DETAILS ON *O-SBL(MCMC)*: A COMPRESSIVE SENSING ALGORITHM FOR SPARSE SIGNAL RECOVERY FOR THE SMV/MMV PROBLEM USING SPARSE BAYESIAN LEARNING AND MARKOV CHAIN MONTE CARLO INFERENCE

Mohammad Shekaramiz, Todd K. Moon, and Jacob H. Gunther

Electrical and Computer Engineering Department and Information Dynamics Laboratory
Utah State University
{mohammad.shekaramiz@aggiemail.usu.edu, todd.moon@usu.edu, jake.gunther@usu.edu}

This work is supported by the Grant NASA NNX13 AD 39 A.

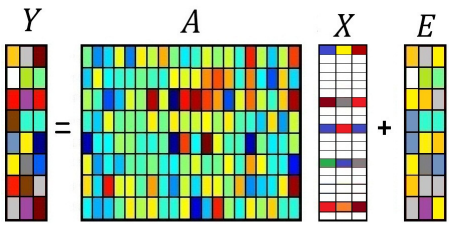

Figure 1: Example of the MMV structure.

ABSTRACT

This report provides details on O-SBL(MCMC) algorithm for the recovery of jointly-sparse signals for the multiple measurement vector (MMV) problem. For the MMVs with this structure, the solution matrix, which is a collection of sparse vectors, is expected to exhibit joint sparsity across the columns. The notion of joint sparsity here means that the columns of the solution matrix share common supports. This algorithm employs a sparse Bayesian learning (SBL) model to encourage the joint sparsity structure across the columns of the solution. While the proposed algorithm is constructed for the MMV problems, it can also be applied to the single measurement vector (SMV) problems. Part of this work has been published in [1, 2].

Keywords— Compressive sensing, Sparse Bayesian learning (SBL), single measurement vector (SMV), multiple measurement vectors (MMVs).

1. INTRODUCTION

Single- and multiple measurement vector (SMV and MMV) problems are computational inverse problems in the compressive sensing (CS) area. The idea behind compressive sensing is the possibility of representing a sparse or compressible signal from a small set of non-adaptive linear measurements [3]. In linear CS, the P-dimensional signal $\mathbf{x} \in \mathbb{R}^P$ is modeled by the linear equation $\mathbf{y} = \Phi\mathbf{x}$, where $\mathbf{y} \in \mathbb{R}^M$ is the measurement vector (with $M \ll P$) and $\Phi \in \mathbb{R}^{M\times P}$ is a wide sensing matrix. In the CS context, it is further assumed that $\mathbf{x}$ is sparse under some proper basis Ψ, i.e., $\mathbf{x} = \Psi\mathbf{x}_s$, where $\mathbf{x}_s$ denotes a sparse vector. A sparse vector contains few non-zero components. Combining the two above equations, we obtain $\mathbf{y} = A\mathbf{x}_s$, where $A = \Phi\Psi$ [4]. Since matrix A is wide, the model is underdetermined and CS instead looks for a sparse (if not the most sparse) solution $\hat{\mathbf{x}}_s$ such that $\mathbf{y} = A\hat{\mathbf{x}}_s$ [5]. The SMV is a CS problem when A is known and the measurements are contaminated with noise $\mathbf{e}$, i.e., $\mathbf{y} = A\mathbf{x}_s + \mathbf{e}$. The case where Y and X_s are matrices is called the MMV problem, i.e., $Y = AX_s + E$. In the basic MMV model, it is assumed that all the columns of the solution matrix X_s share joint sparsity, meaning that they have the same unknown non-zero locations. Fig. 1 shows an example of the MMV structure.

Bayesian learning models are able to incorporate prior knowledge on the characteristics of the underlying signal. Starting with prior knowledge, these models update their belief about the underlying features of interest in an unsupervised manner based on the observations. Regarding the sparse recovery of SMV and MMVs, existing SBL algorithms can be categorized into the two following approaches. The first and most common approach to impose sparsity on the solution is achieved by modeling each component of the solution with a zero-mean Gaussian prior accompanied with a Gamma distribution on the precision (inverse of variance) of the corresponding component [6–8]. The second SBL approach is to use a spike-and-slab prior [9–12]. These models have been applied to the SMV problem. For example, Hernandez *et al*., proposed the generalized spike-and-slab prior which is suitable for situations where prior information on the groups of components in the solution (that are expected to be jointly zero or jointly non-zero) is available [10].

