Efficient Recovery of Structured Sparse Signals via Approximate Message Passing with Structured Spike and Slab Prior

Xiangming Meng[1], Sheng Wu[2,*], Michael Riis Andersen[3], Jiang Zhu[4], Zuyao Ni[2]

[1] Huawei Technologies Co. Ltd., Shanghai 201206, China
[2] Tsinghua Space Center, Tsinghua University, Beijing 100084, China
[3] Department of Computer Science, Aalto University P.O. Box 15400, FI-00076, Finland
[4] Ocean College, Zhejiang University, Zhoushan 316021, China
* The corresponding author, email: peach.shengsheng@gmail.com

Abstract: Due to limited volume, weight and power consumption, micro-satellite has to reduce data transmission and storage capacity by image compression when performs earth observation missions. However, the quality of images may be unsatisfied. This paper considers the problem of recovering sparse signals by exploiting their unknown sparsity pattern. To model structured sparsity, the prior correlation of the support is encoded by imposing a transformed Gaussian process on the spike and slab probabilities. Then, an efficient approximate message-passing algorithm with structured spike and slab prior is derived for posterior inference, which, combined with a fast direct method, reduces the computational complexity significantly. Further, a unified scheme is developed to learn the hyperparameters using expectation maximization (EM) and Bethe free energy optimization. Simulation results on both synthetic and real data demonstrate the superiority of the proposed algorithm.

Keywords: compressed sensing; structured sparsity; spike and slab prior; approximate message passing; expectation propagation.

I. INTRODUCTION

Due to limited volume, weight and power consumption, micro-satellite has to reduce data transmission and storage capacity by image compression when performs earth observation missions. However, the quality of images may become unsatisfied. It is well known that compressed sensing (CS) can reconstruct sparse signals accurately from under-sampled linear measurements [1]. CS makes it possible to compress onboard images of micro-satellites maximally, thereby minimizing the required data transmission and storage space, as image can be recovered by CS to meet the requirement of missions. In the last decade, CS technology has been applied in many areas such as imaging processing, machine learning, radar detection, and computer science. Moreover, exploiting the sparsity of the target signal in wireless communications has been studied intensively in recent years [2], [3]. Typical examples include channel estimation [4]–[6], multiuser detection [7], and low-resolution analog-to-digital converters [8]–[10]. To this

Received: Feb. 8, 2018
Revised: Mar. 12, 2018
Editor: Linling Kuang

end, plethora of methods have been studied in the past years. The Bayesian interpretation of sparse reconstruction involves maximum a posteriori (MAP) inference with some sparsity-promoting priors, e.g., Laplace prior [11], automatic relevance determination [12], Dirichlet process prior [13], and spike and slab prior [14]. Among various Bayesian methods, approximate message passing (AMP) [15] is one state-of-the-art algorithm for sparse reconstruction. AMP can be seen as a large system limit approximation of belief propagation [16] and is deeply related to the seminal Thouless-Anderson-Palmer (TAP) equations in spin glass theory [17]. Moreover, to deal with general linear mixing problems, AMP has been extended to generalized AMP (GAMP) [18], which greatly enables the wide applicability of the AMP framework in sparse reconstruction.

While many practical signals can be described as sparse, they often exhibit an underlying structure, such as clustered sparsity, i.e., the nonzero coefficients occur in clusters, which is also known as group sparse or block sparse [19], [20]. In such settings, the nearby coefficients exhibit dependencies and exploiting such intrinsic structure beyond simple sparsity can significantly boost the reconstruction performance [20]. From the optimization perspective, various regularizations that model specific sparsity pattern are proposed, e.g., group LASSO [21], StructOMP [22]. From the Bayesian perspective, a number of methods have been developed to use structured priors to model both sparsity and cluster patterns simultaneously. The main effort of these algorithms lies in constructing a hierarchical prior model, e.g., Markov tree [23], structured spike and slab [24]–[26], hierarchical Gamma-Gaussian [27] to encode the structured sparsity pattern.

In this paper, using the structured spike and slab prior [24]–[26] with high flexibility in modeling the sparsity pattern, an efficient message passing algorithm, termed as AMP with structured spike and slab prior (AMP-SSS), is proposed to recover structured sparse signals with no prior knowledge of the sparsity pattern. Different from [24]–[26], which used expectation propagation (EP) [28], [29] to perform approximate Bayesian inference, this paper resorts to AMP [15] and a fast direct method [30], so that the computational complexity of sparse reconstruction with spike and slab prior is significantly reduced. In practice, the sparsity pattern is *unknown*, and the original AMP cannot be directly applied to recover signals with structured prior distribution. To address these problems, following the Turbo AMP framework [31], [32], an efficient method to learn the hyperparameters of structured spike and slab prior using expectation maximization (EM) [14], [33] and Bethe free energy [4], [34], [35] optimization is proposed. It is important to note that though this paper considers the linear Gaussian model, the proposed method can be extended to generalized linear models in a straightforward way using the GAMP [36], [37]. To test the effectiveness of the proposed method, various experiments on both synthetic and real data are performed, showing that it achieves excellent performance in recovering structured sparse signals with Gaussian process prior.

The rest of this paper is organized as follows. In Section II, the generalized linear model with structured spike and slab prior is described, which encodes the structured sparsity using a transformed Gaussian process. In Section III, the posterior of the proposed model is computed using the framework of AMP. Section IV presents a unified learning scheme of the hyperparameters via EM and Bethe free energy minimization. To reduce the computational complexity, in Section V, a novel method, namely fast direct method, is described. Extensive experiments are conducted in Section VI to demonstrate the efficiency of our method. Finally, some conclusions and future directions are made in Section VII.

II. SYSTEM MODEL

Consider the generalized linear model (GLM) with structured prior as shown in figure 1. The input unknown signal vector $\mathbf{x} \in \mathbb{R}^N$ is generated following a structured prior distribution $p_0(\mathbf{x})$. The signal vector $\mathbf{x}$ then passes through a linear transform

$$\mathbf{z} = \mathbf{A}\mathbf{x}, \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{M\times N}$ is the measurement matrix and is assumed to be known. The output observation vector $\mathbf{y}$ is obtained through a component-wise random mapping, which is described by a factorized conditional distribution

$$p(\mathbf{y}\,|\,\mathbf{z}) = \prod_{m=1}^{M} p(y_m \,|\, z_m) = \prod_{m=1}^{M} p\left(y_m \,\Big|\, \sum_{i=1}^{N} A_{mi}x_i \right). \tag{2}$$

The GLM arises in various problems in signal processing, communications, and machine learning. The classic linear Gaussian model,

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w}, \tag{3}$$

where $\mathbf{w} \in \mathbb{C}^M \sim \mathcal{N}\left(\mathbf{w}; 0, \sigma_n^2 \mathbf{I}_M\right)$ is a special case of GLM with $p(y_m \,|\, z_m) = \mathcal{N}\left(y_m; z_m, \sigma_n^2\right)$, where $\mathcal{N}(\mathbf{x}; \mathbf{m}, \mathbf{C})$ denotes the Gaussian distribution of $\mathbf{x}$ with mean $\mathbf{m}$ and covariance $\mathbf{C}$ and $\mathbf{I}_K$ denotes the identity matrix of dimension K. This paper only considers the linear Gaussian model (3). Extension to GLM is straightforward.

To model the structure of the sparsity pattern of signal $\mathbf{x}$, the authors in [24] proposed a structured spike and slab prior inspired by Gaussian processes [38]. Specifically, the hierarchical prior distribution of $\mathbf{x}$ reads

$$\begin{aligned} p(\mathbf{x}\,|\,\mathbf{s}) &= \prod_{i=1}^{N} p(x_i \,|\, s_i) \\ &= \prod_{i=1}^{N} (1-s_i)\delta(x_i) + s_i f(x_i), \end{aligned} \tag{4}$$

$$p(\mathbf{s}|\ \boldsymbol{\gamma}) = \prod_{i=1}^{N} \mathrm{Ber}\left(s_i \,|\, \phi\left(\gamma_i\right)\right), \tag{5}$$

$$p^{\mathrm{a}}(\boldsymbol{\gamma}) = \mathcal{N}\left(\boldsymbol{\gamma}; \hat{\boldsymbol{\gamma}}^{\mathrm{a}}, \boldsymbol{\Sigma}^{\mathrm{a}}\right), \tag{6}$$

where $\mathbf{s} \in \{0,1\}^N$ is the hidden support vector, $\delta(\cdot)$ is the Dirac delta function, $f(x_i)$ is the distribution of the nonzero entry x_i, $\phi(\cdot)$ is the standard Normal cumulative distribution function (CDF) i.e., $\phi(x) = \int_{-\infty}^{x} \mathcal{N}(t; 0, 1)\,dt$, $\mathrm{Ber}(s\,|\,p)$ denotes Bernoulli distribution function with $p(s=1\,|\,p) = p$, and $p^{\mathrm{a}}(\boldsymbol{\gamma})$ is the a priori probability of $\boldsymbol{\gamma}$. Furthermore, the prior covariance matrix $\boldsymbol{\Sigma}^{\mathrm{a}}$ is constructed using kernel functions, which further constitute a set of hyperparameters. In this paper, the squared exponential kernel function is taken as an example. That is, the (i,j)th element of $\boldsymbol{\Sigma}^{\mathrm{a}}$ is defined as

$$\left(\boldsymbol{\Sigma}^{\mathrm{a}}\right)_{ij} = \kappa \exp\left[-\frac{(i-j)^2}{2s^2} \right], \tag{7}$$

so that this kind of covariance matrix include hyperparameters κ and s. Due to the marginal characteristic of multivariate Gaussian distributions, $p^{\mathrm{a}}(\gamma_i) = \mathcal{N}\left(\gamma_i; \gamma_i^{\mathrm{a}}, \Sigma_i^{\mathrm{a}}\right)$, where γ_i^{a} and Σ_i^{a} are the ith element of $\boldsymbol{\gamma}^{\mathrm{a}}$ and the ith diagonal element of $\boldsymbol{\Sigma}^{\mathrm{a}}$, respectively. With $p^{\mathrm{a}}(\gamma_i)$, the marginal prior probability of x_i being active can be calculated as

$$\begin{aligned} p(s_i = 1) &= \int \mathrm{Ber}\left(s_i \,|\, \phi(\gamma_i)\right) p^{\mathrm{a}}(\gamma_i)\, d\gamma_i \\ &= \phi\left(\frac{\hat{\gamma}_i^{\mathrm{a}}}{\sqrt{1+\Sigma_i^{\mathrm{a}}}} \right) \end{aligned} \tag{8}$$

Note that the choice of $f(x_i)$ in (4) is flexible, which is an advantage of spike and slab prior. This paper only focuses on Gaussian distribution, i.e., $f(x_i) = \mathcal{N}(x_i; \hat{x}^{\mathrm{a}}, \tau^{\mathrm{a}})$. The structured spike and slab prior can encode prior information about the sparsity pattern. Specifically, the mean value vector $\hat{\boldsymbol{\gamma}}^{\mathrm{a}}$ controls the expected degree of sparsity while the covariance matrix $\boldsymbol{\Sigma}^{\mathrm{a}}$ determines the prior correlation of the support [24]–[26]. As in Gaussian processes, the covariance matrix $\boldsymbol{\Sigma}^{\mathrm{a}}$ can be constructed using various kernel functions such as radial basis function (RBF) [38]. The joint posterior distribution of $\mathbf{x}$, $\mathbf{s}$, and $\boldsymbol{\gamma}$ can be written as

