 16 components without lost when the SNR is greater or equal to 5 db and resp. 4 components when the SNR is greater to 1 db. Fig. 5.7 shows that the reconstruction is more difficult when the number of components increases.

Fig. 5.7 (c) is obtained by calculating the SNR of the reconstructed signal $\hat{\boldsymbol{s}}$. We observe a quite linear behavior. For 4 components the gain is of 17 db whereas for 16 components we gain 11.5 db. Finally, Fig. 5.7 (d) exposes the quadratic error of the peaks amplitude. There are here two cases. When all the components are found this error is linear (see Fig. 5.7 (d) the bottom curve when SNR>1) but it increases more rapidly when some components are not found (see Fig. 5.7 (d) the bottom curve when SNR<1).

6. Conclusion. In this paper, we have defined an iterative algorithm based on the descent gradient principle and adapted to the context of variational Bayesian methods. The main interest of this algorithm is that it converges faster than the classical Bayesian methods and allows an use on large dimensional datasets. We have furthermore give its implementation in the case of white noise model when the prior information enhances some sparse behavior. A small tomographic application allows us to compare our method with classical ones. We see that even in small cases, our algorithm can be faster than classical ones. A second simulation part, corresponding

[1]the significant coefficients are obtained by applying a threshold equal to 0.2 on the coefficients vector. This threshold is equal to the third of the minimum value of the true non zero coefficients.

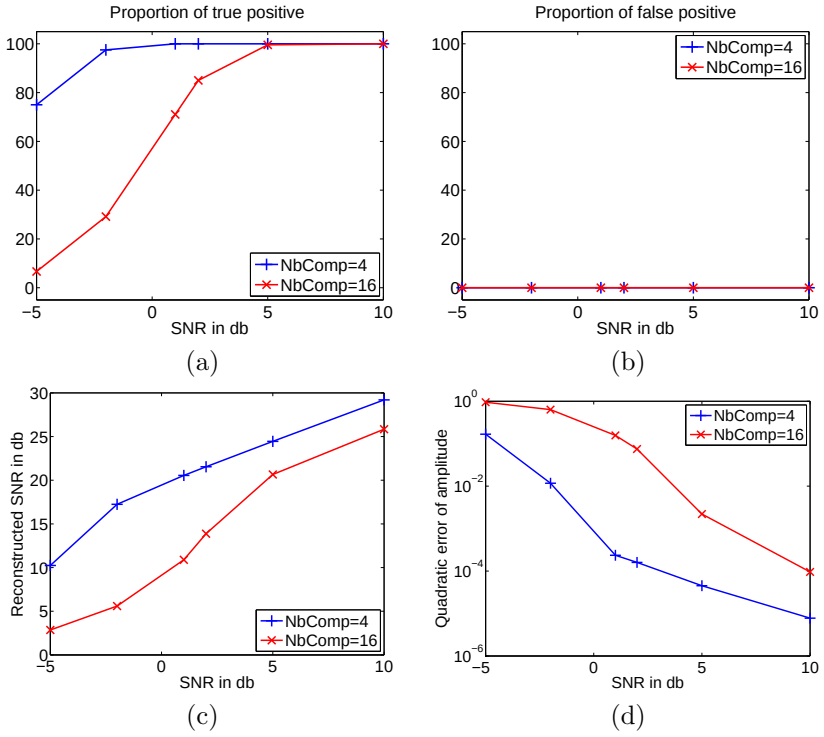


FIG. 5.7. *Study of Robustness versus noise, each point of the curve being is calculated by averaging 30 reconstructions with components randomly chosen: (a) True positive proportion of the significant reconstructed coefficients, (b) False positive proportion of the significant reconstructed coefficients, (c)* SNR *of the estimated signal* $\hat{\boldsymbol{s}}$*, (d) Quadratic error of the peaks amplitude*

to a dictionary identification allows to understand the behavior of our method for large dimensional problems. Once again this method has good reconstruction properties in this case.

7. Annexe.

7.1. Proof of Lemma 2.3. In order to ensure that for small values of α, we have $F(\mu^\alpha) > F(\mu^k)$, we show that the right part of Equation (2.15) is positive.

Let the notations be given by Section 2.

First, one can notice that

$$\begin{aligned}
\mathrm{d}F_{\mu^k}(\mu^\alpha - \mu^k) &= \int_{\mathbb{R}^N} \mathrm{d}f(\mu^k, \boldsymbol{x})(h_\alpha(\boldsymbol{x}) - 1)\mathrm{d}\mu^k(\boldsymbol{x}) \\
&= \frac{1}{\alpha}\int_{\mathbb{R}^N} (\alpha \mathrm{d}f(\mu^k, \boldsymbol{x}) + \ln K_k(\alpha))(h_\alpha(\boldsymbol{x}) - 1)\mathrm{d}\mu(\boldsymbol{x}) - \int_{\mathbb{R}^N} \frac{\ln K_k(\alpha)}{\alpha}(h_\alpha(\boldsymbol{x}) - 1)\mathrm{d}\mu^k(\boldsymbol{x}) \\
&= \frac{1}{\alpha}\int_{\mathbb{R}^N} (\alpha \mathrm{d}f(\mu^k, \boldsymbol{x}) + \ln K_k(\alpha))(h_\alpha(\boldsymbol{x}) - 1)\mathrm{d}\mu^k(\boldsymbol{x}),
\end{aligned}$$

as $\frac{\ln K_k(\alpha)}{\alpha}$ is constant and $\int_{\mathbb{R}^N}(h_\alpha(\boldsymbol{x}) - 1)\mathrm{d}\mu(\boldsymbol{x}) = 0$.