2. IDEA BEHIND THE PROPOSED ALGORITHM

We present the OSBL(MCMC) algorithm to solve the MMV problem for sparse signals. Our model falls within the second sparse Bayesian modeling category mentioned earlier in Section 1, i.e., the spike-and-slab-like prior. We establish a simple hierarchical Bayesian model for solving the general form of the MMV problem. In this model, we use Bernoulli-Gaussian prior, which approximates the spike-and-slab prior, but in a sense that instead of employing spikes in the model, we have a binary vector that is to be learned to determine the supports of the sparse solution. Related algorithms can be found in [9, 13, 14]. In terms of Gaussian-Bernoulli modeling of the sparse signal, our model is close to the simplified form of [9, 14] and in terms implementation, it uses MCMC implementation with Gibbs sampler techniques as in [9, 13]. A binary matrix was used as a part of the Bayesian modeling in [13] for separating the foreground (sparse) component and the background (low-rank) component from the collection of noisy frames of a video recording. The difference between our model and [13] is that here we use a binary vector to learn the supports of the solution for the *MMV problem with the joint sparsity structure*.

3. OSBL(MCMC): A SPARSE BAYESIAN LEARNING MODELING AND ALGORITHM FOR SMV/MMVS

Here, we construct a hierarchical sparse Bayesian learning (SBL) algorithm for the sparse recovery of basic MMVs with the joint sparsity structure that is expected to occur across the columns of the solution matrix. The model can also be applied to the SMV problem by simply removing the joint sparsity structure and replacing the measurement and solution matrices with vectors. We refer to this SBL algorithm as ordinary SBL (O-SBL).

The supports of the solution are modeled by the binary vector $\mathbf{s}$. Therefore, the sparse solution is described by $\mathbf{s} \circ X$, where $\mathbf{s}$ and X account for the support and the solution-values, respectively, and $\circ$ denotes element-by-element multiplication (Hadamard product) applied across the columns of X. Therefore, the model for the MMV problem is

$$Y = A(\mathbf{s} \circ X) + E, \tag{1}$$

where $Y \in \mathbb{R}^{M\times N}$, $A \in \mathbb{R}^{M\times P}$, $\mathbf{s} \in \{0,1\}^{P\times 1}$, $X \in \mathbb{R}^{P\times N}$, and $E \in \mathbb{R}^{M\times N}$. The matrix Y contains N columns of observed noisy data, A denotes the known sensing matrix, $\mathbf{s}$ is an unknown binary support-learning vector, X is an unknown solution-values matrix, and E represents the noise in the measurements. In the product $\mathbf{s} \circ X$, when $N > 1$, the support vector $\mathbf{s}$ deals across the columns of X. In other words, the term $\mathbf{s} \circ X$ is simply equivalent to $\text{diag}\{\mathbf{s}\} \cdot X$, where '$\cdot$' is the regular matrix product and $\text{diag}\{\cdot\}$ creates a diagonal matrix from its argument vector.

A representation of a hierarchical Bayesian graphical model of the problem used in the development of our algorithm is portrayed in Fig. 2. The shaded node Y shows the observations and the small solid nodes represent the hyperparameters. Each unshaded node denotes a random variable (or a group of random variables) [15]. The support-learning component $\mathbf{s}$ in (1) is a binary vector and we model the elements of $\mathbf{s}$ as Bernoulli random variables, whose probabilities are governed by the prior $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \ldots, \gamma_P]^T$; that is,

$$s_p \sim \text{Bernoulli}(\gamma_p),\ \gamma_p \sim \text{Beta}(\alpha_0, \beta_0), p = 1, \ldots, P. \tag{2}$$

In order to favor the sparsity structure in $\mathbf{s}$, on the basis of experimentation we set $\alpha_0 = \frac{1}{P}$ and $\beta_0 = 1 - \alpha_0$, as suggested in [13].