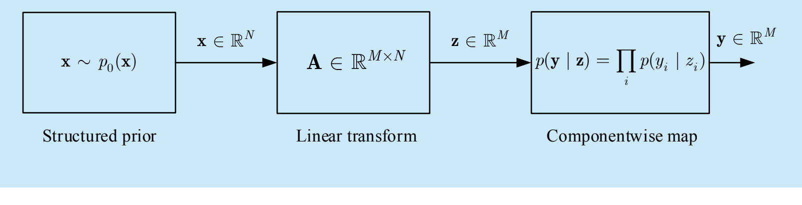

$\mathbf{x} \sim p_0(\mathbf{x})$
$\mathbf{x} \in \mathbb{R}^N$
$\mathbf{A} \in \mathbb{R}^{M\times N}$
$\mathbf{z} \in \mathbb{R}^M$
$p(\mathbf{y} \mid \mathbf{z}) = \prod_i p(y_i \mid z_i)$
$\mathbf{y} \in \mathbb{R}^M$
Structured prior
Linear transform
Componentwise map

Fig. 1. *Generalized linear model with structured prior [36].*

$$\begin{aligned} & p(\mathbf{x},\mathbf{s},\boldsymbol{\gamma}\mid\mathbf{y}) \\ & =\frac{1}{Z}p(\mathbf{y}\mid\mathbf{z}=\mathbf{A}\mathbf{x})\,p(\mathbf{x}\mid\mathbf{s})\,p(\mathbf{s}\mid\boldsymbol{\gamma})\,p^{\mathrm{a}}(\boldsymbol{\gamma}) \\ & =\frac{1}{Z}\underbrace{\prod_{m=1}^{M}\underbrace{p\left(y_m\mid\sum_{i=1}^{N}A_{mi}x_i\right)}_{f_{am}}}_{f_a}\underbrace{\prod_{i=1}^{N}\underbrace{\mathrm{Ber}\left(s_i\mid\phi(\gamma_i)\right)}_{f_{ci}}}_{f_c} \\ & \times\underbrace{\prod_{i=1}^{N}\underbrace{(1-s_i)\delta(x_i)+s_i\mathcal{N}(x_i;\mu_0,\tau_0)}_{f_{bi}}}_{f_b}\underbrace{p^{\mathrm{a}}(\boldsymbol{\gamma})}_{f_d} \end{aligned} \qquad (9)$$

where Z is the normalization constant.

The goal is to estimate $\mathbf{x}$ from the noisy observations $\mathbf{y}$ using the minimum mean square error (MMSE) criterion. It is well known that the MMSE estimate of x_i is the posterior mean, i.e., $\hat{x}_i=\int x_i p(x_i\mid\mathbf{y})dx_i$, where $p(x_i\mid\mathbf{y})$ is the marginal posterior distribution defined as

$$p(x_i\mid\mathbf{y})\ =\sum_{\mathbf{s}}\iint p(\mathbf{x},\mathbf{s},\boldsymbol{\gamma}\mid\mathbf{y})d\mathbf{x}_{\backslash i}d\boldsymbol{\gamma}, \qquad (10)$$

where $\mathbf{x}_{\backslash i}$ denotes all variables in $\mathbf{x}$ excluding x_i. Direct computation of (10) requires high-dimensional summations and integrals, rendering the complexity of exact calculation prohibitively high. The factorization in (9) can be explicitly encoded by a factor graph, one kind of undirected bipartite graph that connects the distribution factors in (9) with the random variables that constitute their arguments [16], as shown in figure 2. The round nodes denote the random variables while the square nodes denote the factors in (9). In fact, since the overall factor graph in figure 2 has loops, exact inference is NP-hard [39]. As such, we resort to approximate inference methods.

III. Efficient Reconstruction

3.1 AMP and GAMP

Before proceeding to deal with the case with structured priors, first take a look at the simple case with separable priors, i.e.,

$$p_0(\mathbf{x})=\prod_{i=1}^{N}p_0(x_i). \qquad (11)$$

As shown in figure 3, the factor graph of the joint distribution with separable priors is a subgraph of that with structured priors. Consequently, we can utilize the efficient algorithms such as AMP and GAMP algorithms to perform optimal reconstruction in the subgraph of figure 2. To deal with arbitrary separable priors, AMP has been extended to Bayesian AMP (B-AMP) [18], [40]. Specifically, our method resorts to the Bayesian form AMP, B-AMP, which is summarized in Algorithm 1. For de-

Algorithm 1. AMP [18], [40].

```
Input:  $\mathbf{y}\ \ \mathbf{A}\ \ \{p_0(x_i)\}_{i=1}^{N}$
Define:  $p(x\mid R,\Sigma)=\frac{p_0(x)\mathcal{N}(x;R,\Sigma)}{\int p_0(x)\mathcal{N}(x;R,\Sigma)dx}$
Define:  $g_a(R,\Sigma)=\int xp(x\mid R,\Sigma)dx$
Define:  $g_c(R,\Sigma)=\int|x|^2\ p(x\mid R,\Sigma)dx-|g_a(R,\Sigma)|^2$
Initialization: $\hat{x}_i^1=\int x_i p_0(x_i)dx$, $\mathrm{v}_i^1=\int|x_i-\hat{x}_i^1|^2\ p_0(x_i)dx$, $V_m^0=1, Z_m^0=y_m$, $l=1$.
for $i=1\rightarrow N$ do
    $V_m^l=\sum_i|A_{mi}|^2\ v_i^l$
    $Z_m^l=\sum_i A_{mi}\hat{x}_i^l-\frac{V_m^l}{\sigma^2+V_m^{l-1}}\left(y_m-Z_m^{l-1}\right)$
for $m=1\rightarrow N$ do
    $\Sigma_i^l=\frac{1}{\sum_{m=1}^{M}\frac{|A_{mi}|^2}{\sigma^2+V_m^l}}$
    $R_i^l=\hat{x}_i^t+\Sigma_i^t\sum_{m=1}^{M}\frac{A_{mi}(y_m-Z_m^t)}{\sigma^2+V_m^t}$
    $\hat{x}_i^{l+1}=g_a\left(R_i^l,\Sigma_i^l\right)$, $\hat{v}_i^{l+1}=g_c\left(R_i^l,\Sigma_i^l\right)$
end for
```

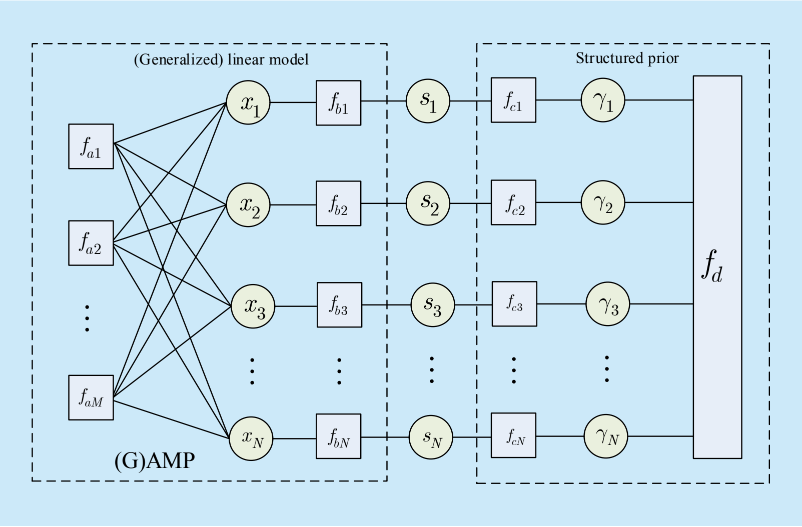


Fig. 2. *Factor graph of the joint distribution.*

tailed derivation, the readers are referred to [18], [40].

3.2 The turbo AMP approach

The AMP algorithm cannot be directly applied to reconstruct the structured signal of the form (4)-(6) due to its structured prior. To address this problem, this paper resorts to the Turbo approach proposed in [31], [32], [41]. In particular, the factor graph in figure 2 is divided into two subgraphs and then alternate between message passing within subgraphs and message passing between subgraphs in a turbo-like manner until they reach a common agreement, i.e., the iteration converges. Specifically, probabilistic beliefs of the hidden support elements **s** are exchanged between the two subgraphs in figure 2, the left one exploiting the observation model and the right one exploiting the structured spike and slab prior. One full round of alternation is designated as a "turbo iteration".

Denote by $\mu^t_{c\to s}(s_i)$ the message from factor node f_{ci} to variable node s_i at the *t*th turbo iteration while the message in the opposite direction is denoted by $\mu^t_{s\to c}(s_i)$. The other messages in factor graph shown in figure 2 follows the same notation. The messages $\mu^t_{c\to s}(s_i), i=1,\ldots,N$ can be viewed as soft estimates of the support elements and treated as priors for the left subgraph. Thanks to the decoupling characteristic of the turbo approach, these tentative priors can be seen as separable priors. As a result, the left subgraph reduces to the ordinary linear Gaussian model for which AMP can be applied. After convergence of AMP, the left subgraph yields messages $\mu^t_{b\to s}(s_i)$, which are then treated as priors for the right subgraph. Then, we update the soft estimates of the support elements using EP without inner iterations, as detailed in section 3.3. The resulting leftward message $\mu^{t+1}_{c\to s}(s_i)$ are treated as priors for AMP on the left subgraph at the next turbo iteration. This process continues until convergence or a maximum number of turbo iterations. The final estimate of signals $\mathbf{x}$ is the output of AMP in the last turbo iteration.

3.3 Message computation

This subsection addresses the computation of messages in the factor graph in a full round of single turbo iteration. Note that the message passing within the left subgraph is performed using AMP as described in Tab. I, so that our main focus in this section is on the message computation within the right subgraph and that between the two subgraphs. Specifically, at the *t*th turbo iteration, we assume that the message from factor node f_{di} to variable γ_i is $\mu^t_{d\to\gamma}(\gamma_i) = \mathcal{N}(\gamma_i; \hat{\gamma}^t_{d\to\gamma_i}, \Sigma^t_{d\to\gamma_i})$. Then, we will obtain the updated message $\mu^{t+1}_{d\to\gamma}(\gamma_i)$ in the following text, which is also demonstrated in Algorithm II.

At the start of turbo iteration, i.e., $t=1$, $\mu^t_{d\to\gamma}(\gamma_i)$ is initialized as the marginal probability of the joint prior $p^{\mathrm{a}}(\boldsymbol{\gamma})$ defined in (6), i.e., $\hat{\gamma}^1_{d\to\gamma_i} = \gamma^{\mathrm{a}}_i$, and $\Sigma^1_{d\to\gamma_i} = \Sigma^{\mathrm{a}}_i$. Since $\mu^t_{\gamma\to c}(\gamma_i) = \mu^t_{d\to\gamma}(\gamma_i)$, the message from f_{ci} to s_i is calculated as

$$m^t_{c\to s}(s_i=1) = \int \phi(\gamma_i) m^t_{d\to\gamma}(\gamma_i) d\gamma_i = \phi\left(\frac{\mu^t_{d\to\gamma_i}}{\sqrt{1+\Sigma^t_{d\to\gamma_i}}}\right), \quad (12)$$

$$m^t_{c\to s}(s_i=0) = (1-\phi(\gamma_i)) m^t_{d\to\gamma_i}(\gamma_i) d\gamma_i = 1-\phi\left(\frac{\mu^t_{d\to\gamma_i}}{\sqrt{1+\Sigma^t_{d\to\gamma_i}}}\right), \quad (13)$$

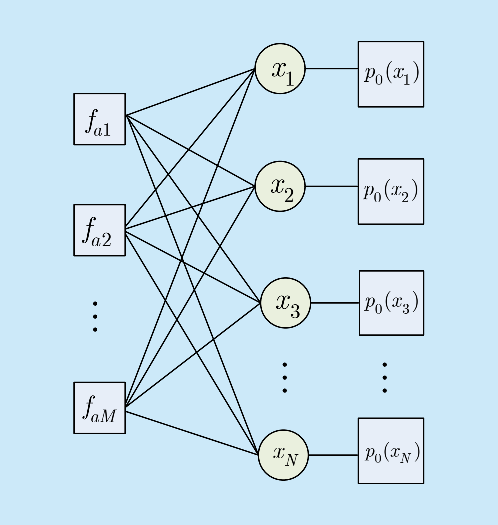


Fig. 3. *Factor graph of the joint distribution with separable prior.*

which implies that

$$m^t_{c\to s}\left(s_i\right) = \mathrm{Ber}\left(s_i \mid \phi\left(\frac{\mu^t_{d\to\gamma_i}}{\sqrt{1+\Sigma^t_{d\to\gamma_i}}}\right)\right). \quad (14)$$