For the second term one has

$$-\int_{\mathbb{R}^N} \ln(h_\alpha(\mu^k, \boldsymbol{x}))\mathrm{d}\mu^\alpha(\boldsymbol{x}) = -\int_{\mathbb{R}^N} (\alpha \mathrm{d}f(\mu^k, \boldsymbol{x}) + \ln K_k(\alpha))h_\alpha \mathrm{d}\mu^k(\boldsymbol{x}) = -\alpha\int_{\mathbb{R}^N} \mathrm{d}f(\mu^k, \boldsymbol{x})h_\alpha \mathrm{d}\mu^k(\boldsymbol{x}) - \ln K_k(\alpha).$$

But Jensen's inequality ensures that

$$\ln K_k(\alpha) \leq -\alpha \int_{\mathbb{R}^N} \mathrm{d}f(q^k, \boldsymbol{x}) \mathrm{d}\mu^k(\boldsymbol{x}). \tag{7.1}$$

Which leads to

$$-\int_{\mathbb{R}^N} \ln(h_\alpha(\mu^k, \boldsymbol{x})) \mathrm{d}\mu^\alpha(\boldsymbol{x}) \geq -\alpha \int_{\mathbb{R}^N} \mathrm{d}f(\mu^k, \boldsymbol{x})(h_\alpha(\boldsymbol{x}) - 1) \mathrm{d}\mu^k(\boldsymbol{x}) \tag{7.2}$$

Finally,

$$\begin{aligned}
&\mathrm{d}F_{\mu^k}(\mu^\alpha - \mu^k) - L\|h_\alpha - 1\|^2_{L^2(\mu^k)} - \int_{\mathbb{R}^N} (\alpha \mathrm{d}f(\mu^k, \boldsymbol{x}) + \ln K_k(\alpha)) \mathrm{d}\mu^\alpha(\boldsymbol{x}) \\
&\geq \int_{\mathbb{R}^N} \left((\alpha \mathrm{d}f(\mu^k, \boldsymbol{x}) + \ln K_k(\alpha))(\frac{1}{\alpha} - 1) - L(h_\alpha(\boldsymbol{x}) - 1) \right) (h_\alpha(\boldsymbol{x}) - 1) \mathrm{d}\mu^k(\boldsymbol{x}) \\
&= \int_{\{x : \alpha \mathrm{d}f(\mu^k, \boldsymbol{x}) + \ln K_k(\alpha) \geq 0\}} \left((\alpha \mathrm{d}f(\mu^k, \boldsymbol{x}) + \ln K_k(\alpha))(\frac{1}{\alpha} - 1) - L(h_\alpha(\boldsymbol{x}) - 1) \right) (h_\alpha(\boldsymbol{x}) - 1) \mathrm{d}\mu^k(\boldsymbol{x}) \\
&+ \int_{\{x : \alpha \mathrm{d}f(\mu^k, \boldsymbol{x}) + \ln K_k(\alpha) < 0\}} \left((\alpha \mathrm{d}f(\mu^k, \boldsymbol{x}) + \ln K_k(\alpha))(\frac{1}{\alpha} - 1) - L(h_\alpha(\boldsymbol{x}) - 1) \right) (h_\alpha(\boldsymbol{x}) - 1) \mathrm{d}\mu^k(\boldsymbol{x})
\end{aligned} \tag{7.3}$$

Let us consider each integrals appearing in Eq. (7.3) separately. First, let us notice that if $\alpha \mathrm{d}f(\mu^k, \boldsymbol{x}) + \ln K_k(\alpha) < 0$ then so is $h_\alpha - 1$. Furthermore, for every $\alpha > 0$ and $\boldsymbol{x} \in \mathbb{R}^N$, we have $h_\alpha - 1 \geq \alpha \mathrm{d}f(\mu^k, \boldsymbol{x}) + \ln K_k(\alpha)$. Hence if $\boldsymbol{x} \in \mathbb{R}^N$ is such that $\alpha \mathrm{d}f(\mu^k, \boldsymbol{x}) + \ln K_k(\alpha) < 0$ then

$$\left((\alpha \mathrm{d}f(\mu^k, \boldsymbol{x}) + \ln K_k(\alpha))(\frac{1}{\alpha} - 1) - L(h_\alpha(\boldsymbol{x}) - 1) \right) (h_\alpha(\boldsymbol{x}) - 1) \geq (h_\alpha(\boldsymbol{x}) - 1)^2 \left(\frac{1}{\alpha} - 1 - L \right),$$

which is positive as soon as $\alpha \leq \frac{1}{1+L}$.

