entropy

MDPI

Article

Compressive Sensing via Variational Bayesian Inference under Two Widely Used Priors: Modeling, Comparison and Discussion

Mohammad Shekaramiz [1,*] and Todd K. Moon [2]

[1] Machine Learning & Drone Lab, Electrical and Computer Engineering Program, Engineering Department, Utah Valley University, 800 West University Parkway, Orem, UT 84058, USA
[2] Electrical and Computer Engineering Department, Utah State University, 4120 Old Main Hill, Logan, UT 84322, USA
* Correspondence: mshekaramiz@uvu.edu; Tel.: +1-801-863-4665

Abstract: Compressive sensing is a sub-Nyquist sampling technique for efficient signal acquisition and reconstruction of sparse or compressible signals. In order to account for the sparsity of the underlying signal of interest, it is common to use sparsifying priors such as Bernoulli–Gaussian-inverse Gamma (BGiG) and Gaussian-inverse Gamma (GiG) priors on the components of the signal. With the introduction of variational Bayesian inference, the sparse Bayesian learning (SBL) methods for solving the inverse problem of compressive sensing have received significant interest as the SBL methods become more efficient in terms of execution time. In this paper, we consider the sparse signal recovery problem using compressive sensing and the variational Bayesian (VB) inference framework. More specifically, we consider two widely used Bayesian models of BGiG and GiG for modeling the underlying sparse signal for this problem. Although these two models have been widely used for sparse recovery problems under various signal structures, the question of which model can outperform the other for sparse signal recovery under no specific structure has yet to be fully addressed under the VB inference setting. Here, we study these two models specifically under VB inference in detail, provide some motivating examples regarding the issues in signal reconstruction that may occur under each model, perform comparisons and provide suggestions on how to improve the performance of each model.

Keywords: compressive sensing; signal recovery; variational Bayes inference; sparse Bayesian learning; prior modeling; hyperparameters; graphical Bayesian representation

Citation: Shekaramiz, M.; Moon, T.K. Compressive Sensing via Variational Bayesian Inference under Two Widely Used Priors: Modeling, Comparison and Discussion. *Entropy* **2023**, *25*, 511. https://doi.org/10.3390/e25030511

Academic Editor: Carlos M. Travieso-González

Received: 30 January 2023
Revised: 6 March 2023
Accepted: 14 March 2023
Published: 16 March 2023

1. Introduction

Compressive sensing (CS) involves efficient signal acquisition and reconstruction techniques in a sub-Nyquist sampling sense. The CS framework can capture the vital information of the underlying signal via a small number of measurements while retaining the ability to reconstruct the signal. CS operates under the assumption that the signal is compressible or sparse, and the number and location of dominating components are unknown in most cases [1–3]. Compressibility or sparsity means that the signal has few dominating elements under some proper basis. CS has been used in a variety of applications such as the single-pixel camera, missing pixels and inpainting removal of images, biomedical such as heart rate estimation, internet of things (IoT), geostatistical data analysis, seismic tomography, communications such as blind multi-narrowband signals sampling and recovery, the direction of arrival (DoA) estimation, spectrum sharing of radar and communication signals, wireless networks and many more [4–27]. In the linear CS framework, the problem is posed as

$$\mathbf{y} = A\mathbf{x}_s + \mathbf{e}, \tag{1}$$

where $\mathbf{y} \in \mathbb{R}^M$ contains the measurements, $\mathbf{x}_s \in \mathbb{R}^N$ is the sparse signal of interest, $\mathbf{e}$ is the noise representing either the measurement noise or the insignificant coefficients of $\mathbf{x}_s$ and, generally, $M \ll N$ [1,2]. The measurement matrix can be defined as $A = \Phi\Psi$, where Φ is the sensing design matrix and Ψ is a proper sparsifying basis. There exist various approaches to solve for $\mathbf{x}_s$ in (1) including greedy-based, convex-based, thresholding-based and sparse Bayesian learning (SBL) algorithms [27–64]. Typically, the performance of CS reconstruction is determined in terms of the mean-squared reconstruction error. In this paper, we are also interested in the more demanding requirements of the probability of detection and the false alarm of the nonzero components. This is of more interest to CS applications such as blind multinarrowband signals, spectrum sharing RADAR, etc. [11–15].

The focus of this paper is on sparse Bayesian learning (SBL) for the CS problem. Bayesian learning models are flexible in incorporating prior knowledge of the characteristics of the underlying signal into the model. Bayesian learning also provides a distribution of the hidden variables, which is more informative than the point estimate approaches. A prior favoring the sparsity or compressibility in $\mathbf{x}_s$ can be represented in the SBL framework via Gaussian-inverse Gamma (GiG), Laplace-inverse Gamma (LiG), Bernoulli–Gaussian-inverse Gamma (BGiG), often referred to as spike-and-slab prior, etc. [27,46–59]. The inference on parameters and hidden variables in these models is usually made using Markov chain Monte Carlo (MCMC) and variational Bayes (VB) [27,45–52]. In this paper, we focus on the two most commonly used SBL prior models for solving the inverse problem of compressive sensing: Bernoulli–Gaussian-inverse Gamma (BGiG) prior and Gaussian-inverse Gamma (GiG). These models have been widely used, along with some additional priors, for sparse recovery of signals or images with block-sparsity/clustering patterns, sparse signals with correlated coefficients or other structured patterns [26,27,48–51,62,63].

We use VB inference to estimate the variables and parameters of the SBL model. VB is preferred over MCMC because MCMC is computationally expensive, though it can numerically approximate exact posterior distributions with a sufficient amount of computation. The convergence diagnostic of MCMC requires additional work, such as measuring the potential scale reduction factor (PSRF) for all the hidden variables and parameters of the model or monitoring their trace plots [45,50–52,65]. In contrast, VB inference can lead to a reasonable approximation of the exact posteriors, using less computation than MCMC and less effort to monitor the convergence [45,51,66,67]. In this paper, we present the derivation of the update rules of the parameters and variables using VB inference for both the BGiG and GiG models. (Portions of this derivation have been previously presented in [68,69]). Although these prior models have been widely used in various applications of compressive sensing, the study of the overall performance of these models under VB inference has yet to be thoroughly investigated. The preference for one model over the other becomes crucial when dealing with moderate or low sampling ratios, which we discuss in this paper. Here, we study the issues associated with each model via some motivational examples. Pre-/postprocessing approaches will then be discussed to tackle the issues. Finally, the overall performance of BGiG and GiG is compared.

The remainder of this work is organized as follows. In Section 2, we present a brief background on VB inference. We study Bernoulli–Gaussian-inverse Gamma modeling for CS using VB in Section 3. Some motivational examples are provided to show the issues with this approach. Section 4 represents Gaussian-inverse Gamma modeling, the associated update rules using VB inference and a motivational example of the issue that may occur using this approach. In Section 5, we study the improvement of the performances of the models after some pre-/postprocessing along with simulation results and comparisons. Section 6 concludes this work.

2. Variational Bayesian Inference

Variational Bayes (VB) is an effective approach to approximate intractable integrals that may arise in Bayesian inference. The main idea behind variational methods is to use a family of distributions over the latent variables with their own variational param-

eters. VB is a fast alternative to sampling methods such as Markov chain Monte Carlo (MCMC) and Sequential Monte Carlo (SMC) for performing approximate Bayesian inference [70,71]. For a probabilistic model with unknown parameters θ and hidden variables $\mathbf{x}$, the posterior distribution of the unknowns, given a set of observations $\mathbf{y}$, can be written as $p(\mathbf{x},\theta|\mathbf{y}) = p(\mathbf{x},\theta,\mathbf{y})/p(\mathbf{y})$. Finding the exact posterior in closed form to perform the inference would be a challenge, as the marginal distribution $p(\mathbf{y}) = \int p(\mathbf{y},\mathbf{x},\theta)d\mathbf{x}d\theta$ is often intractable. As an efficient approximation method for such inference problems, VB provides an analytical approximation to the posterior $p(\mathbf{x},\theta|\mathbf{y})$. VB approximates the joint density $p(\mathbf{x},\theta|\mathbf{y})$ via a variational distribution $Q_{x,\theta}(\mathbf{x},\theta)$, i.e., $p(\mathbf{x},\theta|\mathbf{y}) \approx Q_{x,\theta}(\mathbf{x},\theta)$. VB assumes that the distribution Q can be fully factorized with respect to the unknown parameters and hidden variables, i.e.,

$$\begin{aligned} Q_{x,\theta}(\mathbf{x},\theta) &= q_x(\mathbf{x})q_\theta(\theta) \\ &= \prod_{i=1}^{I} q_x(x_i) \prod_{j=1}^{J} q_\theta(\theta_j), \end{aligned}$$

where I and J are the number of unknown parameters and hidden variables, respectively. This independence assumption in VB further simplifies the search for a closed-form solution to the approximation of the actual posterior. We desire to select the variational distribution $Q^*_{x,\theta}(\mathbf{x},\theta)$ as close as possible to $p(\mathbf{x},\theta|\mathbf{y})$, where the closeness metric for distribution $Q_{x,\theta}(\mathbf{x},\theta)$ is formulated as minimizing the Kullback–Leibler (KL) divergence of the approximation $Q_{x,\theta}(\mathbf{x},\theta)$ and the true posterior $p(\mathbf{x},\theta|\mathbf{y})$ as

$$\begin{aligned} Q^\star_{x,\theta}(\mathbf{x},\theta) &= \underset{Q_{x,\theta}(\mathbf{x},\theta)}{\arg\min}\, \mathrm{KL}\big(Q_{x,\theta}(\mathbf{x},\theta)||p(\mathbf{x},\theta|\mathbf{y})\big) \\ &= \underset{Q_{x,\theta}(\mathbf{x},\theta)}{\arg\min} \int Q_{x,\theta}(\mathbf{x},\theta) \log \frac{Q_{x,\theta}(\mathbf{x},\theta)}{p(\mathbf{x},\theta|\mathbf{y})} d\mathbf{x}d\theta. \end{aligned}$$

The quantity $\log p(\mathbf{y})$ can be written as $\log p(\mathbf{y}) = \log\{\int p(\mathbf{x},\theta,\mathbf{y})d\mathbf{x}d\theta\}$. Then, defining

$$F\big(Q_{x,\theta}(\mathbf{x},\theta)\big) = \int Q_{x,\theta}(\mathbf{x},\theta) \log \frac{p(\mathbf{x},\theta,\mathbf{y})}{Q_{x,\theta}(\mathbf{x},\theta)} d\mathbf{x}d\theta.$$

It is straightforward to show that

$$\log p(\mathbf{y}) = F\big(Q_{x,\theta}(\mathbf{x},\theta)\big) + \mathrm{KL}\big(Q_{x,\theta}(\mathbf{x},\theta), p(\mathbf{x},\theta|\mathbf{y})\big).$$

Since (by Jensen's inequality) $KL(Q_{x,\theta}(\mathbf{x},\theta) \geq 0$, $\log(p(\mathbf{y}) \geq F(Q_{x,\theta}(\mathbf{x},\theta)$. Since $\log(p(\mathbf{y})$ is constant with respect to $Q_{x,\theta}$, minimizing the KL-divergence between the actual posterior distribution and the variational distribution is equivalent to maximizing the lower bound $F(\cdot)$ [66,67]. Since the term $p(\mathbf{y})$ in $p(\mathbf{x},\theta|\mathbf{y}) = p(\mathbf{x},\theta,\mathbf{y})p(\mathbf{y})$ does not involve the variational distribution $Q_{x,\theta}(\mathbf{x},\theta)$, this term can be ignored when maximizing $F(\cdot)$. The lower bound $F(\cdot)$ on the model log-marginal likelihood can be iteratively optimized until the convergence by the following update rules [66,72].
VB-E step:

$$q_x^{[t+1]}(\mathbf{x}) \propto \exp\left\{\mathrm{E}_{q_\theta^{[t]}}[\log p(\mathbf{x},\mathbf{y}|\theta)]\right\} \tag{2}$$

VB-M step:

$$q_\theta^{[t+1]}(\theta) \propto p(\theta) \exp\left\{\mathrm{E}_{q_x^{[t+1]}}[\log p(\mathbf{x},\mathbf{y}|\theta)]\right\} \tag{3}$$

This results in an iterative algorithm analogous to the expectation-maximization (EM) approach.

3. Bernoulli–Gaussian-Inverse Gamma Modeling and SBL(BGiG) Algorithm

In the inverse problem of CS defined in (1), the goal is to recover the sparse vector $\mathbf{x}_s$. In the Bernoulli–Gaussian-inverse Gamma model, the sparse solution is defined as

$$\mathbf{x}_s = (\mathbf{s} \circ \mathbf{x}), \tag{4}$$

where $\mathbf{s}$ is a binary support vector indicating the non-zero locations in the solution, $\mathbf{x}$ represents values of the solution and $\circ$ is Hadamard (element-by-element) product [47]. We refer to the algorithm associated with this Bayesian modeling based on VB inference as SBL(BGiG). SBL using VB inference for the clustered pattern of sparse signals has already been investigated in the recent literature [45,50,51,58]. In this paper, however, we intend to focus on the ordinary SBL using VB inference modeling without promoting any structure on the supports other than sparsity itself. We show that when the sampling ratio is moderate or low (with respect to the sparsity level), the reconstruction performance becomes sensitive to selecting the support-related hyperparameters.

We define a set of priors as follows [47,68,69]. We model the elements of vector $\mathbf{s}$ as

$$s_n \sim \text{Bernoulli}(\gamma_n), \gamma_n \sim \text{Beta}(\alpha_0, \beta_0), \forall n, \tag{5}$$

where α_0 and β_0 are the support-related hyperparameters. Setting α_0 and β_0 to small values and with $\alpha_0 \ll \beta_0$ encourages $\mathbf{s}$ to be sparse on average. The prior on the solution value vector is defined as

$$\mathbf{x} \sim \mathcal{N}(0, \tau^{-1} I_N), \quad \tau \sim \text{Gamma}(a_0, b_0). \tag{6}$$

Here, τ is the precision value. Finally, the prior on the noise is

$$\mathbf{e} \sim \mathcal{N}(0, \epsilon^{-1} I_M), \quad \epsilon \sim \text{Gamma}(\theta_0, \theta_1), \tag{7}$$

where θ_0 and θ_1 are set to small positive values.

3.1. Update Rules of SBL(BGiG) Using VB Inference

According to the VB algorithm defined in (2) and (3), the update rule of the variables and parameters of the BGiG model can be simplified as follows [68]. The details of these derivations appear in Appendix A.1.

- Update rule for the support vector $\mathbf{s}$

$$q(s_n|-) \sim \text{Bernoulli}(\frac{1}{1+c_n\kappa_n}), \ \forall n = 1, \ldots, N,$$

where conditioning on $-$ denotes conditioning on all relevant variables and observations. Therefore,

$$\tilde{s}_n = \frac{1}{1+c_n\kappa_n}, \ \forall n = 1, \ldots, N, \tag{8}$$

where

$$\begin{aligned} c_n &:= e^{\psi(\beta_{1,n}) - \psi(\alpha_{1,n})}, \\ \kappa_n &:= e^{\frac{1}{2}\tilde{\epsilon}\left(\|\mathbf{a}_n\|_2^2(\tilde{x_n}^2 + \sigma_{\tilde{x}_n}^2) - 2\tilde{x_n}\mathbf{a}_n^T\tilde{\mathbf{y}}^{-n}\right)}, \\ \tilde{y}_m^{-n} &:= y_m - \sum_{l \neq n}^{N} a_{ml}\tilde{s}_l\tilde{x}_l. \end{aligned} \tag{9}$$

Here, $\tilde{\mathbf{x}} := < \mathbf{x} >_{q_x}$, ψ is the digamma function (the logarithmic derivative of the gamma function), and $\tilde{\mathbf{y}}^{-n} = [\tilde{y}_1^{-n}, \ldots, \tilde{y}_M^{-n}]^T$.

- Update rule for the solution value matrix $\mathbf{x}$

$$q(\mathbf{x}|-) \sim \mathcal{N}(\tilde{\mathbf{x}}, \Sigma_{\tilde{x}}),$$

where

$$\Sigma_{\tilde{x}} = \left(\tilde{\tau} I_N + \tilde{\epsilon}\tilde{\Phi}\right)^{-1} \text{ and } \tilde{\mathbf{x}} = \tilde{\epsilon}\Sigma_{\tilde{x}}\text{diag}(\tilde{\mathbf{s}})A^T\mathbf{y}, \tag{10}$$

and where diag($\mathbf{s}$) denotes a diagonal matrix with the components of $\mathbf{s}$ on its main diagonal, and

$$\tilde{\Phi} := \left[(A^T A) \circ \left(\tilde{\mathbf{s}}\tilde{\mathbf{s}}^T + \text{diag}(\tilde{\mathbf{s}} \circ (1 - \tilde{\mathbf{s}}))\right)\right]. \tag{11}$$

- Update rule for γ_n

$$q(\gamma_n|-) \sim \text{Beta}(\alpha_{1,n}, \beta_{1,n}),\ \forall n = 1, \ldots, N.$$

Therefore,

$$\tilde{\gamma_n} = \frac{\alpha_{1,n}}{\alpha_{1,n} + \beta_{1,n}},\ \forall n = 1, \ldots, N, \tag{12}$$

where

$$\alpha_{1,n} := \alpha_0 + \tilde{s}_n \text{ and } \beta_{1,n} := \beta_0 + 1 - \tilde{s}_n. \tag{13}$$

- Update rule for the solution precision τ

$$q(\tau|-) \sim \text{Gamma}\big(a_0 + \frac{N}{2}, b_0 + \frac{1}{2}(\|\tilde{\mathbf{x}}\|_2^2 + \text{Tr}\,(\Sigma_{\tilde{x}}))\big),$$

where $\Sigma_{\tilde{x}} = \text{diag}\,(\sigma_{\tilde{x}_1}^2, \ldots, \sigma_{\tilde{x}_N}^2)$ and $\text{Tr}(A)$ is the trace of matrix A. Thus

$$\tilde{\tau} = \frac{a_0 + \frac{N}{2}}{b_0 + \frac{1}{2}\left(\|\tilde{\mathbf{x}}\|_2^2 + \sum_{n=1}^{N}\sigma_{\tilde{x}_n}^2\right)}. \tag{14}$$

- Update rule for the noise precision ϵ

$$q(\epsilon|-) \sim \text{Gamma}(\theta_0 + \frac{M}{2}, \theta_1 + \frac{1}{2}\tilde{\Psi}),$$

where

$$\tilde{\Psi} := \left(\mathbf{y}^T\mathbf{y} - 2(\tilde{\mathbf{x}} \circ \tilde{\mathbf{s}})^T A^T\mathbf{y} + \text{Tr}\left((\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T + \Sigma_{\tilde{x}})\tilde{\Phi}\right)\right). \tag{15}$$

This yields to the following update rule for the precision of the noise component

$$\tilde{\epsilon} = \frac{\theta_0 + \frac{M}{2}}{\theta_1 + \frac{1}{2}\tilde{\Psi}}. \tag{16}$$

The stopping criterion of the algorithm is made based on the log-marginalized likelihood. We define the stopping condition in terms of $L := \log\{p(\mathbf{y}|\mathbf{s}, \epsilon, \tau)\}$. The marginalized likelihood can be written as

$$p(\mathbf{y}|\mathbf{s}, \epsilon, \tau) = \int p(\mathbf{y}|\mathbf{x}, \mathbf{s}, \epsilon)p(\mathbf{x}|\tau I_N)d\mathbf{x}.$$

After some simplification, the negative log-likelihood is proportional to

$$-L \propto \log|\Sigma_0^{-1}| + \mathbf{y}^T\Sigma_0\mathbf{y},$$

where

$$\Sigma_0 = (\tilde{\epsilon}^{-1} I_M + \tilde{\tau}^{-1} A\tilde{S}^2 A^T)^{-1} \tag{17}$$

and $\tilde{S} := \text{diag}\{\tilde{\mathbf{s}}\}$. Therefore, the stopping condition can be made as

$$\Delta L_n^{[t]} := |\Delta L^{[t]}|/|L^{[t-1]}| \leq T_0, \tag{18}$$

for some small value of threshold T_0 [50], where

$$L^{[t]} := \log\Sigma_0^{[t]} - \mathbf{y}^T\Sigma_0^{[t]}\mathbf{y}. \tag{19}$$

and

$$\begin{aligned}\Delta L^{[t]} &:= L^{[t]} - L^{[t-1]} \\ &= \log|\frac{\Sigma_0^{[t]}}{\Sigma_0^{[t-1]}}| + \mathbf{y}^T(\Sigma_0^{[t-1]} - \Sigma_0^{[t]})\mathbf{y}.\end{aligned} \tag{20}$$

Figure 1 illustrates the graphical Bayesian representation of the BGiG model, which is an undirected graph. The shaded node **y** shows the observations (measurements), and the small solid nodes represent the hyperparameters. Each unshaded node denotes a random variable (or a group of random variables).