From (12) and (13), and the message passing rule, $\mu^t_{s\to b}\left(s_i\right) = \mu^t_{c\to s}\left(s_i\right)$. Then, the message from f_b to x_i is obtained as

$$\begin{aligned}\mu^t_{b\to x}\left(x_i\right) &= \sum_{s_i} p\left(x_i \mid s_i\right)\mu^t_{s\to b}\left(s_i\right)\\ &= \left(1-\lambda_i^t\right)\delta\left(x_i\right) + \lambda_i^t \mathcal{N}\left(x_i;\hat{x}^{\mathrm{a}},\tau^{\mathrm{a}}\right),\end{aligned} \quad (15)$$

where

$$\lambda_i^t = \phi\left(\frac{\hat{\gamma}^t_{d\to\gamma_i}}{\sqrt{1+\Sigma^t_{d\to\gamma_i}}}\right). \quad (16)$$

At this step, the message from the right subgraph to the left subgraph has been accomplished. Taking $\mu^t_{b\to x_i}\left(x_i\right)$ as the prior distribution for x_i, i.e., $p^{\mathrm{a}}\left(x_i\right) = \mu^t_{b\to x}\left(x_i\right)$, the left subgraph is thus equivalent to a linear Gaussian model with separable priors, for which AMP can be directly applied. The detailed implementation of AMP within the left subgraph is illustrated in Tab. I and is omitted here, except for the definition of $g_a\left(R_i^l,\Sigma_i^l\right)$ and $g_c\left(R_i^l,\Sigma_i^l\right)$ at iteration t for the prior in (15). For notational brevity, the l iteration index within AMP is omitted. At the t th turbo iteration, for each inner iteration of AMP, the marginal posterior distribution of x_i becomes

$$\begin{aligned}&p^t\left(x_i \mid R_i,\Sigma_i\right)\\ &= \frac{\left(1-\lambda_i^t\right)\delta\left(x_i\right) + \lambda_i^t\mathcal{N}\left(x_i;\hat{x}^{\mathrm{a}},\tau^{\mathrm{a}}\right)}{Z\left(R_i,\Sigma_i\right)}\mathcal{N}\left(x_i;R_i,\Sigma_i\right)\\ &= \left(1-\pi_i^t\right)\delta\left(x_i\right) + \pi_i^t\mathcal{N}\left(x_i;m_i,V_i\right),\end{aligned} \quad (17)$$

where

$$\begin{aligned}&Z\left(R_i,\Sigma_i\right)\\ &= \int\left[\left(1-\lambda_i^t\right)\delta\left(x_i\right) + \lambda_i^t\mathcal{N}\left(x_i;\hat{x}^{\mathrm{a}},\tau^{\mathrm{a}}\right)\right]\mathcal{N}\left(x_i;R_i,\Sigma_i\right)dx_i\\ &= \left(1-\lambda_i^t\right)\mathcal{Z}_i^z + \lambda_i^t\mathcal{Z}_i^{nz},\end{aligned} \quad (18)$$

$$\mathcal{Z}_i^z = \frac{\exp\left(-\frac{R_i^2}{2\Sigma_i}\right)}{\sqrt{2\pi\Sigma_i}}, \quad (19)$$

$$\mathcal{Z}_i^{nz} = \frac{1}{\sqrt{2\pi\left(\Sigma_i+\tau^{\mathrm{a}}\right)}}\exp\left(-\frac{\left(R_i-\hat{x}\right)^2}{2\left(\Sigma_i+\tau^{\mathrm{a}}\right)}\right), \quad (20)$$

$$V_i = \frac{\tau^{\mathrm{a}}\Sigma_i}{\Sigma_i+\tau^{\mathrm{a}}}, \quad (21)$$

$$m_i = \frac{\tau^{\mathrm{a}}R_i + \Sigma_i\hat{x}^{\mathrm{a}}}{\Sigma_i+\tau^{\mathrm{a}}}, \quad (22)$$

$$\pi_i^t = \frac{\lambda_i^t\mathcal{Z}_i^{nz}}{\lambda_i^t\mathcal{Z}_i^{nz} + \left(1-\lambda_i^t\right)\mathcal{Z}_i^z}. \quad (23)$$

Note that the normalization constant $Z\left(R_i,\Sigma_i\right)$ is written in the form of two sub terms related to the zero support $\left(1-\lambda_i^t\right)\mathcal{Z}_i^z$ and the active support $\lambda_i^t\mathcal{Z}_i^{nz}$ as in [42]. Then, by definition, the posterior mean and variance are

$$g_a\left(R_i,\Sigma_i\right) = \frac{\lambda_i^t\mathcal{Z}_i^{nz}m_i}{Z\left(R_i,\Sigma_i\right)}, \quad (24)$$

$$g_c\left(R_i,\Sigma_i\right) = \frac{\lambda_i^t\mathcal{Z}_i^{nz}\left(m_i^2+V_i\right)}{Z\left(R_i,\Sigma_i\right)} - g_a^2\left(R_i,\Sigma_i\right). \quad (25)$$

To avoid potential numerical problem, it is better to rewrite

$$\frac{\mathcal{Z}_i^{nz}}{\left(1-\lambda_i^t\right)\mathcal{Z}_i^z + \lambda_i^t\mathcal{Z}_i^{nz}} = \frac{1}{\left(1-\lambda_i^t\right)\frac{\mathcal{Z}_i^z}{\mathcal{Z}_i^{nz}} + \lambda_i^t},$$

where

$$\begin{aligned}\ln\frac{\mathcal{Z}_i^{nz}}{\mathcal{Z}_i^z} &= \ln\frac{\exp\left(-\frac{\left(R_i-\mu_0\right)^2}{2\left(\Sigma_i+\tau^{\mathrm{a}}\right)}\right)}{\sqrt{2\pi\left(\Sigma_i+\tau^{\mathrm{a}}\right)}}\frac{\sqrt{2\pi\Sigma_i}}{\exp\left(-\frac{R_i^2}{2\Sigma_i}\right)}\\ &= \frac{1}{2}\ln\frac{\Sigma_i}{\Sigma_i+\tau^{\mathrm{a}}} + \frac{R_i^2}{2\Sigma_i} - \frac{\left(R_i-\hat{x}^{\mathrm{a}}\right)^2}{2\left(\Sigma_i+\tau^{\mathrm{a}}\right)}.\end{aligned}$$

Now we focus on the computation of messages $\mu^t_{b\to s}\left(s_i\right)$ from f_{bi} to s_i after convergence of AMP. From the marginal posterior distribution in (17), the posterior support probability is

$$p^{\mathrm{post}}\left(s_i=1\right) = \frac{\lambda_i^t\mathcal{Z}_i^{nz}}{Z\left(R_i,\Sigma_i\right)}, \quad (26)$$

$$p^{\mathrm{post}}\left(s_i=0\right) = \frac{\left(1-\lambda_i^t\right)\mathcal{Z}_i^z}{Z\left(R_i,\Sigma_i\right)}. \quad (27)$$

In the logarithmic domain, we have

$$\ln\frac{p^{\mathrm{post}}\left(s_i=1\right)}{p^{\mathrm{post}}\left(s_i=0\right)} = \underbrace{\ln\frac{\lambda_i^t}{\left(1-\lambda_i^t\right)}}_{prior} + \underbrace{\ln\frac{\mathcal{Z}_i^{nz}}{\mathcal{Z}_i^z}}_{extrinsic}. \quad (28)$$

From (26) and (27), note that given the prior support probability of $p(s_i) = \mathrm{Ber}(s_i \mid \lambda_i^t)$, the extrinsic information of the support s_i proposed by the left subgraph is

$$p^{\mathrm{ext}}(s_i) = \mathrm{Ber}(s_i \mid \eta_i^t), \tag{29}$$

where

$$\eta_i^t = \frac{\mathcal{Z}_i^{nz}}{\mathcal{Z}_i^{nz} + \mathcal{Z}_i^{z}}. \tag{30}$$

Therefore, the message from node f_{bi} to variable s_i is

$$\mu_{b \to s}^t(s_i) = \mathrm{Ber}(s_i \mid \eta_i^t). \tag{31}$$

To compute $\mu_{c \to \gamma}^t(\gamma_i)$, i.e., the message from f_{ci} to γ_i, the EP [28] method is resorted. Before proceeding to the detailed message computation, the definition of probability distribution projection operation induced by the Kullback-Leibler divergence is given. Mathematically, the projection of a particular distribution p into a distribution set Φ is defined as

$$\mathrm{Proj}_{\Phi}[p] = \arg\min_{q \in \Phi} D(p \,\|\, q), \tag{32}$$

where $D(p \,\|\, q)$ denotes the Kullback-Leibler divergence. If $p \in \Phi$, then the projection reduces to the identity mapping, i.e., $q = p$.

The joint posterior probability of s_i and γ_i is

$$\begin{aligned} p(s_i, \gamma_i) &\propto \mu_{b \to s}^t(s_i) f_{ci}(s_i, \gamma_i) \mu_{\gamma \to c}^t(\gamma_i) \\ &\propto \mathrm{Ber}(s_i \mid \eta_i^t) \mathrm{Ber}(s_i \mid \phi(\gamma_i)) \mathcal{N}(\gamma_i; \hat{\gamma}_{d \to \gamma_i}^t, \Sigma_{d \to \gamma_i}^t), \end{aligned} \tag{33}$$

where $\propto$ denotes identity between two distributions up to a normalization constant. The tentative marginal posterior probability of γ_i can be evaluated as

$$\begin{aligned} q^t(\gamma_i) = \frac{1}{Z_{\gamma_i}} \sum_{s_i} &\mathrm{Ber}(s_i \mid \eta_i^t) \mathrm{Ber}(s_i \mid \phi(\gamma_i)) \\ &\mathcal{N}(\gamma_i; \hat{\gamma}_{d \to \gamma_i}^t, \Sigma_{d \to \gamma_i}^t), \end{aligned} \tag{34}$$

where Z_{γ_i} is the normalization constant

$$\begin{aligned} &Z_{\gamma_i} \\ &= \int \sum_{s_i} \mathrm{Ber}(s_i \mid \eta_i^t) \mathrm{Ber}(s_i \mid \phi(\gamma_i)) \mathcal{N}(\gamma_i; \hat{\gamma}_{d \to \gamma_i}^t, \Sigma_{d \to \gamma_i}^t) d\gamma_i \\ &= \sum_{s_i} \int \mathrm{Ber}(s_i \mid \eta_i^t) \mathrm{Ber}(s_i \mid \phi(\gamma_i)) \mathcal{N}(\gamma_i; \hat{\gamma}_{d \to \gamma_i}^t, \Sigma_{d \to \gamma_i}^t) d\gamma_i \\ &= \eta_i^t \int \phi(\gamma_i) \mathcal{N}(\gamma_i; \hat{\gamma}_{d \to \gamma_i}^t, \Sigma_{d \to \gamma_i}^t) d\gamma_i \\ &\quad + (1 - \eta_i^t) \int (1 - \phi(\gamma_i)) \mathcal{N}(\gamma_i; \hat{\gamma}_{d \to \gamma_i}^t, \Sigma_{d \to \gamma_i}^t) d\gamma_i. \end{aligned}$$

For notational brevity, let us define

$$C_i^t \triangleq \int \phi(\gamma_i) \mathcal{N}(\gamma_i; \hat{\gamma}_{d \to \gamma_i}^t, \Sigma_{d \to \gamma_i}^t) d\gamma_i = \phi(\alpha_i^t), \tag{35}$$

$$\alpha_i^t \triangleq \frac{\hat{\gamma}_{d \to \gamma_i}^t}{\sqrt{1 + \Sigma_{d \to \gamma_i}^t}}. \tag{36}$$