Consider now that $\boldsymbol{x} \in \mathbb{R}^N$ is such that $\alpha \mathrm{d}f(\mu^k, \boldsymbol{x}) + \ln K_k(\alpha) \geq 0$. The Mean Value Theorem applied to the function exponential ensures that one can find, for every $\boldsymbol{x}$ and α a $\theta \in (0, \alpha \mathrm{d}f(\mu^k, \boldsymbol{x}) + \ln K_k(\alpha))$ such that

$$h_\alpha(\mu^k, \boldsymbol{x}) = e^{\alpha \mathrm{d}f(\mu^k, \boldsymbol{x}) + \ln K_k(\alpha)} = 1 + (\alpha \mathrm{d}f(\mu^k, \boldsymbol{x}) + \ln K_k(\alpha)) e^\theta.$$

This entails that

$$\left((\alpha \mathrm{d}f(\mu^k, \boldsymbol{x}) + \ln K_k(\alpha))(\frac{1}{\alpha} - 1) - L(h_\alpha(\boldsymbol{x}) - 1) \right) = (\alpha \mathrm{d}f(\mu^k, \boldsymbol{x}) + \ln K_k(\alpha))(\frac{1}{\alpha} - 1 - Le^\theta).$$

Furthermore, Jensen's inequality ensures that

$$0 \leq \alpha \mathrm{d}f(\mu^k, \boldsymbol{x}) + \ln K_k(\alpha) \leq \alpha \left(\mathrm{d}f(\mu^k, \boldsymbol{x}) - \int_{\mathbb{R}^N} \mathrm{d}f(\mu^k, \boldsymbol{x}) \mathrm{d}\mu^k(\boldsymbol{x}) \right).$$

Thus

$$\mathrm{d}f(\mu^k, \boldsymbol{x}) - \int_{\mathbb{R}^N} \mathrm{d}f(\mu^k, \boldsymbol{x}) \mathrm{d}\mu^k(\boldsymbol{x}) \geq 0.$$

And

$$1 \leq e^{\theta} \leq h_\alpha(\mu^k, \boldsymbol{x}) \leq e^{\alpha\left(\mathrm{d}f(\mu^k,\boldsymbol{x}) - \int_{\mathbb{R}^N} \mathrm{d}f(\mu^k,\boldsymbol{x})\mathrm{d}\mu^k(\boldsymbol{x})\right)},$$

which leads to

$$\frac{1}{\alpha} - 1 - Le^{\alpha\left(\mathrm{d}f(\mu^k,\boldsymbol{x}) - \int_{\mathbb{R}^N} \mathrm{d}f(\mu^k,\boldsymbol{x})\mathrm{d}\mu^k(\boldsymbol{x})\right)} \leq \frac{1}{\alpha} - 1 - Le^{\theta} \leq \frac{1}{\alpha} - 1 - L.$$

Concerning the left part of the previous equation one can notice that the function of α defined here is such that there exists a value $\alpha_0 > 0$ such that for every $\alpha \leq \alpha_0$,

$$\frac{1}{\alpha} - 1 - Le^{\alpha\left(\mathrm{d}f(\mu^k,\boldsymbol{x}) - \int_{\mathbb{R}^N} \mathrm{d}f(\mu^k,\boldsymbol{x})\mathrm{d}\mu^k(\boldsymbol{x})\right)} \geq 0.$$

Finally one has that there exists $\alpha_0 \geq 0$ such that

$$\forall \alpha \leq \alpha_0 \;\; \mathrm{d}F_{\mu^k}(\mu^\alpha - \mu^k) - L\|h_\alpha - 1\|^2_{L^2(\mu^k)} - \int_{\mathbb{R}^N} (\alpha \mathrm{d}f(\mu^k, \boldsymbol{x}) + \ln K_k(\alpha))\mathrm{d}\mu^\alpha(\boldsymbol{x}) \geq 0. \tag{7.4}$$

7.2. Optimization of the parameter α. In order to obtain the optimal value of $\alpha > 0$, we have to optimize

$$g_k(\alpha) = F(q^\alpha) = \int_{\mathbb{R}^N}\int_{\mathbb{R}^N} \ln p(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) q^\alpha(\boldsymbol{x})\tilde{q}^k(\boldsymbol{z})\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{z} - \int_{\mathbb{R}^N}\int_{\mathbb{R}^N} \ln(q^\alpha(\boldsymbol{x})\tilde{q}^k(\boldsymbol{z}))q^\alpha(\boldsymbol{x})\tilde{q}^k(\boldsymbol{z})\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{z}.$$

Let us compute $F(q^\alpha)$. To achieve this, we first notice that

$$\ln p(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) = \tilde{C} - \frac{\boldsymbol{x}^T\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{x} - 2\boldsymbol{x}^T\boldsymbol{H}^T\boldsymbol{y} + \boldsymbol{y}^T\boldsymbol{y}}{2\sigma_b^2} + \sum_i \frac{1}{2}\ln(z_i/\sigma_s^2) - \frac{z_i x_i^2}{2\sigma_s^2} + (\tilde{a}_i - 1)\ln(z_i) - z_i\tilde{b}_i,$$

where $\tilde{C}$ is a positive constant. Thus

$$\begin{aligned}\int_{\mathbb{R}^N}\int_{\mathbb{R}^N} \ln p(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) q^\alpha(\boldsymbol{x})\tilde{q}^k(\boldsymbol{z})\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{z} = \tilde{C} &- \int_{\mathbb{R}^N}\int_{\mathbb{R}^N} \frac{\boldsymbol{x}^T\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{x} - 2\boldsymbol{x}^T\boldsymbol{H}^T\boldsymbol{y}}{2\sigma_b^2} \prod q_j^\alpha(x_j)\tilde{q}^k(\boldsymbol{z})\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{z} \\ &+ \sum_i \int_{\mathbb{R}} \frac{1}{2}\ln(z_i/\sigma_s^2)\tilde{q}_i^k(z_i)\mathrm{d}z_i + \sum_i \int_{\mathbb{R}} \Big((\tilde{a}_i - 1)\ln(z_i) - z_i\tilde{b}_i\Big)\tilde{q}^k(z_i)\mathrm{d}z_i \\ &- \sum_i \int_{\mathbb{R}^2} \frac{z_i x_i^2}{2\sigma_s}\tilde{q}_i^k(z_i)\mathrm{d}x_i\mathrm{d}z_i\end{aligned}$$