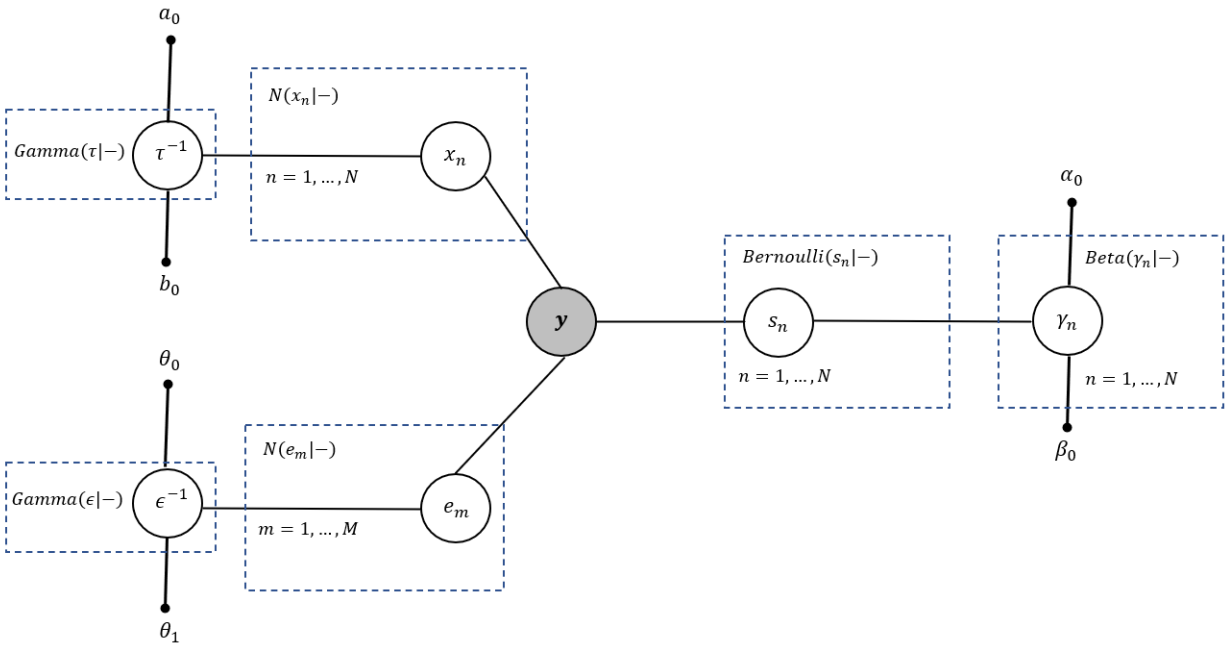


Figure 1. Graphical Bayesian representation of the BGiG model.

The flowchart representation of the algorithm is shown in Figure 2 motivated by the graphical approach in [47,73]. According to the pseudocode in Algorithm 1 and the flowchart in Figure 2, first, the hyperparameters of the model are set. The support-related hyperparameters α_0 and β_0 are suggested to be set to small values with $\alpha_0 \ll \beta_0$ to encourage **s** to be sparse on the average. The hyperparameters a_0 and b_0 on the precision of the solution-value vector are also initialized and suggested to be small not to bias the estimation when the measurements are incorporated. The hyperparameters θ_0 and θ_1 on the precision of the noise are recommended to be of order 10^{-6} for high SNRs. For moderate and low SNRs, higher values are recommended. In the next step, all the main variables of the model are drawn i.i.d. from their corresponding prior distributions defined in (5)–(7). Then, the stopping condition is computed based on the log-marginalized likelihood in (19). In the main loop, all of the main variables of the model are updated via the expected values obtained from the VB inference. Specifically, we first update the support vector and the solution value components; then, the precisions of the solution vector and the noise are updated. Finally, the stopping criterion is computed through the measure of the log-marginalized likelihood of the observations. The pseudocode of the algorithm is provided below.

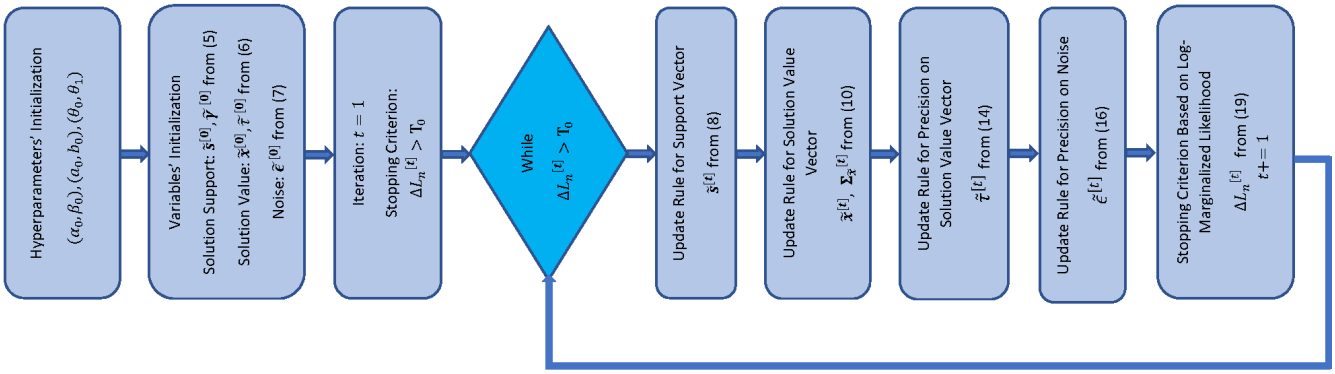


Figure 2. Flowchart of SBL(BGiG) algorithm.

Algorithm 1: SBL(BGiG) Algorithm

$\hat{\mathbf{x}}_s = \tilde{\mathbf{x}} \circ \tilde{\mathbf{s}}$
$\left[\tilde{\mathbf{x}}, \tilde{\mathbf{s}}\right] = \textbf{SBL-BGiG}(Y, A)$
Set the hyperparameters, i.e., (α_0, β_0), (a_0, b_0), and (θ_0, θ_1)
% Variables Initialization
Draw $\tilde{\mathbf{s}}$ and $\tilde{\gamma}$ from (5)
Draw $\tilde{\mathbf{x}}$ and $\tilde{\tau}$ from (6)
Draw $\tilde{\epsilon}$ from (7)
t = 1 % Iterator
Compute $L^{[\mathrm{t}]}$ from (19) and set $L^{[0]} = 0$
% Main Loop for Estimations
While $\frac{|L^{[\mathrm{t}]} - L^{[\mathrm{t}-1]}|}{|L^{[\mathrm{t}-1]}|} \geq T_0$. For example $T_0 = 10^{-6}$.
Compute $\tilde{s}_n$ from (8), $\forall n = 1, \ldots, N$ % (Support vector component)
Compute $\Sigma_{\tilde{x}}$ and $\tilde{\mathbf{x}}$ from (10) % (Solution-value matrix component)
Compute $\alpha_{1,n}$ and $\beta_{1,n}$ from (13) $\forall n = 1, \ldots, N$ % (Parameters of the hyperprior γ)
t Compute $\tilde{\tau}$ from (14) % (Precision on the solution)
Compute $\tilde{\epsilon}$ from (16) % (Precision on the noise)
Compute $L^{[\mathrm{t}]}$ from (19) and then t = t + 1
End While

3.2. Issues with SBL(BGiG)

In this section, we show that the estimated solution using SBL(BGiG) algorithm is sensitive to support-related hyperparameters, i.e., α_0 and β_0 in (5). We provide an example under three cases to demonstrate this issue. We generated a random scenario, where the true solution $\mathbf{x}_s \in \mathbf{R}^{100}$ has the sparsity level of $k = 25$, that is, the true $\mathbf{x}$ (or $\mathbf{s}$) has k active elements. The active elements of $\mathbf{s}$ were drawn randomly. The nonzeros of $\mathbf{x}_s$, corresponding to the active locations of $\mathbf{s}$, were drawn from $\mathcal{N}(0, \sigma_x^2)$, with $\sigma_x^2 = 1$. Each entry of the sensing matrix A was drawn i.i.d. from the Gaussian distribution $\mathcal{N}(0, 1)$, then normalized, so each column has the Euclidian norm of 1. The elements of measurement noise were drawn from $\mathcal{N}(0, \sigma^2)$ with $SNR = 25$ dB, where $SNR := 20\log_{10}(\sigma_x/\sigma)$. The hyperparameters of τ and ϵ were set to $a_0 = b_0 = 10^{-3}$ and $\theta_0 = \theta_1 = 10^{-6}$, respectively. In Cases 1–3, we set the pair (α_0, β_0) with low emphasis on the prior $(0.01, 0.99)$, moderate emphasis $(0.1, 0.9)$ and fairly high emphasis $(1.4, 2)$, respectively.

From the top to the bottom row of Figures 3–5, we illustrate the estimated results with the number of measurements set to 80, 60 and 40 (that is, the sample ratio λ is 0.80, 0.60, and 0.40), respectively. In each row of Figures 3–5 from left to right, we show the comparison between the measurements $\mathbf{y}$ and the computed measurements based on $\hat{\mathbf{y}} = A(\tilde{\mathbf{s}} \circ \tilde{\mathbf{x}})$, the true signal $\mathbf{x}_s = \mathbf{s} \circ \mathbf{x}$ and the reconstructed signal $\hat{\mathbf{x}}_s = \tilde{\mathbf{s}} \circ \tilde{\mathbf{x}}$, the true support

vector $\mathbf{s}$ and the estimated support vector $\tilde{\mathbf{s}}$ and the evolution of the estimated supports with respect to the iterations in the SBL(BGiG) algorithm.

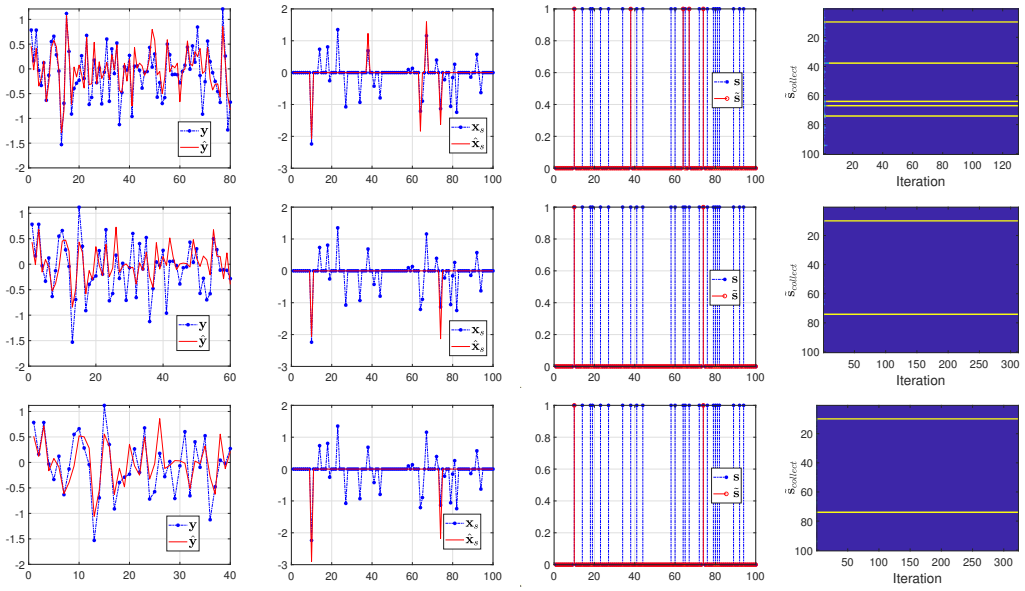


Figure 3. Case 1: $(\alpha_0, \beta_0) = (0.01, 0.99)$. From top to bottom, the rows show the results of SBL(BGiG) for the sampling ratio $\lambda = 0.80, 0.60, 0.40$, respectively.

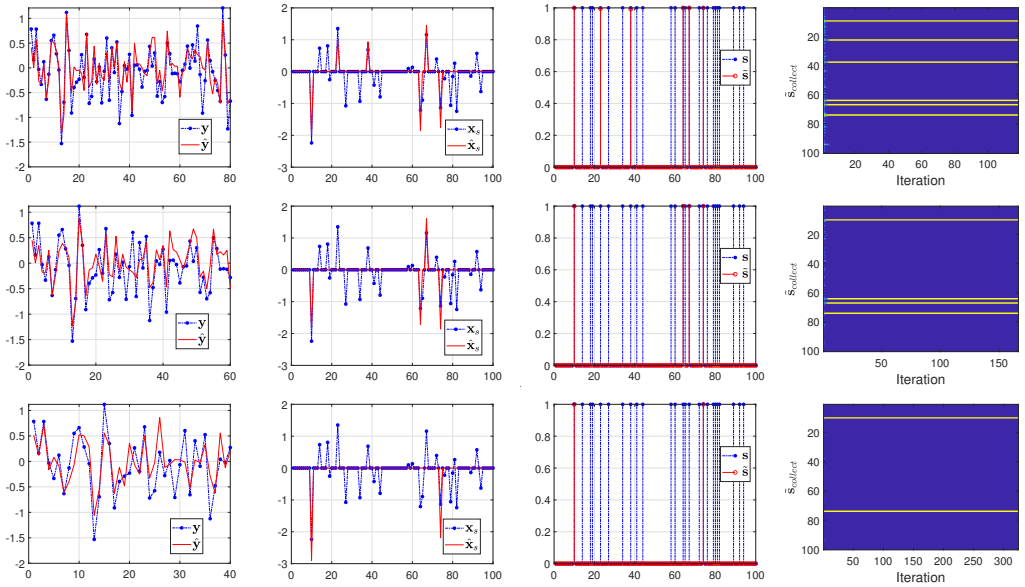


Figure 4. Case 2: $(\alpha_0, \beta_0) = (0.1, 0.9)$. From top to bottom, the rows show the results of SBL(BGiG) for the sampling ratio $\lambda = 0.80, 0.60, 0.40$, respectively.

According to Figure 3, the setting for (α_0, β_0) in Case 1 fails to provide perfect results even for high sampling ratios. Similarly, Figure 4 shows that the settings for (α_0, β_0) in Case 2 do not provide encouraging results even for high sampling ratios. Specifically, it turns out that Case 1 and Case 2 provide sparse solutions for the sampling ratios within the range $[0, 1]$, where $\lambda = 1$ means $M = N$.

According to Figure 5, setting (α_0, β_0) to $(1.4, 2)$ seems to be a reasonable choice for high sampling ratios (over 70%), while it is not a good choice for the lower sampling ratios. This issue can be seen in the supports plot in the 2nd and 3rd row of Figure 5. One may argue that the estimated support vector $\hat{\mathbf{s}}$ can be filtered via some threshold value (such as 0.3) for $\lambda = 0.6$. However, thresholding will adversely affect the detection rate, and setting

the threshold depends on our understanding of the signal characteristics. Furthermore, we should account for the effect of the filtered supports since their corresponding estimated components in $\hat{\mathbf{x}}_s$ contribute to fitting the model to the measurements.

In Table 1, we summarize the performance of the generated example for Cases 1–3, where P_D, P_{FA} and $NMSE$ denote the detection rate and false alarm rate in support recovery and the normalized mean-squared error between the true and the estimated sparse signal. This also shows that the algorithm fails to provide reasonable results for the sampling ratio of $\lambda = 0.4$.

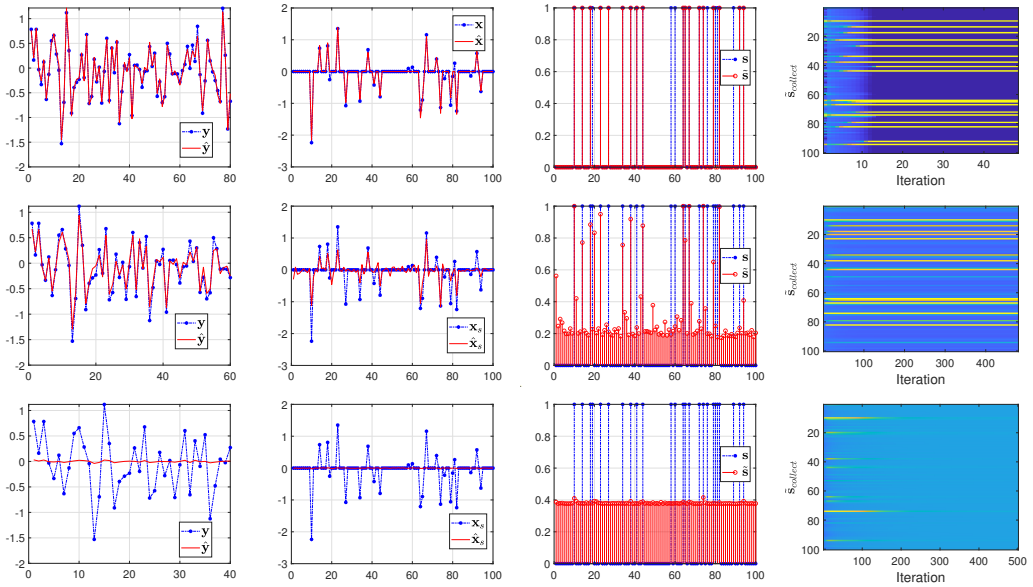


Figure 5. Case 3: $(\alpha_0, \beta_0) = (1.4, 2)$. From top to bottom, the rows show the results of SBL(BGiG) for the sampling ratio $\lambda = 0.80, 0.60, 0.40$, respectively.