Then, we have

$$Z_{\gamma_i} = \eta_i^t C_i^t + (1 - \eta_i^t)(1 - C_i^t). \tag{37}$$

According to the massage-passing update rule of EP [28], we get

$$\mu_{c \to \gamma}^t(\gamma_i) \propto \frac{\mathrm{Proj}_{\Phi}[q^t(\gamma_i)]}{\mu_{\gamma \to c}^t(\gamma_i)}. \tag{38}$$

Choosing the Gaussian distribution set Φ, then the projection operation $q = \mathrm{Proj}_{\Phi}[p]$ reduces to moment matching, i.e., the mean and variance are the same with respect to distribution p and q. So the mean and variance of γ_i with respect to $q^t(\gamma_i)$ need to be computed. Specifically, the mean can be evaluated by

$$\begin{aligned} \mathrm{E}_{q^t(\gamma_i)}(\gamma_i) = &\frac{\eta_i^t}{Z_{\gamma_i}} \int \gamma_i \phi(\gamma_i) \mathcal{N}(\gamma_i; \hat{\gamma}_{d \to \gamma_i}^t, \Sigma_{d \to \gamma_i}^t) d\gamma_i \\ &+ \frac{1 - \eta_i^t}{Z_{\gamma_i}} \int \gamma_i [1 - \phi(\gamma_i)] \mathcal{N}(\gamma_i; \hat{\gamma}_{d \to \gamma_i}^t, \Sigma_{d \to \gamma_i}^t) d\gamma_i. \end{aligned} \tag{39}$$

For notational brevity, we follow the derivation in [38] and define

$$\begin{aligned} D_i^t &\triangleq \int \gamma_i \phi(\gamma_i) \mathcal{N}(\gamma_i; \hat{\gamma}_{d \to \gamma_i}^t, \Sigma_{d \to \gamma_i}^t) d\gamma_i \\ &= C_i^t \hat{\gamma}_{d \to \gamma_i}^t + \frac{\Sigma_{d \to \gamma_i}^t \mathcal{N}(\alpha_i^t; 0, 1)}{\sqrt{1 + \Sigma_{d \to \gamma_i}^t}}, \end{aligned} \tag{40}$$

where C_i^t and α_i^t are defined as in (35) and (36), respectively. Thus

$$\mathrm{E}_{q^t(\gamma_i)}(\gamma_i) = \frac{1}{Z_{\gamma_i}} \left[\eta_i^t D_i^t + (1 - \eta_i^t)(\alpha_i^t - D_i^t) \right]. \tag{41}$$

To calculate the variance of γ_i, we first evaluate its second moment

$$\begin{aligned} \mathrm{E}_{q^t(\gamma_i)}(\gamma_i^2) = &\frac{\eta_i^t}{Z_{\gamma_i}} \int \gamma_i^2 \phi(\gamma_i) \mathcal{N}(\gamma_i; \hat{\gamma}_{d \to \gamma_i}^t, \Sigma_{d \to \gamma_i}^t) d\gamma_i \\ &+ \frac{1 - \eta_i^t}{Z_{\gamma_i}} \int \gamma_i^2 (1 - \phi(\gamma_i)) \mathcal{N}(\gamma_i; \hat{\gamma}_{d \to \gamma_i}^t, \Sigma_{d \to \gamma_i}^t) d\gamma_i. \end{aligned} \tag{42}$$

As in [38], the integral can be calculated as

$$
\begin{aligned}
W_i^t &= \int \gamma_i^2 \phi(\gamma_i) \mathcal{N}\left(\gamma_i; \hat{\gamma}_{d\to\gamma_i}^t, \Sigma_{d\to\gamma_i}^t\right) d\gamma_i \\
&= 2\hat{\gamma}_{d\to\gamma_i}^t D_i^t + C_i^t \left(\Sigma_i^t - \left(\hat{\gamma}_{d\to\gamma_i}^t\right)^2\right) \\
&\quad - \frac{\left(\Sigma_{d\to\gamma_i}^t\right)^2 \alpha_i^t \mathcal{N}\left(\alpha_i^t; 0, 1\right)}{1 + \Sigma_{d\to\gamma_i}^t},
\end{aligned}
\tag{43}
$$

so that

$$
\mathrm{E}_{q^t(\gamma_i)}\left(\gamma_i^2\right) = \frac{\eta_i W_i^t + (1-\eta_i)\left(\left(\hat{\gamma}_{d\to\gamma_i}^t\right)^2 + \Sigma_{d\to\gamma_i}^t - W_i^t\right)}{Z_{\gamma_i}}.
\tag{44}
$$

Therefore, the posterior variance of γ_i can be easily calculated as

$$
\mathrm{Var}_{q^t(\gamma_i)}\left(\gamma_i\right) = \mathrm{E}_{q^t(\gamma_i)}\left(\gamma_i^2\right) - \mathrm{E}_{q^t(\gamma_i)}^2\left(\gamma_\mathrm{i}\right).
\tag{45}
$$

Having derived the posterior mean and variance of γ_i, the message from factor node f_{ci} to variable node γ_i is updated as

$$
\begin{aligned}
\mu_{c\to\gamma_i}^t\left(\gamma_i\right) &\propto \frac{\mathcal{N}\left(\gamma_i; \mathrm{E}_{q^t(\gamma_i)}\left(\gamma_i\right), \mathrm{Var}_{q^t(\gamma_i)}\left(\gamma_i^2\right)\right)}{\mathcal{N}\left(\gamma_i; \hat{\gamma}_{d\to\gamma_i}^t, \Sigma_{d\to\gamma_i}^t\right)} \\
&\propto \mathcal{N}\left(\gamma_i; \hat{\gamma}_{c\to\gamma_i}^t, \Sigma_{c\to\gamma_i}^t\right),
\end{aligned}
\tag{46}
$$

where

$$
\frac{1}{\Sigma_{c\to\gamma_i}^t} = \frac{1}{\mathrm{Var}_{q^t(\gamma_i)}\left(\gamma_i\right)} - \frac{1}{\Sigma_{d\to\gamma_i}^t},
\tag{47}
$$

$$
\hat{\gamma}_{c\to\gamma_i}^t = \Sigma_{c\to\gamma_i}^t \left(\frac{\mathrm{E}_{q^t(\gamma_i)}\left(\gamma_i\right)}{\mathrm{Var}_{q^t(\gamma_i)}\left(\gamma_i\right)} - \frac{\hat{\gamma}_{d\to\gamma_i}^t}{\Sigma_{d\to\gamma_i}^t}\right).
\tag{48}
$$

Then, combining all the messages from factor node f_{ci} to $\gamma_i, i = 1, \ldots, N$, and the Gaussian process prior (6), the updated posterior distribution of $\boldsymbol{\gamma}$ at the $t+1$ th iteration is obtained as

$$
\begin{aligned}
q^{t+1}\left(\boldsymbol{\gamma}\right) &\propto \mathcal{N}\left(\boldsymbol{\gamma}; \hat{\boldsymbol{\gamma}}^\mathrm{a}, \boldsymbol{\Sigma}^\mathrm{a}\right) \prod_{i=1}^N \mu_{c\to\gamma_i}^t\left(\gamma_i\right) \\
&\propto \mathcal{N}\left(\boldsymbol{\gamma}; \hat{\boldsymbol{\gamma}}^\mathrm{a}, \boldsymbol{\Sigma}^\mathrm{a}\right) \mathcal{N}\left(\boldsymbol{\gamma}; \hat{\boldsymbol{\gamma}}_{c\to\gamma}^t, \boldsymbol{\Sigma}_t^t\right) \\
&\propto \mathcal{N}\left(\boldsymbol{\gamma}; \hat{\boldsymbol{\gamma}}^{t+1}, \boldsymbol{\Sigma}^{t+1}\right),
\end{aligned}
\tag{49}
$$

where,

$$
\boldsymbol{\Sigma}^{t+1} = \left(\left(\boldsymbol{\Sigma}^\mathrm{a}\right)^{-1} + \left(\boldsymbol{\Sigma}_{c\to\gamma}^t\right)^{-1}\right)^{-1},
\tag{50}
$$

$$
\hat{\boldsymbol{\gamma}}^{t+1} = \boldsymbol{\Sigma}^{t+1}\left(\left(\boldsymbol{\Sigma}^\mathrm{a}\right)^{-1} \hat{\boldsymbol{\gamma}}^\mathrm{a} + \left(\boldsymbol{\Sigma}_{c\to\gamma}^t\right)^{-1} \hat{\boldsymbol{\gamma}}_{c\to\gamma}^t\right),
\tag{51}
$$

and $\boldsymbol{\Sigma}_{c\to\gamma}^t = \mathrm{diag}\left(\Sigma_{c\to\gamma_1}^t, \ldots, \Sigma_{c\to\gamma_N}^t\right)$ and $\hat{\boldsymbol{\gamma}}_{c\to\gamma}^t = \left(\hat{\gamma}_{c\to\gamma_1}^t, \ldots, \hat{\gamma}_{c\to\gamma_N}^t\right)^\mathrm{T}$.

Finally, the message from factor node f_d to variable node γ_i at the $(t+1)$ th iteration is updated as

$$
\begin{aligned}
\mu_{d\to\gamma}^{t+1}\left(\gamma_i\right) &\propto \frac{\mathcal{N}\left(\gamma_i; \hat{\gamma}_i^{t+1}, \Sigma_i^{t+1}\right)}{\mathcal{N}\left(\gamma_i; \hat{\gamma}_{c\to\gamma_i}^t, \Sigma_{c\to\gamma_i}^t\right)} \\
&\propto \mathcal{N}\left(\gamma_i; \hat{\gamma}_{d\to\gamma_i}^{t+1}, \Sigma_{d\to\gamma_i}^{t+1}\right),
\end{aligned}
\tag{52}
$$

where

$$
\Sigma_{d\to\gamma_i}^{t+1} = \left(\frac{1}{\Sigma_i^{t+1}} - \frac{1}{\Sigma_{c\to\gamma_i}^t}\right)^{-1},
\tag{53}
$$

$$
\hat{\gamma}_{d\to\gamma_i}^{t+1} = \Sigma_{d\to\gamma_i}^{t+1}\left(\frac{\hat{\gamma}_i^{t+1}}{\Sigma_i^{t+1}} - \frac{\mu_{c\to\gamma_i}^t}{\Sigma_{c\to\gamma_i}^t}\right),
\tag{54}
$$

and $\hat{\gamma}_i^{t+1}$, Σ_i^{t+1} are the ith element of $\hat{\boldsymbol{\gamma}}^{t+1}$ and the ith diagonal element of $\boldsymbol{\Sigma}^{t+1}$, respectively.

IV. Learning of Hyperparameters

In practice, since the prior parameters that encode prior distribution of $\boldsymbol{\gamma}$ are unknown, they need to be learned from data. In the sequel, expectation maximization (EM) algorithm is utilized to learn these hyperparameters.