We have thus five terms:

- $$\begin{aligned}A &= -\int_{\mathbb{R}^N\times\mathbb{R}^N} \frac{\boldsymbol{x}^T\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{x}}{2\sigma_b^2}\prod_j q_j^\alpha(x_j)\tilde{q}^k(\boldsymbol{z})\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{z}\\ &= -\frac{1}{2\sigma_b^2}\int_{\mathbb{R}^N} \frac{\boldsymbol{x}^T\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{x}}{2\sigma_b^2}\prod_j q_i j^\alpha(x_j)\mathrm{d}\boldsymbol{x}\\ &= -\frac{1}{2\sigma_b^2}\int_{\mathbb{R}^N}\sum_{l=1}^N\sum_{p=1}^N x_l x_p h_{lp}\prod_j q_j^\alpha(x_j)\mathrm{d}\boldsymbol{x}\\ &= -\frac{1}{2\sigma_b^2}\int_{\mathbb{R}^N}\sum_{l=1}^N\sum_{p\neq l}^N x_l x_p h_{lp}\prod_j q_j^\alpha(x_j)\mathrm{d}\boldsymbol{x} - \frac{1}{2\sigma_b^2}\int_{\mathbb{R}^N}\sum_{p=1}^N x_p^2 h_{pp}\prod_j q_j^\alpha(x_j)\mathrm{d}\boldsymbol{x}\\ &= -\frac{1}{2\sigma_b^2}\left(\sum_{l=1}^N\sum_{p\neq l}^N \boldsymbol{m}_\alpha(l)\boldsymbol{m}_\alpha(p)h_{lp} + \sum_{p=1}^N h_{pp}(\boldsymbol{\sigma}_\alpha^2(p)+\boldsymbol{m}_\alpha^2(p))\right)\\ &= -\frac{1}{2}\left(\boldsymbol{m}_\alpha^T\frac{\boldsymbol{H}^T\boldsymbol{H}}{\sigma_b^2}\boldsymbol{m}_\alpha + \sum_{p=1}^N h_{pp}\boldsymbol{\sigma}_\alpha^2(p)\right)\end{aligned}$$

  Where $(h_{lp})_{1\le l\le N,1\le p\le N}$ are the coefficients of $\boldsymbol{H}^T\boldsymbol{H}$.
- $$\begin{aligned}B &= \int_{\mathbb{R}^N\times\mathbb{R}^N}\frac{\boldsymbol{x}^T\boldsymbol{H}^T\boldsymbol{y}}{\sigma_b^2}\prod q_i^\alpha(x_i)\tilde{q}^k(\boldsymbol{z})\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{z}\\ &= \boldsymbol{m}_\alpha^T\frac{\boldsymbol{H}^T\boldsymbol{y}}{\sigma_b^2}.\end{aligned}$$
- $$\begin{aligned}C &= \sum_i\int_{\mathbb{R}}\frac{1}{2}\ln(z_i)\tilde{q}^k(z_i)\mathrm{d}z_i + \sum_i\int_{\mathbb{R}}\Big((\tilde{a}_i-1)\ln(z_i)\mathrm{d}z_i\\ &= \sum_{i=1}^N\int_{\mathbb{R}}(\tilde{a}_i-\frac{1}{2})\ln(z_i)\tilde{q}^k(z_i)\mathrm{d}z_i.\end{aligned}$$
- $$D = -\sum_i\frac{1}{2}\ln(\sigma_s^2) + \int_{\mathbb{R}} z_i\tilde{b}_i\tilde{q}^k(z_i)\mathrm{d}z_i = -\sum_{i=1}^N\tilde{b}_i\frac{\boldsymbol{a}(i)}{\boldsymbol{b}_k(i)} - \frac{N}{2}\ln(\sigma_s^2).$$
- $$\begin{aligned}E &= -\sum_i\int_{\mathbb{R}^2}\frac{z_i x_i^2}{2\sigma_s^2}q_i^\alpha(x_i)\tilde{q}^k(z_i)\mathrm{d}x_i\mathrm{d}z_i\\ &= -\sum_i\int_{\mathbb{R}}\frac{x_i^2}{2\sigma_s^2}q_i^\alpha(x_i)\mathrm{d}x_i\int_{\mathbb{R}} z_i\tilde{q}^k(z_i)\mathrm{d}z_i\\ &= -\frac{\boldsymbol{a}(i)}{2\boldsymbol{b}_k(i)\sigma_s^2}\int_{\mathbb{R}} x_i^2 q_i^\alpha(x_i)dx_i\\ &= -\sum_i\frac{\boldsymbol{a}(i)(\boldsymbol{\sigma}_\alpha^2(i)+\boldsymbol{m}_\alpha^2(i))}{2\boldsymbol{b}_k(i)\sigma_s^2}.\end{aligned}$$

Furthermore, concerning the entropy we have

$$\mathcal{H}(\mathbf{X},\mathbf{Z}) = \sum_{i=1}^{N} \mathcal{H}(q^\alpha(x_i)) + \sum_{i=1}^{N} \mathcal{H}(\tilde{q}^k(z_i)).$$

And,

$$\begin{aligned}
\mathcal{H}(q^\alpha(x_i)) &= \frac{1}{2}(1+\ln(2\pi\boldsymbol{\sigma}_\alpha^2(i))),\\
\mathcal{H}(\tilde{q}^k(z_i)) &= -\int_{\mathbb{R}} (\boldsymbol{a}(i)-1)\ln(z_i)\tilde{q}^k(z_i)\mathrm{d}z_i + Const.
\end{aligned}$$

