DETAILS ON *AMP-B-SBL*: AN ALGORITHM FOR RECOVERY OF CLUSTERED SPARSE SIGNALS USING APPROXIMATE MESSAGE PASSING [1–3]

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.

ABSTRACT

Solving the inverse problem of compressive sensing in the context of single measurement vector (SMV) problem with an unknown block-sparsity structure is considered. For this purpose, we propose a sparse Bayesian learning (SBL) algorithm simplified via the approximate message passing (AMP) framework. In order to encourage the block-sparsity structure, we incorporate the concept of total variation, called Sigma-Delta, as a measure of block-sparsity on the support set of the solution. The AMP framework reduces the computational load of the proposed SBL algorithm and as a result makes it faster compared to the message passing framework. Furthermore, in terms of the mean-squared error between the true and the reconstructed solution, the algorithm demonstrates an encouraging improvement compared to the other algorithms.

Keywords— Compressed sensing (CS), sparse Bayesian learning (SBL), approximate message passing (AMP), clustered sparsity, single measurement vector (SMV).

1. INTRODUCTION

The SMV problem is a set of linear noisy measurements from a sparse signal $\mathbf{x}$ and is modeled as $\mathbf{y} = A\mathbf{x} + \mathbf{e}$, where $\mathbf{x} \in \mathbb{R}^N$ is the signal of interest to be reconstructed and $\mathbf{e}$ denotes the noise. In this model, $A \in \mathbb{R}^{M \times N}$ is a known wide sensing matrix with $M \ll N$. In some practical applications the non-zero entries of the sparse signal $\mathbf{x}$ appear in clusters, so they more or less clump together. This feature has been referred to as block-sparsity or clustered pattern in the literature [4–9]. One example is magnetoencephalography, which seeks the locations where most brain activities are produced. Such activities exhibit contiguity i.e., they occur in localized regions [6].

For the purpose of promoting the recovery performance of the SMV problem with an unknown block-sparsity framework, Zhang and Rao proposed a sparse Bayesian learning (SBL) algorithm that incorporates intra-block correlation (correlation structure in each block) [5]. In this model, they defined a Gaussian-distributed prior with zero-mean and the covariance that depends on the multiplication of a nonnegative scaling parameter followed by a covariance matrix for each block. Such blocks are assumed to be uncorrelated with each other. In order to simplify the model, reduce the complexity, and suppress the overfitting of the parameters in the model, they further considered the same underlying covariance matrix with different scaling parameters for the blocks as a prior. This matrix is updated via the expectation-maximization (EM) algorithm. In [8], a hierarchical Bayesian approach was proposed to deal with the block-sparse multiple measurement vectors (MMVs) problem. In this case, a prior is incorporated to encourage both contiguity and sparsity in the solution. This prior is based on the total variation of the support of the solution, referred to as Sigma-Delta ($\Sigma\Delta$). The Sigma-Delta parameter is defined as follows

$$(\Sigma\Delta)_{\mathbf{s}} = \sum_{n=2}^{N} |s_n - s_{n-1}|, \tag{1}$$

where $\mathbf{s}$ is the support learning vector of the solution and has binary values with '1' denoting the active entry of $\mathbf{x}$. The prior on $\mathbf{s}$ depends on the term $e^{-\alpha(\Sigma\Delta)_{\mathbf{s}}}$ where $\alpha > 0$. The larger the weight α is, the more clumpy the solution becomes. Empirical receiver operating curves (ROC) indicate that the performance of the algorithm was satisfactory and encouraging. However, the runtime of the algorithm was not fast. In [9], we proposed the SDsRandOMP algorithm which is essentially a sparse version of RandOMP [10] and incorporates the Sigma-Delta parameter to encourage the block-sparsity. This algorithm is more or less greedy-based in which the behavior of Sigma-Delta is modeled by a Gamma distribution with $(\Sigma\Delta)_{\mathbf{s}} \sim \Gamma(1, \theta)$. Although this algorithm is faster than the SBLs, it requires more information i.e., noise variance.

Recently, research in this area has turned to reducing the computational complexity of their algorithms while preserving the success in the support recovery at a high rate. Such work includes using approximate message passing (AMP) and the expectation-maximization (EM) for the hyperparameters of the SBL algorithms [11–13].

The AMP is a simplified version of the message passing where the number of messages to be propagated is reduced based on Taylor series approximation, averaging over the messages passed to the same node, and the central limit theorem [11, 12]. In this case, Al-Shoukairi and Rao incorporated the AMP to their earlier SBL model for both the SMV and MMV problem to reduce the runtime of their algorithms [11]. They avoided imposing a Bernoulli-Gaussian prior on the solution vector and instead simply used a zero-mean Gaussian prior. This simplification causes all the distributions on the factor graph to become Gaussian and results in fewer computations when using message passing.

In this work, we propose a new SBL algorithm for the block-sparse SMV problem based on the AMP-SBL algorithm proposed in [11] . For this purpose, we use our measure of clumpiness over the supports of the solution (Sigma-Delta) proposed in [8, 9] to the AMP-SBL. We refer to our new algorithm as AMP-B-SBL where "-B-" denotes the block-sparsity. To demonstrate the performance of our algorithm, we compare the proposed algorithm with two other algorithms in terms of the normalized mean-squared error between the true and the reconstructed solution. Furthermore, to show the of behavior of the algorithm in the signal reconstruction, a random block-sparse case scenario for the SMV has been made. Since it benefits from the AMP, this algorithm turns out to be faster than our previously proposed SBLs [8, 9].

2. AMP-B-SBL FOR SOLVING SMV

In order to solve for the SMV problem with an unknown block-sparsity structure, we combine our measure of clumpiness of the solution (Sigma-Delta) proposed in [8, 9] with the AMP-SBL algorithm introduced in [11]. Here, we define the Sigma-Delta as

$$(\Sigma\Delta)_{(\text{support of } \mathbf{x})} = \sum_{n=2}^{N} |b(x_n, T) - b(x_{n-1}, T)|, \tag{2}$$

for some small threshold T. The function $b(.,.)$ in (2) returns a binary value and is defined as follows

$$b(x_n, T) = \begin{cases} 1 & \text{if } |x_n| > T \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

In our previous SBL model [8] we did not have such soft thresholding (3), and regardless of having a very small or large value on x_n, the corresponding support s_n became active ($s_n = 1$). In contrast, here we simply discard such small values by setting the corresponding $s_n = 0$ under the assumption that the corresponding component x_n does not have considerable contribution in the solution. For not discarding the important portions of the signal of interest, the threshold T is set to a small value.

Based on the assumption that the solution vector $\mathbf{x}$ is sparse, we consider an i.i.d. zero-mean Gaussian prior on the components of $\mathbf{x}$. The variance on x_n is defined as α_n which accounts for learning the block-sparsity structure in the solution. These distributions are defined below.

$$\begin{aligned} &\forall n = 1, \ldots, N, \quad x_n \sim \mathcal{N}(0, \alpha_n) \\ &\alpha_n \sim \mathcal{N}\big(e^{\left\{\frac{(\Sigma\Delta)|_{b(x_n,.)=0} - (\Sigma\Delta)|_{b(x_n,.)=1} - 1}{\theta_1}\right\}}, \theta_2\big), \end{aligned} \tag{4}$$

where $(\Sigma\Delta)|_{b(x_n,.)=1}$ is the Sigma-Delta evaluation of the supports of $\mathbf{x}$ for the case where s_n is forced to be active, θ_1 is the emphasizing parameter on the measure of clumpiness, and θ_2 is the prior variance on the variance of the variable x_n. The rationale behind assuming this prior on α_n is described in Tab. 1, in which *cte* denotes a constant value. As an example of Tab. 1, consider the case where forcing either $s_n = 0$ or $s_n = 1$ does not make any change in the evaluation of Sigma-Delta. In this case, though it promotes the clumpiness in the solution, it discourages the solution to be sparse in the iterative learning process. Therefore, α_n needs to be decreased.

Table 1: Behavior of α_n with respect to Sigma-Delta

$(\Sigma\Delta)\|_{b(x_n,.)=1}$	$(\Sigma\Delta)\|_{b(x_n,.)=0}$	α_n
cte	*cte*	$\downarrow$
$\uparrow$	*cte*	$\downarrow$
cte	$\downarrow$	$\downarrow$
cte	$\uparrow$	$\uparrow$
$\downarrow$	*cte*	$\uparrow$

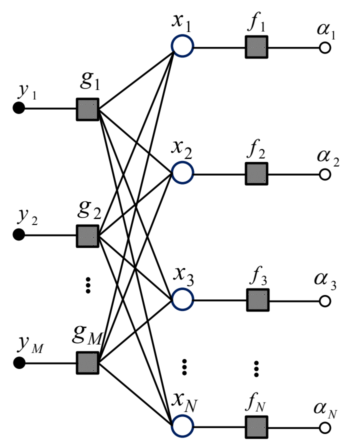

Figure 1: Factor graph for B-SMV.

According to the prior distributions defined in (4), the joint probability distribution of our model becomes

$$p(\mathbf{y}, \mathbf{x}, \boldsymbol{\alpha}, \theta_1, \theta_2, \sigma^2) \propto p(\mathbf{y}|\mathbf{x}, \sigma^2 I_N) \prod_{n=1}^{N} \Big(p(x_n; 0, \alpha_n) p(\alpha_n; e^{\left\{\frac{(\Sigma\Delta)|_{b(x_n,.)=0} - (\Sigma\Delta)|_{b(x_n,.)=1} - 1}{\theta_1}, \right\}}, \theta_2) \Big), \tag{5}$$

where the measurement noise is assumed to be $\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \sigma^2 I_M)$. In this model, all the distributions of the joint, conditional, and posterior densities become Gaussian. Therefore, we only need to pass the mean and variance of the messages rather than computing and propagating the full actual messages. The factor graph of the model for the block-sparse SMV (B-SMV) problem is illustrated in Fig. 1. In the graphical model Fig. 1, the large circle nodes represent the random variables of interest, the shaded squares show the function nodes, the small shaded circles are the observations, and the small unshaded circles denote the hyperparameters [14]. Since this main layer of our algorithm is the same as the one introduced in [11, 13], here we use the same notations as [11]. Function nodes shown in Fig. 1 are defined as follows

$$\begin{aligned} g_m &:= p(y_m|\mathbf{x}, \boldsymbol{\alpha}), \quad m = 1, 2, \ldots, M \\ f_n &:= p(x_n; 0, \alpha_n), \quad n = 1, 2, \ldots, N. \end{aligned} \tag{6}$$

Notice that the proposed algorithm adds an additional layer to [11] by incorporating the prior (4) in order to encourage the block-sparsity. This change in the model does not appear in the factor graph Fig. 1. Below, we describe the reasoning behind the proposed model for the B-SMV problem. As a prior knowledge, it is expected that the solution vector $\mathbf{x}$ to be sparse. Therefore, in order to encourage the sparsity, we assume a zero-mean Gaussian distribution for $\mathbf{x}$ with the variance α_n, $n = 1, 2, ..., N$ on its components. The supports of the solution are then specified by the binary function $b(., T)$ where T is a predetermined threshold. In other words, based on the threshold we discard the small-valued components of $\mathbf{x}$ from being considered as the support of the solution. The smaller

α_n is, the higher probability it provides to x_n to become zero. Based on the previous results in [8, 9], we observed that having more clumpy supports in the solution causes higher value of α and vice versa. Therefore, here we made the mean of the Gaussian prior on α_n be proportional to the term $\exp\left\{(\Sigma\Delta)|_{b(x_n,T)=0} - (\Sigma\Delta)|_{b(x_n,.)=1}\right\}$.