These experiments suggest that there is no fixed setting for (α_0, β_0) capable of performing reasonably well for all sampling ratios and thus, selecting the hyperparameters (α_0, β_0) should be made with care.

Table 1. Performance results of SBL(BGiG) for Cases 1–3.

Case 1: ($\alpha_0 = 0.01, \beta_0 = 0.99$)				**Case 2: ($\alpha_0 = 0.1, \beta_0 = 0.9$)**				**Case 3: ($\alpha_0 = 1.4, \beta_0 = 2.0$)**			
λ	P_D	P_{FA}	**NMSE (dB)**	λ	P_D	P_{FA}	**NMSE (dB)**	λ	P_D	P_{FA}	**NMSE (dB)**
0.8	0.20	0	−2.367	0.8	0.24	0	−3.109	0.8	0.72	0	−16.264
0.6	0.08	0	−1.326	0.6	0.16	0	−2.197	0.6	1	0	−5.226
0.4	0.08	0	−1.181	0.4	0.08	0	−1.181	0.4	1	1	−0.088

Continuing this examination, in Figures 6–8, we illustrate the negative log-marginalized likelihood, the noise precision estimation and the estimated precision on the generated true solution in Cases 1–3, respectively. The horizontal axis shows the iterations until the stopping rule is met.

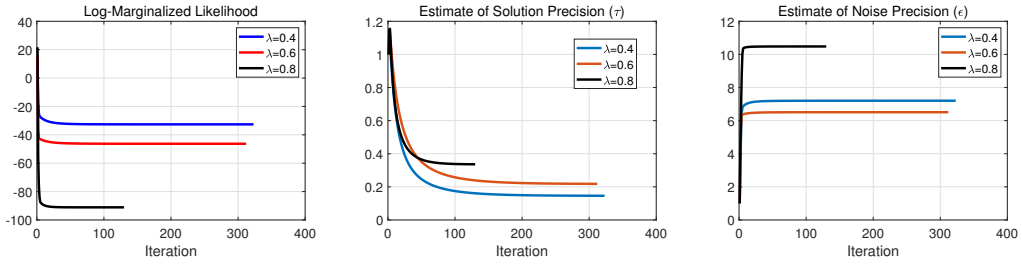


Figure 6. Case 1: Performance evaluation of SBL(BGiG).

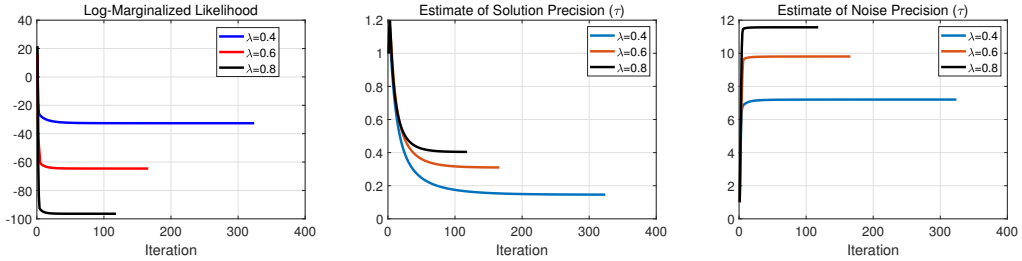


Figure 7. Case 2: Performance evaluation of SBL(BGiG).

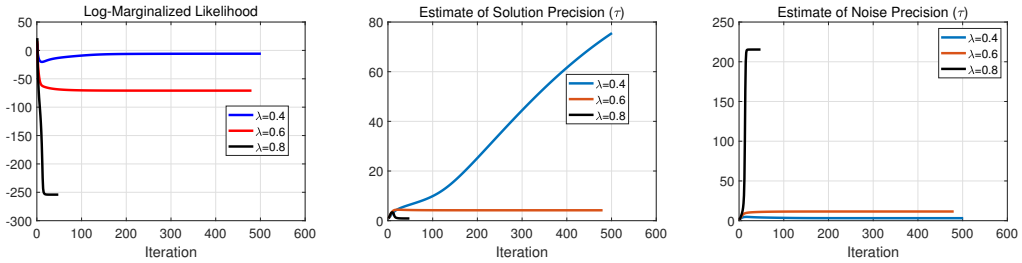


Figure 8. Case 3: Performance evaluation of SBL(BGiG).

As expected, as the sampling ratio increases, the algorithm requires fewer iterations to meet its stopping condition. This can be seen on the negative log-marginalized likelihood plots in Figures 6–8. In these experiments, the actual precision of the solution components was set to $\tau = 1$, and the actual noise precision was set to $\epsilon = 316.2$.

For Cases 1 and 2, according to Figures 6–8, the estimated precisions on both the noise and solution components were far off from the actual ones even for $\lambda = 0.8$. Thus, it resulted in poor performance in signal recovery for Cases 1 and 2 (see Figures 3 and 4).

For Case 3, the estimated precisions on the noise and the solution components were acceptable for $\lambda = 0.8$ but far off from the actual ones for lower sampling ratios (see Figure 8). The main issue of the failures can be found in the update rule of the support learning vector $\tilde{\mathbf{s}}$ defined in (8). It is important to balance between the terms c_n and κ_n, where c_n imposes the effect of hyper-prior on $\mathbf{s}$ accompanied by the current estimate of s_n. In contrast, κ_n imposes the contribution of the current estimates of noise precision, solution and other supports in fitting the model to the measurements. Therefore, if we impose a substantial weight on the sparsity via c_n, the solution tends to neglect the effect of κ_n and vice versa. This is why we had sparse (with poor performance) in Cases 1 and 2 for all the represented sampling ratios and nonsparse (with poor performance) for moderate and lower sampling ratios in Case 3. These results suggest that the algorithm and its update rules are sensitive to the selection of hyperparameters on the Gamma prior on the support vector $\mathbf{s}$. The main issue can be seen in (9), where the selection of the hyperparameters α_0 and β_0 resulted in a large or small value in c_n due to the digamma function.

4. Gaussian-Inverse Gamma Modeling and SBL(GiG) Algorithm

In this section, we consider the Gaussian-inverse Gamma (GiG) model. In this model, each component x_n of the solution is modeled by zero-mean Gaussian with the precision

τ_n. The main difference between this model and the model defined in Section 3 is that the GiG model does not have the support vector $\mathbf{s}$; instead, different precisions are considered on the components of the solution vector $\mathbf{x}_s$ in (1). A simpler version of GiG can also be used by defining the same precision τ for all the components of $\mathbf{x}_s$.

Here, we rather use different precisions to make the GiG model have almost the same complexity as the BGiG model in terms of the parameters to be learned. The set of priors in this model is defined as follows.

$$x_n \sim \mathcal{N}(0, \tau_n^{-1}),\ \tau_n \sim \text{Gamma}(a_0, b_0),\ \forall n, \tag{21}$$

where a_0 and b_0 denote the shape and rate of the Gamma distribution, respectively. The entries of the noise component $\mathbf{e}$ are defined the same as (7), i.e.,

$$\mathbf{e} \sim \mathcal{N}(0, \epsilon^{-1} I_M),\ \epsilon \sim \text{Gamma}(\theta_0, \theta_1),$$

where θ_0 and θ_1 are set to small positive values. The estimation of the parameters in this model is carried out using VB inference, as discussed below.

4.1. Update Rules of SBL(GiG) Using VB Inference

According to the VB algorithm described in (2) and (3), the update rule of the variables and parameters of the GiG model can be simplified as follows. The details of these derivations appear in Appendix A.2.

- Update rule for the precision τ_n on x_n using VB

$$q(\tau_n) \sim \text{Gamma}\big(a_0 + \frac{1}{2}, b_0 + \frac{1}{2}(\tilde{x}_n^2 + \sigma_{\tilde{x}_n}^2)\big),\ \forall n = 1, \ldots, N.$$

Thus,

$$\tilde{\tau_n} = \frac{a_0 + \frac{1}{2}}{b_0 + \frac{1}{2}(\tilde{x}_n^2 + \sigma_{\tilde{x}_n}^2)},\ \forall n = 1, 2, \ldots, N. \tag{22}$$

- Update rule for the noise precision ϵ using VB

$$q(\epsilon) \sim \text{Gamma}(\theta_0 + \frac{M}{2}, b_0 + \frac{1}{2}\tilde{\Psi})$$

which yields

$$\tilde{\epsilon} = \frac{\theta_0 + \frac{M}{2}}{\theta_1 + \frac{1}{2}\tilde{\Psi}}, \tag{23}$$

where

$$\tilde{\Psi} := \mathbf{y}^T\mathbf{y} - 2\tilde{\mathbf{x}}^T A^T \mathbf{y} + \text{Tr}\left((\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T + \Sigma_{\tilde{x}}) A^T A\right). \tag{24}$$

- Update rule for the solution vector $\mathbf{x}$ using VB

$$q_x(\mathbf{x}) \sim \mathcal{N}(\tilde{\mathbf{x}}, \Sigma_{\tilde{x}}), \tag{25}$$

where

$$\Sigma_{\tilde{x}} := (\tilde{T} + \tilde{\epsilon} A^T A)^{-1} \text{ and } \tilde{\mathbf{x}} := \tilde{\epsilon} \Sigma_{\tilde{x}} A^T \mathbf{y}, \tag{26}$$

and

$$\tilde{T} := \text{diag}\{[\tilde{\tau}_1, \ldots, \tilde{\tau}_N]\}.$$

We set the stopping rule of the algorithm using the marginalized likelihood (evidence) defined as

$$p(\mathbf{y}|\epsilon, \tau) = \int p(\mathbf{y}|\mathbf{x}, \epsilon, \tau) p(\mathbf{x}|\tau) d\mathbf{x}.$$

After simplification and for the comparison purposes of $L^{[t]}$ with $L^{[t-1]}$ in the updating process, we have

$$L^{[t]} \propto \log |\Sigma_0^{[t]}| - \mathbf{y}^T \Sigma_0^{[t]} \mathbf{y},$$

where Σ_0 is defined as

$$\Sigma_0 := (\tilde{\epsilon}^{-1} I_M + \tilde{T}^{-1} A A^T)^{-1}. \tag{27}$$

Therefore, similar to SBL(BGiG), the stopping condition can be made as

$$\Delta L_n^{[t]} := |\Delta L^{[t]}| / |L^{[t-1]}| \leq T_0, \tag{28}$$

for some small value of threshold T_0.

Figure 9 illustrates the graphical Bayesian representation of the GiG model, which is an undirected graph. Similar to Figure 1, the shaded node **y** shows the observations, the small solid nodes represent the hyperparameters and the unshaded nodes denote the random variables.

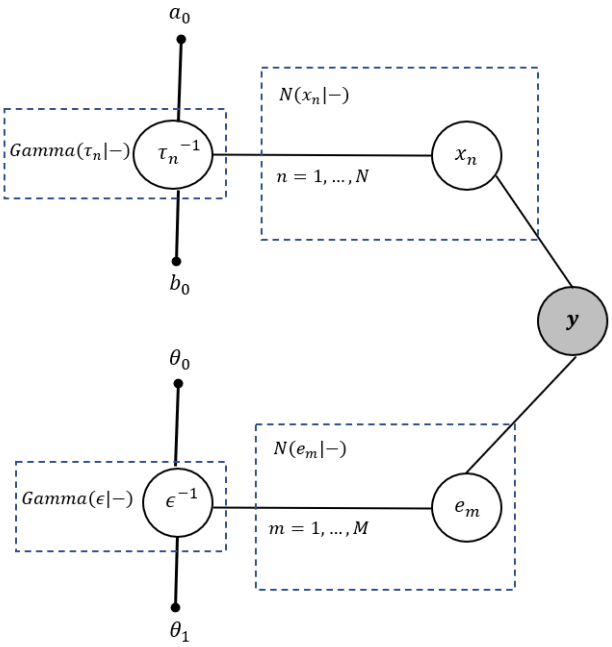


Figure 9. Graphical Bayesian representation of the GiG model.

The flowchart representation of the algorithm is shown in Figure 10. According to the pseudocode in Algorithm 2 and the flowchart in Figure 10, first, the hyperparameters of the model are set. The hyperparameters a_0 and b_0 on the precision of the solution-value vector are initialized and suggested to be small. Similar to SBL(BGiG), the hyperparameters θ_0 and θ_1 on the precision of the noise are recommended to be of order 10^{-6} for high SNRs. All the main variables of the model are drawn i.i.d. from their corresponding prior distributions defined in (22)–(26). Then, the stopping condition is computed based on (28). In the main loop, all the main variables of the model are updated via the expected values obtained from the VB inference through (22)–(26). The pseudocode of the algorithm is provided below.

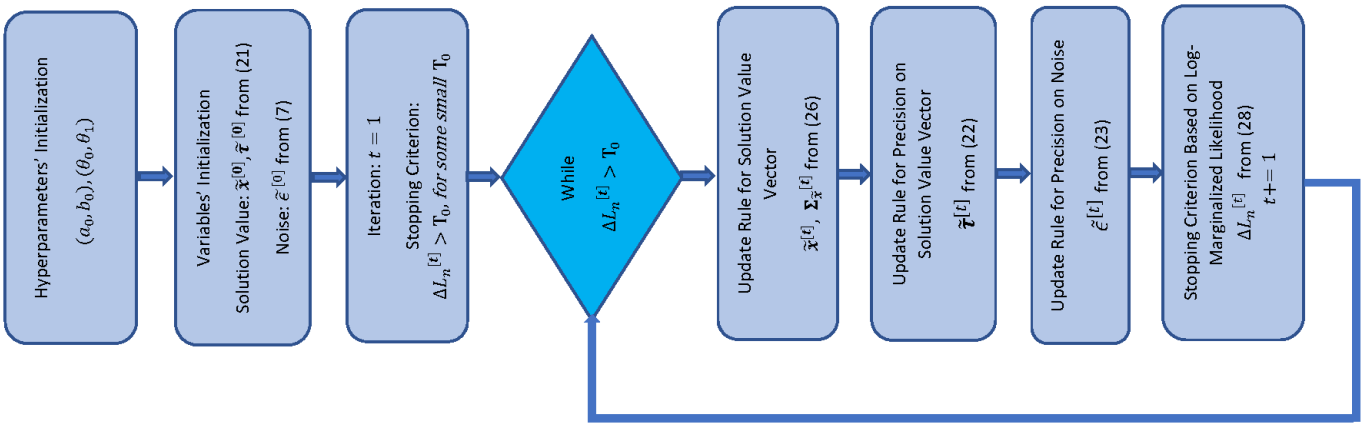


Figure 10. Flowchart of SBL(GiG) algorithm.

Algorithm 2: SBL(GiG) Algorithm

$\tilde{\mathbf{x}}_s = \textbf{SBL-GiG}(Y, A)$
Set the hyperparameters, i.e., (a_0, b_0) and (θ_0, θ_1)
% Variables' Initialization
Draw $\tilde{\mathbf{x}}_s$ and $\tilde{\boldsymbol{\sigma}}$ from (21)
Draw $\tilde{e}$ from (7)
t = 1 % Iterator
Compute $\tilde{L}^{[t]}$ from (28) and (27), and set $\tilde{L}^{[0]} = 0$
% Main Loop for Estimations
t = 1
While $\frac{|L^{[t]} - L^{[t-1]}|}{|L^{[t-1]}|} \geq T_0$. For example $T_0 = 10^{-6}$.
Compute $\Sigma_{\tilde{x}}$ and $\tilde{\mathbf{x}}_s$ from (26) % (Solution-value matrix component)
Compute $\tilde{T}$ from (22) % (Precisions on the solution)
Compute $\tilde{e}$ from (23) % (Precision on the noise)
Compute $L^{[t]}$ from (28) and (27), and then t = t + 1
End While

4.2. Issues with SBL(GiG)

An issue with the SBL(GiG) algorithm is that the solution becomes nonsparse since it does not incorporate a binary vector $\mathbf{s}$ (hard-thresholding or soft-thresholding if the expected value is used) as we had in SBL(BGiG). This may have no major effect on the signal reconstruction for high sampling ratios. However, the nonsparseness effect appears in low sampling ratios by misleading the algorithm to wrongly activate many components in the estimated signal yet providing a good fit of the model to the measurements. Here, we use the same example as we made for the SBL(BGiG) model with the same sensing matrix A, measurement vector $\mathbf{y}$ and noise $\mathbf{e}$. Notice that in the SBL(BGiG) model, we considered the same precision τ on all the components of the solution value vector $\mathbf{x}$ support vector $\mathbf{s}$. In contrast, the SBL(GiG) model does not have the support learning vector; instead, we assume that each component of the solution vector has different precision τ_n. It turns out that SBL(GiG) is not very sensitive to the selection of the hyperparameters as the SBL(BGiG). Thus, here, we show the results for one case scenario for the hyperparameters. We use the same setting for the parameters of ϵ in the hyper prior as before, i.e., $\theta_0 = \theta_1 = 10^{-6}$, and the same parameters for all the precisions τ_n of the solution component, i.e., $a_0 = b_0 = 10^{-3}$. In Figures 11 and 12, we illustrate the results after applying the SBL(GiG) algorithm. In Figure 11, from left to right, we show the results for sampling ratios of λ =0.8, 0.6, and 0.40, respectively. The first row shows the comparison of $\mathbf{y}$ with $\hat{\mathbf{y}} = A\tilde{\mathbf{x}}_s$, the second row

shows the true solution $\mathbf{x}_s$ and the estimated solution $\tilde{\mathbf{x}}_s$, and the third row demonstrates the estimated precisions on the solution components. In Figure 12, we demonstrate the negative log-marginalized likelihood comparison and the estimated noise precision against the true noise precision for the sampling ratios of $\lambda = 0.8$, 0.6 and 0.4.