From Equation (4.7) we have $\boldsymbol{a}(i) = \tilde{a}_i + \frac{1}{2}$ and

$$C+\sum_{i=1}^{N}\mathcal{H}(\tilde{q}^k(z_i)) = \sum_{i=1}^{N}\sum_{i=1}^{N}\int_{\mathbb{R}}(\tilde{a}_i-\frac{1}{2})\ln(z_i)\tilde{q}^k(z_i)\mathrm{d}z_i - \int_{\mathbb{R}}(\tilde{a}_i+\frac{1}{2}-1)\ln(z_i)\tilde{q}^k(z_i)\mathrm{d}z_i + Const = Const$$

Finally, the negative free entropy is

$$F(q^\alpha) = A + B + D + E + \sum_i \mathcal{H}(x_i) + Const, \tag{7.5}$$

Let us determine the critical values of $g_k(\alpha) = F(q^\alpha)$ and their signs.

As one can see, the derivative of $g_k(\alpha)$ is closely related to the derivatives of $\boldsymbol{\sigma}_\alpha^2(i)$ and $\boldsymbol{m}_\alpha(i)$ which, for $i = 1,\ldots,N$, are given by

$$\frac{\mathrm{d}\boldsymbol{\sigma}_\alpha^2(i)}{\mathrm{d}\alpha} = \frac{-\boldsymbol{\sigma}_k^2(i)\boldsymbol{\sigma}_r^2(i)(\boldsymbol{\sigma}_k^2(i)-\boldsymbol{\sigma}_r^2(i))}{(\boldsymbol{\sigma}_r^2(i)+\alpha(\boldsymbol{\sigma}_k^2(i)-\boldsymbol{\sigma}_r^2(i)))^2}, \tag{7.6}$$

whereas

$$\frac{\mathrm{d}\boldsymbol{m}_\alpha(i)}{\mathrm{d}\alpha} = \frac{\boldsymbol{\sigma}_r^2(i)\boldsymbol{\sigma}_k^2(i)(\boldsymbol{m}_r(i)-\boldsymbol{m}_k(i))}{(\boldsymbol{\sigma}_r^2(i)+\alpha(\boldsymbol{\sigma}_k^2(i)-\boldsymbol{\sigma}_r^2(i)))^2}. \tag{7.7}$$

And the second order derivatives are

$$\frac{\mathrm{d}^2\boldsymbol{\sigma}_\alpha^2(i)}{\mathrm{d}\alpha^2} = \frac{2\boldsymbol{\sigma}_k^2(i)\boldsymbol{\sigma}_r^2(i)(\boldsymbol{\sigma}_k^2(i)-\boldsymbol{\sigma}_r^2(i))^2}{(\boldsymbol{\sigma}_r^2(i)+\alpha(\boldsymbol{\sigma}_k^2(i)-\boldsymbol{\sigma}_r^2(i)))^3} \tag{7.8}$$

and,

$$\frac{\mathrm{d}^2\boldsymbol{m}_\alpha(i)}{\mathrm{d}\alpha^2} = \frac{2\boldsymbol{\sigma}_r^2(i)\boldsymbol{\sigma}_k^2(i)(\boldsymbol{m}_r(i)-\boldsymbol{m}_k(i))(\boldsymbol{\sigma}_r^2(i)-\boldsymbol{\sigma}_k^2(i))}{(\boldsymbol{\sigma}_r^2(i)+\alpha(\boldsymbol{\sigma}_k^2(i)-\boldsymbol{\sigma}_r^2(i)))^3}. \tag{7.9}$$

This entails that the first order derivative of g satisfies

$$\begin{aligned}
g'(\alpha) = &\sum_{i=1}^{N} -\frac{1}{\sigma_b^2}\sum_{p=1}^{N}\frac{\mathrm{d}\boldsymbol{m}_\alpha(i)}{\mathrm{d}\alpha}\boldsymbol{m}_\alpha(p)h_{ip}\\
&-\frac{h_{ii}}{2\sigma_b^2}\frac{\mathrm{d}\boldsymbol{\sigma}_\alpha^2(i)}{\mathrm{d}\alpha} + \frac{\mathrm{d}\boldsymbol{m}_\alpha(i)}{\mathrm{d}\alpha}\left(\frac{\boldsymbol{H}^T\boldsymbol{y}}{\sigma_b^2}\right)_i - \frac{\boldsymbol{a}(i)}{2\boldsymbol{b}_k(i)\sigma_s^2}\frac{\mathrm{d}\boldsymbol{\sigma}_\alpha^2(i)}{\mathrm{d}\alpha} - \frac{\boldsymbol{a}(i)\boldsymbol{m}_\alpha(i)}{\boldsymbol{b}_k(i)\sigma_s^2}\frac{\mathrm{d}\boldsymbol{m}_\alpha(i)}{\mathrm{d}\alpha} + \frac{\mathrm{d}\boldsymbol{\sigma}_\alpha^2(i)/\mathrm{d}\alpha}{4\pi\boldsymbol{\sigma}_\alpha^2(i)}
\end{aligned} \tag{7.10}$$