Below, we represent the message that propagates from each function node to a variable node and vice versa in Fig. 1. The derived messages in (7)-(9) become similar to those derived in [11, 15].

- Message from a function node to a variable node:

$$M_{g_m \to x_n} \propto \mathcal{N}(a_{mn}x_n; z_{mn}, c_{mn}), \tag{7}$$

where

$$z_{mn} = y_m - \sum_{q \neq n} a_{mq}\mu_q \quad \text{and} \quad c_{mn} = \sigma^2 + \sum_{q \neq n} |a_{mq}|^2 \Sigma_q.$$

- Message from a variable node to a function node:

$$M_{x_n \to g_m} \propto \mathcal{N}\big(x_n; \sum_{l \neq m} \frac{a_{nl} z_{ln} \alpha_n}{c_n + \alpha_n \sum_{l \neq m} a_{ln}^2}, \frac{c_n \alpha_n}{c_n + \alpha_n \sum_{l \neq m} a_{ln}^2}\big),$$

where c_{nl} (under the large-system-limit) is approximated by [15]

$$c_{nl} \simeq c_n := \frac{1}{M} \sum_{m=1}^{M} c_{mn}.$$

Using the fact that we have already normalized the sensing matrix A with respect to its columns, we can further approximate

$$\sum_{l \neq m} a_{ln}^2 \simeq \sum_{m=1}^{M} a_{mn}^2 = 1.$$

Therefore,

$$M_{x_n \to g_m} \propto \mathcal{N}\big(x_n; \sum_{l \neq m} a_{nl} z_{ln} \frac{\alpha_n}{c_n + \alpha_n}, \frac{c_n \alpha_n}{c_n + \alpha_n}\big). \tag{8}$$

- Estimating the posterior distribution on x_n:

$$p(x_n|\mathbf{y}) \propto p(x_n; \alpha_n) \prod_{m=1}^{M} p(y_m|x_n) \propto M_{f_n \to x_n} \prod_{m=1}^{M} M_{g_m \to x_n}.$$

After simplification, we obtain [11]

$$p(x_n|\mathbf{y}) \propto \mathcal{N}(x_n; \mu_n, \nu_n),$$

where

$$\mu_n = \sum_{m=1}^{M} a_{mn} z_{mn} \left(\frac{\alpha_n}{c_n + \alpha_n}\right), \quad \text{and} \quad \nu_n = \frac{c_n \alpha_n}{c_n + \alpha_n}. \tag{9}$$

Below, we provide the update rules for the parameters of our algorithm using the expectation-maximization (EM) technique. As was mentioned earlier, since we add a layer to the AMP-SBL algorithm to account for the block-sparsity structure, the update rules become different from [11]. Notice that as a prior we consider $\mathbf{x}$ is $x_n \sim \mathcal{N}(0, \alpha_n), \forall n = 1, \ldots, N$. Let us define

$$(\Sigma\Delta)_{n,k} := (\Sigma\Delta)|_{b(x_n,.)=k}, \forall k = 0, 1. \tag{10}$$

- Update rule for α_n:

$$\begin{aligned}
{\alpha_n}^{[k+1]} =& \arg\min_{\alpha_n} \log(\alpha_n) + \frac{1}{\alpha_n}\mathbb{E}_{\mathbf{x}|{\alpha_n}^{[k]},-}[x_n^2] + \frac{1}{\theta_2}\big(\alpha_n - e^{\{\frac{(\Sigma\Delta)_{n,0}-(\Sigma\Delta)_{n,1}-1}{\theta_1}\}}\big)^2 \\
=& \arg\min_{\alpha_n} \log(\alpha_n) + \frac{\mu_n^2+\nu_n}{\alpha_n} + \frac{1}{\theta_2}\big(\alpha_n - e^{\{\frac{(\Sigma\Delta)_{n,0}-(\Sigma\Delta)_{n,1}-1}{\theta_1}\}}\big)^2,
\end{aligned} \tag{11}$$

resulting to solve for α_n in

$$\alpha_n^3 - e^{\left\{\frac{(\Sigma\Delta)_{n,0}-(\Sigma\Delta)_{n,0}-1}{\theta_1}\right\}}\alpha_n^2 + \frac{\theta_2}{2}\alpha_n - \frac{\theta_2}{2}(\mu_n^2+\nu_n) = 0, \tag{12}$$

where μ_n and ν_n were defined in (9). A solution among all the three possible roots for (12) which minimizes (11) is the update rule of α_n for the next iteration.

- Update rule for σ^2:

$$\begin{aligned}
\sigma^{2[k+1]} &= \arg\max_{\sigma^2} \mathbb{E}_{\mathbf{x}|\mathbf{y},\theta_1,\theta_2,\boldsymbol{\alpha},\sigma^{2[k]}}\left[p(\mathbf{y},\mathbf{x},\boldsymbol{\alpha},\theta_1,\theta_2,\sigma^2)\right] \\
&= \arg\min_{\sigma^2} 2M\log\sigma + \frac{1}{\sigma^2}\mathbb{E}_{\mathbf{x}|\mathbf{y},-}\left[\|\mathbf{y}-A\mathbf{x}\|_2^2\right]
\end{aligned}$$

Therefore,

$$\begin{aligned}
\sigma^{2[k+1]} &= \frac{\|\mathbf{y}-A\boldsymbol{\mu}_x\|_2^2 + \mathrm{tr}(\Sigma_x A^T A)}{M} \\
&= \frac{\|\mathbf{y}-A\boldsymbol{\mu}_x\|_2^2 + \sum_{n=1}^N \|\mathbf{a}_n\|_2^2\Sigma_x}{M},
\end{aligned} \tag{13}$$

where $\boldsymbol{\mu}_{x-} := [\mu_1,\ldots,\mu_N]^T$ and $\Sigma_x := \mathrm{diag}\{\nu_1,\ldots,\nu_N\}$.

- Update rule for θ_2:

$${\theta_2}^{[k+1]} = \arg\min_{\theta_2} \mathbb{E}_{\mathbf{x}|\mathbf{y},{\theta_2}^{[k]},-}\big[N\log(\theta_2) + \frac{1}{\theta_2}\sum_{n=1}^N(\alpha_n - e^{\{(\Sigma\Delta)_{n,0}-(\Sigma\Delta)_{n,1}-1\}})^2\big].$$

Therefore,

$${\theta_2}^{[k+1]} = \frac{\sum_{n=1}^N(\alpha_n - e^{\{(\Sigma\Delta)_{n,0}-(\Sigma\Delta)_{n,1}-1\}})^2}{N}. \tag{14}$$

Finally, based on the AMP algorithm in [11], the pseudo code of AMP-B-SBL is described as follows. The stopping condition of the algorithm can be based on a predetermined number of iterations or convergence of the solution to a tight bound.

AMP-B-SBL Algorithm for the block-sparse SMV problem:

- **Definitions**

$F_n(k_n, \alpha_n, c) = k_n \frac{\alpha_n}{c+\alpha_n}$
$G_n(\alpha_n, c) = \frac{c.\alpha_n}{c+\alpha_n}$
$F'_n(\alpha_n, c) = \frac{\alpha_n}{c+\alpha_n}$

- **Message updates**

For $n = 1, 2, \ldots, N$
$k_n = \sum_{m=1}^{m} a^{\star}_{mn} z_m + \mu_n$
$\mu_n = F_n(k_n, \alpha_n, c)$
$\nu_n = G_n(\alpha_n, c)$
End
$c = \sigma^2 + \frac{1}{M} \sum_{n=1}^{N} \nu_n$
$z_m = y_m - \sum_{n=1}^{N} a_{mn}\mu_n + \frac{z_m}{M} \sum_{n=1}^{N} F'_n(\alpha_n, c), \forall m = 1, \ldots, M$

- **Parameter updates**

% Updating $\boldsymbol{\alpha}$:
$\forall n = 1, 2, \ldots, N$, solve for α_n in

$$\alpha_n^3 - e^{\left\{\frac{(\Sigma\Delta)_{n,0} - (\Sigma\Delta)_{n,1} - 1}{\theta_1}\right\}} \alpha_n^2 + \frac{\theta_2}{2}\alpha_n - \frac{\theta_2(\mu_n^2 + \nu_n)}{2} = 0$$

which is the minimizer of

$$f(\alpha_n) = \log(\alpha_n) + \frac{\mu_n^2 + \nu_n}{\alpha_n} + \frac{1}{\theta_2}\left(\alpha_n - e^{\left\{\frac{(\Sigma\Delta)_{n,0} - (\Sigma\Delta)_{n,1} - 1}{\theta_1}\right\}}\right)^2$$

% Updating the noise variance σ^2:
$\sigma^{2[k+1]} = \frac{\|\mathbf{y} - A\boldsymbol{\mu}_x\|_2^2 + \sum_{n=1}^{N} \|\mathbf{a}_n\|_2^2 \nu_n}{M}$
% Updating the variance of $\boldsymbol{\alpha}$:
$\theta_2{}^{[k+1]} = \frac{1}{N} \sum_{n=1}^{N} \left(\alpha_n - e^{\left\{\frac{(\Sigma\Delta)_{n,0} - (\Sigma\Delta)_{n,1} - 1}{\theta_1}\right\}}\right)^2$

In the above algorithm, the hyperparameter θ_1 is a tuning parameter.

3. SIMULATION RESULTS

This section contains the simulation results on both synthetic and real-data for the proposed algorithm against some of the existing algorithms in this area.