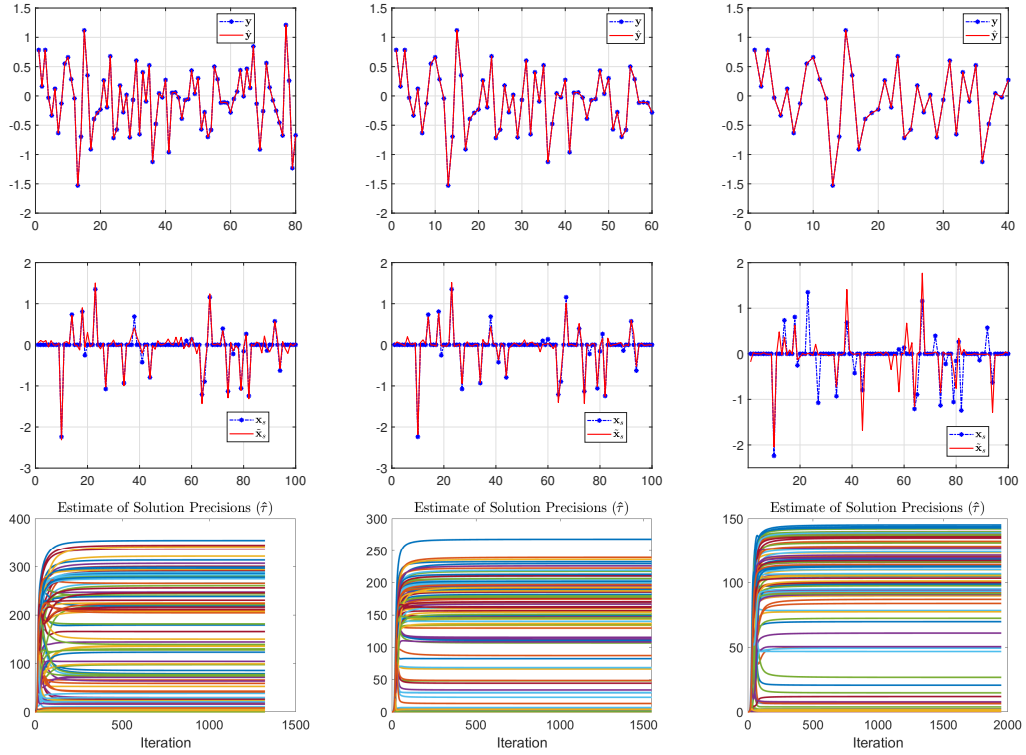


Figure 11. From left to right, we show the results for sampling ratios of λ =0.8, 0.6 and 0.40, respectively. The first row shows the comparison of $\mathbf{y}$ with $\hat{\mathbf{y}} = A\tilde{\mathbf{x}}_s$, the second row shows the true solution $\mathbf{x}_s$ and the estimated solution $\tilde{\mathbf{x}}_s$, and the third row demonstrates the estimated precisions on the solution components.

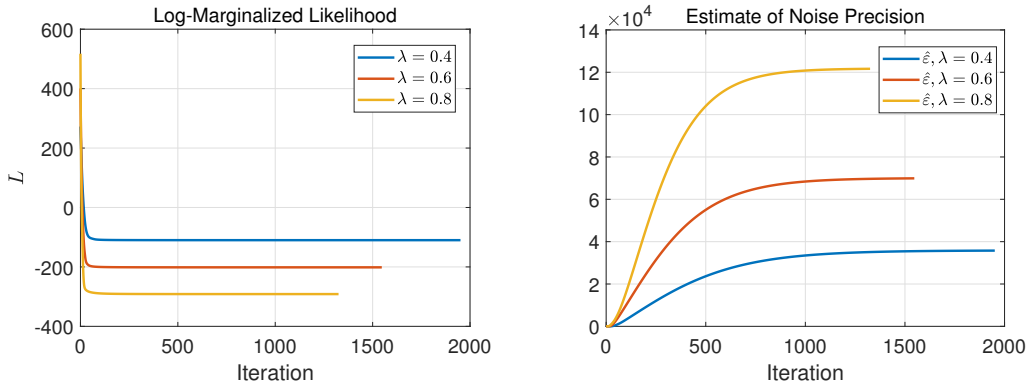


Figure 12. The behavior of negative marginalized log-likelihood and the precision on the noise using SBL(GiG) for the sampling ratios of 0.4, 0.6 and 0.80.

From the results shown in Figures 11 and 12, we observe that the recovered signal tends to become nonsparse. This effect is illustrated in the second row of Figure 11. This can also be observed in the precision estimations of the solution components. More specifically, the true nonzero components in our simulations were drawn from a zero-mean Gaussian with the precision of $\tau_n = 1$. Thus, the ideal precision estimation would be within the two classes of values of 1 and infinity or very large values. However, the estimated results in our simulation do not show such a classification. As the sampling ratio decreases,

the solution estimate has poor performance, due not only to the reduction in the number of measurements but also the nonsparseness behavior.

5. Preprocessing versus Postprocessing and Simulations

In this section, we show that in order to improve the performance of Bernoulli–Gaussian-inverse Gamma modeling using the SBL(BGiG) algorithm, we need to perform a preprocessing step. The results in Section 4 suggest one can perform some postprocessing for the SBL(GiG) algorithm to improve the reconstruction performance. Below, we provide more details for each of these algorithms.

5.1. Pre-Processing for the SBL(BGiG) Algorithm

Based on the observations made on the performance of SBL(BGiG) in Section 3.2, we showed that the pair of hyperparameters (α_0,β_0) should be selected with care. In other words, obtaining good performance with this algorithm needs some preprocessing to assess an appropriate setting for the parameters. For a more rigorous study, here, we perform a grid search on the hyperparameters (α_0, β_0) to see whether we can find some common pattern in selecting these parameters for all sampling ratios. The grid search runs the algorithm for different values of α_0 and β_0 with the search range of $[0.1, 2]$ with the resolution of 0.1. For each (α_0, β_0) within this range, we ran 200 random trials and then averaged the results. The settings of these trials are represented in Table 2.

Table 2. Settings for preprocessing analysis and simulations on SBL(BGiG).

α_0	β_0	a_0	b_0	θ_0	θ_1	Sparsity	γ	N
$[0.1, 2]$	$[0.1, 2]$	10^{-3}	10^{-3}	10^{-6}	10^{-6}	25	(5)	100
$\mathbf{s}$	τ	$\mathbf{x}$	$\mathbf{x}_s$	M	ϵ	$\mathbf{e}$	A	$\mathbf{y}$
(5)	(6)	(6)	$\mathbf{x}_s = \mathbf{x} \circ \mathbf{s}$	$5:N$	316	(7)	$[A]_{mn} \sim \mathcal{N}(0,1)$	$A\mathbf{x}_s + \mathbf{e}$

We generated a random scenario, where the true solution $\mathbf{x}_s \in \mathbf{R}^{100}$ has the sparsity level of $k = 25$. The active elements of $\mathbf{s}$ were drawn randomly. The nonzeros of $\mathbf{x}_s$ were drawn from $\mathcal{N}(0, \sigma_x^2)$, with $\sigma_x^2 = 1$. Each entry of the sensing matrix A was drawn i.i.d. from the Gaussian distribution $\mathcal{N}(0, 1)$, then normalized. The elements of measurement noise were drawn from $\mathcal{N}(0, \sigma^2)$ with $SNR = 25$ dB. The results were examined to see what values of (α_0, β_0) provided the highest performance in the detection rate vs. and false alarm rate. The simulation was executed for a range of *sampling ratios* in the range $[0.05, 1]$ with the step size of 0.05. The results are demonstrated in Figure 13. In this figure, we also provide the results of performing a random Sobol search for (α_0, β_0). A Sobol sequence is a low discrepancy quasirandom sequence. The two right plots in Figure 13 show the results for the best setting of (α_0, β_0).

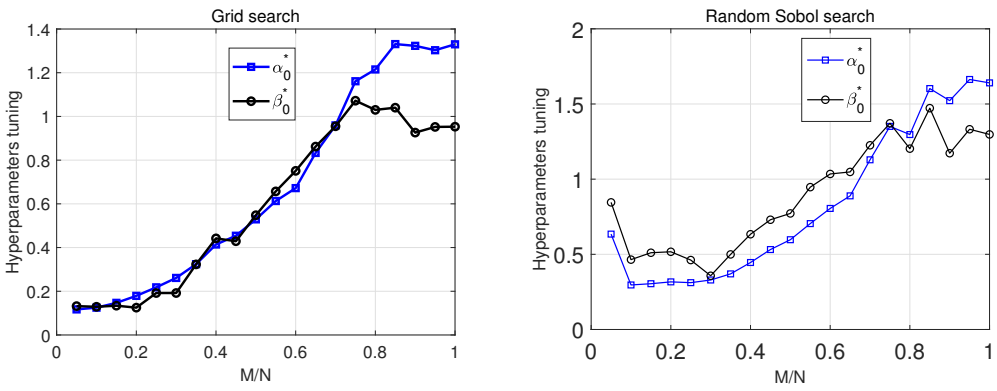


Figure 13. *Cont.*

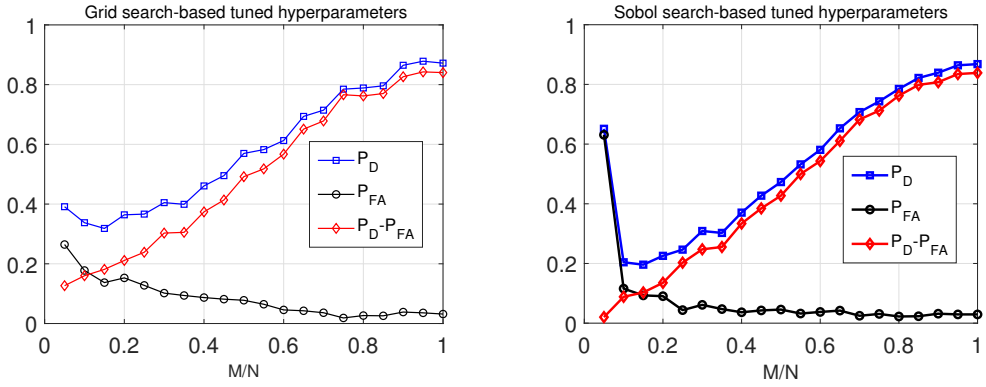
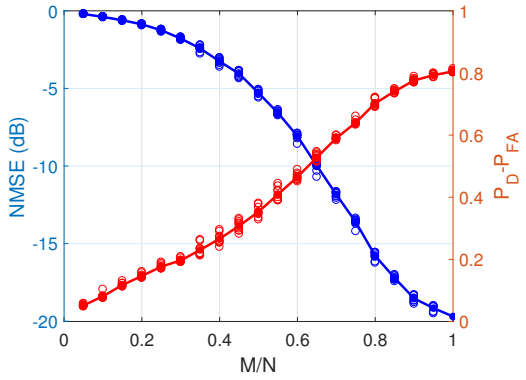
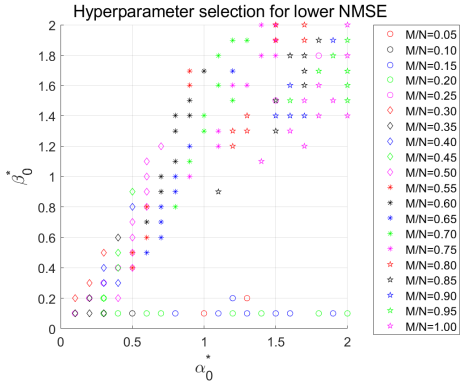


Figure 13. Performance evaluation of SBL(BGiG) using grid and random Sobol search.

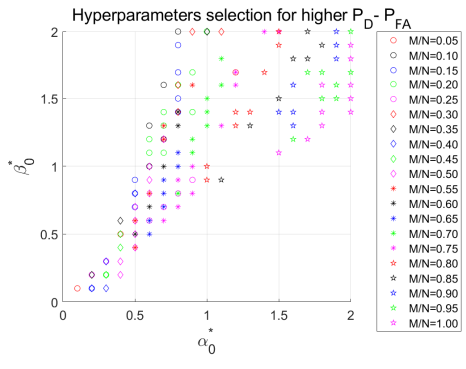
It should be clear from Figure 13 that there is no fixed setting for these parameters in order to get the best performance for all sampling ratios. The two plots on the right of Figure 13 illustrate the performance based on the best values of these hyperparameters, which provided the best performance, i.e., tuned hyperparameters. We also examined the grid search results for the top 10 highest performances for each sampling ratio, where performance is in terms of $P_D - P_{FA}$ and the normalized mean-squared error (NMSE). In Figure 14a, we demonstrate the top 10 highest performances based on NMSE and $P_D - P_{FA}$ for different sampling ratios. In Figure 14b,c, we illustrate the values of (α_0, β_0), which led to the performances shown in Figure 14a for different sampling ratios. Figure 15 details the top 10 values of (α_0, β_0) vs. sampling ratio.



(**a**)

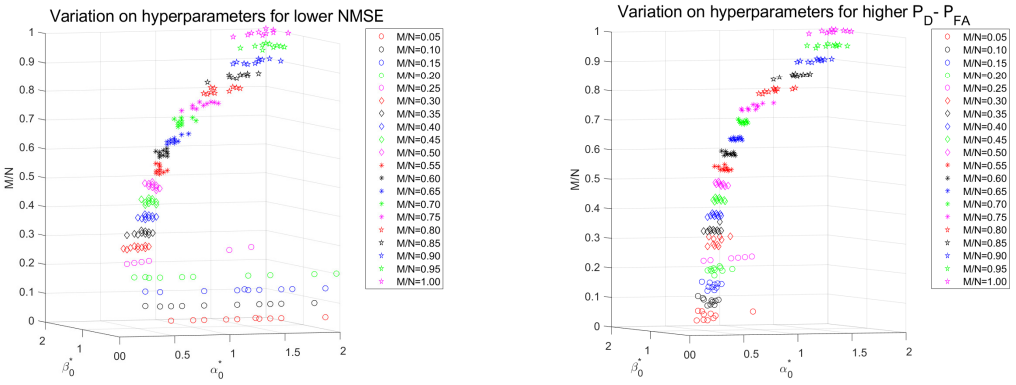


(**b**)



(**c**)

Figure 14. (**a**) Overall performance (**b**) Top 10 (α_0, β_0) with lowest NMSE (**c**) Top 10 (α_0, β_0) with highest $P_D - P_{FA}$.



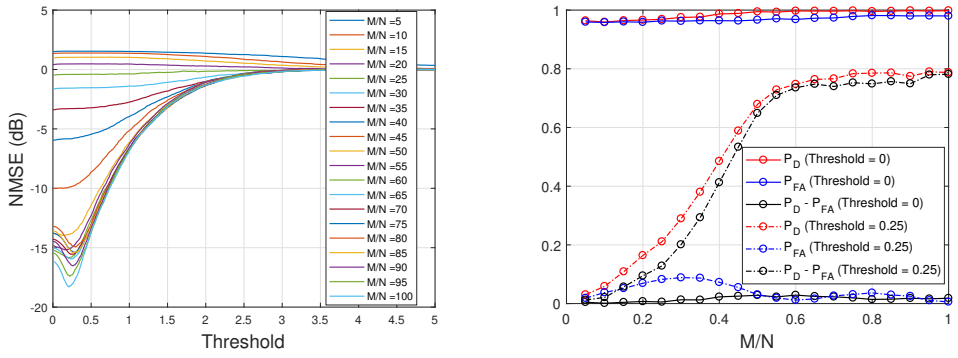
(**a**) (**b**)

Figure 15. (**a**) Top 10 (α_0, β_0) with lowest NMSE vs. sampling ratio (**b**) Top 10 (α_0, β_0) with highest $P_D - P_{FA}$ vs. sampling ratio.

According to Figure 14b,c, there is no specific pattern for these hyperparameters. Figure 15 also shows that hyperparameters need to be carefully selected.

5.2. Post-Processing for the SBL(GiG) Algorithm

Since the SBL(GiG) algorithm does not include the binary support vector **s**, as SBL(BGiG) possesses, the resulting solution tends to become nonsparse. This leads to a high detection rate for the location of active supports and a high false alarm rate. Thus, as the sampling ratio decreases, there is a high chance that this algorithm overwhelms the locations of the true solution. Therefore, SBL(GiG) requires some postprocessing to discard the components with low amplitudes. This problem becomes of great importance for applications where detecting the correct nonzero components is more crucial than the magnitudes of the nonzeros in the signal. This effect can be seen in Figure 16b. The curves with solid lines in this plot show the detection and false alarm rate in support recovery and the difference between the rates. This issue can be resolved by some postprocessing such as data-driven threshold tuning. That way, the amplitudes in the reconstructed signal with lower values than the threshold can be discarded. For this purpose, we set up 200 random trials, the same way as the one explained for SBL(BGiG), and then evaluate the performance in terms of NMSE by varying the threshold. Figure 16b shows the averaged results of 200 trials. The settings of these trials are represented in Table 3.



(**a**) (**b**)

Figure 16. Performance of SBL(GiG). (**a**) NMSE of SBL(GiG) vs. threshold. (**b**) Performance of SBL(GiG) before and after postprocessing.

Table 3. Settings for preprocessing analysis and simulations on SBL(GiG).

a_0	b_0	θ_0	θ_1	Sparsity	N
10^{-3}	10^{-3}	10^{-6}	10^{-6}	25	100
τ_n	$\mathbf{x}_s$	M	$\mathbf{e}$	A	$\mathbf{y}$
(22)	(25)	$5:N$	(23)	$[A]_{mn} \sim \mathcal{N}(0,1)$	$A\mathbf{x}_s + \mathbf{e}$

In Figure 16a, we observe that the postprocessing does not benefit us so much in terms of the reconstruction error for low and moderate sampling ratios. However, there is a threshold of around 0.25, for which the postprocessing step reduced the reconstruction error by approximately 3 dB. We set the threshold to 0.25 and ran 200 random trials by applying SBL(GiG) and evaluating the performance based on the detection and false alarm rate in support recovery. According to Figure 16b, the additional post-processing step provides reasonable performance.

Finally, in Figure 17, we compare the performance of the SBL(BGiG) algorithm (with performing the preprocessing step) with the SBL(GiG) algorithm (after performing post-processing). We see that Bernoulli–Gaussian-inverse Gamma implemented via SBL(BGiG) provides better performance for low and high sampling ratios. In contrast, Gaussian-inverse Gamma modeling implemented via SBL(GiG) performs much better for the moderate sampling ratios.

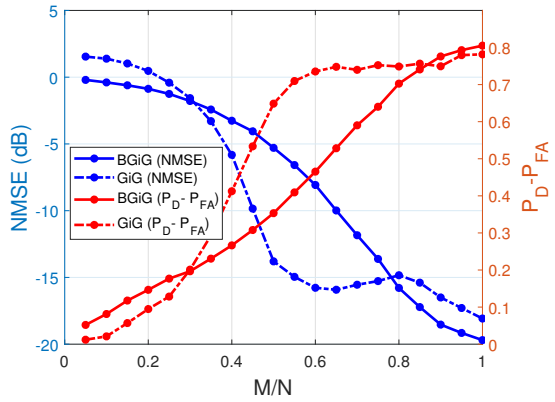


Figure 17. Performance of SBL(BGiG) and SBL(GiG) after preprocessing and postprocessing, respectively.