The columns of the solution-value matrix $X = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ in (1) are assumed to be drawn *i.i.d.* according to the normal-gamma distribution

$$\mathbf{x}_n \sim \mathcal{N}(0, \tau^{-1} I_P), \tau \sim \text{Gamma}(a_0, b_0), n = 1, \ldots, N, \tag{3}$$

where a_0 and b_0 denote the shape and rate of the Gamma distribution, respectively. For the purpose of reducing the model complexity, we use the same precision τ for all the components of X. Moreover, due to the lack of prior

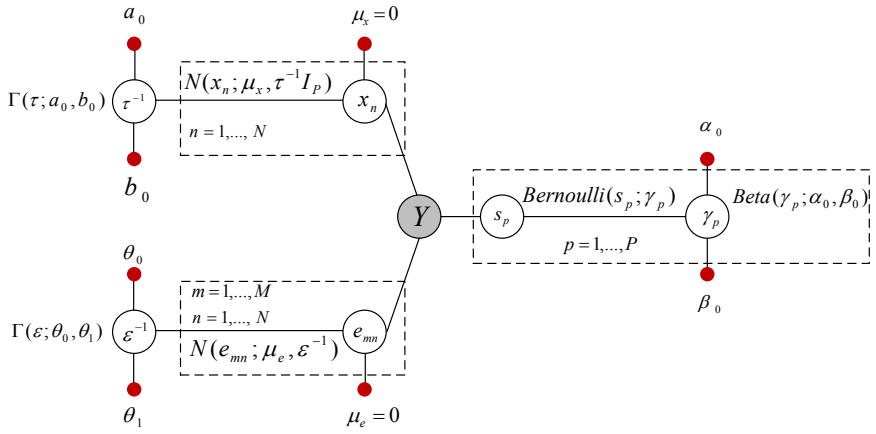

Figure 2: Graphical model of the Bayesian formulation (1).

knowledge on the entries of X, we experimentally set the hyper-parameters in (3) to $a_0 = b_0 = 10^{-3}$, endowing X *a priori* with a fairly high variance.

The entries of the noise component E are assumed to be drawn *i.i.d.* from a Gaussian distribution with the precision ε^{-1}. In our model, the precision ε^{-1} is unknown and will be learned via inference, so that

$$\begin{aligned} &e_{mn} \sim \mathcal{N}(0, \varepsilon^{-1}),\ m = 1, \ldots, M,\ n = 1, \ldots, N, \\ &\varepsilon \sim \text{Gamma}(\theta_0, \theta_1). \end{aligned} \tag{4}$$

The hyper-parameters in (4) are set to $\theta_0 = \theta_1 = 10^{-3}$. This setting may vary under the desired precision or prior knowledge on the noise variance.

Referring to the graphical model in Fig. 2, the joint probability distribution of the model can be written as

$$p(Y, \mathbf{s}, X) \propto p(Y|\mathbf{s}, X, \varepsilon)\big(\prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{0}, \tau^{-1} I_P)\big) p(\mathbf{s}|\boldsymbol{\gamma}) p(\boldsymbol{\gamma}; \alpha_0, \beta_0) p(\tau; a_0, b_0) p(\varepsilon; \theta_0, \theta_1). \tag{5}$$

The marginalized posterior distributions for the variables of interest are represented below. In these descriptions, conditioning on $-$, as in $(s_p|-)$, is used to denote the inference on s_p conditioning upon all relevant variables (including the observations).

- $(s_p|-) \sim \text{Bernoulli}(\dfrac{q_1}{q_0 + q_1})$, (6)

where $q_1 = \gamma_p e^{-\frac{\varepsilon}{2}(\|\mathbf{a}_p\|_2^2 \sum_{n=1}^{N} x_{pn}^2 - 2\mathbf{a}_p^T \sum_{n=1}^{N} x_{pn} \tilde{\mathbf{y}}_n^{-p})}$ and $q_0 = 1 - \gamma_p$.

In the above equation, $\tilde{\mathbf{y}}_n^{-p} = [\tilde{y}_{1n}^{-p}, ..., \tilde{y}_{mn}^{-p}]^T$ and the term $\tilde{y}_{mn}^{-p}$ is defined as $\tilde{y}_{mn}^{-p} = y_{mn} - \sum_{l \neq p}^{P} a_{ml} s_l x_{ln}$. For more detail, please see the Appendix.

- $(\gamma_p|-) \propto p(s_p|\gamma_p) p(\gamma_p; \alpha_0, \beta_0)$

  $\propto \gamma_p^{s_p} (1 - \gamma_p)^{1 - s_p} \gamma_p^{\alpha_0 - 1} (1 - \gamma_p)^{\beta_0 - 1}$

  Therefore, $(\gamma_p|-) \sim \text{Beta}(\alpha_0 + s_p\ ,\ \beta_0 + 1 - s_p)$.