3.1. Simulations on the Synthetically Generated Data

Here, we demonstrate the performance of the proposed algorithm compared to two versions the SBL algorithm borrowed from [16] and the approximate message passing SBL (AMP-SBL) [11]. For the simulation purposes, the elements of vector $\mathbf{x}_{np}$ are drawn *i.i.d.* from Gaussian distribution with zero-mean and variance $\sigma_x^2 = 1$, where $\mathbf{x}_{np}$ denotes a non-sparse solution vector. The supports of the solution are binary and randomly drawn from a

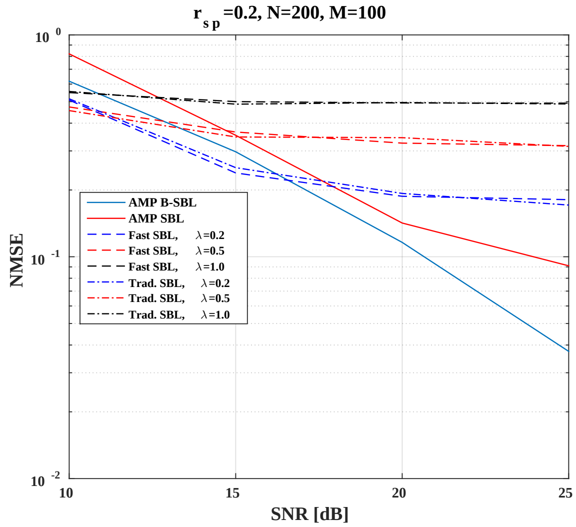

Figure 2: Performance when varying the SNR.

Bernoulli distribution in such a way to exhibit a random block-sparsity structure. The true block-sparse solution is then constructed from $\mathbf{x} = \mathbf{s} \circ \mathbf{x}_{np}$, where "$\circ$" denotes Hadamard product. The sensing matrix A is 100×200 and is randomly drawn from $\mathcal{N}(0, 1)$ and then it is normalized with respect to its columns. The elements of the noise component are drawn from $e_{mn} \sim \mathcal{N}(0, \sigma_n^2)$. We vary σ_n^2 to illustrate the performance over different SNRs. The measurement vector $\mathbf{y}$ is 100×1 and is computed from $\mathbf{y} = A\mathbf{x} + \mathbf{e}$. We evaluate the performance of the proposed algorithm using the normalized mean-squared error defined as

$$NMSE := \mathbb{E}\|\hat{\mathbf{x}} - \mathbf{x}\|_2^2 / \mathbb{E}\|\mathbf{x}\|_2^2,$$

where $\hat{\mathbf{x}}$ and $\mathbf{x}$ denote the estimated solution and the true block-sparse solution, respectively. In Fig. 2 we show the simulation results based on 100 randomly generated SMVs with the aforementioned features and then averaging over the obtained results. All of the compared algorithms in Fig. 2 are set to 1000 iterations. In this figure, we set $\sigma_x^2 = 1$ and change the noise variance σ_n^2 for evaluating the performance at different SNRs. In the legend of Fig. 2, "AMP-B-SBL" and "AMP-SBL" denote our algorithm and the algorithm proposed in [11], respectively. Also, "Trad.-SBL" and "Fast-SBL" are the traditional SBL and the fast EM-based version of the traditional SBL, respectively [16]. The term λ is the emphasis parameter on the sparsity over the error term, and r_{sp} is defined as the sparsity level divided by the length of the original signal i.e., $r_{sp} = K_{sp}/N$. We chose K_{sp} in such a way that on the overage $r_{sp} \simeq 0.2$. It can be seen from Fig. 2 that the normalized mean-squared error for both Trad.-SBL and Fast-SBL are the same for the same SNR values and almost remain constant as the SNR varies. Our algorithm shows less NMSE compared to those algorithms and also compared to the AMP-SBL. The reason that our algorithm works better than the AMP-SBL is due to the fact that the AMP-B-SBL accounts for the unknown block-sparsity in the solution.

Below, we also apply AMP-B-SBL to a specific example. In this case, we randomly generated a block-sparse signal and the corresponding set of measurements as described earlier. In our case, the true supports of the solution (non-zeros of $\mathbf{s}$) are $\{28 : 30, 89 : 95, 106, 117, 118, 128 : 131, 135, 146 : 151, 156 : 162, 172, 190, 191, 200\}$ with $\{28 : 30\}$ denoting the 28th through 30th entries of $\mathbf{s}$. For the initialization of the parameters, we set $\theta_1 = 100$, $c^{[0]} = 5$, $\theta_2^{[0]} = 0.002$, and the threshold $T = 0.001$. Fig. 3 and Fig. 4 illustrate the obtained $\alpha_n, \forall n = 1, 2, \ldots, N$ after 1000 iterations for the SNR=10 [dB] and 25 [dB], respectively.

In Fig. 5 and Fig. 6 we demonstrate the comparison between the true and the estimated solution of AMP-B-SBL algorithm for our case scenario. It can be seen from Fig. 5 that for low SNR, our algorithm almost successfully

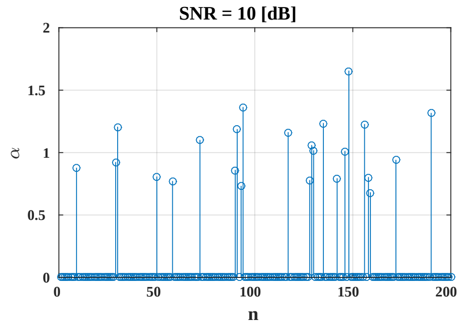

Figure 3: Estimated $\boldsymbol{\alpha}$ for SNR=10[dB] using AMP-B-SBL.

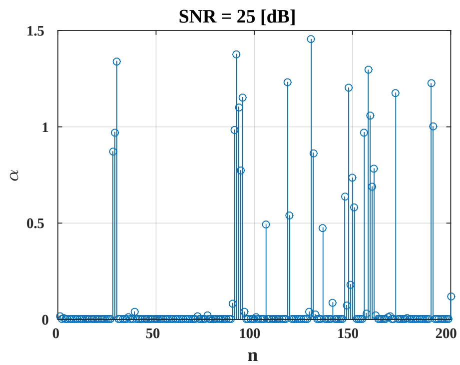

Figure 4: Estimated $\boldsymbol{\alpha}$ for SNR=25[dB] using AMP-B-SBL.

finds the active entries of the solution. There are also some false supports found by the algorithm and those caused the estimated amplitudes of the solution differ from the true ones. According to Fig. 6, the performance of the algorithm for SNR=25[dB] is satisfactory in terms of both the support detection and the non-zero amplitudes of the true solution. Based on Fig. 6, although the AMP-B-SBL also found some extra supports which are not in the true solution, such false supports clumped together and have very small amplitudes. Such clumpiness is due to the fact that the AMP-B-SBL encourages the Sigma-Delta measure of the solution to be small. Also, in Fig. 7 and Fig. 8 we show the support recovery of our example as it is being updated for the SNRs of 10 [dB] and 25 [dB], respectively.

We further demonstrate the performance of AMP-B-SBL algorithm compared to orthogonal matching pursuit (OMP) [17], basis pursuit denoising (BPDN) [18], MFOCUSS [19], and AMP-SBL [11] algorithms for solving the SMV problem. In all the simulations, we used the default settings for MFOCUSS algorithm. For the OMP, the stopping condition is set to $\sqrt{0.5N\sigma^2}$. In AMP-B-SBL, we set $c^{[10]} = 10$, $\alpha_n = 2 \times 10^{-4}, \forall n = 1, \ldots, N$, $T = 0.001$, $\theta_1 = 8$, $\theta_2^{[0]} = 0.6$, and the number of iterations is set to 1000. For the AMP-SBL algorithm, we set $c^{[0]} = 10$, $\gamma_n = 2 \times 10^{-4}, \forall n = 1, \ldots, N$, and the number of iterations is set to 1000. Finally, the stopping criterion of $\|\mathbf{y} - A\hat{\mathbf{x}}\|_2 < 0.75$ was used for the BPDN algorithm.

In the first set of experiments, we generate 200 independent trials and then average over the obtained results. In this case, we evaluate the performance of the algorithms via detection and false alarm rate in support estimation, and also the normalized mean-squared error (NMSE) between the true and the estimation solutions.

In Fig. 3.9, the empirical results of detection rate *vs.* the ratio $\lambda = M/N$ are illustrated. In these simulations, we discarded those estimated supports which their corresponding amplitudes become less than 0.01. In Fig. 3.9 we see that AMP-B-SBL shows the highest detection rate for $\lambda \in [0.05, 0.2]$. For $0.25 \leq \lambda \leq 0.55$ AMP-B-SBL, AMP-SBL, BPDN, and MFOCUSS demonstrate almost the same detection rate. Finally, for $\lambda \geq 0.55$ AMP-B-SBL,

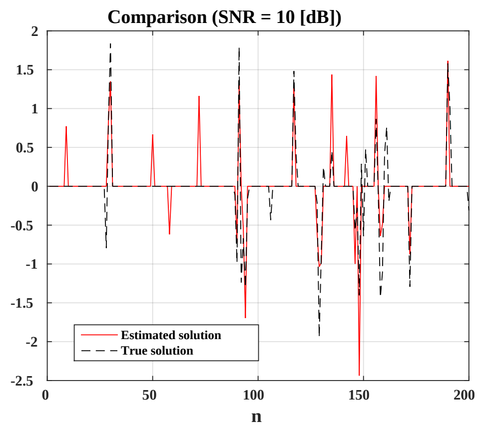

Figure 5: Comparison of $\mathbf{x}$ with $\hat{\mathbf{x}}$ for SNR=10[dB].

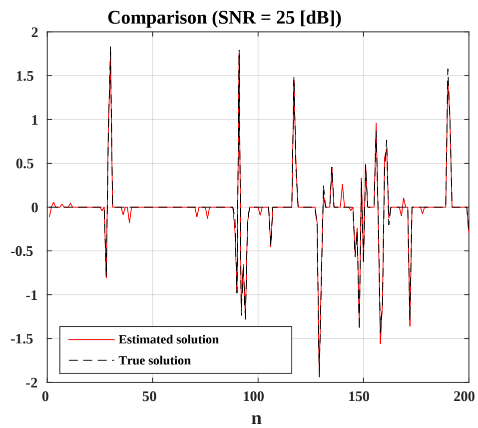

Figure 6: Comparison of $\mathbf{x}$ with $\hat{\mathbf{x}}$ for SNR=25[dB].

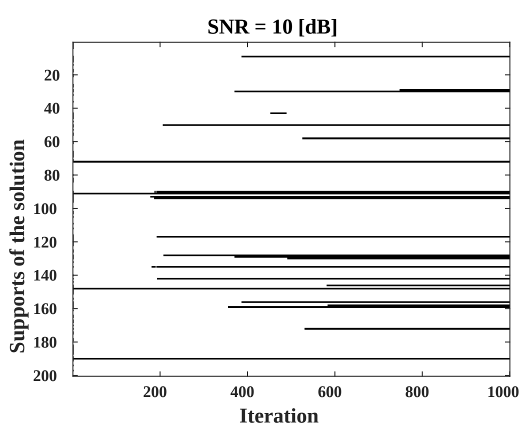

Figure 7: Support recovery using AMP-B-SBL for SNR=10[dB].