The hidden variables are chosen as $\mathbf{x}$, $\mathbf{s}$, and $\boldsymbol{\gamma}$. let $\boldsymbol{\theta} = \{\boldsymbol{\sigma}_{\boldsymbol{n}}^2, \boldsymbol{\gamma}^\mathrm{a}, \boldsymbol{\Sigma}^\mathrm{a}, \hat{\boldsymbol{x}}^\mathrm{a}, \boldsymbol{\tau}^\mathrm{a}\}$ denote hyperparameters and $\boldsymbol{\theta}^t$ denote the estimation at the tth iteration. The EM algorithm alternates between the following two steps:

$$
Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^t\right) = \mathrm{E}\left\{\ln p\left(\mathbf{x}, \mathbf{s}, \boldsymbol{\gamma}, \mathbf{y}\right) \mid \mathbf{y}; \boldsymbol{\theta}^t\right\},
\tag{55}
$$

$$
\boldsymbol{\theta}^{t+1} = \arg\max_{\theta} Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^t\right),
\tag{56}
$$

where $\mathrm{E}\{\cdot \mid \mathbf{y}; \boldsymbol{\theta}^t\}$ denotes expectation conditioned on observations $\mathbf{y}$ under parameters $\boldsymbol{\theta}^t$, i.e., the expectation is with respect to the posterior conditional distribution $p\left(\mathbf{x}, \mathbf{s}, \boldsymbol{\gamma} \mid \mathbf{y}, \boldsymbol{\theta}^t\right)$. The exact computation of $p\left(\mathbf{x}, \mathbf{s}, \boldsymbol{\gamma} \mid \mathbf{y}, \boldsymbol{\theta}^t\right)$ is intractable in practice. Fortunately, the Turbo AMP framework offers an efficient approximation, whereby the E step in (55) can be readily calculated as

$$
\begin{aligned}
&Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^t\right) \\
&= \mathrm{E}\left\{\ln p\left(\mathbf{y} \mid \mathbf{z}\right) p\left(\mathbf{x} \mid \mathbf{s}\right) p\left(\mathbf{s} \mid \boldsymbol{\gamma}\right) p^\mathrm{a}\left(\boldsymbol{\gamma}\right) \mid \mathbf{y}; \boldsymbol{\theta}^t\right\} \\
&= \mathrm{E}\Bigg\{\sum_{a=1}^M \ln p\left(y_a \mid z_a\right) + \sum_{i=1}^N \ln p\left(x_i \mid s_i\right) + \sum_{i=1}^N \ln p\left(s_i \mid \gamma_i\right) \\
&\quad - \frac{1}{2}\left(\boldsymbol{\gamma} - \hat{\boldsymbol{\gamma}}^\mathrm{a}\right)^\mathrm{T} \left(\boldsymbol{\Sigma}^\mathrm{a}\right)^{-1} \left(\boldsymbol{\gamma} - \hat{\boldsymbol{\gamma}}^\mathrm{a}\right) - \frac{1}{2}\ln\left(\det \boldsymbol{\Sigma}^\mathrm{a}\right) \mid \mathbf{y}; \boldsymbol{\theta}^t\Bigg\}.
\end{aligned}
\tag{57}
$$

The joint optimization of $\boldsymbol{\theta}$ is impractical and thus the incremental EM update rule is adopted, i.e., one element is updated at a time while holding the other parameters fixed.

4.1 Learning noise variance in linear Gaussian case

In this case, $p\left(y_m \mid z_m\right)=\mathcal{N}\left(y_m;\sum_i A_{mi}x_i,\sigma_n^2\right)$, the update of σ_n^2 follows

$$\left(\sigma_n^2\right)^{t+1}=\arg\max_{\sigma_n^2>0} Q\left(\boldsymbol{\theta},\boldsymbol{\theta}^t\right). \tag{58}$$

Maximizing $Q\left(\boldsymbol{\theta},\boldsymbol{\theta}^t\right)$ with respect to σ_n^2 results in

$$\left(\sigma_n^2\right)^{t+1}=\frac{1}{M}\sum_{m=1}^{M}\left[\left(y_m-\sum_i A_{mi}\hat{x}_i^t\right)^2+V_m^t\right]. \tag{59}$$

Since $p\left(y_m \mid z_m\right)=\mathcal{N}\left(y_m;z_m,\sigma_n^2\right)$, the update equation (59) can be rewritten in the form of statistics of z_m, which can be calculated to be

$$\left(\sigma_n^2\right)^{t+1}=\frac{1}{M}\sum_{m=1}^{M}\left[\left(y_m-\hat{z}_m^t\right)^2+\left(v_m^z\right)^t\right], \tag{60}$$

where $\hat{z}_m^t$ and $\left(v_m^z\right)^t$ in the linear Gaussian case are calculated to be

$$\hat{z}_m^t=\frac{V_m^t y_m+\left(\sigma_n^2\right)^t Z_m^t}{\left(\sigma_n^2\right)^t+V_m^t}, \tag{61}$$

$$\left(v_m^z\right)^t=\frac{V_m^t\left(\sigma_n^2\right)^t}{\left(\sigma_n^2\right)^t+V_m^t}, \tag{62}$$

where Z_m^t and V_m^t are the values of Z_m^l and V_m^l in Tab. I after the *t*th turbo iteration. It can be noted that using statistics of z_m leads to recursive update of noise variance. The update equation (60) is used in Section VI.

4.2 Learning $\hat{x}^{\mathrm{a}}$ and τ^{a}

Following the derivation in [14], $Q\left(\boldsymbol{\theta},\boldsymbol{\theta}^t\right)$ is maximized with respect to $\hat{x}^{\mathrm{a}}$ and τ^{a} and the update equations are

$$\hat{x}^{\mathrm{a},t+1}=\frac{\sum_i \pi_i^t m_i}{\sum_i \pi_i^t}, \tag{63}$$

$$\tau^{\mathrm{a},t+1}=\frac{1}{\sum_i \pi_i^t}\sum_i \pi_i^t\left[\left(\hat{x}^{\mathrm{a},t}-m_i\right)^2+V_i\right]. \tag{64}$$

4.3 Learning $\boldsymbol{\gamma}^{\mathrm{a}}$

For $\boldsymbol{\gamma}^{\mathrm{a}}$, the EM update is

$$\boldsymbol{\gamma}^{\mathrm{a},t+1}=\arg\max_{\boldsymbol{\gamma}^{\mathrm{a}}} Q\left(\boldsymbol{\theta},\boldsymbol{\theta}^t\right) \tag{65}$$

Setting the derivative of $Q\left(\boldsymbol{\theta},\boldsymbol{\theta}^t\right)$ with respect to $\boldsymbol{\gamma}^{\mathrm{a}}$ to zero results in

$$\frac{\partial}{\partial\boldsymbol{\gamma}^{\mathrm{a}}}Q\left(\boldsymbol{\theta},\boldsymbol{\theta}^t\right)=\mathrm{E}\left\{\left(\boldsymbol{\Sigma}^{\mathrm{a}}\right)^{-1}\left(\boldsymbol{\gamma}-\boldsymbol{\gamma}^{\mathrm{a}}\right)\mid \mathbf{y};\boldsymbol{\theta}^t\right\}=0. \tag{66}$$

so that $\boldsymbol{\gamma}^{\mathrm{a}}$ is updated as

$$\boldsymbol{\gamma}^{\mathrm{a},t+1}=\mathrm{E}_{q^t(\boldsymbol{\gamma})}\left(\boldsymbol{\gamma}\right) \tag{67}$$

where $q^t\left(\boldsymbol{\gamma}\right)=\sum_i q^t\left(\gamma_i\right)$ and $q^t\left(\gamma_i\right)$ is defined in (34).

4.4 Learning hyperparameters of $\boldsymbol{\Sigma}^{\mathrm{a}}$

To learn these parameters of $\boldsymbol{\Sigma}^{\mathrm{a}}$ defined by (7), the EM update can be also applied. However, the computation is a bit tedious. By defining

$$\left(\boldsymbol{\Sigma}\right)_{ij}=\exp\left[-\frac{(i-j)^2}{2s^2}\right], \tag{68}$$

we have $\boldsymbol{\Sigma}^{\mathrm{a}}=\kappa\Sigma$ from (7) and

$$\begin{aligned}&Q\left(\boldsymbol{\theta},\boldsymbol{\theta}^t\right)\\&=\mathrm{E}\left\{\sum_{a=1}^{M}\ln p\left(y_a\mid z_a\right)+\sum_{i=1}^{N}\ln p\left(x_i\mid s_i\right)+\sum_{i=1}^{N}\ln p\left(s_i\mid\gamma_i\right)\right.\\&\left.-\frac{1}{2\kappa}\left(\boldsymbol{\gamma}-\boldsymbol{\gamma}^{\mathrm{a}}\right)^{\mathrm{T}}\left(\boldsymbol{\Sigma}^{\mathrm{a}}\right)^{-1}\left(\boldsymbol{\gamma}-\boldsymbol{\gamma}^{\mathrm{a}}\right)-\frac{1}{2}\ln\left(\kappa^N\det\boldsymbol{\Sigma}\right)\mid\mathbf{y};\boldsymbol{\theta}^t\right\}.\end{aligned} \tag{69}$$

Setting the derivative of $Q\left(\boldsymbol{\theta},\boldsymbol{\theta}^t\right)$ with respect to κ to zero results in

$$\begin{aligned}&\frac{\partial}{\partial\kappa}Q\left(\boldsymbol{\theta},\boldsymbol{\theta}^t\right)\\&=\mathrm{E}\left\{\frac{1}{2\kappa^2}\left(\boldsymbol{\gamma}-\boldsymbol{\gamma}^{\mathrm{a}}\right)^{\mathrm{T}}\left(\boldsymbol{\Sigma}^{\mathrm{a}}\right)^{-1}\left(\boldsymbol{\gamma}-\boldsymbol{\gamma}^{\mathrm{a}}\right)-\frac{N}{2\kappa}\mid\mathbf{y};\boldsymbol{\theta}^t\right\}\\&=0,\end{aligned} \tag{70}$$

After some algebra, we get

$$\begin{aligned}\kappa^{t+1}&=\frac{1}{N}\mathrm{E}\left\{\left(\boldsymbol{\gamma}-\boldsymbol{\gamma}^{\mathrm{a}}\right)^{\mathrm{T}}\left(\boldsymbol{\Sigma}^{\mathrm{a}}\right)^{-1}\left(\boldsymbol{\gamma}-\boldsymbol{\gamma}^{\mathrm{a}}\right)\mid\mathbf{y};\boldsymbol{\theta}^t\right\}\\&=\frac{1}{N}\mathrm{tr}\left\{\left(\boldsymbol{\Sigma}^{\mathrm{a}}\right)^{-1}\mathrm{E}\left[\left(\boldsymbol{\gamma}-\boldsymbol{\gamma}^{\mathrm{a}}\right)\left(\boldsymbol{\gamma}-\boldsymbol{\gamma}^{\mathrm{a}}\right)^{T}\mid\mathbf{y};\boldsymbol{\theta}^t\right]\right\}\\&=\frac{1}{N}\sum_{i=1}^{N}\xi_{ii}\mathrm{Var}_{q^t(\gamma_i)}\left(\gamma_i\right),\end{aligned} \tag{71}$$

where ξ_{ij} is the $\left(i,j\right)$ th element of $\boldsymbol{\Sigma}^{-1}$ and the last equality is due to the update of $\boldsymbol{\gamma}^{\mathrm{a}}$ in (67).

As for the learning of length-scale parameter s, however, there is no closed form update equations. To address this problem, a free energy view is adopted in optimizing s. Following similar derivation in [35], the Bethe free energy after the tth turbo iteration can be calculated as

$$\begin{aligned} F_{\text{Bethe}}^{t}(s) = & \sum_{m=1}^{M} \frac{\left(y_m - \sum_{i=1}^{N} A_{mi} \hat{x}_i^t \right)^2}{2\left(\sigma_n^2\right)^t} + \frac{M}{2} \ln 2\pi \left(\sigma_n^2\right)^t \\ & + \sum_{m=1}^{M} \frac{1}{2} \ln \left(1 + \frac{1}{\left(\sigma_n^2\right)^t} \sum_{i=1}^{N} A_{mi}^2 \hat{v}_i^t \right) \\ & - \sum_{i=1}^{N} \left(\ln Z\left(R_i, \Sigma_i\right) + \frac{\hat{v}_i^t + \left(\hat{x}_i^t - R_i\right)^2}{2\Sigma_i} \right), \end{aligned} \tag{72}$$

where the values $\hat{x}_i^t, \hat{v}_i^t, R_i, \Sigma_i, \left(\sigma_n^2\right)^t$ are obtained from AMP-SSS (refer to Algorithm 1 and Algorithm 2). At first glance, it seems that the Bethe free energy is not related to the parameters. In fact, $F_{\text{Bethe}}^{t}(s)$ depends on s implicitly through $\hat{x}_i^t, \hat{v}_i^t, R_i, \Sigma_i, \left(\sigma_n^2\right)^t$. For each values of s, the associated $F_{\text{Bethe}}^{t}(s)$ is obtained as in (72) and the optimal s corresponds to the minimum free energy, i.e.,

$$\hat{s} = \arg\max_{s} F_{\text{Bethe}}^{t}(s). \tag{73}$$

Though the search space for s is huge, numerical results show that the performance is insensitive to s. As a result, the search space can be restricted to a small set with typical values. Denote by $\mathbb{S}$ the search space set. It is empirically demonstrated that $\mathbb{S} = \{10^{-3}, 10^{-2}, 1, 10, 50, 100, 500\}$ performs quite well most of the time and is used in Section VI.