- $(x_{pn}|-) \sim \mathcal{N}(\mu_{x_{pn}}, \Sigma_{x_{pn}})$, where $\mu_{x_{pn}} = \varepsilon s_p \Sigma_{x_{pn}} \mathbf{a}_p^T \tilde{\mathbf{y}}_n^{-p}$ and $\Sigma_{x_{pn}} = (\tau + \varepsilon s_p^2 \|\mathbf{a}_p\|_2^2)^{-1}$.

- $(\tau|-) \propto p(X|0,\tau^{-1})p(\tau;a_0,b_0)$

  $\propto \tau^{\frac{NP}{2}} e^{-\frac{\tau}{2}\|X\|_F^2} \tau^{a_0-1} e^{-b_0\tau}$

  Therefore, $(\tau|-) \sim \text{Gamma}(a_0+\frac{NP}{2}, b_0+\frac{1}{2}\|X\|_F^2)$, where $\|.\|_F$ denotes the Frobenius norm. Finally,

- $(\varepsilon|-) \propto p(Y|\mathbf{s},X,\varepsilon)p(\varepsilon;\theta_0,\theta_1)$

  $\propto \varepsilon^{\frac{MN}{2}} e^{-\frac{1}{2}\varepsilon\|Y-A(\mathbf{s}\circ X)\|_F^2} \varepsilon^{\theta_0-1} e^{-\varepsilon\theta_1}$

  Thus, $(\varepsilon|-) \sim \text{Gamma}(\theta_0+\frac{MN}{2}, \theta_1+\frac{1}{2}\|Y-A(\mathbf{s}\circ X)\|_F^2)$.

In the implementation, the above posterior densities are approximated via Markov chain Monte Carlo (MCMC) using Gibbs sampling technique. The idea behind this approach is to approximate each posterior density by a collection of samples drawn iteratively from the conditional posterior distribution of the corresponding random variable given the most recent estimated values of all the other parameters [13]. The pseudocode description the O-SBL(MCMC) algorithm is represented below.

O-SBL(MCMC) Algorithm:

$\{\Theta^{(i)}\}_{i=1}^{N_{\text{collect}}} = \textbf{O-SBL}(Y, A, \Theta_0, N_{\text{burn-in}}, N_{\text{collect}})$
For Iter $= 1$ to $N_{\text{burn-in}} + N_{\text{collect}}$
  % Support-learning vector component
  For $p = 1$ to P
    $\tilde{y}_{mn}^{-p} = y_{mn} - \sum_{l\neq p}^{P} a_{ml}s_l x_{ln},\ \forall m = 1 \text{ to } M, \forall n = 1 \text{ to } N$
    $q_0 = 1-\gamma_p$
    $q_1 = \gamma_p e^{-\frac{\varepsilon}{2}(\|\mathbf{a}_p\|_2^2 \sum_{n=1}^{N} x_{pn}^2 - 2\mathbf{a}_p^T \sum_{n=1}^{N} x_{pn}\tilde{\mathbf{y}}_n^{-p})}$
    $(s_p|-) \sim \text{Bernoulli}(\frac{q_1}{q_0+q_1})$
    % Solution-value matrix component
    For $l = 1$ to P
      $\sigma_x^2 = (\tau + \varepsilon s_l^2 \|\mathbf{a}_l\|_2^2)^{-1}$
      $\bar{\boldsymbol{\mu}} = \varepsilon s_l \sigma_x^2 \mathbf{a}_l$
      $\tilde{\mathbf{y}}_n^{-l} = \mathbf{y}_n - A(\mathbf{s}\circ\mathbf{x}_n) + s_l x_{l,n}\mathbf{a}_l,\ \forall n = 1 \text{ to } N$
      $(x_{l,n}|-) \sim \mathcal{N}(\bar{\boldsymbol{\mu}}^T\tilde{\mathbf{y}}_n^{-l}, \sigma_x^2),\ \forall n = 1 \text{ to } N$
    End For {l}
    $(\gamma_p|-) \sim \text{Beta}(\alpha_0 + s_p\ ,\ \beta_0 + 1 - s_p)$
  End For {p}
  $(\tau|-) \sim \text{Gamma}(a_0+\frac{NP}{2}, b_0+\frac{1}{2}\|X\|_F^2)$
  $(\varepsilon|-) \sim \text{Gamma}(\theta_0+\frac{MN}{2}, \theta_1+\frac{1}{2}\|Y-A(\mathbf{s}\circ X)\|_F^2)$
  $\Theta^{(\text{Iter}-N_{\text{burn-in}})} \leftarrow \Theta, \quad \forall \text{Iter} > N_{\text{burn-in}}$
End For {Iter}