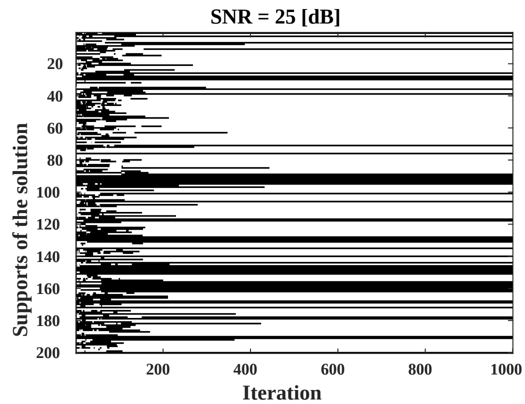

Figure 8: Support recovery using AMP-B-SBL for SNR=25[dB].

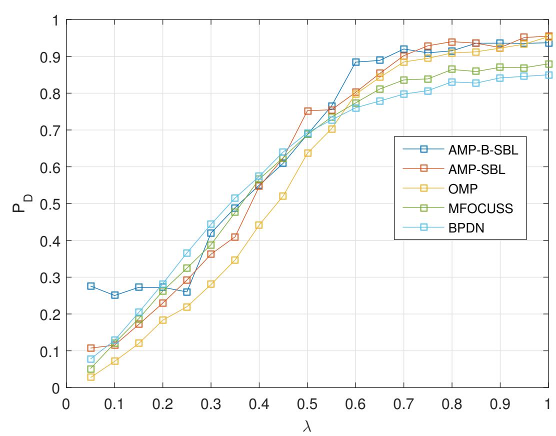

Figure 9: Comparison in terms of detection rate.

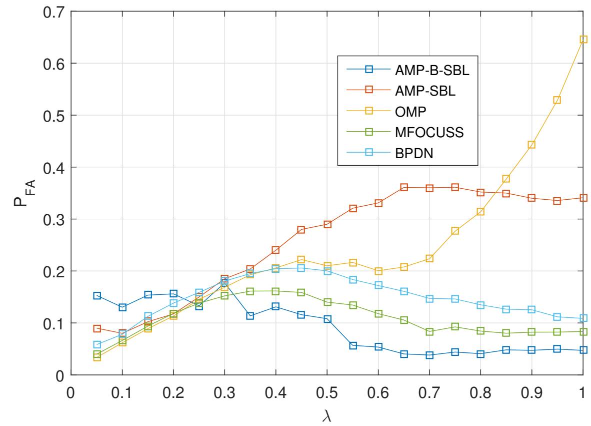

Figure 10: Comparison of false alarm rate in support recovery.

AMP-SBL, OMP, and MFOCUSS show approximately the same performance.

In Fig. 3.10, the false alarm rate (the rate of deciding on wrong supports in the solution) *vs.* λ is illustrated. According to Fig. 3.10, we observe that the proposed algorithm has a higher false alarm rate compared to the other algorithms for $\lambda = [0, 0.2]$. Comparing the detection rate with the false alarm rate of AMP-B-SBL for such range of λ shows the trade off that exists between the detection rate and false alarm rate. For $0.2 < \lambda \leq 0.3$ all the algorithms show the same performance. However, AMP-B-SBL illustrates the *lowest* false alarm rate for $\lambda > 0.3$. Fig. 3.11 demonstrates the overall performance of the algorithms as a combination of support detection, false alarm rate, and the number of measurements.

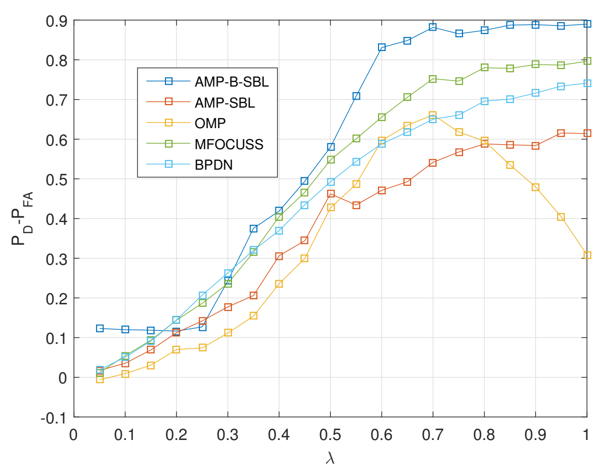

Figure 11: Comparison based on the difference between the experimental detection and false alarm rate.

From Fig. 3.11, we observe that the best performance belongs to the AMP-B-SBL algorithm. This shows that incorporating the measure of contiguity in the supports into the SBL algorithm boosts the recovery performance for the clustered pattern sparse signals. Finally, Fig. 3.12 shows the performance comparison in terms of normalized mean-squared error between the true and estimated solutions. According to Fig. 3.12, AMP-B-SBL outperforms the other algorithms in terms of estimating the true solution.

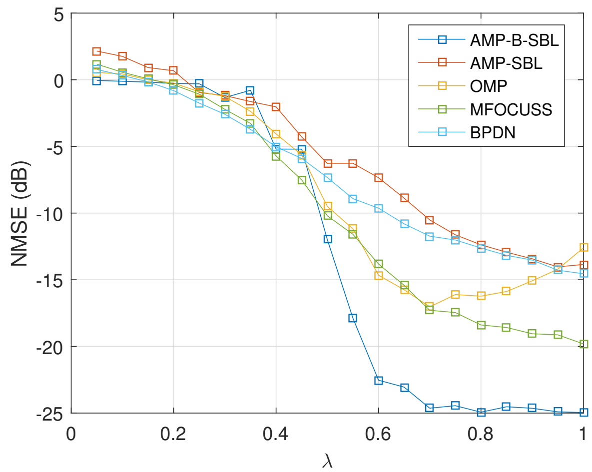

Figure 12: Comparison in terms of normalized mean-squared error between the true and estimated solution.

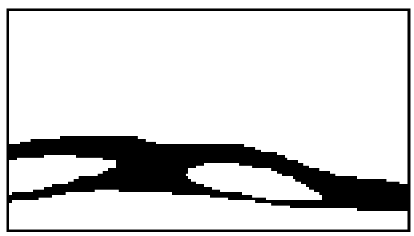

Figure 13: True Image.

3.2. Simulations on Real-Data

Here, we further evaluate the performance of the algorithms via the following example. In this case we use the image of 112×200 pixels, where we consider the black pixels as the "interesting" locations. The image is shown in Fig. 13.

In the simulations for this image, the value of 1 is assigned to the pixels with black color and 0 to the white ones. Then, the matrix corresponding to the image is reshaped and the vector $\mathbf{x} \in \mathbf{R}^{11200 \times 1}$ is constructed. The number of measurements are set to 5040, i.e., $\lambda = 0.45$. The sensing matrix is constructed in the same way we described earlier. Then, the measurements are obtained from $\mathbf{y} = A\mathbf{x} + \mathbf{e}$ with SNR=25dB. All the algorithms are fed with the measurement vector $\mathbf{y}$ and the sensing matrix A. The reconstructed images are illustrated in Fig. 3.14. The threshold of 0.01 is used for all the algorithms, we applied the threshold of 0.01, meaning that we discarded those estimated supports that corresponded to non-zero elements with the absolute value of less than 0.01. Below, we also include the reconstruction based on CLUSSMCMC algorithm [20]. In order to compare the reconstructed images illustrated in Fig. 3.14, we compare the obtained results based on the NMSE and peak-SNR (PSNR) as shown in Tab. 2. According to Tab. 2, we observe that CLUSSMCMC provides the best performance in terms of NMSE and PSNR. However, it is much slower compared to other algorithms. In contrast, AMP-B-SBL algorithm provides high performance for NMSE, PSNR, and the speed of the algorithm. By comparing the results of AMP-SBL with AMP-B-SBL, we also see that using the measure of clumpiness into the SBL algorithm definitely improves the reconstruction performance.

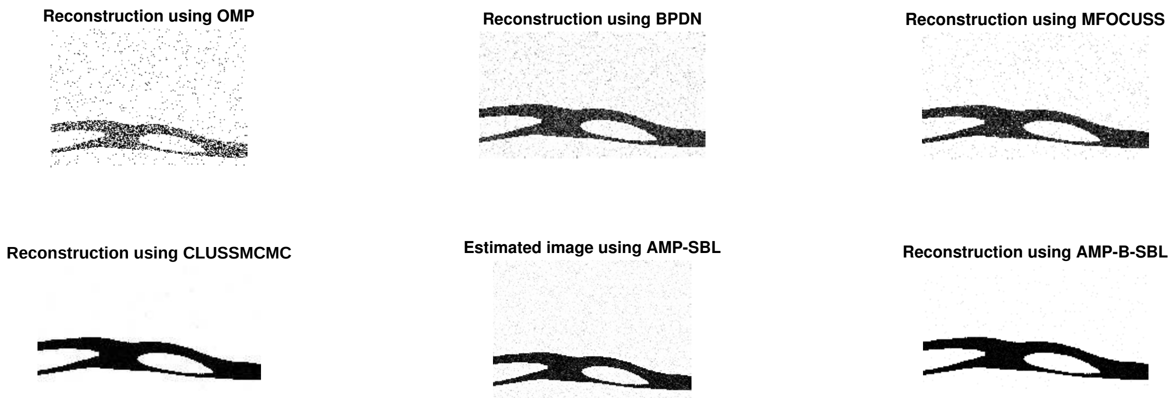

Figure 14: Results of reconstructed images for OMP, BPDN, MFOCUSS, CLUSSMCMC, AMP-SBL, and AMP-B-SBL.

Table 2: NMSE and PSNR comparison in image reconstruction.

Algorithm	NMSE (dB)	PSNR (dB)	speed
OMP	-1.7124	10.9425	Very Fast
BPDN	-8.7905	17.0205	Very Fast
MFOCUSS	-10.6522	18.8822	Slow
CLUSSMCMC	-27.42	35.6504	Slow
AMP-SBL	-13.4642	22.6943	Fast
AMP-B-SBL	-24.5600	32.7901	Fast

4. SUMMARY

A new algorithm for solving the block-sparse SMV problem is proposed. By assuming Gaussian distributions for all the variables, we were able to use the factor graph which only needs to propagate the mean and variance of the full messages. We further reduced the computational load of the problem using the approximate message passing. It was shown that dependence of the solution variances on Sigma-Delta encourages the solution to have a block-sparsity structure and also reduces the normalized mean-squared error between the true and the estimated sparse signal.