Whereas,

$$g''(\alpha) = \sum_{i=1}^{N} -\frac{1}{\sigma_b^2} \sum_{p=1}^{N} \left(\frac{\mathrm{d}^2 \boldsymbol{m}_\alpha(i)}{\mathrm{d}\alpha^2} \boldsymbol{m}_\alpha(p) + \frac{\mathrm{d}\boldsymbol{m}_\alpha(i)}{\mathrm{d}\alpha} \frac{\mathrm{d}\boldsymbol{m}_\alpha(p)}{\mathrm{d}\alpha} \right) h_{ip} - \frac{h_{ii}}{2\sigma_b^2} \frac{\mathrm{d}^2 \boldsymbol{\sigma}_\alpha^2(i)}{\mathrm{d}\alpha^2} + \frac{\mathrm{d}^2 \boldsymbol{m}_\alpha(i)}{\mathrm{d}\alpha^2} \left(\frac{\boldsymbol{H}^T \boldsymbol{y}}{\sigma_b^2} \right)_i$$
$$- \frac{\boldsymbol{a}(i)}{2\boldsymbol{b}_k(i)\sigma_s^2} \frac{\mathrm{d}^2 \boldsymbol{\sigma}_\alpha^2(i)}{\mathrm{d}\alpha^2} - \frac{\boldsymbol{a}(i)\boldsymbol{m}_\alpha(i)}{\boldsymbol{b}_k(i)\sigma_s^2} \frac{\mathrm{d}^2 \boldsymbol{m}_\alpha(i)}{\mathrm{d}\alpha^2} - \frac{\boldsymbol{a}(i)}{\boldsymbol{b}_k(i)\sigma_s^2} \left(\frac{\mathrm{d}\boldsymbol{m}_\alpha(i)}{\mathrm{d}\alpha} \right)^2 + \frac{\mathrm{d}^2 \boldsymbol{\sigma}_\alpha^2(i)/\mathrm{d}\alpha^2 \boldsymbol{\sigma}_\alpha^2(i) - \mathrm{d}\boldsymbol{\sigma}_\alpha^2(i)/\mathrm{d}\alpha}{4\pi(\boldsymbol{\sigma}_\alpha^2(i))^2} \tag{7.11}$$

In this case, the approximated critical value of F is given by α such that

$$\alpha_{subopt} = -\frac{\frac{dF(q^\alpha)}{d\alpha}\Big|_{\alpha=0}}{\frac{\mathrm{d}^2 F(q^\alpha)}{d\alpha^2}\Big|_{\alpha=0}}.$$

Finally, we consider $\alpha_{subopt} = -\frac{g'(0)}{g''(0)}$, where

$$g'(0) = -\left(\frac{\mathrm{d}\boldsymbol{m}_\alpha}{\mathrm{d}\alpha}\Big|_{\alpha=0} \right)^T \left(\frac{\boldsymbol{H}^T \boldsymbol{H} \boldsymbol{m}_k - \boldsymbol{H}^T \boldsymbol{y}}{\sigma_b^2} \right) - \sum_{i=1}^{N} \frac{\mathrm{d}\boldsymbol{m}_\alpha(i)}{\mathrm{d}\alpha}\Big|_{\alpha=0} \frac{\boldsymbol{a}(i)\boldsymbol{m}_k(i)}{\boldsymbol{b}_k(i)\sigma_s^2}$$
$$+ \frac{1}{2} \sum_{i=1}^{N} \frac{\boldsymbol{\sigma}_k^2(i)}{\boldsymbol{\sigma}_r^2(i)} \left(\boldsymbol{\sigma}_r^2(i) - \boldsymbol{\sigma}_k^2(i) \right) \left(-\frac{h_{ii}}{\sigma_b^2} - \frac{\boldsymbol{a}(i)}{\boldsymbol{b}_k(i)\sigma_s^2} + \frac{1}{2\pi \boldsymbol{\sigma}_k^2(i)} \right) \Bigg] . \tag{7.12}$$