In the above algorithm, the term Θ_0 contains the set of initial values of the parameters of interest. The initialization of the parameters is performed by drawing random samples from the corresponding prior distributions defined in (2) to (4). We then run the O-SBL algorithm for the number of $N_{\text{burn-in}}$ iterations. $N_{\text{burn-in}}$ is set experimentally based on the number of iterations required for reaching the stable Markov chain using potential scale reduction factor (PSRF) and multiple-PSRF (MPSRF) in [16, 17]. We do not collect any samples during the burn-in period. Once the algorithm iterates for $N_{\text{burn-in}}$ times, we perform N_{collect} more iterations to collect

the set of samples. For example, the estimate of the solution matrix X is computed from the sample mean i.e., $\tilde{X} = 1/N_{\text{collect}} \sum_{n=N_{\text{burn-in}}+1}^{N_{\text{burn-in}}+N_{\text{collect}}} \hat{X}^{[n]}$, where $\hat{X}^{[n]}$ denotes the collected samples for the solution matrix obtained from the corresponding approximated posterior distribution at the nth iteration. As an alternative, one may use the samples obtained at the last iteration of the collection period as the estimate of the variable of interest. For example, $\tilde{X} = \hat{X}^{[N_{\text{burn-in}}+N_{\text{collect}}]}$.

4. SUMMARY

The sparse recovery of signals via SMV and MMV models was considered. The O-SBL(MCMC) algorithm simultaneously learns both the supports and solution-value matrix for the MMVs with the joint sparsity structure. Although O-SBL provides encouraging results, it is computationally expensive due to the MCMC implementation. In future work we will consider the alternative approaches to MCMC implementation such as variational Bayes inference method.

5. APPENDIX

In this section we provide details on the inference and update rules of the variables of the O-SBL algorithm.

In this section, we provide the inference on the full posterior distributions for the model parameters to obtain approximations for the probability distributions of $\mathbf{s}$ and X for O-SBL algorithm. According to the graphical model of the O-SBL algorithm illustrated in Fig. 2, and the equations (1–6), the marginalized posterior distributions for the variables and parameters of the model can be obtained from the below descriptions.

- Posterior of the Support-Learning Component:

Below we describe the derivation of the inference equation for s_p provided in (6). According to the joint probability distribution (5), we have

$$p(\mathbf{s}|-) \sim p(Y|\mathbf{s}, X)p(\mathbf{s}|\gamma_p), \tag{7}$$

where

$$p(s_p|\gamma_p) \sim \text{Bernoulli}(\gamma_p), \quad \forall p = 1, 2, ..., P. \tag{8}$$

Therefore,

$$P(s_p|\gamma_p) \propto \gamma_p^{s_p}(1-\gamma_p)^{1-s_p}, \quad \forall p = 1, 2, ..., P$$

and

$$p(Y|\mathbf{s}, X, \Sigma_x) \propto \prod_{i=1}^{N} p(\mathbf{y}_i|\mathbf{s}, \mathbf{x}_i, \Sigma_x).$$

By expanding the above likelihood, we then have

$$\log\{p(Y|\mathbf{s}, X, \Sigma_x)\} \propto \varepsilon\big((y_{11} - \sum_{k=1}^{P} a_{1k}s_p x_{k1})^T(\star) + ... + (y_{m1} - \sum_{k=1}^{P} a_{mk}s_p x_{k1})^T(\star)) + ...$$

$$+((y_{1n} - \sum_{k=1}^{P} a_{1k}s_p x_{kn})^T(\star) + .. + (y_{mn} - \sum_{k=1}^{p} a_{mk}s_p x_{kn})^T(\star)\big),$$

where $(B)^T A(B)$ is denoted by $(B)^T A(\star)$. Equivalently,

$$\log\{p(Y|s_p, X, \Sigma_x)\} \propto$$

$$\varepsilon\big((a_{1k}s_p x_{k1} - (y_{11} - \sum_{l\neq k}^{p} a_{1l}s_l x_{l1}))^T(\star) + ... + (a_{mk}s_p x_{k1} - (y_{m1} - \sum_{l\neq k}^{P} a_{ml}s_l x_{l1}))^T(\star)) + ...$$

$$((a_{1k}s_px_{kn} - (y_{1n} - \sum_{l\neq k}^{P} a_{1l}s_lx_{ln}))^T(\star) + ... + (a_{mk}s_px_{kn} - (y_{mn} - \sum_{l\neq k}^{p} a_{ml}s_lx_{ln}))^T(\star)\big).$$