5. APPENDIX

In this section, we provide more details on the update rule of the parameters and variables of the proposed model and the AMP-B-SBL algorithm.

5.1. Factor graph

The factor graph of our model for the B-SMV problem is illustrated in Fig. 15. In such graphical probabilistic model, the large circle nodes represent the random variables and the shaded boxes show the function nodes. The small circles on the right hand-side represent the set of hyper-parameters and the shaded circles on the left denote the set of observations.

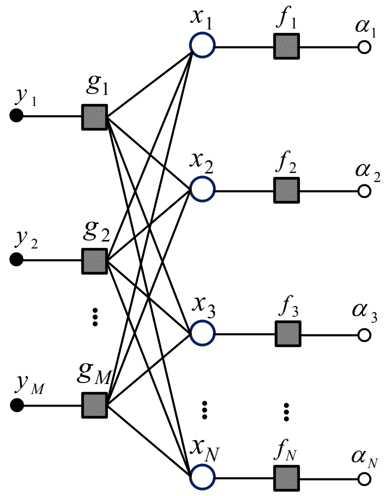

Figure 15: Factor graph for the B-SMV problem.

In Fig. 15, the term g_m denotes the likelihood function based on the observation y_m and is defined as

$$g_m := p(\mathbf{y}|\mathbf{x}). \tag{15}$$

Assume that we have M observations collected into $\mathbf{y} = [y_1, \ldots, y_M]^T$ and the model is $\mathbf{y} = A\mathbf{x} + \mathbf{e}$ with $\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \sigma^2 I_N)$. Therefore,

$$p(\mathbf{y}|\mathbf{x}; \sigma^2) = (2\pi\sigma^2)^{-\frac{M}{2}} e^{\{-\frac{1}{2\sigma^2}\|\mathbf{y}-A\mathbf{x}\|_2^2\}}. \tag{16}$$

The prior distribution on the solution vector $\mathbf{x}$ is defined as $x_n \sim \mathcal{N}(0, \alpha_n), \forall n = 1, \ldots, N$. In this factor graph, the function node f_n is defined as follows

$$f_n := p(x_n; 0, \alpha_n), \ \forall n = 1, \ldots, N.$$

5.2. Message passing from a function node to a variable node

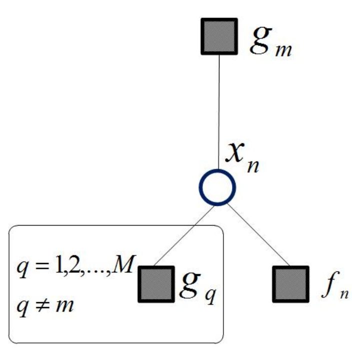

Figure 16: Message passing from a function node to a variable node when there also exist other function nodes connected to the same variable node.

According to Fig. 16, the message propagates from the function node g_m to the variable node x_n can be obtained as

$$M_{g_m \to x_n}(x_n) \propto \int_{x_{-n}} g_m\big(\text{Ne}(g_m)\big) \prod_{q \neq n} M_{x_q \to g_m}(x_q), \tag{17}$$

where $\int_{x_{-n}}$ denotes the integration over all $x_i, i \neq n$.

Suppose that the message (containing the mean and variance) of a variable node $x_n, \forall n = 1, \ldots, N$ to a function node $g_m, \forall m = 1, \ldots, M$ is as follows

$$M_{g_m \to x_n} \propto \mathcal{N}(a_{mn} x_n; z_{mn}, c_{mn}), \tag{18}$$

where

$$\begin{aligned} z_{mn} &= y_m - \sum_{q \neq n} a_{mq} \mu_q \\ c_{mn} &= \sigma^2 + \sum_{q \neq n} |a_{mq}|^2 \Sigma_q. \end{aligned}$$

Proof:

For simplicity and without loss of generality, suppose that the vector $\mathbf{x}$ has only two elements and is defined as $\mathbf{x} := [x_1, x_2]^T$. Suppose that we are interested in finding the message passes from the function node g_m to the variable node x_1. Let's have the following definitions

$$X_1 := a_{m1} x_1, \quad \bar{y}_m := y_m - X_1.$$

Now the posterior inference on the variable x_1 can be obtained from the followings

$$\begin{aligned} p(X_1|-) &\propto \int \exp\{-\frac{1}{2\sigma^2}(y_m - a_{m1}x_1 - a_{m2}x_2)^2 - \frac{1}{2}\sum_{q \neq 1}^{2} \frac{1}{\Sigma_q}(x_q - \mu_q)^2\} dx_2 \\ &\propto \int \exp\{-\frac{1}{2\sigma^2}(a_{m2}x_2 - \bar{y}_m)^2 - \frac{1}{2\Sigma_2}(x_2 - \mu_2)^2\} dx_2 \\ &\propto \int \exp\{-\frac{1}{2\sigma^2}(a_{m2}^2 x_2^2 - 2a_{m2}\bar{y}_m x_2 + \bar{y}_m^2) - \frac{1}{2\Sigma_2}(x_2^2 - 2\mu_2 x_2)\} dx_2 \\ &\propto \exp\{-\frac{1}{2\sigma^2}\bar{y}_m^2\} \int \exp\{-\frac{1}{2}((\frac{a_{m2}^2}{\sigma^2} + \frac{1}{\Sigma_2})x_2^2 - 2x_2(\frac{a_{m2}}{\sigma^2}\bar{y}_m + \frac{\mu_2}{\Sigma_2}))\} dx_2. \end{aligned}$$

Now, we factor out the term $\frac{a_{mn}^2}{\sigma^2} + \frac{1}{\Sigma_2}$ (inside the integral) and complete the square with respect to the variable x_2.

$$p(X_1|-) \propto \exp^{\{-\frac{1}{2\sigma^2}\bar{y}_m^2\}} \times$$
$$\int \exp(\{-\frac{1}{2}((\frac{a_{m2}^2}{\sigma^2} + \frac{1}{\Sigma_2})(x_2 - (\frac{a_{m2}^2}{\sigma^2} + \frac{1}{\Sigma_2})^{-1}(\frac{a_{m2}}{\sigma^2}\bar{y}_m + \frac{\mu_2}{\Sigma_2}))^2 - (\frac{a_{m2}^2}{\sigma^2} + \frac{1}{\Sigma_2})^{-1}(\frac{a_{m2}}{\sigma^2}\bar{y}_m + \frac{\mu_2}{\Sigma_2})^2)\} dx_2$$

Notice that the term

$$\int \exp\{-\frac{1}{2}((\frac{a_{m2}^2}{\sigma^2} + \frac{1}{\Sigma_2})(x_2 - (\frac{a_{m2}^2}{\sigma^2} + \frac{1}{\Sigma_2})^{-1}(\frac{a_{m2}}{\sigma^2}\bar{y}_m + \frac{\mu_2}{\Sigma_2}))^2)\} dx_2$$

is the integration of a probability distribution function under its whole support and is equal to 1. Therefore,

$$\begin{aligned} p(X_1|-) &\propto \exp\{-\frac{1}{2}\big(\frac{1}{\sigma^2}\bar{y}_m^2 - (\frac{a_{m2}^2}{\sigma^2} + \frac{1}{\Sigma_2})^{-1}(\frac{a_{m2}}{\sigma^2}\bar{y}_m + \frac{\mu_2}{\Sigma_2})^2\big)\} \\ &\propto \exp\{-\frac{1}{2}\big(\frac{1}{\sigma^2}\bar{y}_m^2 - (\frac{a_{m2}^2}{\sigma^2} + \frac{1}{\Sigma_2})^{-1}(\frac{a_{m2}^2}{\sigma^4}\bar{y}_m^2 + 2\frac{a_{m2}}{\sigma^2}\frac{\mu_2}{\Sigma_2}\bar{y}_m)\big)\} \\ &\propto \exp\{-\frac{1}{2\sigma^2}\big((y_m - X_1)^2 - (\frac{a_{m2}^2}{\sigma^2} + \frac{1}{\Sigma_2})^{-1}(\frac{a_{m2}^2}{\sigma^2}(y_m - X_1)^2 + 2a_{m2}\frac{\mu_2}{\Sigma_2}(y_m - X_1))\big)\} \\ &\propto \exp\{-\frac{1}{2\sigma^2}\big(X_1^2(1 - (\frac{a_{m2}^2}{\sigma^2} + \frac{1}{\Sigma_2})^{-1}\frac{a_{m2}^2}{\sigma^2}) - 2X_1(y_m - (\frac{a_{m2}^2}{\sigma^2} + \frac{1}{\Sigma_2})^{-1}(\frac{a_{m2}^2}{\sigma^2}y_m + a_{m2}\frac{\mu_2}{\Sigma_2}))\big)\} \end{aligned}$$

Therefore,

$$\begin{aligned} &p(X_1|-) = p(a_{m1}x_1|-) \propto \\ &e^{\big\{-\frac{1}{2}\big(\frac{1}{\sigma^2}(1-(\frac{a_{m2}^2}{\sigma^2}+\frac{1}{\Sigma_2})^{-1}\frac{a_{m2}^2}{\sigma^2})\big)\big(X_1-\frac{\sigma^2}{\sigma^2}(1-(\frac{a_{m2}^2}{\sigma^2}+\frac{1}{\Sigma_2})^{-1}\frac{a_{m2}^2}{\sigma^2})^{-1}(y_m-(\frac{a_{m2}^2}{\sigma^2}+\frac{1}{\Sigma_2})^{-1}(\frac{a_{m2}^2}{\sigma^2}y_m+a_{m2}\frac{\mu_2}{\Sigma_2}))\big)^2\big\}}. \end{aligned} \tag{19}$$

5.2.1. Finding the variance

Based on (19), the variance of X_1 can be found as follows

$$\begin{aligned} \Sigma_{X_1} &= \big(\frac{1}{\sigma^2}(1 - (\frac{a_{m2}^2}{\sigma^2} + \frac{1}{\Sigma_2})^{-1}\frac{a_{m2}^2}{\sigma^2})\big)^{-1} \\ &= \sigma^2\big(1 - (\frac{a_{m2}^2}{\sigma^2} + \frac{1}{\Sigma_2})^{-1}\frac{a_{m2}^2}{\sigma^2}\big)^{-1}. \end{aligned} \tag{20}$$