Algorithm 2. AMP with structured spike and slab priors (AMP-SSS).

```
Initialization: $\mu_{d\to\gamma_i}^1 = \mu_{0,i}, \Sigma_{d\to\gamma_i}^1 = \Sigma_{0,ii}$, $i = 1, \ldots, N$, $\mathbf{x}^0 = 0$.
for $t = 1, \ldots, T_{\max}$ do
    Compute $\lambda_i^t$ via (16);
    Perform AMP within the left subgraph;
    Compute $\mathcal{Z}_i^z$, $\mathcal{Z}_i^{nz}$ and $\eta_i^t$ via (19), (20) and (30);
    Compute $\Sigma_{c\to\gamma_i}^t$ and $\mu_{c\to\gamma_i}^t$ via (47) and (48); Compute $\Sigma_p^{t+1}$ and $\Sigma_p^{t+1}$ via (50) and (51);
    Update $\Sigma_{d\to\gamma_i}^{t+1}$ and $\mu_{d\to\gamma_i}^{t+1}$ via (53) and (54).
    If $\| \mathbf{x}^{t+1} - \mathbf{x}^t \| / \| \mathbf{x}^t \| < \epsilon_{\text{turbo}}$, break;
end for
```

V. Complexity Reduction via Fast Direct Method

In this section, the computation complexity of the proposed algorithm is analyzed. The turbo message schedule implies that the complexity consists of two parts: AMP operation and update of soft support by the structured prior. The complexity of AMP is $\mathcal{O}(MN)$ [18], [36]. However, the update of soft support needs matrix inversions as shown in (50) and (51), whose direct solution scales as $\mathcal{O}\left(N^3\right)$, which is prohibitively high in large-scale setting. To make this algorithm applicable in high-dimensional settings, the complexity needs to be further reduced. Note that (50) and (51) can be rewritten as

$$\mathbf{\Sigma}^{t+1} = \mathbf{\Sigma}_{c\to\gamma}^{t} \left(\mathbf{\Sigma}^{\text{a}} + \mathbf{\Sigma}_{c\to\gamma}^{t} \right)^{-1} \mathbf{\Sigma}^{\text{a}}, \tag{74}$$

$$\begin{aligned} \hat{\boldsymbol{\gamma}}^{t+1} = & \mathbf{\Sigma}_{c\to\gamma} \left(\mathbf{\Sigma}^{\text{a}} + \mathbf{\Sigma}_{c\to\gamma}^{t} \right)^{-1} \hat{\boldsymbol{\gamma}}^{\text{a}} \\ & + \mathbf{\Sigma}^{\text{a}} \left(\mathbf{\Sigma}^{\text{a}} + \mathbf{\Sigma}_{c\to\gamma}^{t} \right)^{-1} \hat{\boldsymbol{\gamma}}_{c\to\gamma}^{t}. \end{aligned} \tag{75}$$

Therefore, the computational bottleneck lies in the inversion of matrix $\mathbf{\Sigma}^{\text{a}} + \mathbf{\Sigma}_{c\to\gamma}^{t}$. In [43], a kind of fast direct method is proposed to compute the inversion of matrix of the form $\mathbf{C} = \mathbf{D} + \mathbf{K}$ with complexity $\mathcal{O}\left(N \log_2 N\right)$, where $\mathbf{D}$ is a general diagonal matrix with non-negative constants just like $\mathbf{\Sigma}_{c\to\gamma}^{t}$, $\mathbf{K}$ is computed using a specified covariance kernel. Since $\mathbf{\Sigma}_{c\to\gamma}^{t}$ is a diagonal matrix, and that only the diagonal terms of $\mathbf{\Sigma}_{p}^{t+1}$ need to be computed, the complexity of (74) scales as $\mathcal{O}\left(N^2\right)$. The matrix-by-matrix product in (75) need not be computed directly. Instead, do the matrix-by-vector multiplications first, so that the complexity in (75) is $\mathcal{O}\left(N^2\right)$ as well. Therefore, the overall complexity of the proposed algorithm then reduces to $\mathcal{O}\left(MN + N^2\right)$ ignoring $N \log_2 N$ term per iteration.

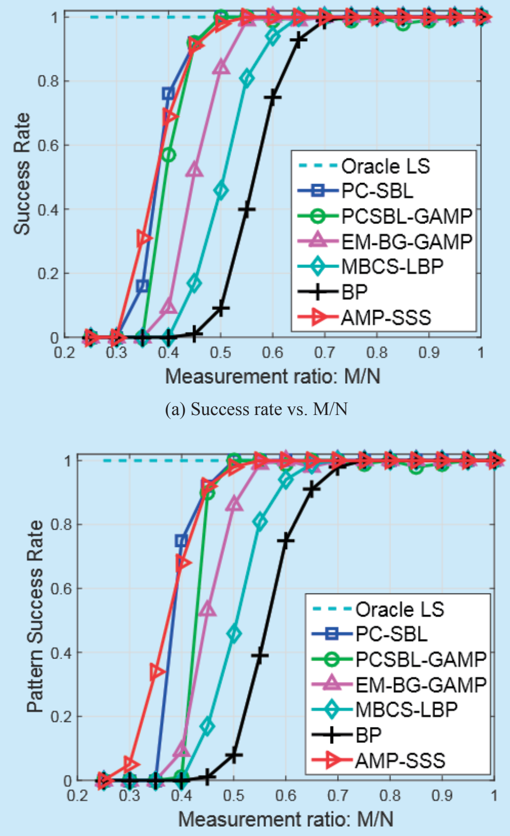
VI. Results and Discussion

In this section, a series of numerical experiments are conducted to investigate the efficiency of the proposed algorithm, referred to as AMP-SSS. Comparisons are made to algorithms without consideration on structures, such as Basis Pursuit (BP) [44], and EM-BG-GAMP [14], as well as some state-of-the-art methods without prior knowledge of the sparsity pattern but taking into account structures, e.g., MBCS-LBP [27], PC-SBL [45] and its AMP version PCSBL-GAMP [46]. The performance of oracle least squares estimator (Oracle LS) which knows the true support is given as the benchmark. Note that since the proposed AMP-SSS requires no prior knowledge of the sparsity pattern, e.g., sparse ratio, number of groups, etc., no comparisons are to those algorithms that need partial or full knowledge of the sparsity pattern.

Throughout the experiments, no prior knowledge of the sparsity pattern, e.g., sparsity ratio, number of nonzero groups, is known except for Oracle LS. The maximum number of iterations for PCSBL-GAMP, and EM-BG-GAMP is set to be $T_{\max} = 200$, and the tolerance value of termination is $\epsilon_{\text{toc}} = 10^{-6}$. For AMP-SSS, there are two iterative loops. The inner maximum number of AMP iterations is set to be $L_{\max} = 50$ and the outer maximum number of turbo iterations is set to be $T_{\max} = 50$. The tolerance values for the inner AMP and outer turbo iterations are set to be $\epsilon_{\text{amp}} = 10^{-6}$ and $\epsilon_{\text{turbo}} = 10^{-6}$, respectively. To avoid divergence, the damping technique is used for AMP-SSS, and the damping factor is set to be 0.3. For other algorithms, the default settings are used. The elements of measurement matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$ are independently generated following standard Gaussian distribution and the columns are normalized to unit norm. The success rate is defined as the ratio of the number of successful trials to the total number of experiments, where a trial is successful if the normalized mean square error (NMSE) is less than -50 dB, where $\text{NMSE} = 20\log_{10}\left(\|\hat{\mathbf{x}} - \mathbf{x}\|_2 / \|\mathbf{x}\|_2\right)$ with $\hat{\mathbf{x}}$ being the recovered signal. The pattern recovery success rate is defined as the ratio of the number of successful trials to the total number of experiments, where a trial is successful if the support is exactly recovered. A coefficient whose magnitude is less than 10^{-4} is deemed as a zero coefficient.

6.1 Sparse gaussian data

Synthetic block-sparse signals are generated in a similar way as [45], [47], where K nonzero elements are partitioned into L blocks with random sizes and random locations. Set N = 100, K = 25, L = 4 and the nonzero elements are generated independently following Gaussian distribution with mean $\mu_0 = 3$ and variance $\tau_0 = 1$. The results are averaged over 100 independent runs. Figure 4(a) and figure 4(b) depict the success rate and pattern recovery



(a) Success rate vs. M/N

(b) Pattern success rate vs. M/N

Fig. 4. *Block sparse Gaussian signal reconstruction, noiseless case.*

success rate, respectively. It can be seen that in both scenarios, AMP-SSS performs nearly the same as PC-SBL and slightly better than PCSBLGAMP. In the noisy setting, figure 5 shows the average NMSE of different algorithms when the signal to noise ratio (SNR) is 50 dB, where $SNR = 20\log_{10}\left(\|\mathbf{Ax}\|_2 / \|\mathbf{w}\|_2\right)$. Note that AMP-SSS and PC-SBL outperform other methods both in terms of NMSE in the noisy case.

6.2 Sparse binary data

The synthetic block-sparse binary signals, i.e., the elements are either 0 or 1, are generated in a similar way as the sparse Gaussian case. Set N = 100, K = 25, L = 4. The results are averaged over 100 independent runs.

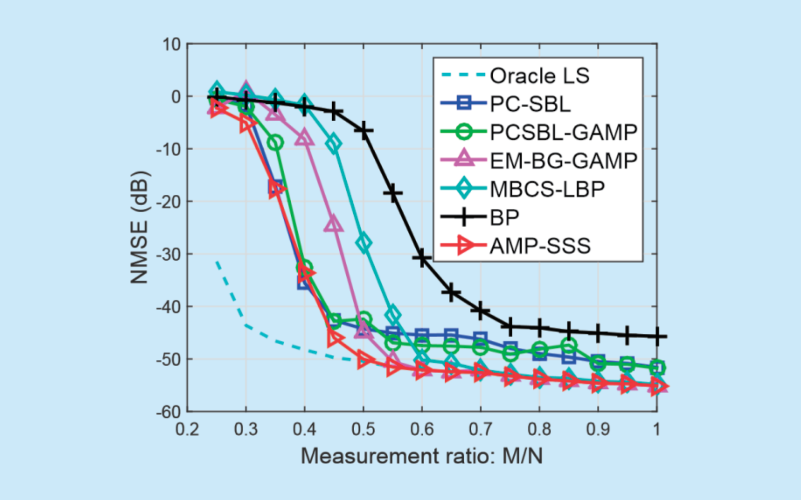


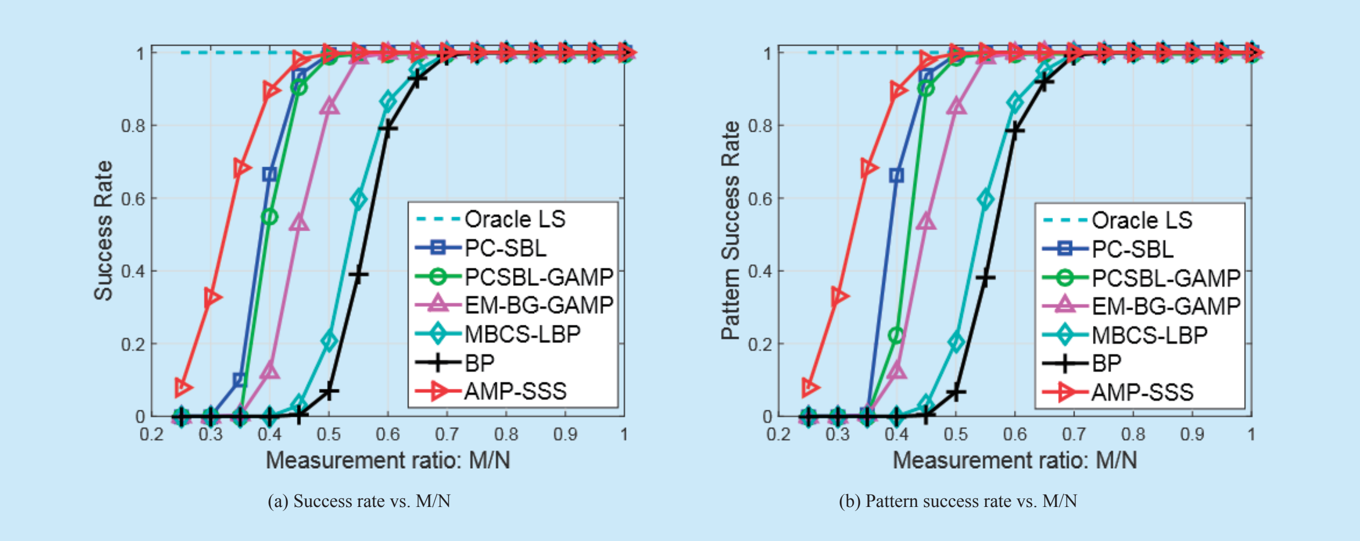
Fig. 5. *NMSE vs. M/N for block sparse Gaussian signals, SNR = 50 dB.*