$$\begin{aligned}
g'(0) &= \sum_{i=1}^{N} -\frac{1}{\sigma_b^2} \sum_{p=1}^{N} \frac{\boldsymbol{\sigma}_k^2(i)}{\boldsymbol{\sigma}_r^2(i)} (\boldsymbol{m}_r(i) - \boldsymbol{m}_k(i)) \boldsymbol{m}_0(p) h_{ip} - \frac{h_{ii}}{2\sigma_b^2} \frac{\boldsymbol{\sigma}_k^2(i)}{\boldsymbol{\sigma}_r^2(i)} (\boldsymbol{\sigma}_r^2(i) - \boldsymbol{\sigma}_k^2(i)) \\
&+ \frac{\boldsymbol{\sigma}_k^2(i)}{\boldsymbol{\sigma}_r^2(i)} (\boldsymbol{m}_r(i) - \boldsymbol{m}_k(i)) \left(\frac{\boldsymbol{H}^T \boldsymbol{y}}{\sigma_b^2} \right)_i - \frac{a_i^k b_i^k}{2\sigma_s^2} \frac{\boldsymbol{\sigma}_k^2(i)}{\boldsymbol{\sigma}_r^2(i)} (\boldsymbol{\sigma}_r^2(i) - \boldsymbol{\sigma}_k^2(i)) - \frac{\boldsymbol{\sigma}_k^2(i)}{\boldsymbol{\sigma}_r^2(i)} (\boldsymbol{m}_r(i) - \boldsymbol{m}_k(i)) \frac{\boldsymbol{m}_0(i)}{\sigma_s^2} \\
&+ \frac{\boldsymbol{\sigma}_r^2(i) - \boldsymbol{\sigma}_k^2(i)}{4\pi \boldsymbol{\sigma}_r^2(i)} \\
&= \sum_{i=1}^{N} \frac{\boldsymbol{\sigma}_k^2(i)}{\boldsymbol{\sigma}_r^2(i)} \left[(\boldsymbol{m}_r(i) - \boldsymbol{m}_k(i)) \left(-\frac{(\boldsymbol{H}^T \boldsymbol{H} \boldsymbol{m}_k - \boldsymbol{H}^T \boldsymbol{y})_i}{\sigma_b^2} - \frac{\boldsymbol{m}_k(i)}{\sigma_s^2} \right) \right. \\
&\left. + (\boldsymbol{\sigma}_r^2(i) - \boldsymbol{\sigma}_k^2(i)) \left(-\frac{h_{ii}}{2\sigma_b^2} - \frac{a_i^k b_i^k}{2\sigma_s^2} + \frac{1}{4\pi \boldsymbol{\sigma}_k^2(i)} \right) \right] \\
&= -\left(\frac{d\boldsymbol{m}_\alpha}{d\alpha}\Big|_{\alpha=0} \right)^T \left(\frac{\boldsymbol{H}^T \boldsymbol{H} \boldsymbol{m}_k - \boldsymbol{H}^T \boldsymbol{y}}{\sigma_b^2} + \frac{\boldsymbol{m}_k}{\sigma_s^2} \right) + \frac{1}{2} \sum_{i=1}^{N} \frac{\boldsymbol{\sigma}_k^2(i)}{\boldsymbol{\sigma}_r^2(i)} (\boldsymbol{\sigma}_r^2(i) - \boldsymbol{\sigma}_k^2(i)) \left(-\frac{h_{ii}}{\sigma_b^2} - \frac{a_i^k b_i^k}{\sigma_s^2} + \frac{1}{2\pi \boldsymbol{\sigma}_k^2(i)} \right) \Bigg] .
\end{aligned} \tag{7.13}$$

And,

$$g''(0) = -\left(\left.\frac{\mathrm{d}^2\boldsymbol{m}_\alpha}{\mathrm{d}\alpha^2}\right|_{\alpha=0}\right)^T\left(\frac{\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{m}_k - \boldsymbol{H}^T\boldsymbol{y}}{\sigma_b^2}\right) - \sum_{i=1}^{N}\left.\frac{\mathrm{d}^2\boldsymbol{m}_\alpha(i)}{\mathrm{d}\alpha^2}\right|_{\alpha=0}\frac{\boldsymbol{a}(i)\boldsymbol{m}_k(i)}{\boldsymbol{b}_k(i)\sigma_s^2}$$
$$-\left(\left.\frac{\mathrm{d}\boldsymbol{m}_\alpha}{\mathrm{d}\alpha}\right|_{\alpha=0}\right)^T\left(\frac{\boldsymbol{H}^T\boldsymbol{H}}{\sigma_b^2} - \frac{\mathrm{Diag}(\boldsymbol{a})(\mathrm{Diag}(\boldsymbol{b}))^{-1}}{\sigma_s^2}\right)\left.\frac{\mathrm{d}\boldsymbol{m}_\alpha}{\mathrm{d}\alpha}\right|_{\alpha=0}$$
$$+\sum_{i=1}^{N}\frac{\boldsymbol{\sigma}_k^2(i)}{(\boldsymbol{\sigma}_r^2(i))^2}(\boldsymbol{\sigma}_r^2(i)-\boldsymbol{\sigma}_k^2(i))^2\left(-\frac{h_{ii}}{\sigma_b^2} - \frac{\boldsymbol{a}(i)}{\boldsymbol{b}_k(i)\sigma_s^2} + \frac{1}{2\pi\boldsymbol{\sigma}_k^2(i)}\right) - \sum_{i=1}^{N}\frac{\boldsymbol{\sigma}_k^2(i)}{\boldsymbol{\sigma}_r^2(i)}\frac{\boldsymbol{\sigma}_r^2(i)-\boldsymbol{\sigma}_k^2(i)}{4\pi\boldsymbol{\sigma}_k^2(i)}. \tag{7.14}$$

REFERENCES

[1] R. Amiri, M. Alaee, H. Rahmani, and M. Firoozmand, *Chirplet based denoising of reflected radar signals*, Asia International Conference on Modelling & Simulation **0** (2009), 304–308.

[2] S. D. Babacan, R. Molina, and A. K. Katsaggelos, *Variationnal Bayesian Blind Deconvolution Using a Total Variation Prior*, IEEE Trans. Image Processing **18** (2009), no. 1, 12–26.

[3] N. Bali and A. Mohammad-Djafari, *Bayesian approach with hidden Markov modeling and mean field approximation for hyperspectral data analysis*, IEEE Trans. on Image Processing **17** (2008), no. 2, 217–225.

[4] G. Chantas, N. Galatsanos, A. Likas, and M. Saunders, *Variational Bayesian image restoration based on a product of t-distributions image prior*, IEEE Trans. Image Processing **17** (2008), no. 10, 1795–1805.

[5] R. Chartrand and W. Brendt, *A Gradient Descent Solution to the Monge-Kantorovich Problem*, Applied Math. Sciences (2009).

[6] R. A. Choudrey, *Variational methods for bayesian independent component analysis*, Ph.D. thesis, University of Oxford, 2002.

[7] G. Demoment, *Image reconstruction and restoration: Overview of common estimation structure and problems*, IEEE Trans. Acoust. Speech, Signal Processing **assp-37** (1989), no. 12, 2024–2036.