Based on the matrix inversion lemma, we have

$$(A - BD^{-1}C)^{-1} = A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1}, \tag{21}$$

where for our case, $A = 1, B = 1, C = \frac{a_{m2}^2}{\sigma^2}$, and $D = (\frac{a_{m2}^2}{\sigma^2} + \frac{1}{\Sigma_2})$. By substituting (21) into (20) we will have

$$\begin{aligned} \Sigma_{X_1} &= \sigma^2\big(1 + (\frac{a_{m2}^2}{\sigma^2} + \frac{2}{\Sigma_2} - \frac{a_{m2}^2}{\sigma^2})^{-1}\frac{a_{m2}^2}{\sigma^2}\big) \\ &= \sigma^2(1 + \Sigma_2\frac{a_{m2}^2}{\sigma^2}). \end{aligned}$$

Therefore,

$$\Sigma_{X_1} = \sigma^2 + a_{m2}^2\Sigma_2. \tag{22}$$

5.2.2. Finding the mean

According to (19), the mean of the random variable X_1 i.e., $a_{m1}E[x_1]$ is

$$
\begin{aligned}
\mu_{X_1} &= \big(1-(\frac{a_{m2}^2}{\sigma^2}+\frac{1}{\Sigma_2})^{-1}\frac{a_{m2}^2}{\sigma^2}\big)^{-1}\big(y_m-(\frac{a_{m2}^2}{\sigma^2}+\frac{1}{\Sigma_2})^{-1}\frac{a_{m2}^2}{\sigma^2}y_m-(\frac{a_{m2}^2}{\sigma^2}+\frac{1}{\Sigma_2})^{-1}a_{m2}\frac{\mu_2}{\Sigma_2}\big)\\
&= \big(1-(\frac{a_{m2}^2}{\sigma^2}+\frac{1}{\Sigma_2})^{-1}\frac{a_{m2}^2}{\sigma^2}\big)^{-1}\big(y_m(1-(\frac{a_{m2}^2}{\sigma^2}+\frac{1}{\Sigma_2})^{-1}\frac{a_{m2}^2}{\sigma^2})-(\frac{a_{m2}^2}{\sigma^2}+\frac{1}{\Sigma_2})^{-1}a_{m2}\frac{\mu_2}{\Sigma_2}\big)\\
&= y_m-\big(1-(\frac{a_{m2}^2}{\sigma^2}+\frac{1}{\Sigma_2})^{-1}\frac{a_{m2}^2}{\sigma^2}\big)^{-1}(\frac{a_{m2}^2}{\sigma^2}+\frac{1}{\Sigma_2})^{-1}a_{m2}\frac{\mu_2}{\Sigma_2}\\
&= y_m-\Big((\frac{a_{m2}^2}{\sigma^2}+\frac{1}{\Sigma_2})\big(1-(\frac{a_{m2}^2}{\sigma^2}+\frac{1}{\Sigma_2})^{-1}\frac{a_{m2}^2}{\sigma^2}\big)\Big)^{-1}a_{m2}\frac{\mu_2}{\Sigma_2}\\
&= y_m-\big((\frac{a_{m2}^2}{\sigma^2}+\frac{1}{\Sigma_2})-\frac{a_{m2}^2}{\sigma^2}\big)^{-1}a_{m2}\frac{\mu_2}{\Sigma_2}\\
&= y_m-\Sigma_2 a_{m2}\frac{\mu_2}{\Sigma_2}.
\end{aligned}
$$

Therefore,

$$
\mu_{x_1} = y_m - a_{m2}\mu_2. \tag{23}
$$

Finally, it is straight forward to show that the generalized form of the message that propagates from the function node g_m to the variable node x_n is as follows

$$
M_{g_m\to x_n} \propto \mathcal{N}(a_{mn}x_n; z_{mn}, c_{mn}),
$$

where

$$
\begin{aligned}
z_{mn} &= y_m - \sum_{q\neq n} a_{mq}\mu_q\\
c_{mn} &= \sigma^2 + \sum_{q\neq n} |a_{mq}|^2\Sigma_q.
\end{aligned}
$$

5.3. Product of messages passed from function nodes to a variable node

Based on what we obtained in (18), we then have

$$
\begin{aligned}
\prod_{l\neq m} M_{g_l\to x_n} &\propto \exp\big\{-\frac{1}{2}\sum_{l\neq m}\frac{(a_{ln}x_n-z_{ln})^2}{c_{ln}}\big\}\\
&\propto \exp\big\{-\frac{1}{2}\big((\sum_{l\neq m}\frac{a_{ln}^2}{c_{ln}})x_n^2-2x_n(\sum_{l\neq m}\frac{a_{ln}z_{ln}}{c_{ln}})\big)\big\}.
\end{aligned}
$$

Therefore,

$$
\prod_{l\neq m} M_{g_l\to x_n} \propto \mathcal{N}\big(x_n; \frac{\sum_{l\neq m}\frac{a_{ln}z_{ln}}{c_{ln}}}{\sum_{l\neq m}\frac{a_{ln}^2}{c_{ln}}}, \frac{1}{\sum_{l\neq m}\frac{a_{ln}^2}{c_{ln}}}\big). \tag{24}
$$

In the large system limit we can approximate c_{ln} by

$$
c_{ln} \simeq c_n := \frac{1}{M}\sum_{m=1}^{M} c_{mn}. \tag{25}
$$

Under the assumption that the sensing matrix A have already normalized with respect to its columns, we then can have the following approximation

$$\sum_{l\neq m} a_{ln}^2 \simeq \sum_{m=1}^{M} a_{mn}^2 = 1. \tag{26}$$

Substituting (25) and (26) into (24) yields

$$\prod_{l\neq m} M_{g_l\to x_n} \propto \mathcal{N}\big(x_n; \sum_{l\neq m} a_{ln}z_{ln}, c_{ln}\big). \tag{27}$$

5.4. Message passing from a variable node to a function node

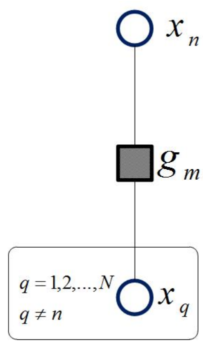

Figure 17: Message passing from a variable node to a function node when there also exist other variable nodes connected to the same function node.

According to Fig. 17, the message propagates from the variable node x_n to the function node g_m can be obtained as follows

$$M_{x_n\to g_m} \propto M_{f_n\to x_n} \prod_{l\neq m} M_{g_l\to x_n}, \tag{28}$$

where $\prod_{l\neq m} M_{g_l\to x_n}$ was obtained in (24) and $M_{f_n\to x_n} \propto \mathcal{N}(x_n; 0, \alpha_n)$. Therefore,

$$\begin{aligned} M_{x_n\to g_m} &\propto \exp\Big\{-\frac{1}{2}\big(\alpha_n^{-1}x_n^2 + (\sum_{l\neq m}\frac{a_{ln}^2}{c_{nl}})(x_n - \frac{\sum_{l\neq m}\frac{a_{nl}z_{nl}}{c_{nl}}}{\sum_{l\neq m}\frac{a_{nl}^2}{c_{nl}}})^2\big)\Big\} \\ &\propto \exp\Big\{-\frac{1}{2}\Big((\alpha_n^{-1} + \sum_{l\neq m}\frac{a_{ln}^2}{c_{nl}})x_n^2 - 2x_n\big((\sum_{l\neq m}\frac{a_{ln}^2}{c_{nl}})\frac{\sum_{l\neq m}\frac{a_{nl}z_{nl}}{c_{nl}}}{\sum_{l\neq m}\frac{a_{ln}^2}{c_{nl}}})\big)\Big)\Big\}. \end{aligned}$$

Using (25), we then have

$$\begin{aligned} M_{x_n\to g_m} &\propto \exp\Big\{-\frac{1}{2}\big((\frac{1}{\alpha_n} + \frac{1}{c_n}\sum_{l\neq m} a_{ln}^2)x_n^2 - 2x_n(\frac{1}{c_n}\sum_{l\neq m} a_{nl}c_{nl})\big)\Big\} \\ &\propto \exp\Big\{-\frac{1}{2}\big((\frac{c_n + \alpha_n\sum_{l\neq m} a_{ln}^2}{c_n\alpha_n})x_n^2 - 2x_n(\frac{1}{c_n}\sum_{l\neq m} a_{nl}c_{nl})\big)\Big\} \\ &\propto \exp\Big\{-\frac{1}{2}(\frac{c_n + \alpha_n\sum_{l\neq m} a_{ln}^2}{c_n\alpha_n})(x_n - \frac{c_n\alpha_n}{c_n + \alpha_n\sum_{l\neq m} a_{ln}^2}\frac{1}{c_n}\sum_{l\neq m} a_{nl}z_{nl})^2\Big\}. \end{aligned}$$

Therefore, the message that goes from the variable node x_n to the function node g_m is as follows

$$M_{x_n \to g_m} \propto \mathcal{N}\big(x_n; \sum_{l \neq m} a_{nl} z_{ln} \frac{\alpha_n}{c_n + \alpha_n \sum_{l \neq m} a_{ln}^2}, \frac{c_n \alpha_n}{c_n + \alpha_n \sum_{l \neq m} a_{ln}^2}\big). \tag{29}$$

Finally, substituting (26) into (29) leads to the message passing below.

$$M_{x_n \to g_m} \propto \mathcal{N}\big(x_n; \sum_{l \neq m} a_{nl} z_{ln} \frac{\alpha_n}{c_n + \alpha_n}, \frac{c_n \alpha_n}{c_n + \alpha_n}\big). \tag{30}$$

5.5. Estimating the posterior on the variables of interest

Here, we estimate the posterior inference for the variable x_n.

$$\begin{aligned} p(x_n|\mathbf{y}) &\propto p(x_n; \alpha_n) p(\mathbf{y}|x_n, \sigma^2) \\ &\propto p(x_n; \alpha_n) \prod_{m=1}^{M} p(y_m|x_n) \\ &\propto M_{f_n \to x_n} \prod_{m=1}^{M} M_{g_m \to x_n} \end{aligned}$$

Using (24), it is straightforward to show that

$$\prod_{m=1}^{M} M_{g_m \to x_n} \propto \mathcal{N}\big(x_n; \frac{\sum_{m=1}^{M} \frac{a_{mn} z_{mn}}{c_{mn}}}{\sum_{m=1}^{M} \frac{a_{mn}^2}{c_{mn}}}, \frac{1}{\sum_{m=1}^{M} \frac{a_{mn}^2}{c_{mn}}}\big). \tag{31}$$

Based on (29) and (31), we finally get the followings

$$p(x_n|\mathbf{y}) \propto \mathcal{N}(x_n; \mu_{x_n|-}, \Sigma_{x_n|-}), \tag{32}$$

where

$$\begin{aligned} \mu_{x_n|-} &= \sum_{m=1}^{M} a_{mn} z_{mn} \big(\frac{\alpha_n}{c_n + \alpha_n \sum_{m=1}^{M} a_{mn}^2}\big), \\ \Sigma_{x_n|-} &= \frac{c_n \alpha_n}{c_n + \alpha_n \sum_{m=1}^{M} a_{mn}^2}. \end{aligned}$$

Again, substituting (25) and (26) into (32), yields

$$\begin{aligned} \mu_{x_n|-} &= \sum_{m=1}^{M} a_{mn} z_{mn} \big(\frac{\alpha_n}{c_n + \alpha_n}\big), \\ \Sigma_{x_n|-} &= \frac{c_n \alpha_n}{c_n + \alpha_n}. \end{aligned}$$

5.6. Update rule for the hyper-parameters

Below, we describe the update equations for our model using the EM algorithm.