Let's define

$$\tilde{y}_{ji}^{-k} \stackrel{\text{def}}{=} y_{ji} - \sum_{l\neq k}^{p} a_{jl}s_lx_{li}, \;\; \forall j = 1, 2, ..., M, \;\; \forall i = 1, 2, ..., N, \;\; \forall k = 1, 2, ..., P. \tag{9}$$

Therefore,

$$\log\{p(Y|s_p, X, \Sigma_x)\} \propto$$
$$\varepsilon\big(((x_{k1}^2 \sum_{i=1}^{M} a_{ik}^2)s_p^2 - 2(x_{k1} \sum_{i=1}^{M} a_{ik}\tilde{y}_{i1}^{-k})s_p) + ... + ((x_{kn}^2 \sum_{i=1}^{m} a_{ik}^2)s_p^2 - 2(x_{kn} \sum_{i=1}^{M} a_{ik}\tilde{y}_{in}^{-k})s_p)\big),$$

Resulting in

$$\log\{p(Y|s_p, X, \Sigma_x)\} \propto \varepsilon\big((\|\mathbf{a}_k\|_2^2 \sum_{i=1}^{N} x_{ki}^2)s_p^2 - 2(x_{k1} \sum_{i=1}^{M} a_{ik}\tilde{y}_{i1}^{-k} + ... + x_{kn} \sum_{i=1}^{M} a_{ik}\tilde{y}_{in}^{-k})s_p\big). \tag{10}$$

Let's also define

$$\mathbf{a}_k^T\tilde{\mathbf{y}}_1^{-k} \stackrel{\text{def}}{=} \sum_{i=1}^{M} a_{ik}\tilde{y}_{i1}^{-k}, \tag{11}$$

where $\tilde{\mathbf{y}}_1^{-k} = [\tilde{y}_{11}^{-k}, ..., \tilde{y}_{m1}^{-k}]^T$. Substituting (11) into (10) yields

$$\log\{p(Y|s_p, X, \Sigma_x)\} \propto \varepsilon\big((\|\mathbf{a}_k\|_2^2 \sum_{i=1}^{N} x_{ki}^2)s_p^2 - 2(\mathbf{a}_k^T(\sum_{i=1}^{N} x_{ki}\tilde{\mathbf{y}}_i^{-k}))s_p\big).$$

Finally, we obtain the following results

$$p(s_p|-) \sim \text{Bernoulli}(\frac{q_1}{q_0 + q_1}), \;\; \forall p = 1, 2, ..., P, \tag{12}$$

where

$$q_0 \stackrel{\text{def}}{=} 1 - \gamma_p \tag{13}$$

and

$$q_1 \stackrel{\text{def}}{=} \gamma_p e^{-\frac{\varepsilon}{2}\left((\|\mathbf{a}_k\|_2^2 \sum_{i=1}^{N} x_{ki}^2) - 2(\mathbf{a}_k^T(\sum_{i=1}^{N} x_{ki}\tilde{\mathbf{y}}_i^{-k}))\right)}. \tag{14}$$

- Posterior of γ_p:

The posterior γ_p can be obtained from the relationship below.

$$p(\gamma_p|-) \propto p(s_p|\gamma_p)p(\gamma_p|\alpha_0, \beta_0), \;\; \forall p = 1, 2, ..., P, \tag{15}$$

where

$$p(\gamma_p|\alpha_0, \beta_0) \propto (\gamma_p)^{\alpha_0 - 1}(1 - \gamma_p)^{\beta_0 - 1}, \tag{16}$$

which yields to

$$p(\gamma_p|-) \propto (\gamma_p)^{(\alpha_0 + s_p) - 1}(1 - \gamma_p)^{(\beta_0 + 1 - s_p) - 1}. \tag{17}$$

Thus

$$p(\gamma_p|-) \sim \text{Beta}(\alpha_0 + s_p \;,\; \beta_0 + 1 - s_p), \;\; \forall p = 1, 2, ..., P. \tag{18}$$