6. Conclusions

We investigated solving the inverse problem of compressive sensing using VB inference for two sparse Bayesian models of Bernoulli–Gaussian-inverse Gamma (BGiG) and Gaussian-inverse Gamma (GiG). The issues of each approach were discussed and the performance between the two models was compared. Specifically, we showed the behavior of these models and algorithms when the sampling ratio is low and moderate as well as the importance of selecting the hyperparameters of BGiG model with care. We further provided some intuition for performing additional pre/post-processing steps, depending on the selected model for better performance.

Based on our study on the synthetic data and considering the overall performance of both algorithms and the complexity in additional pre-/postprocessing, we observed that for moderate sampling ratios, SBL(GiG) is performing better than SBL(BGiG) modeling when using VB for sparse signals with no specific pattern in the supports. In contrast, SBL(BGiG) provided better perfomance for low and high sampling ratios. Finally, a rigorous comparison is required to study in the future under real-world scenarios and various applications. The MATLAB codes for GiG and BGiG modeling are available at https:

//github.com/MoShekaramiz/Compressive-Sensing-GiG-versus-BGiG-Modeling.git, accessed on 15 December 2022.

Author Contributions: Methodology, M.S. and T.K.M.; Formal analysis, M.S. and T.K.M.; Investigation, M.S.; Resources, M.S.; Writing—original draft, M.S.; Writing—review & editing, M.S. and T.K.M.; Visualization, M.S.; Supervision, T.K.M. All authors have read and agreed to the published version of the manuscript

Funding: This research received no external funding.

Data Availability Statement: https://github.com/MoShekaramiz/Compressive-Sensing-GiG-versus-BGiG-Modeling.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

In this section, we provide details on deriving the update rules of the parameters and variables for both models and the associated algorithms.

Appendix A.1. Bernoulli–Gaussian-Inverse-Gamma Modeling and the SBL(BGiG)

- Update rule for the precision τ of the solution value vector $\mathbf{x}$

$$\begin{aligned} q(\tau) &\propto p(\tau; a_0, b_0) e^{(<\log p(\mathbf{x}|\tau I_N)>_{q_x})} \\ &\propto \tau^{a_0-1} e^{-b_0\tau} e^{\left(<\log\left\{\prod_{n=1}^{N} p(x_n;\tau^{-1})\right\}>_{q_x}\right)} \\ &\propto \tau^{a_0-1} e^{-b_0\tau} e^{\left\{<\log\{\tau^{\frac{N}{2}} e^{\{-\frac{\tau}{2}\|\mathbf{x}\|_2^2\}}\}>_{q_x}\right\}} \\ &\propto \tau^{(a_0+\frac{N}{2})-1} e^{-(b_0+\frac{1}{2}<\|\mathbf{x}\|_2^2>_{q_x})\tau}, \end{aligned}$$

where

$$< \|\mathbf{x}\|_2^2 >_{q_x} = < \mathbf{x}^T\mathbf{x} >_{q_x} = \mathrm{Tr}(< \mathbf{x}\mathbf{x}^T >_{q_x}) = \|\tilde{\mathbf{x}}\|_2^2 + \sum_{n=1}^{N} \sigma_{\tilde{x}_n}^2,$$

and $\tilde{\mathbf{x}} := < \mathbf{x} >_{q_x}$. Therefore,

$$q(\tau) \sim \mathrm{Gamma}\big(a_0 + \frac{N}{2}, b_0 + \frac{1}{2}(\|\tilde{\mathbf{x}}\|_2^2 + \sum_{n=1}^{N} \sigma_{\tilde{x}_n}^2)\big).$$

Finally, considering the point estimate on τ as the expected value of the Gamma distribution in $q(\tau)$, the update rule for τ can be defined as

$$\tilde{\tau} = \frac{a_0 + \frac{N}{2}}{b_0 + \frac{1}{2}\left(\|\tilde{\mathbf{x}}\|_2^2 + \sum_{n=1}^{N} \sigma_{\tilde{x}_n}^2\right)}.$$

- Update rule for the noise precision ϵ

$$\begin{aligned} q(\epsilon) &\propto p(\epsilon; \theta_0, \theta_1) e^{\left\{<\log p(\mathbf{y}|\mathbf{x},\mathbf{s},\epsilon)>_{q_x q_s}\right\}} \\ &\propto \epsilon^{\theta_0-1} e^{-\theta_1\epsilon} e^{\left\{<\log\left\{\epsilon^{\frac{M}{2}} e^{\left\{-\frac{1}{2}\epsilon\|\mathbf{y}-A(\mathbf{s}\circ\mathbf{x})\|_2^2\right\}}\right\}>_{q_x q_s}\right\}} \\ &\propto \epsilon^{(\theta_0+\frac{M}{2})-1} e^{-(\theta_1+\frac{1}{2}<\|\mathbf{y}-A(\mathbf{s}\circ\mathbf{x})\|_2^2>_{q_x q_s})\epsilon}, \end{aligned}$$

where

$$\begin{aligned} < \|\mathbf{y} - A(\mathbf{s} \circ \mathbf{x})\|_2^2 >_{q_x q_s} &=< \|\mathbf{y} - AS\mathbf{x})\|_2^2 >_{q_x q_s} \\ &= \mathbf{y}^T\mathbf{y} - 2 < \mathbf{x}^T S A^T \mathbf{y} >_{q_x q_s} + < \mathbf{x}^T S A^T A S \mathbf{x} >_{q_x q_s} \\ &= \mathbf{y}^T\mathbf{y} - 2 < \mathbf{x} >_{q_x}^T < S >_{q_s} A^T\mathbf{y} + < \mathbf{x}^T S A^T A S \mathbf{x} >_{q_x q_s} \\ &= \mathbf{y}^T\mathbf{y} - 2(\tilde{\mathbf{x}} \circ \tilde{\mathbf{s}})^T A^T \mathbf{y} + < \mathbf{x}^T M_s \mathbf{x} >_{q_x q_s}, \end{aligned}$$

where $S = \text{diag}\,\{\mathbf{s}\}$, $M_s := SA^TAS$, and

$$\begin{aligned} < \mathbf{x}^T M_s \mathbf{x} >_{q_x q_s} &= \text{Tr}\,(< \mathbf{x}\mathbf{x}^T >_{q_x} < M_s >_{q_s}) \\ &= \text{Tr}\,\big((\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T + \Sigma_{\tilde{x}}) < M_s >_{q_s}\big), \end{aligned}$$

where $\Sigma_{\tilde{x}} = \text{diag}\{\sigma^2_{\tilde{x}_1}, \ldots, \sigma^2_{\tilde{x}_N}\}$, and

$$\begin{aligned} < M_s >_s &=< SA^TAS >_{q_s} \\ &=< (A^TA) \circ (\mathbf{s}\mathbf{s}^T) >_{q_s} \\ &= (A^TA) \circ < (\mathbf{s}\mathbf{s}^T) >_{q_s} \\ &= (A^TA) \circ \Big(\tilde{\mathbf{s}}\tilde{\mathbf{s}}^T + \text{diag}\,\big(\tilde{\mathbf{s}} \circ (1 - \tilde{\mathbf{s}})\big)\Big). \end{aligned}$$

Therefore,

$$< \mathbf{x}^T M_s \mathbf{x} >_{q_x q_s} = \text{Tr}\,\Big((\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T + \Sigma_{\tilde{x}})\big((A^TA) \circ (\tilde{\mathbf{s}}\tilde{\mathbf{s}}^T + \text{diag}\,\{\tilde{\mathbf{s}} \circ (1 - \tilde{\mathbf{s}})\})\big)\Big).$$

As a result,

$$q(\epsilon) \sim \text{Gamma}\big(\theta_0 + \frac{M}{2}, \theta_1 + \frac{1}{2}\tilde{\Psi}\big),$$

where

$$\begin{aligned} \tilde{\Psi} &:=< \|\mathbf{y} - A(\mathbf{s} \circ \mathbf{x})\|_2^2) >_{q_x q_s} \\ &= \mathbf{y}^T\mathbf{y} - 2(\tilde{\mathbf{x}} \circ \tilde{\mathbf{s}})^T A^T \mathbf{y} + \text{Tr}\,\Big((\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T + \Sigma_{\tilde{x}})\big((A^TA) \circ (\tilde{\mathbf{s}}\tilde{\mathbf{s}}^T + \text{diag}\,\{\tilde{\mathbf{s}} \circ (1 - \tilde{\mathbf{s}})\})\big)\Big). \end{aligned}$$

Finally, the update rule for the precision of the noise can be written as

$$\tilde{\epsilon} = \frac{\theta_0 + \frac{M}{2}}{\theta_1 + \frac{1}{2}\tilde{\Psi}},$$

Remark A1. *Notice that* $\text{Tr}\,(X^TY) = \sum_{i,j}(X \circ Y)_{ij} = \mathbf{1}^T(X \circ Y)\mathbf{1}$. *Therefore,*

$$\begin{aligned} &\text{Tr}\,\big((\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T + \Sigma_{\tilde{x}})((A^TA) \circ (\tilde{\mathbf{s}}\tilde{\mathbf{s}}^T + \text{diag}\,\{\tilde{\mathbf{s}} \circ (1 - \tilde{\mathbf{s}})\}))\big) \\ &= \mathbf{1}^T\big((\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T + \Sigma_{\tilde{x}}) \circ (A^TA) \circ (\tilde{\mathbf{s}}\tilde{\mathbf{s}}^T + \text{diag}\,\{\tilde{\mathbf{s}} \circ (1 - \tilde{\mathbf{s}})\}))\big)\mathbf{1}, \end{aligned}$$

where $\mathbf{1} = [1, \ldots, 1]^T$. *Thus,* $\tilde{\Psi}$ *can be written as*

$$\tilde{\Psi} := \mathbf{y}^T\mathbf{y} - 2(\tilde{\mathbf{x}} \circ \tilde{\mathbf{s}})^T A^T \mathbf{y} + \mathbf{1}^T\big((\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T + \Sigma_{\tilde{x}}) \circ (A^TA) \circ (\tilde{\mathbf{s}}\tilde{\mathbf{s}}^T + \text{diag}\,\{\tilde{\mathbf{s}} \circ (1 - \tilde{\mathbf{s}})\}))\big)\mathbf{1}.$$

- Update rule for γ_n

$$\begin{aligned}
q(\gamma_n) &\propto p(\gamma_n;\alpha_0,\beta_0)e^{(<\log\{p(\mathbf{x},\mathbf{s},\mathbf{y}|\theta)\}>_{q_x q_s})} \\
&\propto \gamma_n^{\alpha_0-1}(1-\gamma_n)^{\beta_0-1}e^{\{<\log\{p(s_n|\gamma_n)\}>_{q_x q_s}\}} \\
&\propto \gamma_n^{\alpha_0-1}(1-\gamma_n)^{\beta_0-1}e^{\{<\log\{\gamma_n^{s_n}(1-\gamma_n)^{1-s_n}\}>_{q_{s_n}}\}} \\
&\propto \gamma_n^{\alpha_0-1}(1-\gamma_n)^{\beta_0-1}e^{<s_n>_{q_{s_n}}\log\{\gamma_n\}}e^{(1-<s_n>_{q_{s_n}})\log\{1-\gamma_n\}} \\
&\propto \gamma_n^{\alpha_0-1}(1-\gamma_n)^{\beta_0-1}\gamma_n^{<s_n>_{q_{s_n}}}(1-\gamma_n)^{1-<s_n>_{q_{s_n}}} \\
&\propto \gamma_n^{(\alpha_0+\tilde{s}_n)-1}(1-\gamma_n)^{\beta_0-\tilde{s}_n}.
\end{aligned}$$

Therefore,

$$q_{\gamma_n}(\gamma_n) \sim \text{Beta}(\alpha_{1,n},\beta_{1,n}),\ \forall n = 1,\ldots,N,$$

where $\alpha_{1,n} := \alpha_0 + \tilde{s}_n$ and $\beta_{1,n} := \beta_0 + 1 - \tilde{s}_n$. Finally, the update rule for γ_n can be defined as

$$\tilde{\gamma}_n = \frac{\alpha_{1,n}}{\alpha_{1,n}+\beta_{1,n}}.$$

- Update rule for the solution vector $\mathbf{x}$

$$\begin{aligned}
q_x(\mathbf{x}) &\propto e^{\{<\log\{p(\mathbf{x},\mathbf{s},\mathbf{y}|\theta)\}>_{q_x q_s}\}} \\
&\propto e^{\{<\log\{p(\mathbf{x},\mathbf{s}|\theta)p(\mathbf{y}|\mathbf{x},\mathbf{s},\theta)\}>_{q_\theta q_s}\}} \\
&\propto e^{\{<\log\{p(\mathbf{x}|\theta)\}>_{q_\theta}\}}e^{\{<\log\{p(\mathbf{y}|\mathbf{x},\mathbf{s},\theta)\}>_{q_\theta q_s}\}} \\
&\propto e^{\{<\log\{p(\mathbf{x}|\tau)\}>_{q_\tau}\}}e^{\{<\log\{p(\mathbf{y}|\mathbf{x},\mathbf{s},\epsilon)\}>_{q_\epsilon q_s}\}}.
\end{aligned}$$

To update the elements of $\mathbf{x}$, we have

$$p(\mathbf{y}|\mathbf{x},\mathbf{s},\epsilon) \propto e^{\{-\frac{1}{2}\epsilon\|\mathbf{y}-A(\mathbf{s}\circ\mathbf{x})\|_2^2\}}.$$

Therefore,

$$\begin{aligned}
<\log\{p(\mathbf{y}|\mathbf{x},\mathbf{s},\epsilon)\}>_{q_\epsilon q_s} &\propto -\frac{1}{2}<\epsilon\|\mathbf{y}-A(\mathbf{s}\circ\mathbf{x})\|_2^2>_{q_\epsilon q_s} \\
&\propto -\frac{1}{2}<\epsilon>_{q_\epsilon}<\|\mathbf{y}-A(\mathbf{s}\circ\mathbf{x})\|_2^2>_{q_s} \\
&\propto -\frac{1}{2}\tilde{\epsilon}<|\mathbf{y}-A(\mathbf{s}\circ\mathbf{x})\|_2^2>_{q_s}
\end{aligned}$$

and

$$\begin{aligned}
<\|\mathbf{y}-A(\mathbf{s}\circ\mathbf{x})\|_2^2>_{q_s} &= <\text{Tr}\left(\mathbf{y}\mathbf{y}^T+(\mathbf{x}\circ\mathbf{s})^TA^TA(\mathbf{x}\circ\mathbf{s})-2(\mathbf{x}\circ\mathbf{s})^TA^T\mathbf{y}\right)>_{q_s} \\
&\propto <\text{Tr}\left((\mathbf{x}\circ\mathbf{s})^TA^TA(\mathbf{x}\circ\mathbf{s})-2(\mathbf{x}\circ\mathbf{s})^TA^T\mathbf{y}\right)>_{q_s} \\
&\propto <\text{Tr}\left(\mathbf{x}^TS^TA^TAS\mathbf{x}-2\mathbf{x}^TSA^T\mathbf{y}\right)>_{q_s} \\
&\propto \text{Tr}\left(\mathbf{x}^T<SA^TAS>_{q_s}\mathbf{x}-2\mathbf{x}^T\tilde{S}A^T\mathbf{y}\right).
\end{aligned}$$

This yields to

$$<\|\mathbf{y}-A(\mathbf{s}\circ\mathbf{x})\|_2^2>_{q_s} \propto \mathbf{x}^T<SA^TAS>_{q_s}\mathbf{x}-2\mathbf{x}^T\tilde{S}A^T\mathbf{y},$$

which results in

$$<\log p(\mathbf{y}|\mathbf{x},\mathbf{s},\epsilon)>_{q_\epsilon q_s} \propto -\frac{1}{2}\tilde{\epsilon}(\mathbf{x}^T<SA^TAS>_{q_s}\mathbf{x}-2\mathbf{x}^T\tilde{S}A^T\mathbf{y}).$$

Thus, we can write $q_x(\mathbf{x})$ as

$$\begin{aligned} q_x(\mathbf{x}) &\propto e^{<\log\{p(\mathbf{x}|\tau)\}>_{q_\tau}} e^{<\log\{p(\mathbf{y}|\mathbf{x},\mathbf{s},\epsilon)\}>_{q_\epsilon q_s}} \\ &\propto e^{\{-\frac{1}{2}\tilde{\tau}\mathbf{x}^T\mathbf{x}\}} e^{\{-\frac{1}{2}\tilde{\epsilon}(\mathbf{x}^T<SA^TAS>_{q_s}\mathbf{x}-2\mathbf{x}^T\tilde{S}A^T\mathbf{y})\}} \\ &\propto e^{\{-\frac{1}{2}(\mathbf{x}^T(\tilde{\tau}I_N+\tilde{\epsilon}<SA^TAS>_{q_s})\mathbf{x}-2\tilde{\epsilon}\mathbf{x}^T\tilde{S}A^T\mathbf{y})\}}. \end{aligned}$$

Notice that $SA^TAS = (A^TA)\circ(\mathbf{s}\mathbf{s}^T)$. Since s_n is drawn from a Bernoulli distribution, we have $< s_n^2 >_{q_s} = < s_n >_{q_s} = \tilde{s}_n$, and

$$\tilde{\mathbf{s}}\tilde{\mathbf{s}}^T = \begin{bmatrix} \tilde{s}_1^2 & \tilde{s}_1\tilde{s}_2 & \dots & \tilde{s}_1\tilde{s}_n \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{s}_n\tilde{s}_1 & \tilde{s}_n\tilde{s}_2 & \dots & \tilde{s}_n^2 \end{bmatrix}$$

Therefore,

$$\begin{aligned} < S^TA^TAS >_{q_s} &= (A^TA)\circ(\tilde{\mathbf{s}}\tilde{\mathbf{s}}^T - \mathrm{diag}\{\tilde{\mathbf{s}}\circ\tilde{\mathbf{s}}\} + \mathrm{diag}\{\tilde{\mathbf{s}}\}) \\ &= (A^TA)\circ(\tilde{\mathbf{s}}\tilde{\mathbf{s}}^T + \mathrm{diag}\{\tilde{\mathbf{s}}\circ(1-\tilde{\mathbf{s}})\}), \end{aligned}$$

which yields to

$$q_x(\mathbf{x}) \sim \mathcal{N}(\tilde{\mathbf{x}}, \Sigma_{\tilde{x}}),$$

where

$$\Sigma_{\tilde{x}} = \Big(\tilde{\tau}I_N + \tilde{\epsilon}\big((A^TA)\circ(\tilde{\mathbf{s}}\tilde{\mathbf{s}}^T + \mathrm{diag}\{\tilde{\mathbf{s}}\circ(1-\tilde{\mathbf{s}})\})\big)\Big)^{-1}$$

and

$$\tilde{\mathbf{x}} = \tilde{\epsilon}\Sigma_{\tilde{x}}\tilde{S}A^T\mathbf{y},$$

which $\tilde{\mathbf{x}}$ is the update rule for the solution value vector $\mathbf{x}$.