5.6.1. Update rule for α

In this section, we describe the update rule for the variance α_n of x_n using the expectation-maximization (EM) algorithm. Notice that our prior on the entries of the solution vector $\mathbf{x}$ is $x_n \sim \mathcal{N}(0, \alpha_n), \forall n = 1, \ldots, N$. The reason for considering such prior on $\mathbf{x}$ is to encourage the sparsity in the solution. We further consider the following prior distribution on each α_n and make it depend on the measure of clumpiness i.e., Sigma-Delta

$$\alpha_n \sim \mathcal{N}\big(e^{\{\frac{(\Sigma\Delta)|_{b(x_n,.)=0} - (\Sigma\Delta)|_{b(x_n,.)=1} - 1}{\theta_1}\}}, \theta_2\big), \quad \forall n = 1, \ldots, N. \tag{33}$$

Therefore,

$$\begin{aligned} {\alpha_n}^{[k+1]} &= \arg\max_{\alpha_n} \mathbb{E}_{\mathbf{x}|\mathbf{y},\theta_1,\theta_2,\sigma^2,{\alpha_n}^{[k]}} \big[\log\{p(\mathbf{y}, \mathbf{x}, \boldsymbol{\alpha}, \theta_1, \theta_2, \sigma^2)\}\big] \\ &= \arg\max_{\alpha_n} \mathbb{E}_{\mathbf{x}|-}\big[\log\{p(\mathbf{y}|\mathbf{x}, \sigma^2 I_M)\prod_{n=1}^{N} p(x_n; 0, \alpha_n) p(\alpha_n; e^{\Big\{\frac{(\Sigma\Delta)|_{b(x_n,.)=0} - (\Sigma\Delta)|_{b(x_n,.)=1} - 1}{\theta_1}\Big\}}, \theta_2)\}\big] \end{aligned} \tag{34}$$

Notice that

$$\begin{aligned} &p(\mathbf{y}, \mathbf{x}, \boldsymbol{\alpha}, \theta_1, \theta_2, \sigma^2) \propto (2\pi\sigma^2)^{-\frac{M}{2}} e^{\{-\frac{1}{2\sigma^2}\|\mathbf{y}-A\mathbf{x}\|_2^2\}} \times \\ &\prod_{n=1}^{N} (2\pi\alpha_n)^{-\frac{1}{2}} e^{\{-\frac{1}{2\alpha_n}x_n^2\}} (2\pi\theta_2)^{-\frac{1}{2}} e^{\{-\frac{1}{2\theta_2}\big(\alpha_n - e^{\Big\{\frac{(\Sigma\Delta)|_{b(x_n,.)=0} - (\Sigma\Delta)|_{b(x_n,.)=1} - 1}{\theta_1}\Big\}}\big)^2\}}. \end{aligned} \tag{35}$$

Substituting (35) into (34) and discarding the terms that are independent of α_n yields

$$\begin{aligned} {\alpha_n}^{[k+1]} &= \arg\max_{\alpha_n} \mathbb{E}_{\mathbf{x}|\mathbf{y},\theta_1,\theta_2,\sigma^2,{\alpha_n}^{[k]}} \big[\log\{{\alpha_n}^{-\frac{1}{2}} e^{\{-\frac{1}{2\alpha_n}x_n^2\}} e^{\Big\{-\frac{1}{2\theta_2}\big(\alpha_n - e^{\{\frac{(\Sigma\Delta)|_{b(x_n,.)=0} - (\Sigma\Delta)|_{b(x_n,.)=1} - 1}{\theta_1}\}}\big)^2\Big\}}\}\big] \\ &= \arg\min_{\alpha_n} \mathbb{E}_{\mathbf{x}|\mathbf{y},\theta_1,\theta_2,\sigma^2,{\alpha_n}^{[k]}} \big[\log\{\alpha_n\} + \frac{1}{\alpha_n}x_n^2 + \frac{1}{\theta_2}\big(\alpha_n - e^{\{\frac{(\Sigma\Delta)|_{b(x_n,.)=0} - (\Sigma\Delta)|_{b(x_n,.)=1} - 1}{\theta_1}\}}\big)^2\big] \\ &= \arg\min_{\alpha_n} \log\{\alpha_n\} + \frac{1}{\alpha_n}\mathbb{E}_{\mathbf{x}|\mathbf{y},\theta_1,\theta_2,\sigma^2,{\alpha_n}^{[k]}}[x_n^2] + \frac{1}{\theta_2}\big(\alpha_n - e^{\{\frac{(\Sigma\Delta)|_{b(x_n,.)=0} - (\Sigma\Delta)|_{b(x_n,.)=1} - 1}{\theta_1}\}}\big)^2. \end{aligned}$$

Notice that

$$\mathbb{E}_{\mathbf{x}|\mathbf{y},\theta_1,\theta_2,\sigma^2,{\alpha_n}^{[k]}}[x_n^2] = \mu_{x_n|-}^2 + \Sigma_{x_n|-}.$$

Therefore,

$${\alpha_n}^{[k+1]} = \arg\min_{\alpha_n} f(\alpha_n), \tag{36}$$

where

$$f(\alpha_n) = \log\{\alpha_n\} + \frac{\mu_{x_n|-}^2 + \Sigma_{x_n|-}}{\alpha_n} + \frac{1}{\theta_2}\big(\alpha_n - e^{\{\frac{(\Sigma\Delta)|_{b(x_n,.)=0} - (\Sigma\Delta)|_{b(x_n,.)=1} - 1}{\theta_1}\}}\big)^2$$

We now perform the maximization step of the EM algorithm, which for our case it becomes a minimization problem.

$$\frac{\partial f(\alpha_n)}{\partial \alpha_n} = 0$$

and hence we solve

$$\alpha_n^3 - e^{\left\{\frac{(\Sigma\Delta)|_{b(x_n,.)=0} - (\Sigma\Delta)|_{b(x_n,.)=0^{-1}}}{\theta_1}\right\}} \alpha_n^2 + \frac{\theta_2}{2}\alpha_n - \frac{\theta_2}{2}(\mu_{x_n|-}^2 + \Sigma_{x_n|-}) = 0. \quad (37)$$

Finally, a solution among all the three possible roots for (37) which minimizes (36) is the update rule for ${\alpha_n}^{[k+1]}$.

5.6.2. Update rule for σ^2

Here, we describe the update rule to estimate the noise variance using EM technique.

$$\begin{aligned}
\sigma^{2[k+1]} &= \arg\max_{\sigma^2} \mathbb{E}_{\mathbf{x}|\mathbf{y},\theta_1,\theta_2,\boldsymbol{\alpha},\sigma^{2[k]}}\left[\log\left\{p(\mathbf{y},\mathbf{x},\boldsymbol{\alpha},\theta_1,\theta_2,\sigma^2)\right\}\right] \\
&= \arg\max_{\sigma^2} \mathbb{E}_{\mathbf{x}|\mathbf{y},\theta_1,\theta_2,\boldsymbol{\alpha},\sigma^{2[k]}}\left[\log\left\{(2\pi\sigma^2)^{-\frac{M}{2}} e^{\{-\frac{1}{2\sigma^2}\|\mathbf{y}-A\mathbf{x}\|_2^2\}}\right\}\right]
\end{aligned}$$

Notice that in the above maximization problem, we have discarded all the terms that were independent of σ^2. Therefore,

$$\begin{aligned}
\sigma^{2[k+1]} &= \arg\min_{\sigma^2} \mathbb{E}_{\mathbf{x}|\mathbf{y},\theta_1,\theta_2,\boldsymbol{\alpha},\sigma^{2[k]}}\left[2M\log\{\sigma\} + \frac{1}{\sigma^2}\|\mathbf{y}-A\mathbf{x}\|_2^2\right] \\
&= \arg\min_{\sigma^2} \{2M\log\{\sigma\} + \frac{1}{\sigma^2}\mathbb{E}_{\mathbf{x}|\mathbf{y},\theta_1,\theta_2,\boldsymbol{\alpha},\sigma^{2[k]}}\left[\|\mathbf{y}-A\mathbf{x}\|_2^2\right]\}
\end{aligned} \quad (38)$$

Remark: Here, we simplify the following expectation

$$\begin{aligned}
E_{\mathbf{x}|\mathbf{y},\theta_1,\theta_2,\boldsymbol{\alpha},\sigma^{2[k]}}\left[\|\mathbf{y}-A\mathbf{x}\|_2^2\right] &= \mathbb{E}_{\mathbf{x}|\mathbf{y},-}\left[\|\mathbf{y}-A\big(\boldsymbol{\mu}_{x|-} + (\mathbf{x}-\boldsymbol{\mu}_{x|-})\big)\|_2^2\right] \\
&= \mathbb{E}_{\mathbf{x}|\mathbf{y},-}\left[\|\mathbf{y}-A\boldsymbol{\mu}_{x|-}\|_2^2 + \|A(\mathbf{x}-\boldsymbol{\mu}_{x|-})\|_2^2\right] + ... \\
&\quad (\mathbf{y}-A\boldsymbol{\mu}_{x|-})^T A\mathbb{E}_{\mathbf{x}|\mathbf{y},-}\left[\mathbf{x}-\boldsymbol{\mu}_{x|-}\right] \\
&= \|\mathbf{y}-A\boldsymbol{\mu}_{x|-}\|_2^2 + \mathbb{E}_{\mathbf{x}|\mathbf{y},-}\left[\|A(\mathbf{x}-\boldsymbol{\mu}_{x|-})\|_2^2\right]
\end{aligned}$$