Figure 6(a) and figure 6(b) depict the success rate and pattern recovery success rate, respectively. It can be seen that AMP-SSS achieves the highest success rate and pattern recovery rate at various measurement ratios, implying that the proposed AMP-SSS is very robust to the true prior of the nonzero elements. In addition, compared with PC-SBL and PCSBL-GAMP, AMP-SSS has much more flexibility to encode the distribution of nonzero elements by specifying various distributions on the slab part.

In the noisy setting, figure 7 shows the average NMSE of different algorithms when the signal to noise ratio (SNR) is 50 dB. Note that AMP-SSS outperforms the other algorithms apparently in terms of average NMSE. Note that for measurement ratios higher than 0.5 in the noisy setting, both AMP-SSS and EM-BG-GAMP outperform the Oracle LS estimator in terms of NMSE. This is because both AMP-SSS and EM-BG-GAMP can learn and exploit the distribution characteristic of the binary data, while the Oracle LS estimator cannot.

6.3 2D Hand-written digits

A last series of experiments are carried out to reconstruct 2D images of hand-written digits from the MNIST data set[1]. The MNIST data



(a) Success rate vs. M/N

(b) Pattern success rate vs. M/N

Fig. 6. *Block sparse binary signal reconstruction, noiseless case.*

[1] Data set is available at http://yann.lecun.com/exdb/mnist/

set contains 60,000 digit gray images, each of size 28 × 28 pixels. These images are sparse since most of the pixels are inactive and take the value zero while only a few pixels are active. Moreover, due to the inherent structure of digits, these images also exhibit structured sparsity pattern. One image for each digit is randomly extracted from the MNIST data set. Due to lack of space, the full results are given only for digit 5 in both the noiseless and noisy case with different algorithms. For the remaining digits, average NMSEs in the noisy case at a specific measurement ratio are shown. Note that to recover 2D images, the algorithms MBCS-LBP [27], PC-SBL [45], PCSBL-GAMP [46], and the proposed AMP-SSS are all modified to their 2D versions. The results are averaged over 100 independent runs. Figure 8(a) and figure 8(b) depict the success rate and pattern recovery success rate, respectively, which shows that AMP-SSS achieves the highest (except Oracle LS) success rate and pattern recovery rate at various measurement ratios. Figure 9 shows the average NMSEs at different measurements when the signal to noise ratio (SNR) is 30 dB. Note that AMP-SSS outperforms the other methods (except Oracle LS) in terms of NMSE. The typical recovery results are shown in figure 10 when the measurement ratio $M/N = 0.30$ and $\mathrm{SNR} = 30\ \mathrm{dB}$. In this case AMP-SSS recovers the original image with $\mathrm{NMSE} = -25.04\ \mathrm{dB}$, which is much lower than those of the other methods.

Table I shows the average NMSEs of different algorithms for different digits when the measurement ratio $M/N = 0.30$ and $\mathrm{SNR} = 30\mathrm{dB}$. It can be seen that Oracle LS achieves the lowest NMSE in all cases since it knows the support information. For other algorithms with no prior knowledge of the sparsity pattern, the proposed AMP-SSS performs the best for almost all of the digits. From Table I,

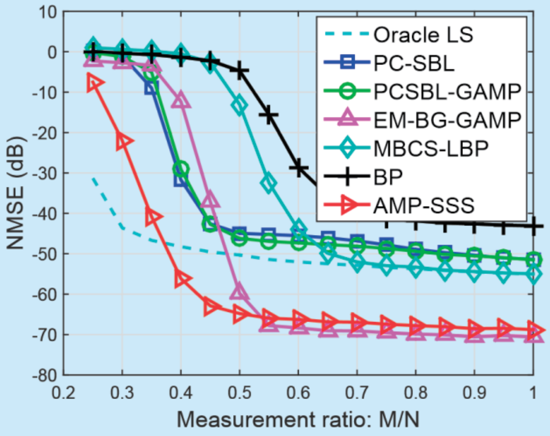
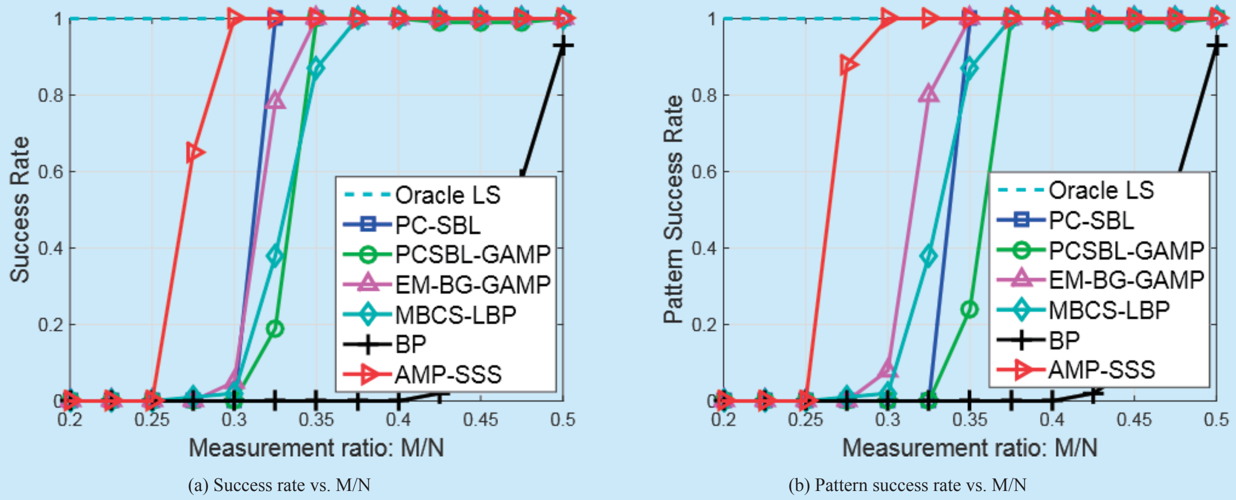


Fig. 7. *NMSE vs. M/N for block sparse binary signals, SNR = 50 dB.*



(a) Success rate vs. M/N

(b) Pattern success rate vs. M/N

Fig. 8. *Digit 5 reconstruction from the MNIST data set, noiseless case.*

it can be seen that the ten digits exhibit quite different sparsity patterns, which implies that the reconstruction performance of AMP-SSS is quite robust to the specific sparsity pattern.

VII. Conclusions

This paper addresses the problem of recovering structured sparse signals using AMP with the structured spike and slab prior. The prior correlation of the support of the solution is encoded by imposing a transformed Gaussian process on the spike and slab probabilities. Under this model, an efficient AMP based algorithm is derived for posterior inference, which reduces the computational complexity significantly. Further, an efficient method is proposed to learn the hyperparameters using EM and Bethe free energy optimization. Various experimental results on both synthetic and real data demonstrate the superior reconstruction performance of the proposed algorithm.

Acknowledgements

This work was partially supported by the National Nature Science Foundation of China (Grant No. 91438206 and 91638205), and was supported by Zhejiang Province Natural Science Foundation of China (Grant No. LQ18F010001).

References

[1] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[2] J. W. Choi, B. Shim, Y. Ding, B. Rao, and D. I. Kim,

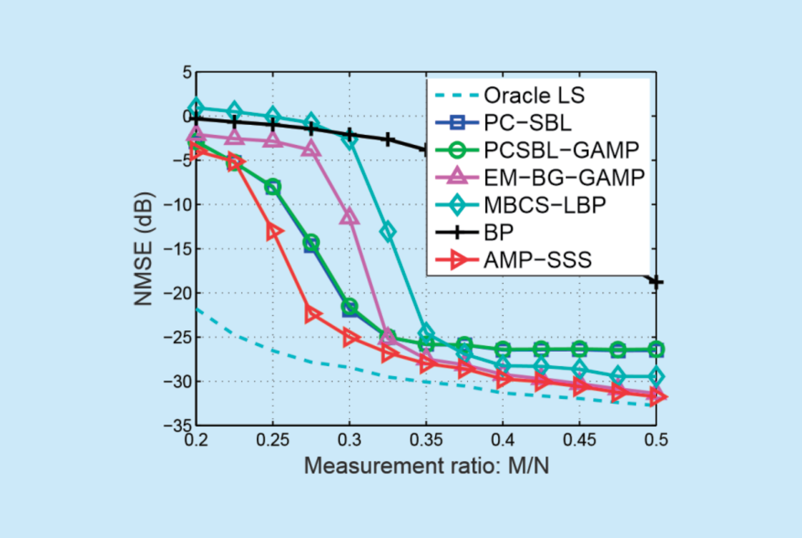


Fig. 9. *NMSE vs. M/N for one digit 5 from the MNIST data set, noisy case. SNR = 30 dB.*

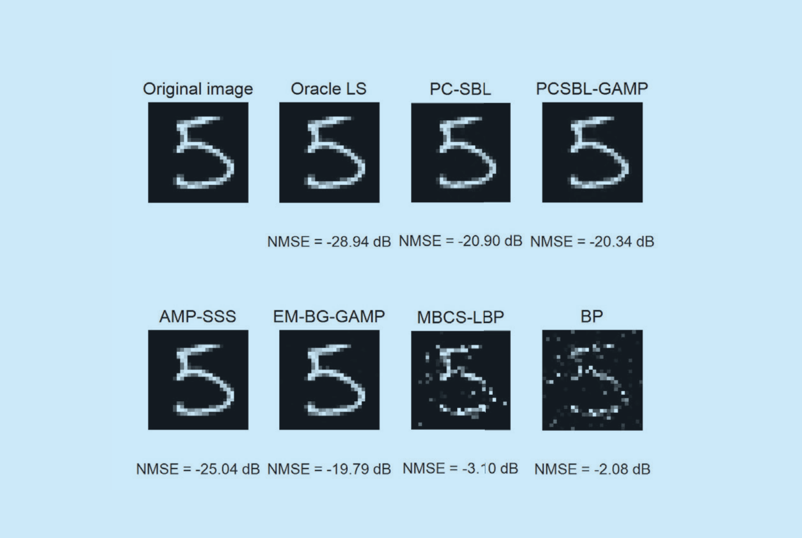
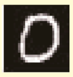


Fig. 10. *Typical recovery results for one digit 5 from the MNIST data set, noisy case. M/N=0.30 and SNR = 30 dB.*

Table I. *The nmse for different digits from mnist data set, noisy case. M/N=0.30 AND SNR = 30 dB.*

Algorithms										
Oracle LS	-25.97	-33.86	-24.99	-28.86	-29.46	-28.94	-29.57	-27.85	-24.69	-25.42
PC-SBL	-12.90	-28.57	-12.64	-25.17	-25.67	-20.90	-25.75	-24.79	-17.13	-20.72
PCSBL-GAMP	-12.68	-28.37	-12.47	-25.18	-25.66	-20.34	-25.72	-24.80	-16.49	-20.12
AMP-SSS	-18.06	-33.26	-18.08	-25.52	-26.33	-25.04	-26.80	-22.73	-18.61	-20.25
EM-BG-GAMP	-3.52	-33.19	-3.68	-5.04	-14.90	-19.79	-21.95	-4.63	-3.39	-3.2864
MBCS-LBP	0.04	-31.64	-0.19	-2.41	-8.02	-3.10	-20.35	-1.58	0.05	0.02
BP	-1.01	-14.58	-1.08	-1.53	-2.15	-2.08	-2.52	-1.49	-0.88	-0.90

“Compressed sensing for wireless communications: Useful tips and tricks,” *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1527–1550, thirdquarter 2017.