[8] B. Frigyik, S. Srivastava, and M. Gupta, *Functional Bregman Divergence and Bayesian Estimation of Distributions*, IEEE Trans. on Information Theory **54** (2008), no. 11, 5130–5139.

[9] A. Globerson, T. Koo, X. Carreras, and M. Collins, *Exponentiated gradient algorithms for log-linear structured prediction*, In Proc. ICML, 2007, pp. 305–312.

[10] Y. Goussard, G. Demoment, and F. Monfront, *Maximum a posteriori dectection-estimation of Bernoulli-Gaussian processes*, J.G. McWhirter ed., Mathematics in Signal Processing II, Clarendon Press, Oxford, UK, 1990.

[11] M.M. Ichir and A. Mohammad-Djafari, *Hidden markov models for wavelet-based blind source separation*, IEEE Trans. on Image Processing **15** (2006), no. 7, 1887–1899.

[12] J. Idier (ed.), *Bayesian approach to inverse problems*, ISTE Ltd and John Wiley & Sons Inc., London, 2008.

[13] S. Jana and P. Moulin, *Optimality of klt for high-rate transform coding of gaussian vector-scale mixtures : Application to reconstruction, estimation, and classification*, IEEE Trans. on Info. Theory (2006).

[14] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, *An Introduction to variational Methods for Graphical Models*, Machine Learning **37** (1999), no. 2, 183–233.

[15] A C Kak and M Slaney, *Principles of computerized tomographic imaging*, IEEE Press, New York, NY, 1988.

[16] H. Kellerer, *Measure theoretic versions of linear programming*, Math. Z. **198** (1988), no. 3, 367–400.

[17] J. Kivinen and M. Warmuth, *Exponentiated gradient versus gradient descent for linear predictors*, Information and Computation **132** (1997), no. 1, 1–63.

[18] M. Kowalski and T. Rodet, *An unsupervised algorithm for hybrid/morphological signal decomposition*, ICASSP 2011 (Prague), no. Id 2155, May 2011.

[19] D. MacKay, *Information theory, inference, and learning algorithms*, Cambridge University Press, 2003.

[20] D. J. C. MacKay, *Ensemble learning and evidence maximization*, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.54.4083, 1995.

[21] J.W Miskin, *Ensemble learning for independent component analysis*, Phd thesis, Cambridge, 2000, `http://www.inference.phy.cam.ac.uk/jwm1003/`.
[22] I. Molchanov, *Tangent sets in the space of measures: with applications to variational analysis*, J. Math. Anal. Appl. **249** (2000), no. 2, 539–552.
[23] I. Molchanov and S. Zuyev, *Steepest descent algorithms in a space of measures*, Statistics and Computing **12** (2002), 115–123.
[24] K. Morton, P. Torrione, and L. Collins, *Variational bayesian learning for mixture autoregressive models with uncertain-order*, IEEE Trans. Signal Processing **59** (2011), no. 6, 2614–2627.
[25] Frank Natterer, *Algorithms in tomography*, The State of the Art in Numerical Analysis, Clarendon Press, duff, i.s. and watson, g.a. ed., 1997.
[26] Y. Nesterov, *Smooth monimization of non-smooth functions*, Mathematic Programming Ser. A **103** (2005), 127–152.
[27] J. Nocedal and S. J. Wright, *Numerical optimization*, Series in Operations Research, Springer Verlag, New York, 2000.
[28] A. Pai, E. Chassande-Mottin, and O. Rabaste, *Best network chirplet chain: Near-optimal coherent detection of unmodeled gravitational wave chirps with a network of detectors*, Phys. Rev. D **77** (2008), no. 062005, 1–22, Also available at gr-qc/0708.3493.
[29] O. Rabaste and T. Chonavel, *Estimation of multipath channels with long impulse response at low SNR via an MCMC method*, IEEE Trans. Signal Processing **55** (2007), no. 4, 1312 – 1325.
[30] C. Robert, *Simulations par la méthode MCMC*, Economica, Paris, France, 1997.
[31] C. Robert and G. Casella, *Monte-Carlo statistical methods*, Springer Texts in Statistics, Springer, New York, NY, 2000.
[32] W. Rudin, *Real and complex analysis*, McGraw-Hill Book Co., New York, 1987.
[33] M. Seeger, *Bayesian inference and optimal design for sparse linear model*, Jour. of Machine Learning Research **9** (2008), 759–813.
[34] M. Seeger and D. Wipf, *Variational bayesian inference techniques*, IEEE signal Processing Magazin **27** (2010), no. 6, 81–91.
[35] V. Smidl and A. Quinn, *The variational bayes method in signal processing*, Springer, 2006.
[36] ———, *Variational bayesian filtering*, IEEE Trans. Signal Processing **56** (2008), no. 10, 5020–5030.
[37] H. Snoussi and A. Mohammad-Djafari, *Bayesian unsupervised learning for source separation with mixture of gaussians prior*, Int. Journal of VLSI Signal Processing Systems **37** (2004), no. 2-3, 263–279.
[38] M. Wainwright and E. Simoncelli, *Scale Mixtures of Gaussians and the statistics of natural images*, NIPS **12** (2000).
[39] G. Wang and Z. Bao, *Inverse synthetic aperture radar imaging of maneuvering targets based on chirplet decomposition*, Opt. Eng. **38** (1999), no. 1534.
[40] M. Zibulevsky and M. Elad, *L1-l2 optimization in signal and image processing*, IEEE Signal Processing Magazine (2010), 76–88.