- Update rule for the support vector $\mathbf{s}$

$$\begin{aligned} q_{s_n}(s_n) &\sim e^{\{<\log\{p(\mathbf{x},\mathbf{s},\mathbf{y}|\theta)\}>_{q_\theta q_x}\}} \\ &\propto e^{\{<\log\{p(\mathbf{x},\mathbf{s}|\theta)p(\mathbf{y}|\mathbf{x},\mathbf{s},\theta)\}>_{q_\theta q_x}\}} \\ &\propto e^{\{<\log\{p(s_n;\gamma_n)\}>_{q_{\gamma_n}}\}} e^{\{<\log\{p(\mathbf{y}|\mathbf{x},\mathbf{s},\epsilon)\}>_{q_{\mathbf{s}-n}q_x q_\epsilon}\}} \\ &\propto e^{\{<\log\{\gamma_n^{s_n}(1-\gamma_n)^{1-s_n}\}>_{q_{\gamma_n}}\}} e^{\{<\log\{p(\mathbf{y}|\mathbf{x},\mathbf{s},\epsilon)\}>_{q_{\mathbf{s}-n}q_x q_\epsilon}\}}, \end{aligned}$$

where

$$e^{<\log\{\gamma_n^{s_n}(1-\gamma_n)^{1-s_n}\}>_{q_{\gamma n}}} = e^{s_n<\log\{\gamma_n\}>_{q_{\gamma n}}} e^{(1-s_n)<\log\{1-\gamma_n\}>_{q_{\gamma n}}},$$

for which

$$< \log\gamma_n >_{q_{\gamma_n}} \sim \mathrm{Beta}(\alpha_{1,n},\beta_{1,n}) = \psi(\alpha_{1,n}) - \psi(\alpha_{1,n}+\beta_{1,n})$$

and

$$< \log\{1-\gamma_n\} >_{q_{\gamma_n}} \sim \mathrm{Beta}(\alpha_{1,n},\beta_{1,n}) = \psi(\beta_{1,n}) - \psi(\alpha_{1,n}+\beta_{1,n}),$$

where $\psi(\cdot)$ is digamma function, the logarithmic derivative of the gamma function, i.e., $\psi(x) = \frac{d}{dx}\log\Gamma(x)$. Therefore,

$$e^{<\log\{\gamma_n^{s_n}(1-\gamma_n)^{1-s_n}\}>_{q_{\gamma n}}} = e^{s_n\left(\psi(\alpha_{1,n})-\psi(\alpha_{1,n}+\beta_{1,n})\right)} e^{(1-s_n)\left(\psi(\beta_{1,n})-\psi(\alpha_{1,n}+\beta_{1,n})\right)}.$$

Also,

$$
\begin{aligned}
e^{<\log\{p(\mathbf{y}|\mathbf{x},\mathbf{s},\epsilon)\}>_{q_{\mathbf{s}^{-n}}q_x q_\epsilon}} &\propto e^{-\frac{1}{2}<\epsilon\|\mathbf{y}-A(\mathbf{s}\circ\mathbf{x})\|_2^2>_{q_{\mathbf{s}^{-n}}q_x q_\epsilon}} \\
&\propto e^{-\frac{1}{2}\tilde{\epsilon}<\|\mathbf{y}-A(\mathbf{s}\circ\mathbf{x})\|_2^2>_{q_{\mathbf{s}^{-n}}q_x}} \\
&\propto e^{-\frac{1}{2}\tilde{\epsilon}<\sum_{m=1}^{M}(y_m-\sum_{n=1}^{N}a_{mn}s_n x_n)^2>_{q_{\mathbf{s}^{-n}}q_x}} \\
&\propto e^{-\frac{1}{2}\tilde{\epsilon}<\left((y_1-\sum_{l\neq n}^{N}a_{1n}s_n x_n)-a_{1n}s_n x_n\right)^2+\cdots+\left((y_M-\sum_{l\neq n}^{N}a_{Ml}s_l x_l)-a_{Mn}s_n x_n\right)^2>_{q_{\mathbf{s}^{-n}}q_x}},
\end{aligned}
$$

where $y_m^{-n} := y_m - \sum_{l\neq n}^{N} a_{ml}s_l x_l$, $\forall m = 1, 2, \ldots, M$. Therefore,

$$
\begin{aligned}
e^{<\log\{p(\mathbf{y}|\mathbf{x},\mathbf{s},\epsilon)\}>_{q_{\mathbf{s}^{-n}}q_x q_\epsilon}} &\propto e^{-\frac{1}{2}\tilde{\epsilon}<\sum_{m=1}^{M}(a_{mn}s_n x_n-y_m^{-n})^2>_{q_{\mathbf{s}^{-n}}q_x}} \\
&\propto e^{-\frac{1}{2}\epsilon\sum_{m=1}^{M}\left(a_{mn}^2 s_n^2<x_n^2>_{q_x}-2a_{mn}s_n<x_n y_m^{-n}>_{q_{\mathbf{s}^{-n}}q_x}\right)} \\
&\propto e^{-\frac{1}{2}\tilde{\epsilon}\left(\|\mathbf{a}_n\|_2^2 s_n^2<x_n^2>_{q_x}-2\sum_{m=1}^{M}a_{mn}s_n<x_n y_m^{-n}>_{q_{\mathbf{x}},q_{\mathbf{s}^{-n}}}\right)} \\
&\propto e^{-\frac{1}{2}\tilde{\epsilon}\left(\|\mathbf{a}_n\|_2^2 s_n^2(\tilde{x}_n^2+\sigma_{\tilde{x}_n}^2)-2s_n\tilde{x_n}\sum_{m=1}^{M}a_{mn}<y_m^{-n}>_{q_{\mathbf{s}^{-n}}q_x}\right)} \\
&\propto e^{-\frac{1}{2}\tilde{\epsilon}\left(\|\mathbf{a}_n\|_2^2(\tilde{x}_n^2+\sigma_{\tilde{x}_n}^2)s_n^2-2s_n\tilde{x}_n\mathbf{a}_n^T<\mathbf{y}^{-n}>_{q_{\mathbf{s}^{-n}}q_x}\right)},
\end{aligned}
$$

where y_m^{-n} contains no x_n component and

$$\mathbf{y}^{-n} := [y_1^{-n}, y_2^{-n}, \ldots, y_M^{-n}].$$

Thus,

$$< y_m^{-n} >_{q_{\mathbf{s}^{-n}}q_x} = < y_m - \sum_{l\neq n}^{N} a_{ml}s_l x_l >_{q_{\mathbf{s}^{-n}}q_x} = y_m - \sum_{l\neq n}^{N} a_{ml}\tilde{s}_l\tilde{x}_l$$

which yields to

$$
\begin{aligned}
\tilde{y}_m^{-n} &:= < y_m^{-n} >_{q_{\mathbf{s}^{-n}}q_x} \\
\tilde{\mathbf{y}}^{-n} &= \mathbf{y} - \sum_{l\neq n}^{N}\tilde{s}_l\tilde{x}_l\mathbf{a}_l.
\end{aligned}
$$

and thus

$$\tilde{\mathbf{y}}^{-n} := < \mathbf{y}^{-n} >_{q_{\mathbf{s}^{-n}}q_x} .$$

Therefore,

$$e^{<\log\{p(\mathbf{y}|\mathbf{x},\mathbf{s},\epsilon)\}>_{q_{\mathbf{s}^{-n}}q_x q_\epsilon}} \propto e^{-\frac{1}{2}\tilde{\epsilon}\left(\left(\|\mathbf{a}_n\|_2^2(\tilde{x}_n^2+\sigma_{\tilde{x}_n}^2)\right)s_n^2-2(\tilde{x}_n\mathbf{a}_n^T\tilde{\mathbf{y}}^{-n})s_n\right)}.$$

Finally,

$$q_{s_n}(s_n) \propto e^{\left\{s_n\left(\psi(\alpha_{1,n})-\psi(\alpha_{1,n}+\beta_{1,n})\right)+(1-s_n)\left(\psi(\beta_{1,n})-\psi(\alpha_{1,n}+\beta_{1,n})\right)-\frac{1}{2}\tilde{\epsilon}\left(\|\mathbf{a}_n\|_2^2(\tilde{x}_n^2+\sigma_{\tilde{x}_n}^2)s_n^2-2\tilde{x}_n\mathbf{a}_n^T\tilde{\mathbf{y}}^{-n}s_n\right)\right\}}.$$

Since s_n is an outcome of a Bernoulli random variable,

$$q_{s_n}(s_n = 0) \propto e^{\left\{\psi(\beta_{1,n})-\psi(\alpha_{1,n}+\beta_{1,n})\right\}}$$

and

$$q_{s_n}(s_n = 1) \propto e^{\left\{\psi(\alpha_{1,n})-\psi(\alpha_{1,n}+\beta_{1,n})-\frac{1}{2}\tilde{\epsilon}\left(\|\mathbf{a}_n\|_2^2(\tilde{x}_n^2+\sigma_{\tilde{x}_n}^2)-2\tilde{x}_n\mathbf{a}_n^T\tilde{\mathbf{y}}^{-n}\right)\right\}}.$$

Therefore,

$$\begin{aligned} q_{s_n}(s_n) &\sim \text{Bernoulli}\Big(\frac{q_{s_n}(s_n=1)}{q_{s_n}(s_n=0)+q_{s_n}(s_n=1)}\Big) \\ &\sim \text{Bernoulli}\Big(\frac{1}{1+\frac{q_{s_n}(s_n=0)}{q_{s_n}(s_n=1)}}\Big), \end{aligned}$$

which yields to

$$\begin{aligned} q_{s_n}(s_n) &\sim \text{Bernoulli}\Big(\frac{1}{1+e^{\psi(\beta_{1,n})-\psi(\alpha_{1,n}+\beta_{1,n})}e^{-\psi(\alpha_{1,n})+\psi(\alpha_{1,n}+\beta_{1,n})+\frac{1}{2}\tilde{\epsilon}\left(\|\mathbf{a}_n\|_2^2(\tilde{x}_n^2+\sigma_{\tilde{x}_n}^2)-2\tilde{x}_n\mathbf{a}_n^T\tilde{\mathbf{y}}^{-n}\right)}}\Big) \\ &\sim \text{Bernoulli}\Big(\frac{1}{1+e^{\left\{\psi(\beta_{1,n})-\psi(\alpha_{1,n})+\frac{1}{2}\tilde{\epsilon}\left(\|\mathbf{a}_n\|_2^2(\tilde{x}_n^2+\sigma_{\tilde{x}_n}^2)-2\tilde{x}_n\mathbf{a}_n^T\tilde{\mathbf{y}}^{-n}\right)\right\}}}\Big). \end{aligned}$$

The update rule for the component s_n can then be written as

$$\tilde{s}_n = \frac{1}{1+e^{\left\{\psi(\beta_{1,n})-\psi(\alpha_{1,n})+\frac{1}{2}\tilde{\epsilon}\left(\|\mathbf{a}_n\|_2^2(\tilde{x}_n^2+\sigma_{\tilde{x}_n}^2)-2\tilde{x}_n\mathbf{a}_n^T\tilde{\mathbf{y}}^{-n}\right)\right\}}}$$

or equivalently,

$$\tilde{s}_n = \frac{1}{1+c_n\kappa_n},\ \forall n=1,\ldots,N,$$

where

$$c_n := e^{\{\psi(\beta_{1,n})-\psi(\alpha_{1,n})\}}$$

and

$$\kappa_n := e^{\left\{\frac{1}{2}\tilde{\epsilon}\left(\|\mathbf{a}_n\|_2^2(\tilde{x}_n^2+\sigma_{\tilde{x}_n^2}^2)-2\tilde{x}_n\mathbf{a}_n^T\tilde{\mathbf{y}}^{-n}\right)\right\}}.$$

- Stopping rule

The stopping rule of the algorithm can be set based on the marginalized likelihood (evidence). We would rather follow the effect of $\mathbf{s}$ on the evidence because if $\mathbf{s}$ is learned, it would be easy to compute $\mathbf{x}_s$. Therefore, we marginalize the distribution on $\mathbf{y}$ and integrate $\mathbf{x}$ out. The details are described below.

$$\begin{aligned} p(\mathbf{y}|\mathbf{s},\epsilon,\tau) &= \int p(\mathbf{y},\mathbf{x}|\mathbf{s},\epsilon,\tau)d\mathbf{x} \\ &= \int p(\mathbf{y}|\mathbf{x},\mathbf{s},\epsilon,\tau)p(\mathbf{x}|\tau)d\mathbf{x} \\ &= \int \frac{1}{(2\pi\epsilon^{-1})^{\frac{M}{2}}}e^{-\frac{1}{2}\epsilon\|\mathbf{y}-A(\mathbf{s}\circ\mathbf{x})\|_2^2}\frac{1}{(2\pi\tau^{-1})^{\frac{N}{2}}}e^{-\frac{1}{2}\tau\|\mathbf{x}\|_2^2}d\mathbf{x} \\ &= \frac{1}{(2\pi)^{\frac{M}{2}}}\epsilon^{\frac{M}{2}}\tau^{\frac{N}{2}}\int\frac{1}{(2\pi)^{\frac{N}{2}}}e^{-\frac{1}{2}\left(\epsilon\left(\mathbf{y}^T\mathbf{y}-2(\mathbf{s}\circ\mathbf{x})^TA^T\mathbf{y}+(\mathbf{s}\circ\mathbf{x})^TA^TA(\mathbf{s}\circ\mathbf{x})\right)+\tau\mathbf{x}^T\mathbf{x}\right)}d\mathbf{x} \\ &= \frac{1}{(2\pi)^{\frac{M}{2}}}\epsilon^{\frac{M}{2}}\tau^{\frac{N}{2}}\int\frac{1}{(2\pi)^{\frac{N}{2}}}e^{-\frac{1}{2}\left(\epsilon\left(\mathbf{y}^T\mathbf{y}-2\mathbf{x}^TSA^T\mathbf{y}+\mathbf{x}^TSA^TAS\mathbf{x}\right)+\tau\mathbf{x}^T\mathbf{x}\right)}d\mathbf{x} \\ &= \frac{1}{(2\pi)^{\frac{M}{2}}}\epsilon^{\frac{M}{2}}\tau^{\frac{N}{2}}e^{-\frac{1}{2}\epsilon\mathbf{y}^T\mathbf{y}}\int\frac{1}{(2\pi)^{\frac{N}{2}}}e^{-\frac{1}{2}\left(\mathbf{x}^T(\epsilon SA^TAS+\tau I_N)\mathbf{x}-2\epsilon\mathbf{x}^TSA^T\mathbf{y}\right)}d\mathbf{x} \end{aligned}$$

$$
\begin{aligned}
p(\mathbf{y}|\mathbf{s},\epsilon,\tau) =& \frac{1}{(2\pi)^{\frac{M}{2}}}\epsilon^{\frac{M}{2}}\tau^{\frac{N}{2}}e^{-\frac{1}{2}\epsilon\mathbf{y}^T\mathbf{y}}|(\tau I_N+\epsilon SA^TAS)^{-1}|^{\frac{1}{2}}\int\frac{1}{(2\pi)^{\frac{N}{2}}}\frac{1}{|(\tau I_N+\epsilon SA^TAS)^{-1}|^{\frac{1}{2}}}\times\ldots\\
& e^{-\frac{1}{2}\left(\mathbf{x}^T(\epsilon SA^TAS+\tau I_N)\mathbf{x}-2\epsilon\mathbf{x}^TSA^T\mathbf{y}\right)}d\mathbf{x}\\
=& \frac{1}{(2\pi)^{\frac{M}{2}}}\epsilon^{\frac{M}{2}}\tau^{\frac{N}{2}}e^{-\frac{1}{2}\epsilon\mathbf{y}^T\mathbf{y}}|(\tau I_N+\epsilon SA^TAS)^{-1}|^{\frac{1}{2}}\int\frac{1}{(2\pi)^{\frac{N}{2}}}\frac{1}{|(\tau I_N+\epsilon SA^TAS)^{-1}|^{\frac{1}{2}}}\times\ldots\\
& e^{-\frac{1}{2}\left(\left(\mathbf{x}-(\tau I_N+\epsilon SA^TAS)^{-1}\epsilon SA^T\mathbf{y}\right)^T(\tau I_N+\epsilon SA^TAS)\left(\mathbf{x}-(\tau I_N+\epsilon SA^TAS)^{-1}\epsilon SA^T\mathbf{y}\right)-\epsilon^2\mathbf{y}^TAS(\tau I_N+\epsilon SA^TAS)^{-1}SA^T\mathbf{y}\right)}d\mathbf{x},
\end{aligned}
$$

which results in

$$
p(\mathbf{y}|\mathbf{s},\epsilon,\tau) = \frac{1}{(2\pi)^{\frac{M}{2}}}\epsilon^{\frac{M}{2}}\tau^{\frac{N}{2}}e^{-\frac{1}{2}\epsilon\mathbf{y}^T\mathbf{y}}|(\tau I_N+\epsilon SA^TAS)^{-1}|^{\frac{1}{2}}e^{\frac{1}{2}\epsilon^2\mathbf{y}^TAS(\tau I_N+\epsilon SA^TAS)^{-1}SA^T\mathbf{y}}.
$$

Thus,

$$
\begin{aligned}
\log p(\mathbf{y}|\mathbf{s},\epsilon,\tau) =& -\frac{M}{2}\log\{2\pi\}+\frac{M}{2}\log\epsilon+\frac{N}{2}\log\tau-\frac{1}{2}\epsilon\mathbf{y}^T\mathbf{y}+\ldots\\
& \frac{1}{2}\log\{|(\tau I_N+\epsilon SA^TAS)^{-1}|\}+\frac{1}{2}\epsilon^2\mathbf{y}^TAS(\tau I_N+\epsilon SA^TAS)^{-1}SA^T\mathbf{y}
\end{aligned}
$$

and

$$
-\frac{1}{2}\epsilon\mathbf{y}^T\mathbf{y}+\frac{1}{2}\epsilon^2\mathbf{y}^TAS(\tau I_N+\epsilon SA^TAS)^{-1}SA^T\mathbf{y} = -\frac{1}{2}\mathbf{y}^T\left(I_M-\epsilon AS(\tau I_N+\epsilon SA^TAS)^{-1}SA^T\right)\mathbf{y}
$$