Notice that $\mathbb{E}_{\mathbf{x}|\mathbf{y},-}\left[\mathbf{x}-\boldsymbol{\mu}_{x|-}\right] = 0$. Therefore,

$$\begin{aligned}
\mathbb{E}_{\mathbf{x}|\mathbf{y},\theta_1,\theta_2,\boldsymbol{\alpha},\sigma^{2[k]}}\left[\|\mathbf{y}-A\mathbf{x}\|_2^2\right] &= \|\mathbf{y}-A\boldsymbol{\mu}_{x|-}\|_2^2 + \mathbb{E}_{\mathbf{x}|\mathbf{y},-}\left[(\mathbf{x}-\boldsymbol{\mu}_{x|-})^T A^T A(\mathbf{x}-\boldsymbol{\mu}_{x|-})\right] \\
&= \|\mathbf{y}-A\boldsymbol{\mu}_{x|-}\|_2^2 + \mathbb{E}_{\mathbf{x}|\mathbf{y},-}\left[\operatorname{tr}\left((\mathbf{x}-\boldsymbol{\mu}_{x|-})^T A^T A(\mathbf{x}-\boldsymbol{\mu}_{x|-})\right)\right] \\
&= \|\mathbf{y}-A\boldsymbol{\mu}_{x|-}\|_2^2 + \mathbb{E}_{\mathbf{x}|\mathbf{y},-}\left[\operatorname{tr}\left((\mathbf{x}-\boldsymbol{\mu}_{x|-})(\mathbf{x}-\boldsymbol{\mu}_{x|-})^T A^T A\right)\right] \\
&= \|\mathbf{y}-A\boldsymbol{\mu}_{x|-}\|_2^2 + \operatorname{tr}\left(\mathbb{E}_{\mathbf{x}|\mathbf{y},-}\left[(\mathbf{x}-\boldsymbol{\mu}_{x|-})(\mathbf{x}-\boldsymbol{\mu}_{x|-})^T A^T A\right]\right).
\end{aligned}$$

Here, we define

$$\Sigma_{x|-} := \mathbb{E}_{\mathbf{x}|\mathbf{y},-}\left[(\mathbf{x}-\boldsymbol{\mu}_{x|-})(\mathbf{x}-\boldsymbol{\mu}_{x|-})^T\right],$$

which is the variance of the posterior inference on $\mathbf{x}$ and is a diagonal matrix. Therefore,

$$\mathbb{E}_{\mathbf{x}|\mathbf{y},\theta_1,\theta_2,\boldsymbol{\alpha},\sigma^{2[k]}}\left[\|\mathbf{y}-A\mathbf{x}\|_2^2\right] = \|\mathbf{y}-A\boldsymbol{\mu}_{x|-}\|_2^2 + \mathrm{tr}(\Sigma_{x|-}A^TA). \tag{39}$$

Substituting (39) into (38) yields

$$\sigma^{2[k+1]} = \arg\min_{\sigma^2} f(\sigma), \tag{40}$$

where

$$f(\sigma) := 2M\log\sigma + \frac{1}{\sigma^2}\big(\|\mathbf{y}-A\boldsymbol{\mu}_{x|-}\|_2^2 + \mathrm{tr}(\Sigma_{x|-}A^TA)\big)$$

For the minimization purposes, we set $\frac{\partial f(\sigma)}{\partial\sigma} = 0$, which leads to

$$2M\frac{1}{\sigma} = \frac{2}{\sigma^3}\big(\|\mathbf{y}-A\boldsymbol{\mu}_{x|-}\|_2^2 + \mathrm{tr}(\Sigma_{x|-}A^TA)\big).$$

Therefore,

$$\begin{aligned}\sigma^{2[k+1]} &= \frac{\|\mathbf{y}-A\boldsymbol{\mu}_{x|-}\|_2^2 + \mathrm{tr}(\Sigma_{x|-}A^TA)}{M}\\ &= \frac{\|\mathbf{y}-A\boldsymbol{\mu}_{x|-}\|_2^2 + \sum_{n=1}^N \|\mathbf{a}_n\|_2^2\Sigma_{x_n|-}}{M}.\end{aligned} \tag{41}$$

In the case where the sensing matrix A is normalized with respected to its columns, (41) simplifies further as follows

$$\sigma^{2[k+1]} = \frac{\|\mathbf{y}-A\boldsymbol{\mu}_{x|-}\|_2^2 + \sum_{n=1}^N(\Sigma_{x_n|-})}{M}. \tag{42}$$

5.6.3. Update rule for θ_2

In this section, we describe the update rule for the variance on α_n using the EM algorithm. Notice that the prior on each α_n has been defined as

$$\alpha_n \sim \mathcal{N}\big(e^{\{\frac{(\Sigma\Delta)|_{b(x_n,.)=0}-(\Sigma\Delta|_{b(x_n,.)=1}-1)}{\theta_1}},\theta_2\}\big),\ \forall n=1,\ldots,N. \tag{43}$$

Based on the EM algorithm we have

$$\begin{aligned}\theta_2{}^{[k+1]} &= \arg\max_{\theta_2} \mathbb{E}_{\mathbf{x}|\mathbf{y},\theta_1,\boldsymbol{\alpha},\sigma^2,\theta_2{}^{[k]}}\big[\log\{p(\mathbf{y},\mathbf{x},\boldsymbol{\alpha},\theta_1,\theta_2,\sigma^2)\}\big]\\ &= \arg\max_{\theta_2} \mathbb{E}_{\mathbf{x}|\mathbf{y},\theta_1,\boldsymbol{\alpha},\sigma^2,\theta_2{}^{[k]}}\big[\log\big(\prod_{n=1}^N(2\pi\theta_2)^{-\frac{1}{2}}e^{\{-\frac{1}{2\theta_2}\left(\alpha_n-e^{\{\frac{(\Sigma\Delta)|_{b(x_n,.)=0}-(\Sigma\Delta)|_{b(x_n,.)=1}-1}{\theta_1}\}}\right)^2\}}\big)\big].\end{aligned}$$

Notice that in the above joint probability inside the expectation, we have discarded all the terms that were independent of θ_2. Therefore,

$$\begin{aligned}\theta_2{}^{[k+1]} &= \arg\min_{\theta_2} \mathbb{E}_{\mathbf{x}|\mathbf{y},\theta_1,\boldsymbol{\alpha},\sigma^2,\theta_2{}^{[k]}}\big[N\log\{\theta_2\} + \frac{1}{\theta_2}\sum_{n=1}^N(\alpha_n - e^{\{\frac{(\Sigma\Delta)|_{b(x_n,.)=0}-(\Sigma\Delta)|_{b(x_n,.)=1}-1}{\theta_1}\}})^2\big]\\ &= \arg\max_{\theta_2} f(\theta_2),\end{aligned}$$

where

$$f(\theta_2) = N\log\{\theta_2\} + \frac{1}{\theta_2}\sum_{n=1}^{N}(\alpha_n - e^{\{\frac{(\Sigma\Delta)|_{b(x_n,.)=0} - (\Sigma\Delta)|_{b(x_n,.)=1} - 1}{\theta_1}\}})^2.$$

For the minimization, we set $\frac{\partial f(\theta_2)}{\partial \theta_2} = 0$, which leads to

$$\frac{N}{\theta_2} = \frac{1}{\theta_2^2}\sum_{n=1}^{N}(\alpha_n - e^{\{\frac{(\Sigma\Delta)|_{b(x_n,.)=0} - (\Sigma\Delta)|_{b(x_n,.)=1} - 1}{\theta_1}\}})^2.$$

Therefore,

$$\theta_2{}^{[k+1]} = \frac{\sum_{n=1}^{N}\left(\alpha_n - e^{\{\frac{(\Sigma\Delta)|_{b(x_n,.)=0} - (\Sigma\Delta)|_{b(x_n,.)=1} - 1}{\theta_1}\}}\right)^2}{N}. \tag{44}$$

6. REFERENCES

[1] M. Shekaramiz, T. K. Moon, and J. H. Gunther, "On the block-sparse solution of single measurement vectors," in *49th Asilomar Conf. of Signals, Systems, and Computers*, pp. 508–512, Nov. 2015.

[2] M. Shekaramiz, T. K. Moon, and J. H. Gunther, "AMP-B-SBL: An algorthm for clustered sparse signals using approximate message passing," in *IEEE Uniqitous Computing, Electronics and Mobile Communication (UEMCON)*, pp. 1–5, Oct. 2016.

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

[4] M. Mishali and Y. C. Eldar, "Reduce and boost: Recovering arbitrary sets of jointly sparse vectors," *IEEE Trans. Sig. Proc.*, vol. 56, pp. 4692–4702, Oct. 2008.

[5] Z. Zhang and B. Rao, "Recovery of block sparse signals using the framework of block sparse Bayesian learning," in *IEEE Int. Conf. on Acoust., Speech and Sig. Proc. (ICASSP)*, pp. 3345–3348, 2012.

[6] H. E. Kwon and B. Rao, "On the benefits of the block-sparsity structure in sparse signal recovery," in *IEEE Int. Conf. on Acoust., Speech and Sig. Proc. (ICASSP)*, pp. 3685–3688, 2012.

[7] Z. Zhang and B. D. Rao, "Extension of SBL algorithms for the recovery of block sparse signals with intra-block correlation," *IEEE Trans. Sig. Proc.*, vol. 61, pp. 2009–2015, Apr. 2013.

[8] 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.

[9] M. Shekaramiz, T. Moon, and J. Gunther, "On the block-sparsity of multiple-measurement vectors," *Presented at the IEEE Sig. Proc. and SP Edu. Workshop, Utah, Aug.*, 2015.

[10] M. Elad and I. Yavneh, "A plurality of sparse representations is better than the sparsest one alone," *IEEE Trans. Info. Theory*, vol. 55, pp. 4701–4714, Oct. 2009.

[11] M. Al-Shoukairi and B. Rao, "Sparse Bayesian learning using approxiamte message passing," in *48th Asilomar Conf. on Sig., Syst. and Compt.*, pp. 1957–1961, Nov. 2014.

[12] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing: I. motivations and construction," in *Proc. Inform. Theory Workshop, (Cairo, Egypt)*, pp. 1–5, Jan. 2010.

[13] J. Ziniel and P. Schniter, "Dynamic compressive sensing of time-varying signals via approximate message passing," *IEEE Trans. Sig. Proc.*, vol. 61, pp. 5270–5284, Nov. 2013.

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

[15] P. Schniter, "Turbo reconstruction of structured sparse signals," in *Conf. Inf. Sci. Syst., Princton, NJ*, pp. 1–5, Mar. 2010.

[16] Z. Zhang, "SBL algorithms matlab code." `https://sites.google.com/site/researchbyzhang/software`. Accessed Sep. 21, 2015.

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

[18] S. Chen and D. Donoho, "Basis pursuit," in *Asilomar Conf. of Signals, Systems, and Computers*, pp. 41–44, Oct. 1994.

[19] S. F. Cotter, B. D. Rao, K. Engan, and K. K. Delgado, "Sparse solutions to linear inverse problem with multiple measurement vectors," *IEEE Trans. Sig. Proc.*, vol. 53, no. 7, pp. 2477–2488, 2005.

[20] 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.