- Posterior of the Solution-Value Matrix X:

Below the equation for finding the posterior x_{LK} is represented.

$$p(x_{LK}|-) \propto p(Y|\mathbf{s}, x_{LK}, 0, \sigma^2_{x_{LK}})p(x_{LK}|X, 0, \tau^{-1}), \tag{19}$$

where

$$p(x_{LK}|X, 0, \tau^{-1}) \propto e^{\frac{-\tau}{2}x^2_{LK}} \ \forall L = 1, 2, ..., P, \ \forall K = 1, 2, ..., N \tag{20}$$

and

$$\log\{p(Y|\mathbf{s}, x_{LK}, 0, \sigma^2_{x_{LK}})\} \propto -\frac{\varepsilon}{2}\big((y_{1k} - \sum_{l=1}^{P} a_{1l}s_l x_{lk})^T(\star) + ... + (y_{mk} - \sum_{l=1}^{p} a_{ml}s_l x_{lk})^T(\star)\big), \tag{21}$$

which

$$\begin{aligned}&\log\{p(x_{LK}|-)\} \propto \\ &-\frac{1}{2}\big((\tau + \varepsilon s^2_L \sum_{i=1}^{M} a^2_{iL})x^2_{LK} - 2(\varepsilon s_L \sum_{i=1}^{M} a_{iL}\tilde{y}^{-L}_{iK})x_{LK}\big), \forall L = 1, ..., P, \forall K = 1, ..., N.\end{aligned} \tag{22}$$

Therefore,

$$p(x_{LK}|-) \sim \mathcal{N}(\mu_{x_{LK}}, \sigma^2_{x_{LK}}), \ \forall L = 1, 2, ..., P, \ \forall K = 1, 2, ..., N, \tag{23}$$

where

$$\mu_{x_{LK}} \overset{\text{def}}{=} \varepsilon s_L \sigma^2_{x_{LK}} \mathbf{a}^T_L \tilde{\mathbf{y}}^{-L}_K$$

and

$$\sigma^2_{x_{LK}} \overset{\text{def}}{=} \big(\tau + \varepsilon(s_L \left\|\mathbf{a}_L\right\|_2)^2\big)^{-1}.$$

- Posterior of τ

$$p(\tau|-) \propto \big(\prod_{i=1}^{P} p(\mathbf{x}_i|\mathbf{0}, T^{-1})\big)p(\tau|a_0, b_0), \tag{24}$$

where $T = \tau I_P$. Also,

$$p(\tau|a_0, b_0) \propto \tau^{a_0 - 1}e^{-b_0\tau}I_{[0,\infty)}(\tau) \tag{25}$$

and

$$\begin{aligned}\prod_{i=1}^{P} p(\mathbf{x}_i|\mathbf{0}, T^{-1}) &\propto \tau^{\frac{NP}{2}} e^{-\frac{\tau}{2}\sum_{i=1}^{N}\mathbf{x}_i^T\mathbf{x}_i} \\ &\propto \tau^{\frac{NP}{2}} e^{-\frac{\tau}{2}\|X\|^2_F},\end{aligned} \tag{26}$$

where $\|.\|_F$ denotes Frobenius norm. By substituting (25) and (26) into (24), we have

$$p(\tau|-) \propto \tau^{(a_0 + \frac{NP}{2}) - 1}e^{-(b_0 + \frac{1}{2}\|X\|^2_F)\tau}I_{[0,\infty)}(\tau) \tag{27}$$

or equivalently,

$$p(\tau|-) \sim \text{Gamma}(a_0 + \frac{NP}{2}\ ,\ b_0 + \frac{1}{2}\left\|X\right\|^2_F). \tag{28}$$

- Posterior of the Noise Precision ε:

$$p(\varepsilon|-) \propto p(Y|\mathbf{s}, X)p(\varepsilon|\theta_0, \theta_0), \tag{29}$$

where

$$p(\varepsilon|\theta_0, \theta_1) \propto \varepsilon^{\theta_0 - 1} e^{-\varepsilon\theta_1}$$

and

$$p(Y|\mathbf{s}, X) \propto \prod_{i=1}^{N} p(\mathbf{y}_i|\mathbf{s}, \mathbf{x}_i, \varepsilon I_M), \tag{30}$$

where

$$p(\mathbf{y}_i|\mathbf{s}, \mathbf{x}_i) \propto \varepsilon^{\frac{M}{2}} e^{-\frac{1}{2}\varepsilon\|\mathbf{y}_i - A(\mathbf{s}\circ\mathbf{x}_i)\|_2^2}, \ \forall i = 1, 2, ..., N. \tag{31}$$