Also,

$$
\begin{aligned}
\frac{N}{2}\log\{\tau\}+\frac{1}{2}\log\{|(\tau I_N+\epsilon SA^TAS)^{-1}|\} &= \frac{1}{2}\log\{|(\tau I_N)(\tau I_N+\epsilon SA^TAS)^{-1}|\}\\
&= -\frac{1}{2}\log\{|(\tau^{-1}I_N)(\tau I_N+\epsilon SA^TAS)|\}\\
&= -\frac{1}{2}\log\{|I_N+\frac{\epsilon}{\tau}SA^TAS|\}\\
&= -\frac{1}{2}\log\{|I_M+\frac{\epsilon}{\tau}AS^2A^T|\}.
\end{aligned}
$$

Thus,

$$
\begin{aligned}
L :=& \log p(\mathbf{y}|\mathbf{s},\epsilon,\tau)\\
=& -\frac{M}{2}\log\{2\pi\}+\frac{M}{2}\log\{\epsilon\}-\frac{1}{2}\log|I_M+\frac{\epsilon}{\tau}AS^2A^T|-\frac{1}{2}\epsilon\mathbf{y}^T\left(I_M-\epsilon AS(\tau I_N+\epsilon SA^TAS)^{-1}SA^T\right)\mathbf{y}.
\end{aligned}
$$

For comparing the changes of $L^{[t]}$ with $L^{[t-1]}$ in the updating process, we have

$$
\begin{aligned}
L &\propto \frac{M}{2}\log\{\epsilon\}+\frac{1}{2}\log\{|(I_M+\frac{\epsilon}{\tau}AS^2A^T)^{-1}|\}-\frac{1}{2}\epsilon\mathbf{y}^T\left(I_M-\epsilon AS(\tau I_N+\epsilon SA^TAS)^{-1}SA^T\right)\mathbf{y}\\
&\propto \frac{1}{2}\left(\log\{|\epsilon I_M|\}+\log\{|(I_M+\frac{\epsilon}{\tau}AS^2A^T)^{-1}|\}\right)-\frac{1}{2}\mathbf{y}^T(\epsilon^{-1}I_M+\frac{1}{\tau}AS^2A^T)^{-1}\mathbf{y}\\
&\propto \frac{1}{2}\log\{|\epsilon^{-1}I_M|^{-1}|I_M+\frac{\epsilon}{\tau}AS^2A^T|^{-1}\}-\frac{1}{2}\mathbf{y}^T(\epsilon^{-1}I_M+\frac{1}{\tau}AS^2A^T)^{-1}\mathbf{y}\\
&\propto \frac{1}{2}\log\{|\epsilon^{-1}I_M(I_M+\frac{\epsilon}{\tau}AS^2A^T)^{-1}|\}-\frac{1}{2}\mathbf{y}^T(\epsilon^{-1}I_M+\frac{1}{\tau}AS^2A^T)^{-1}\mathbf{y}\\
&\propto \log\{|\epsilon^{-1}I_M+\frac{1}{\tau}AS^2A^T|^{-1}\}-\mathbf{y}^T(\epsilon^{-1}I_M+\frac{1}{\tau}AS^2A^T)^{-1}\mathbf{y}.
\end{aligned}
$$

Therefore,

$$L^{[t]} \propto \log|\Sigma_0^{[t]}| - \mathbf{y}^T\Sigma_0^{[t]}\mathbf{y},$$

where

$$\Sigma_0 := (\tilde{\epsilon}^{-1}I_M + \tilde{\tau}^{-1}A\tilde{S}^2A^T)^{-1},$$

which yields to

$$-L \propto \log\{|\Sigma_0^{-1}|\} + \mathbf{y}^T\Sigma_0\mathbf{y}.$$

This means that

$$p(\mathbf{y}|S,\epsilon,\tau) = \frac{1}{(2\pi)^{\frac{M}{2}}}\frac{1}{|\Sigma_0^{-1}|^{\frac{1}{2}}}e^{\{-\frac{1}{2}\mathbf{y}^T\Sigma_0\mathbf{y}\}}$$

or equivalently,

$$p(\mathbf{y}|\mathbf{s},\epsilon,\tau) \sim \mathcal{N}(\mathbf{0},\Sigma_0^{-1}).$$

Therefore, the stopping criterion can be made based on

$$\begin{aligned}\Delta L^{[t]} :=& L^{[t]} - L^{[t-1]}\\ =& \log\{\frac{\Sigma_0^{[t]}}{\Sigma_0^{[t-1]}}\} + \mathbf{y}^T(\Sigma_0^{[t-1]} - \Sigma_0^{[t]})\mathbf{y}.\end{aligned}$$

Appendix A.2. Gaussian-Inverse-Gamma Modeling and the SBL(GiG)

- Update rule for the precision τ_n of the nth component of the solution vector $\mathbf{x}$

$$\begin{aligned}q(\tau_n) &\propto p(\tau_n;a_0,b_0)e^{(<\log p(\mathbf{x}|T)>_{q_{x_n}})}\\ &\propto \tau_n^{a_0-1}e^{-b_0\tau_n}e^{\{<\log\left\{\prod_{n=1}^N p(x_n;\tau_n^{-1})\right\}>_{q_{x_n}}\}}\\ &\propto \tau_n^{a_0-1}e^{-b_0\tau_n}e^{\{<\log\{\tau_n^{\frac{1}{2}}e^{-\frac{\tau_n}{2}x_n^2}\}>_{q_{x_n}}\}}\\ &\propto \tau_n^{a_0+\frac{1}{2}-1}e^{-b_0\tau_n}e^{\{-\frac{\tau_n}{2}<x_n^2>_{q_{x_n}}\}}\\ &\propto \tau_n^{(a_0+\frac{1}{2})-1}e^{-b_0\tau_n}e^{-\frac{\tau_n}{2}(\tilde{x}_n^2+\sigma_{\tilde{x}_n}^2)}\\ &\propto \tau_n^{(a_0+\frac{1}{2})-1}e^{-\left(b_0+\frac{1}{2}(\tilde{x}_n^2+\sigma_{\tilde{x}_n}^2)\right)\tau_n},\end{aligned}$$

  where $T := \text{diag}\{\tau_1,\ldots,\tau_N\}$. Therefore, we can model τ_n as

$$q(\tau_n) \sim \text{Gamma}\big(a_0+\frac{1}{2}, b_0+\frac{1}{2}(\tilde{x}_n^2+\sigma_{\tilde{x}_n}^2)\big).$$

The update rule for τ_n can be then defined as follows.

$$\tilde{\tau}_n = \frac{a_0+\frac{1}{2}}{b_0+\frac{1}{2}(\tilde{x}_n^2+\sigma_{\tilde{x}_n}^2)},\ \forall n = 1,2,\ldots,N$$

- Update rule for the noise precision ϵ

$$\begin{aligned}q(\epsilon) &\propto p(\epsilon;\theta_0,\theta_1)e^{\{<\log p(\mathbf{y}|\mathbf{x},\epsilon)>_{q_x}\}}\\ &\propto \epsilon^{\theta_0-1}e^{-\theta_1\epsilon}e^{\{<\log\{\epsilon^{\frac{M}{2}}e^{(-\frac{1}{2}\epsilon\|\mathbf{y}-A\mathbf{x}\|_2^2)}\}>_{q_x}\}}\\ &\propto \epsilon^{(\theta_0+\frac{M}{2})-1}e^{-\epsilon(\theta_1+\frac{1}{2}<\|\mathbf{y}-A\mathbf{x}\|_2^2>_{q_x})},\end{aligned}$$

where

$$\begin{aligned}<\|\mathbf{y}-A\mathbf{x}\|_2^2>_{q_x} &= \mathbf{y}^T\mathbf{y} - 2<\mathbf{x}>_{q_x}^T A^T\mathbf{y} + <\mathbf{x}^TA^TA\mathbf{x}>q_x\\ &= \mathbf{y}^T\mathbf{y} - 2\tilde{\mathbf{x}}^TA^T\mathbf{y} + <\mathbf{x}^TA^TA\mathbf{x}>_{q_x},\end{aligned}$$

and

$$\begin{aligned} < \mathbf{x}^T A^T A \mathbf{x} >_{q_x} &= \mathrm{Tr}\left(< \mathbf{x}^T A^T A \mathbf{x} >_{q_x} \right) \\ &= \mathrm{Tr}\left(< \mathbf{x}\mathbf{x}^T >_{q_x} A^T A \right) \\ &= \mathrm{Tr}\left((\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T + \Sigma_{\tilde{x}}) A^T A \right). \end{aligned}$$

Therefore,

$$\tilde{\Psi} := < \|\mathbf{y} - A\mathbf{x}\|_2^2 >_{q_x} = \mathbf{y}^T\mathbf{y} - 2\tilde{\mathbf{x}}^T A^T \mathbf{y} + \mathrm{Tr}\left((\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T + \Sigma_{\tilde{x}}) A^T A \right).$$

Therefore, we can model $\hat{\epsilon}$ as

$$q(\epsilon) \sim \mathrm{Gamma}\Big(\theta_0 + \frac{M}{2}, \theta_1 + \frac{1}{2}\tilde{\Psi}\Big).$$

Finally, the update rule for ϵ can be then written as

$$\tilde{\epsilon} = \frac{\theta_0 + \frac{M}{2}}{\theta_1 + \frac{1}{2}\tilde{\Psi}}.$$

- Update rule for the solution vector $\mathbf{x}$

$$\begin{aligned} q_x(\mathbf{x}) &\propto e^{\{ < \log\{p(\mathbf{x},\mathbf{y}|\theta)\} >_{q_\theta} \}} \\ &\propto e^{\{ < \log\{p(\mathbf{x}|\theta)p(\mathbf{y}|\mathbf{x},\theta)\} >_{q_\theta} \}} \\ &\propto e^{\{ < \log\{p(\mathbf{x}|T)\} >_{q_\tau} \}} e^{\{ < \log\{p(\mathbf{y}|\mathbf{x},\epsilon)\} >_{q_\epsilon} \}} \\ &\propto e^{\{ < \log\{p(\mathbf{x}|T)\} >_{q_\tau} \}} e^{\{ -\frac{1}{2}\mathbf{x}^T \tilde{T}\mathbf{x} \}}, \end{aligned}$$

where θ contains the information on the parameters T and ϵ, and $\tilde{T} := \mathrm{diag}\{\tilde{\tau}_1, \ldots, \tilde{\tau}_N\}$. To update the elements of $\mathbf{x}$, we have

$$\begin{aligned} p(\mathbf{y}, \mathbf{x}, \epsilon) &\propto \epsilon^{\frac{M}{2}} e^{\{ -\frac{1}{2}\epsilon\|\mathbf{y} - A\mathbf{x}\|_2^2 \}} \\ &\propto e^{\{ -\frac{1}{2}\epsilon\|\mathbf{y} - A\mathbf{x}\|_2^2 \}}. \end{aligned}$$

Therefore,

$$\begin{aligned} < \log\{p(\mathbf{y}|\mathbf{x},\epsilon)\} >_{q_\epsilon} &\propto -\frac{1}{2} < \epsilon \|\mathbf{y} - A\mathbf{x}\|_2^2 >_{q_\epsilon} \\ &\propto -\frac{1}{2} < \epsilon >_{q_\epsilon} \|\mathbf{y} - A\mathbf{x}\|_2^2 \\ &\propto -\frac{1}{2}\tilde{\epsilon}\|\mathbf{y} - A\mathbf{x}\|_2^2. \end{aligned}$$

Thus, we can write $q_x(\mathbf{x})$ as

$$\begin{aligned} q_x(\mathbf{x}) &\propto e^{< \log\{p(\mathbf{x}|T)\} >_{q_T}} e^{< \log p(\mathbf{y}|\mathbf{x},\theta) >_{q_\theta}} \\ &\propto e^{\{ -\frac{1}{2}\mathbf{x}^T \tilde{T}\mathbf{x} \}} e^{\{ -\frac{1}{2}\tilde{\epsilon}(\mathbf{x}^T A^T A\mathbf{x} - 2\mathbf{x}^T A^T \mathbf{y}) \}} \\ &\propto e^{\{ -\frac{1}{2}\left(\mathbf{x}^T (\tilde{T} + \tilde{\epsilon} A^T A)\mathbf{x} - 2\tilde{\epsilon}\mathbf{x}^T A^T \mathbf{y} \right) \}}. \end{aligned}$$

Finally,

$$q_x(\mathbf{x}) \sim \mathcal{N}(\tilde{\mathbf{x}}, \Sigma_{\tilde{x}}),$$

where

$$\Sigma_{\tilde{x}} := (\tilde{T} + \tilde{\epsilon} A^T A)^{-1} \text{ and } \tilde{\mathbf{x}} := \tilde{\epsilon}\Sigma_{\tilde{x}} A^T \mathbf{y}.$$

- Stopping rule

We set the stopping rule of the algorithm based on the marginalized log-likelihood (evidence) defined as

$$\begin{aligned} p(\mathbf{y}|\epsilon,T) &= \int p(\mathbf{y},\mathbf{x}|\epsilon,T)d\mathbf{x} \\ &\int p(\mathbf{y}|\mathbf{x},\epsilon,T)p(\mathbf{x}|T)d\mathbf{x} \\ &= \int \frac{1}{(2\pi\epsilon^{-1})^{\frac{M}{2}}} e^{-\frac{1}{2}\epsilon\|\mathbf{y}-A\mathbf{x}\|_2^2} \frac{1}{((2\pi)^N|T^{-1}|)^{\frac{1}{2}}} e^{-\frac{1}{2}\mathbf{x}^T T\mathbf{x}} d\mathbf{x} \\ &= \frac{1}{(2\pi)^{\frac{M}{2}}}\epsilon^{\frac{M}{2}}|T|^{\frac{1}{2}} \int \frac{1}{(2\pi)^{\frac{N}{2}}} e^{-\frac{1}{2}\left(\epsilon(\mathbf{y}^T\mathbf{y}-2\mathbf{x}^TA^T\mathbf{y}+\mathbf{x}^TA^TA\mathbf{x})+\mathbf{x}^TT\mathbf{x}\right)} d\mathbf{x} \\ &= \frac{1}{(2\pi)^{\frac{M}{2}}}\epsilon^{\frac{M}{2}}|T|^{\frac{1}{2}} \int \frac{1}{(2\pi)^{\frac{N}{2}}} e^{-\frac{1}{2}\left(\epsilon(\mathbf{y}^T\mathbf{y}-2\mathbf{x}^TA^T\mathbf{y}+\mathbf{x}^TA^TA\mathbf{x})+\mathbf{x}^TT\mathbf{x}\right)} d\mathbf{x} \end{aligned}$$

$$\begin{aligned} &= \frac{1}{(2\pi)^{\frac{M}{2}}}\epsilon^{\frac{M}{2}}|T|^{\frac{1}{2}} e^{-\frac{1}{2}\epsilon\mathbf{y}^T\mathbf{y}} \int \frac{1}{(2\pi)^{\frac{N}{2}}} e^{-\frac{1}{2}\left(\mathbf{x}^T(\epsilon A^TA+T)\mathbf{x}-2\epsilon\mathbf{x}^TA^T\mathbf{y}\right)} d\mathbf{x} \\ &= \frac{1}{(2\pi)^{\frac{M}{2}}}\epsilon^{\frac{M}{2}}|T|^{\frac{1}{2}} e^{-\frac{1}{2}\epsilon\mathbf{y}^T\mathbf{y}}|(T+\epsilon A^TA)^{-1}|^{\frac{1}{2}} \int \frac{1}{(2\pi)^{\frac{N}{2}}} \frac{1}{|(T+\epsilon A^TA)^{-1}|^{\frac{1}{2}}} e^{-\frac{1}{2}\left(\mathbf{x}^T(\epsilon A^TA+T)\mathbf{x}-2\epsilon\mathbf{x}^TA^T\mathbf{y}\right)} d\mathbf{x} \\ &= \frac{1}{(2\pi)^{\frac{M}{2}}}\epsilon^{\frac{M}{2}}|T|^{\frac{1}{2}} e^{-\frac{1}{2}\epsilon\mathbf{y}^T\mathbf{y}}|(T+\epsilon A^TA)^{-1}|^{\frac{1}{2}} \int \frac{1}{(2\pi)^{\frac{N}{2}}} \frac{1}{|(T+\epsilon A^TA)^{-1}|^{\frac{1}{2}}} \times \dots \\ &e^{-\frac{1}{2}\left(\left(\mathbf{x}-(T+\epsilon A^TA)^{-1}\epsilon A^T\mathbf{y}\right)^T(T+\epsilon A^TA)\left(\mathbf{x}-(T+\epsilon A^TA)^{-1}\epsilon A^T\mathbf{y}\right)-\epsilon^2\mathbf{y}^TA(T+\epsilon A^TA)^{-1}A^T\mathbf{y}\right)} d\mathbf{x}. \end{aligned}$$

Thus,

$$\begin{aligned} &\log p(\mathbf{y}|\epsilon,T) = \\ &-\frac{M}{2}\log\{2\pi\} + \frac{M}{2}\log\epsilon + \frac{1}{2}\log|T| - \frac{1}{2}\epsilon\mathbf{y}^T\mathbf{y} + \frac{1}{2}\log\{|(T+\epsilon A^TA)^{-1}|\} + \frac{1}{2}\epsilon^2\mathbf{y}^TA(T+\epsilon A^TA)^{-1}A^T\mathbf{y}. \end{aligned}$$

Notice that

$$-\frac{1}{2}\epsilon\mathbf{y}^T\mathbf{y} + \frac{1}{2}\epsilon^2\mathbf{y}^TA(T+\epsilon A^TA)^{-1}A^T\mathbf{y} = -\frac{1}{2}\mathbf{y}^T\epsilon(I_M - \epsilon A(T+\epsilon A^TA)^{-1}A^T)\mathbf{y}$$

and,

$$\begin{aligned} \frac{1}{2}\log|T| + \frac{1}{2}\log\{|(T+\epsilon A^TA)^{-1}|\} &= \frac{1}{2}\big(\log|T| + \log\{|(T+\epsilon A^TA)^{-1}|\}\big) \\ &= -\frac{1}{2}\log\{|T^{-1}(T+\epsilon A^TA)|\} \\ &= -\frac{1}{2}\log\{|I_N + \epsilon T^{-1}A^TA|\} \\ &= -\frac{1}{2}\log\{|I_M + \epsilon AT^{-1}A^T|\}. \end{aligned}$$

Thus,

$$\begin{aligned} L :&= \log p(\mathbf{y}|\epsilon,T) \\ &= -\frac{M}{2}\log\{2\pi\} + \frac{M}{2}\log\epsilon - \frac{1}{2}\log\{|I_M + \epsilon AT^{-1}A^T|\} - \frac{1}{2}\epsilon\mathbf{y}^T\big(I_M - \epsilon A(T+\epsilon A^TA)^{-1}A^T\big)\mathbf{y}. \end{aligned}$$