[3] Z. Gao, L. Dai, S. Han, C.-L. I, Z. Wang, and L. Hanzo, “Compressive sensing techniques for next-generation wireless communications,” *IEEE Wireless Commun.*, vol. PP, no. 99, pp. 2–11, 2018.

[4] S. Wu, Z. Ni, X. Meng, and L. Kuang, “Block expectation propagation for downlink channel estimation in massive MIMO systems,” *IEEE Commun. Lett.*, vol. 20, no. 11, pp. 2225–2228, Nov 2016.

[5] X. Lin, S. Wu, L. Kuang, Z. Ni, X. Meng, and C. Jiang, “Estimation of sparse massive MIMO-OFDM channels with approximately common support,” *IEEE Commun. Lett.*, vol. 21, no. 5, pp. 1179–1182, May 2017.

[6] X. Lin, S. Wu, C. Jiang, L. Kuang, J. Yan, and L. Hanzo, “Estimation of broadband multiuser millimeter-wave massive MIMO-OFDM channels by exploiting their sparse structure,” *IEEE Trans. Wireless Commun.*, vol. PP, no. 99, pp. 1–14, 2018.

[7] X. Meng, S. Wu, L. Kuang, D. Huang, and J. Lu, “Multi-user detection for spatial modulation via structured approximate message passing,” *IEEE Commun. Lett.*, vol. 20, no. 8, pp. 1527–1530, Aug 2016.

[8] S. Wang, Y. Li, and J. Wang, “Multiuser detection in massive spatial modulation MIMO with low-resolution ADCs,” *IEEE Trans. Wireless Commun.*, vol. 14, no. 4, pp. 2156–2168, April 2015.

[9] S. Wang, L. Zhang, Y. Li, J. Wang, and E. Oki, “Multiuser MIMO transmission aided by massive one-bit magnitude measurements,” *IEEE Trans. Wireless Commun.*, vol. 15, no. 10, pp. 7058–7073, Oct 2016.

[10] H. Cao, J. Zhu, and Z. Xu, “Adaptive one-bit quantization via approximate message passing with nearest neighbour sparsity pattern learning,” *IET Signal Processing*, Jan. 2018.

[11] T. Park and G. Casella, “The Bayesian LASSO,” *Journal of the American Statistical Association*, vol. 103, no. 482, pp. 681–686, 2008.

[12] Z. Zhang and B. D. Rao, “Sparse signal recovery with temporally correlated source vectors using sparse Bayesian learning,” *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 5, pp. 912–926, 2011.

[13] Z. Yuan, C. Zhang, Q. Guo, Z. Wang, X. Lu, and S. Wu, “Combined message passing based SBL with Dirichlet process prior for sparse signal recovery with multiple measurement vectors,” *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2018.

[14] J. P. Vila and P. Schniter, “Expectation-maximization Gaussianmixture approximate message passing,” *IEEE Trans. Signal Process.*, vol. 61, no. 19, pp. 4658–4672, 2013.

[15] D. L. Donoho, A. Maleki, and A. Montanari, “Message-passing algorithms for compressed sensing,” *Proc. Nat. Acad. Sci.*, vol. 106, no. 45, Nov. 2009, pp. 18914–18919.

[16] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[17] M. Mezard and A. Montanari, *Information, physics, and computation*. Oxford University Press, 2009.

[18] S. Rangan, “Generalized approximate message passing for estimation with random linear mixing,” *Proc. IEEE Int. Symp. Inf. Theory*, 2011, pp. 2168–2172.

[19] M. Stojnic, F. Parvaresh, and B. Hassibi, “On the reconstruction of block-sparse signals with an optimal number of measurements,” *IEEE Trans. Signal Process.*, vol. 57, no. 8, pp. 3075–3085, 2009.

[20] Y. C. Eldar, P. Kuppinger, and H. Bölcskei, “Block-sparse signals: Uncertainty relations and efficient recovery,” *IEEE Trans. Signal Process.*, vol. 58, no. 6, pp. 3042–3054, 2010.

[21] X. Lv, G. Bi, and C. Wan, “The group LASSO for stable recovery of block-sparse signal representations,” *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1371–1382, 2011.

[22] J. Huang, T. Zhang, and D. Metaxas, “Learning with structured sparsity,” *The Journal of Machine Learning Research*, vol. 12, pp. 3371–3412, 2011.

[23] S. Som and P. Schniter, “Compressive imaging using approximate message passing and a Markov-tree prior,” *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3439–3448, 2012.

[24] M. R. Andersen, O. Winther, and L. K. Hansen, “Bayesian inference for structured spike and slab priors,” *Proc. Advances in Neural Information Processing Systems*, 2014, pp. 1745–1753.

[25] M. R. Andersen, A. Vehtari, O. Winther, and L. K. Hansen, “Bayesian inference for spatio-temporal spike and slab priors,” *arXiv preprint arXiv:1509.04752*, 2015.

[26] M. R. Andersen, A. Vehtari, O. Winther, and L. K. Hansen, “Bayesian inference for spatio-temporal spike and slab priors,” *Journal of Machine Learning Research*, vol. 18, no. 139, pp. 1–58, 2017.

[27] L. Yu, H. Sun, G. Zheng, and J. P. Barbot, “Model based Bayesian compressive sensing via local beta process,” *Signal Processing*, vol. 108, pp. 259–271, 2015.

[28] T. Minka, “A family of algorithms for approximate Bayesian inference,” Ph.D. dissertation, Massachusetts Institute of Technology, 2001.

[29] X. Meng, S. Wu, L. Kuang, and J. Lu, “An expectation propagation perspective on approximate message passing,” *IEEE Signal Processing Letters*, vol. 22, no. 8, pp. 1194–1197, 2015.

[30] S. Ambikasaran, D. Foreman-Mackey, L. Greengard, D. Hogg, and M. O'Neil, “Fast direct meth-

ods for Gaussian processes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 252–265, Feb 2016.

[31] P. Schniter, "Turbo reconstruction of structured sparse signals," *Proc. IEEE 44th Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2010, pp. 1–6.

[32] J. Ziniel and P. Schniter, "Efficient high-dimensional inference in the multiple measurement vector problem," *IEEE Trans. Signal Process.*, vol. 61, no. 2, pp. 340–354, 2013.

[33] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

[34] F. Krzakala, M. Mézard, F. Sausset, Y. Sun, and L. Zdeborová, "Probabilistic reconstruction in compressed sensing: algorithms, phase diagrams, and threshold achieving matrices," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2012, no. 08, p. P08009, 2012.

[35] F. Krzakala, A. Manoel, E. Tramel, and L. Zdeborová, "Variational free energies for compressed sensing," *Proc. IEEE International Symposium on Information Theory (ISIT)*, Jun. 2014, pp. 1499–1503.

[36] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," *arXiv preprint arXiv:1010.5141v2*, 2012.

[37] X. Meng, S. Wu, and J. Zhu, "A unified Bayesian inference framework for generalized linear models," *IEEE Signal Processing Letters*, vol. 25, no. 3, pp. 398–402, March 2018.

[38] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. the MIT Press, 2006.

[39] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. USA: MIT Press, 2009.

[40] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing: I. motivation and construction," in *IEEE Information Theory Workshop (ITW)*, Jan. 2010, pp. 1–5.

[41] J. Ziniel and P. Schniter, "Dynamic compressive sensing of timevarying signals via approximate message passing," *IEEE Trans. Signal Process.*, vol. 61, no. 21, pp. 5270–5284, 2013.

[42] E. W. Tramel, A. Drémeau, and F. Krzakala, "Approximate message passing with restricted Boltzmann machine priors," *arXiv preprint arXiv:1502.06470*, 2015.

[43] S. Ambikasaran and E. Darve, "An O(NlogN) fast direct solver for partial hierarchically semi-separable matrices," *Journal of Scientific Computing*, vol. 57, no. 3, pp. 477–501, 2013.

[44] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM journal on scientific computing*, vol. 20, no. 1, pp. 33–61, 1998.

[45] J. Fang, Y. Shen, H. Li, and P. Wang, "Pattern-coupled sparse Bayesian learning for recovery of block-sparse signals," *IEEE Trans. Signal Process.*, vol. 63, no. 2, pp. 360–372, 2015.

[46] J. Fang, L. Zhang, and H. Li, "Two-dimensional pattern-coupled sparse Bayesian learning via generalized approximate message passing," *IEEE Transactions on Image Processing*, vol. 25, no. 6, pp. 2920–2930, 2016.

[47] Z. Zhang and B. Rao, "Extension of SBL algorithms for the recovery of block sparse signals with intra-block correlation," *IEEE Trans. on Signal Process.*, vol. 61, no. 8, pp. 2009–2015, 2013.

Biographies

Xiangming Meng, received the B.E. degree from Xidian University, Xi'an, China, in 2011, and the Ph.D. degree in electronic engineering from Tsinghua University, Beijing, China, in 2016. Currently, he is a research engineer at Huawei Technologies, Co. Ltd., Shanghai, China. His research interests include wireless broadband communications, signal processing, and 5G communications.

Sheng Wu, received B.E. and M.E. degrees from Beijing University of Post and Telecommunications, Beijing, China, in 2004 and 2007, respectively, and Ph.D. degree in electronic engineering from Tsinghua University, Beijing, China, in 2014. Currently, he is a postdoctoral researcher in the Tsinghua Space Center at Tsinghua University, Beijing, China. His research interests are mainly in iterative detection and decoding, channel estimation, massive MIMO, and satellite communications.

Michael Riis Andersen, received B.E. degree from Engineering College of Aarhus, Denmark, in 2011, and Ph.D. degree in Machine learning and probabilistic modelling from Technical University of Denmark, Denmark, in 2017. He is currently working as a postdoctoral researcher in the Probabilistic Machine Learning group at Aalto University in Finland. His research interests include probabilistic modelling, approximate inference, and Gaussian processes.

Jiang Zhu, received B.E. from Harbin Engineering University in 2011, Harbin, China, and Ph.D. degrees in electronic engineering from Tsinghua University, Beijing, China, in 2016. From Feb. 2015 to Aug. 2015, he was a visiting student (under supervision of Prof. Blum) with the signal processing and communication laboratory, Lehigh University. Since 2016, he joined Ocean College, Zhejiang University as an Assistant Professor. Dr. Zhu is a member of IEEE.

Zuyao Ni, received the B.E. and M.E. degrees from Zhejiang University, Hangzhou, China, in 1998 and 2001, respectively, and the Ph.D. degree in electronic engineering from Tsinghua University, Beijing, China, in 2006. Since 2007, he has been with Tsinghua University, where he is currently an Associate Professor in the Research Institute of Information Technology. His research interests include wireless broadband communications, signal processing, and satellite communication.