Therefore,

$$p(Y|\mathbf{s}, X) \propto \varepsilon^{\frac{MN}{2}} e^{-\frac{1}{2}\varepsilon\|Y - A(\mathbf{s}\circ X)\|_F^2}. \tag{32}$$

Finally, we can substitute (31) and (30) into (29) and reach to the result below.

$$p(\varepsilon|-) \propto \varepsilon^{(\theta_0 + \frac{MN}{2}) - 1} e^{-(\theta_1 + \frac{1}{2}\|Y - A(\mathbf{s}\circ X)\|_F^2)\varepsilon}.$$

In other words,

$$p(\varepsilon|-) \sim \text{Gamma}\big(\theta_0 + \frac{MN}{2}\ ,\ \theta_1 + \frac{1}{2}\|Y - A(\mathbf{s}\circ X)\|_F^2\big). \tag{33}$$

6. REFERENCES

[1] M. Shekaramiz, T. Moon, and J. Gunther, "Hierarchical Bayesian approach for jointly-sparse solution of multiple-measurement vectors," in *48th Asilomar Conf. on Sig., Syst. and Compt.*, pp. 1962–1966, Nov. 2014.

[2] M. Shekaramiz, *Sparse Signal Recovery Based on Compressive Sensing and Exploration Using Multiple Mobile Sensors*. PhD Dissertation, Utah State University, Digitalcommons, 2018.

[3] D. Donoho, "Compressed sensing," *IEEE Trans. Info. Th.*, vol. 52, pp. 1289–1306, 2006.

[4] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Sig. Proc. Mag.*, vol. 25, no. 2, pp. 21–30, 2008.

[5] M. Elad, *Sparse and Redundant Representation: From Theory to Applications in Signal and Image Processing*. Springer, 2010.

[6] D. Wipf and B. Rao, "Sparse Bayesian learning for basis pursuit selection," *IEEE Trans. Sig. Proc.*, vol. 52, no. 8, pp. 2153–2164, 2004.

[7] Y. Qi, D. Liu, L. Carin, and D. Dunson, "Multi-task compressive sensing with Dirichlet process priors," in *Int. Conf. on Mach. Learn. (ICML)*, pp. 768–775, 2008.

[8] S. Ji, D. Dunson, and L. Carin, "Multitask compressive sensing," *IEEE Trans. Sig. Proc.*, vol. 57, no. 1, pp. 92–106, 2009.

[9] L. Yu, H. Sun, J. P. Barbot, and G. Zheng, "Bayesian compressive sensing for cluster structured sparse signals," *Signal Process.*, vol. 92, no. 1, pp. 259–269, 2012.

[10] D. Hernandez-Lobato, J. M. Hernandez-Lobato, and P. Dupont, "Generalized spike-and-slab priors for Bayesian group feature selection using expectation propagation," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 1891–1945, 2013.

[11] M. R. Anderson, O. Winther, and L. K. Hansen, "Bayesian inference for structured spike and slab priors," in *Adv. Neural Inf. Process. Syst.*, pp. 1745–1753, 2014.

[12] X. Meng, S. Wu, L. Kuang, D. Huang, and J. Lu, "AMP-NNSPL." `arXiv:1601.00543[cs.IT]/`, 2016.

[13] X. Ding, L. He, and L. Carin, "Bayesian robust principal component analysis," *IEEE Trans. Image Proc.*, vol. 20, no. 12, pp. 3419–3430, 2011.

[14] J. Ziniel and P. Schniter, "Dynamic compressive sensing of time-varying signals via approximate message passing," *IEEE Trans. Sig. Proc.*, vol. 61, no. 21, pp. 5270–5284, 2013.

[15] C. M. Bishop, *Pattern Recognition And Machine Learning*. Springer, 2009.

[16] A. Gelman and D. B. Rubin, "Inference from iterative simulation using multiple sequences," *Statistical Sci.*, vol. 7, no. 4, pp. 457–511, 1992.

[17] S. P. Brooks and A. Gelman, "General methods for monitoring convergence of iterative simulations," *J. Comput. Graphical Stat.*, vol. 7, no. 4, pp. 434–455, 1998.