For the comparing $L^{[t]}$ with $L^{[t-1]}$ in the updating process, we have

$$L \propto \frac{M}{2}\log\epsilon + \frac{1}{2}\log\{|(I_M + \epsilon AT^{-1}A^T)^{-1}|\} - \frac{1}{2}\epsilon\mathbf{y}^T\left(I_M - \epsilon A(T+\epsilon A^T A)^{-1}A^T\right)\mathbf{y}.$$

Therefore,

$$I_M - \epsilon A(T+\epsilon A^T)^{-1}A^T = (I_M + \epsilon AT^{-1}A^T)^{-1}.$$

Thus,

$$\begin{aligned} L &\propto \frac{M}{2}\log\epsilon + \frac{1}{2}\log\{|(I_M + +\epsilon AT^{-1}A^T)^{-1}|\} - \frac{1}{2}\epsilon\mathbf{y}^T(I_M + \epsilon AT^{-1}A^T)^{-1}\mathbf{y} \\ &\propto \frac{1}{2}\log\{|\epsilon^{-1}I_M|^{-1}|I_M + \epsilon AT^{-1}A^T|^{-1}\} - \frac{1}{2}\mathbf{y}^T(\epsilon^{-1}I_M + AT^{-1}A^T)^{-1}\mathbf{y} \\ &\propto \frac{1}{2}\log\{|\epsilon^{-1}I_M(I_M + \epsilon AT^{-1}A^T)^{-1}|\} - \frac{1}{2}\mathbf{y}^T(\epsilon^{-1}I_M + AT^{-1}A^T)^{-1}\mathbf{y} \\ &\propto \log\{|\epsilon^{-1}I_M + AT^{-1}A^T|^{-1}\} - \mathbf{y}^T(\epsilon^{-1}I_M + AT^{-1}A^T)^{-1}\mathbf{y}. \end{aligned}$$

Therefore,

$$L^{[t]} \propto \log|\Sigma_0^{[t]}| - \mathbf{y}^T\Sigma_0^{[t]}\mathbf{y},$$

where

$$\Sigma_0 := (\tilde{\epsilon}^{-1}I_M + A\tilde{T}^{-1}A^T)^{-1}.$$

This means that

$$p(\mathbf{y}|\epsilon, T) = \frac{1}{(2\pi)^{\frac{M}{2}}}\frac{1}{|\Sigma_0^{-1}|^{\frac{1}{2}}}e^{\{-\frac{1}{2}\mathbf{y}^T\Sigma_0\mathbf{y}\}}$$

or equivalently,

$$p(\mathbf{y}|\epsilon, T) \sim \mathcal{N}(\mathbf{0}, \Sigma_0^{-1}).$$

Thus, the stopping criterion can be made based on

$$\begin{aligned} \Delta L^{[t]} &:= L^{[t]} - L^{[t-1]} \\ &= \log|\frac{\Sigma_0^{[t]}}{\Sigma_0^{[t-1]}}| + \mathbf{y}^T(\Sigma_0^{[t-1]} - \Sigma_0^{[t]})\mathbf{y}. \end{aligned}$$

References

1. Candes, E.J.; Romberg, J.; Tao, T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inf. Theory* **2006**, *52*, 489–509. [CrossRef]
2. Donoho, D.L. Compressed sensing. *IEEE Trans. Inf. Theory* **2006**, *52*, 1289–1306. [CrossRef]
3. Candes, E.J.; Wakin, M.B. An introduction to compressive sampling. *IEEE Signal Process. Mag.* **2008**, *25*, 21–30. [CrossRef]
4. Duarte, M.; Davenport, M.; Takhar, D.; Laska, J.; Sun, T.; Kelly, K.; Baraniuk, R. Single-pixel imaging via compressive sampling. *IEEE Signal Process. Mag.* **2008**, *25*, 83–91. [CrossRef]
5. Bajwa, W.; Haupt, J.; Sayeed, A.; Nowak, R. Compressed channel sensing: A new approach to estimating sparse multipath channels. *Proc. IEEE* **2010**, *98*, 1058–1076. [CrossRef]
6. Lustig, M.; Donoho, D.; Pauly, J. Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magn. Reson. Med.* **2007**, *58*, 1182–1195. [CrossRef]
7. Kutynoik, G. Theory and applications of compressed sensing. *GAMM-Mitteilungen* **2013**, *36*, 79–101. [CrossRef]
8. Chang, K.; Ding, P.; Li, B. Compressive sensing reconstruction of correlated images using joint regularization. *IEEE Signal Process. Lett.* **2016**, *23*, 449–453. [CrossRef]
9. Wijewardhana, U.L.; Codreanu, M.; Latva-aho, M. Bayesian method for image recovery from block compressive sensing. In Proceedings of the 2016 50th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 6–9 November 2016; pp. 379–383.
10. Qaisar, S.; Bilal, R.M.; Iqbal, W.; Naureen, M.; Lee, S. Compressive sensing: From theory to applications, a survey. *Commun. Netw. J.* **2013**, *15*, 443–456. [CrossRef]
11. Mishali, M.; Eldar, Y.C. Blind multi-band signal reconstruction: Compressed sensing for analog signals. *IEEE Trans. Signal Process.* **2009**, *57*, 993–1009. [CrossRef]

12. Mishali, M.; Eldar, Y.C. Xampling: Signal acquisition and processing in unions of subspaces. *IEEE Trans. Signal Process.* **2011**, *59*, 4719–4734. [CrossRef]
13. Cohen, D.; Mishra, K.V.; Eldar, Y.C. Spectrum sharing Radar: Coexistence via xampling. *IEEE Trans. Aerosp. Electron. Syst.* **2017**, *54*, 1279–1296. [CrossRef]
14. Aubry, A.; Carotenuto, V.; Maio, A.D.; Govoni, M.A.; Farina, A. Experimental analysis of block-sparsity-based spectrum sensing techniques for cognitive Radar. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *57*, 355–370. [CrossRef]
15. Hwang, S.; Seo, J.; Park, J.; Kim, H.; Jeong, B.J. Compressive sensing-based Radar imaging and subcarrier allocation for joint MIMO OFDM Radar and communication system. *Sensors* **2021**, *21*, 2382. [CrossRef]
16. Rani, M.; Dhok, S.B.; Deshmukh, R.B. A systematic review of compressive sensing: Concepts, implementations and applications. *IEEE Access* **2018**, *6*, 4875–4894. [CrossRef]
17. Zhan, Z.; Li, Q.; Huang, J. Application of wavefield compressive sensing in surface wave tomography. *Geophys. J. Int.* **2018**, *213*, 1731–1743. [CrossRef]
18. Da Poian, G.; Rozell, C.J.; Bernardini, R.; Rinaldo, R.; Clifford, G.D. Matched filtering for heart rate estimation on compressive sensing ECG measurements. *IEEE Trans. Biomed. Eng.* **2017**, *65*, 1349–1358. [CrossRef]
19. Djelouat, H.; Zhai, X.; Disi, M.A.; Amira, A.; Bensaali, F. System-on-chip solution for patients biometric: A compressive sensing-based approach. *IEEE Sens. J.* **2018**, *18*, 9629–9639. [CrossRef]
20. Zhang, P.; Wang, S.; Guo, K.; Wang, J. A secure data collection scheme based on compressive sensing in wireless sensor networks. *Ad Hoc Netw.* **2018**, *70*, 73–84. [CrossRef]
21. Sharma, S.K.; Chatzinotas, S.; Ottersten, B. Compressive sparsity order estimation for wideband cognitive radio receiver. *IEEE Trans. Signal Process.* **2014**, *62*, 4984–4996. [CrossRef]
22. Zhao, T.; Wang, Y. Statistical interpolation of spatially varying but sparsely measured 3D geo-data using compressive sensing and variational Bayesian inference. *Math. Geosci.* **2021**, *53*, 1171–1199. [CrossRef]
23. Han, R.; Bai, L.; Zhang, W.; Liu, J.; Choi, J.; Zhang, W. Variational inference based sparse signal detection for next generation multiple access. *IEEE J. Sel. Areas Commun.* **2022**, *40*, 1114–1127. [CrossRef]
24. Tang, V.H.; Bouzerdoum, A.; Phung, S.L. Variational Bayesian compressive multipolarization indoor Radar imaging. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 7459–7474. [CrossRef]
25. Wan, Q.; Fang, J.; Huang, Y.; Duan, H.; Li, H. A Variational Bayesian inference-inspired unrolled deep network for MIMO detection. *IEEE Trans. Signal Process.* **2022**, *70*, 423–437. [CrossRef]
26. Fang, J.; Shen, Y.; Li, H.; Wang, P. Pattern-coupled sparse Bayesian learning for recovery of block-sparse signals. *IEEE Trans. Signal Process.* **2015**, *63*, 360–372. [CrossRef]
27. Shekaramiz, M.; Moon, T.K.; Gunther, J.H. Bayesian compressive sensing of sparse signals with unknown clustering patterns. *Entropy* **2019**, *21*, 247. [CrossRef]
28. Wipf, D.P.; Rao, B.D. Sparse Bayesian learning for basis pursuit selection. *IEEE Trans. Signal Process.* **2004**, *52*, 2153–2164. [CrossRef]
29. Lv, F.; Zhang, C.; Tang, Z.; Zhang, P. Block-sparse signal recovery based on adaptive matching pursuit via spike and slab prior. In Proceedings of the 2020 IEEE 11th Sensor Array and Multichannel Signal Processing Workshop (SAM), Hangzhou, China, 8–11 June 2020; pp. 1–5.
30. Worley, B. Scalable mean-field sparse Bayesian learning. *IEEE Trans. Signal Process.* **2019**, *67*, 6314–6326. [CrossRef]
31. Chen, P.; Zhao, J.; Bai, X. Block inverse-free sparse Bayesian learning for block sparse signal recovery. In Proceedings of the 2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP), Chongqing, China, 11–13 December 2019; pp. 1–4.
32. Hilli, A.A.; Najafizadeh, L.; Petropulu, A. Weighted sparse Bayesian learning (WSBL) for basis selection in linear underdetermined systems. *IEEE Trans. Veh. Technol.* **2019**, *68*, 7353–7367. [CrossRef]
33. Wang, D.; Zhang, Z. Variational Bayesian inference based robust multiple measurement sparse signal recovery. *Digit. Signal Process.* **2019**, *89*, 131–144. [CrossRef]
34. Bayisa, F.L.; Zhou, Z.; Cronie, O.; Yu, J. Adaptive algorithm for sparse signal recovery. *Digit. Signal Process.* **2019**, *87*, 10–18. [CrossRef]
35. Nayek, R.; Fuentes, R.; Worden, K.; Cross, E.J. On spike-and-slab priors for Bayesian equation discovery of nonlinear dynamical systems via sparse linear regression. *Mech. Syst. Signal Process.* **2021**, *161*, 107986. [CrossRef]
36. Li, J.; Zhou, W.; Cheng, C. Adaptive support-driven Bayesian reweighted algorithm for sparse signal recovery. *Signal Image Video Process.* **2021**, *15*, 1295–1302. [CrossRef]
37. Zong-Long, B.; Li-Ming, S.; Jin-Wei, S. Sparse Bayesian learning using adaptive LASSO priors. *Acta Autom. Sin.* **2021**, *45*, 1–16.
38. Mallat, S.; Zhang, Z. Matching pursuits with time-frequency dictionaries. *IEEE Trans. Signal Process.* **1993**, *41*, 3397–3415. [CrossRef]
39. Blumensath, T.; Davies, M.E. Iterative hard thresholding for compressive sensing. *Appl. Comput. Harmon. Anal.* **2009**, *27*, 265–274. [CrossRef]
40. Stankovič, L.; Daković, M.; Vujović, S. Adaptive variable step algorithm for missing samples recovery in sparse signals. *IET Signal Process.* **2014**, *8*, 246–256. [CrossRef]

41. Chen, S.; Donoho, D. Basis pursuit. In Proceedings of the 1994 28th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 31 October–2 November 1994; pp. 41–44.
42. Zhou, W.; Zhang, H.T.; Wang, J. An efficient sparse Bayesian learning algorithm based on Gaussian-scale mixtures. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 3065–3078. [CrossRef]
43. Sant, A.; Leinonen, M.; Rao, B.D. General total variation regularized sparse Bayesian learning for robust block-sparse signal recovery. In Proceedings of the ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 5604–5608.
44. Liu, J.; Wu, Q.; Amin, M.G. Multi-Task Bayesian compressive sensing exploiting signal structures. *Signal Process.* **2021**, 178, 107804. [CrossRef]
45. He, L.; Chen, H.; Carin, L. Tree-structured compressive sensing with variational Bayesian analysis. *IEEE Signal Process. Lett.* **2010**, *17*, 233–236.
46. Ji, S.; Xue, Y.; Carin, L. Bayesian compressive sensing. *IEEE Trans. Signal Process.* **2008**, *56*, 2346–2356. [CrossRef]
47. Shekaramiz, M.; Moon, T.K.; Gunther, J.H. Hierarchical Bayesian approach for jointly-sparse solution of multiple-measurement vectors. In Proceedings of the 2014 48th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 2–5 November 2014; pp. 1962–1966.
48. Oikonomou, V.P.; Nikolopoulos, S.; Kompatsiaris, I. A novel compressive sensing scheme under the variational Bayesian framework. In Proceedings of the 2019 27th European Signal Processing Conference (EUSIPCO), A Coruna, Spain, 2–6 September 2019; pp. 1–5.
49. Wang, L.; Zhao, L.; Yu, L.; Wang, J.; Bi, G. Structured Bayesian learning for recovery of clustered sparse signal. *Signal Process.* **2020**, *166*, 107255. [CrossRef]
50. Yu, L.; Wei, C.; Jia, J.; Sun, H. Compressive sensing for cluster structured sparse signals: Variational Bayes approach. *IET Signal Process.* **2016**, *10*, 770–779. [CrossRef]
51. Babacan, S.D.; Nakajima, S.; Do, M.N. Bayesian group-sparse modeling and variational inference. *IEEE Trans. Signal Process.* **2014**, *62*, 2906–2921. [CrossRef]
52. Yu, L.; Sun, H.; Barbot, J.P.; Zheng, G. Bayesian compressive sensing for cluster structured sparse signals. *Signal Process.* **2012**, *92*, 259–269. [CrossRef]
53. Anderson, M.R.; Winther, O.; Hansen, L.K. Bayesian inference for structured spike and slab priors. In Proceedings of the Advances in Neural Information Processing Systems 27 (NIPS 2014), Montreal, QC, Canada, 8–13 December 2014; pp. 1745–1753.
54. Babacan, S.; Molina, R.; Katsaggelos, A. Bayesian compressive sensing using Laplace priors. *IEEE Trans. Image Process.* **2010**, *19*, 53–63. [CrossRef]
55. Hernandez-Lobato, D.; Hernandez-Lobato, J.M.; Dupont, P. Generalized spike-and-slab priors for Bayesian group feature selection using expectation propagation. *J. Mach. Learn. Res.* **2013**, *14*, 1891–1945.
56. Ji, S.; Dunson, D.; Carin, L. Multitask compressive sensing. *IEEE Trans. Signal Process.* **2009**, *57*, 92–106. [CrossRef]
57. Shekaramiz, M.; Moon, T.K.; Gunther, J.H. Sparse Bayesian learning using variational Bayes inference based on a greedy criterion. In Proceedings of the 2017 51st Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 29 October–1 November 2017; pp. 858–862.
58. Wu, Q.; Fang, S. Structured Bayesian compressive sensing with spatial location dependence via variational Bayesian inference. *Digit. Signal Process.* **2017**, *71*, 95–107. [CrossRef]
59. Wipf, D.P.; Rao, B.D. An empirical Bayesian strategy for solving the simultaneous sparse approximation problem. *IEEE Trans. Signal Process.* **2007**, *55*, 3704–3716. [CrossRef]
60. Tibshirani, R.; Saunders, M.; Rosset, S.; Zhu, J.; Knight, K. Sparsity and smoothness via the fused LASSO. *J. R. Stat. Soc. Ser. B* **2005**, *67*, 91–108. [CrossRef]
61. Blumensath, T.; Davies, M.E. Normalized iterative hard thresholding: Guaranteed stability and performance. *IEEE J. Sel. Top. Signal Process.* **2010**, *4*, 298–309. [CrossRef]
62. Qin, L.; Tan, J.; Wang, Z.; Wang, G.; Guo, X. Exploiting the tree-structured compressive sensing of Wavelet coefficients via block sparse Bayesian learning. *Electron. Lett.* **2018**, *54*, 975–976. [CrossRef]
63. Ambat, S.K.; Chatterjee, S.; Hari, K.V. Fusion of greedy pursuits for compressed sensing signal reconstruction. In Proceedings of the 2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO), Bucharest, Romania, 27–31 August 2012; pp. 1434–1438.
64. Cao, Z.; Dai, J.; Xu, W.; Chang, C. Fast variational Bayesian inference for temporally correlated sparse signal recovery. *IEEE Sigal Process. Lett.* **2021**, *28*, 214–218. [CrossRef]
65. Gelman, A.; Rubin, D.B. Inference from iterative simulation using multiple sequences. *Stat. Sci.* **1992**, *7*, 457–511. [CrossRef]
66. Beal, M. Variational Algorithms for Approximate Bayesian Inference. Ph.D. Dissertation, University College London, London, UK, 2003.
67. Tzikas, D.G.; Likas, A.C.; Galatsanos, N.P. The variational approximation for Bayesian inference. *IEEE Signal Process. Mag.* **2008**, *25*, 131–142. [CrossRef]
68. Shekaramiz, M.; Moon, T.K. Compressive sensing via variational Bayesian inference. In Proceedings of the 2020 Intermountain Engineering, Technology and Computing (IETC), Orem, UT, USA, 2–3 October 2020; pp. 1–6.

69. Shekaramiz, M.; Moon, T.K. Sparse Bayesian learning via variational Bayes fused With orthogonal matching pursuit. In Proceedings of the 2022 Intermountain Engineering, Technology and Computing (IETC), Orem, UT, USA, 13–14 May 2022; pp. 1–5.
70. You, C.; Ormerod, J.T.; Mueller, S. On variational Bayes estimation and variational information criteria for linear regression models. *Aust. N. Z. J. Stat.* **2014**, *56*, 73–87. [CrossRef]
71. Tran, M.N.; Nguyen, T.N.; Dao, V.H. A practical tutorial on variational Bayes. *arXiv* **2021**, arXiv:2103.01327.
72. Fox, C.; Roberts, S. A tutorial on variational Bayesian inference. *Artif. Intell. Rev.* **2011**, *38*, 85–95. [CrossRef]
73. Manipur, I.; Manzo, M.; Granata, I.; Giordano, M.; Maddalena, L.; Guarracino, M.R. Netpro2vec: A graph embedding framework for biomedical applications. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2021**, *19*, 729–740. [CrossRef]