Variance State Propagation for Structured Sparse Bayesian Learning

Mingchen Zhang, Xiaojun Yuan, Senior Member, IEEE, and Zhen-Qing He

arXiv:1910.07352v1 [eess.SP] 16 Oct 2019

Abstract—We propose a compressed sensing algorithm termed variance state propagation (VSP) for block-sparse signals, i.e., sparse signals that have nonzero coefficients occurring in clusters. The VSP algorithm is developed under the Bayesian framework. A hierarchical Gaussian prior is introduced to depict the clustered patterns in the sparse signal. Markov random field (MRF) is introduced to characterize the state of the variances of the Gaussian priors. Such a hierarchical prior has the potential to encourage clustered patterns and suppress isolated coefficients whose patterns are different from their respective neighbors. The core idea of our algorithm is to iteratively update the variances in the prior Gaussian distribution. The message passing technique is employed in the design of the algorithm. For messages that are difficult to calculate, we correspondingly design reasonable methods to achieve approximate calculations. The hyperparameters can be updated within the iteration process. Simulation results demonstrate that the VSP algorithm is able to handle a variety of block-sparse signal recovery tasks and presents a significant advantage over the existing methods.

Index Terms—compressed sensing, variance state propagation, sparse Bayesian learning, block-sparse signal recovery

I. INTRODUCTION

In recent years, the compressed sensing (CS) technique, as a new signal acquisition scheme beyond Nyquist sampling, has attracted great interest with a wide range of applications in signal processing [1], [2] and wireless communications [3], [4]. Mathematically, given a measurement matrix $\mathbf{A} \in \mathbb{C}^{M \times N}$ $(M \ll N)$, the goal of CS is to reconstruct an unknown sparse signal $\mathbf{x} \in \mathbb{C}^N$ from the noise-corrupted linear measurements

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w} \tag{1}$$

where $w \in \mathbb{C}^M$ is an additive noise. This problem has been well studied and a variety of algorithms that with guaranteed recovery performance have been proposed, including orthogonal matching pursuit (OMP) [5], compressive sampling matching pursuit (CoSaMP) [6], basis pursuit method [7], and sparse Bayesian learning (SBL) methods [8].

Since the advent of compressed sensing, much research attention has been paid to the reconstruction of a type of sparse signals with additional structure, i.e., sparse signals with the nonzero entries appearing in clusters, namely, *block-sparse* signals. Block-sparsity arises naturally in a variety of practical signals. For instance, in wireless communications, due to the effect of limited local scattering in the propagation environment, the massive multi-input multi-output (MIMO) channel in the virtual angular domain exhibits a block sparsity [9], [10]. In a video surveillance system, foreground moving objects usually occupy a small portion of the camera view, leading to natural block sparsity. Block-sparse patterns also appear in the multiple measurement vector (MMV) problem that deals with the recovery of multiple sparse signal vectors sharing a common nonzero support [11], [12].

For the reconstruction of block-sparse signals, algorithms making an explicit use of the additional block-sparse pattern can yield provably better reconstruction performance than the conventional CS algorithms in which the signals for recovery are assumed to be randomly sparse. A number of CS algorithms are specifically designed for the recovery of block-sparse signals, e.g., Block-OMP [13], mixed ℓ^2/ℓ^1 norm-minimization [11], and group LASSO [14]. These algorithms require strong prior knowledge of the block structure, such as the locations and the lengths of the associated blocks, which are often unavailable in practical applications. Algorithms for structure-agnostic block-sparse signal recovery are also developed. For example, in [15], a hierarchical Bayesian Bernoulli-Gaussian prior model was adopted to model both the sparse prior and the cluster prior, and a Markov chain Monte Carlo (MCMC) sampling method is employed in the inference; in [16], the block-sparse pattern is modeled by a Boltzman machine, and a greedy method was used to simplify the maximum *a posteriori* probability (MAP) estimator; in [17], the components of the signal are modeled by a number of overlapping blocks, and an expanded block sparse Bayesian learning (EBSBL) was proposed to adaptively exploit intra-block correlation; in [18], a pattern-coupled hierarchical Gaussian framework was proposed to encourage block-sparse patterns, where the sparsity of each coefficient is controlled by the linear combination of the hyperparameters of itself and its neighbors. Although they require little or even do not require the prior information of the sparse patterns, these algorithms typically perform far away from the genie bound in which the location of nonzeros is known a priori.

Due to its outstanding performance and low complexity, the message passing technique has been employed in the design of CS algorithms for a decade. Representative message passing based CS algorithms include approximate message passing (AMP) [19], generalized approximate message passing (GAMP) [20], expectation-maximization Gaussian-mixture approximate message passing (EM-GM-AMP) [21], and turbo compressed sensing (Turbo-CS) [22]. Message passing based CS algorithms for block-sparse signals have also been proposed in [23]–[25]. In [23], structured turbo compressed sensing (STCS) was developed for massive MIMO channel estimation. By combining a Markov prior into the Turbo-CS framework, STCS fully utilizes the knowledge of block sparsity and shows

M. Zhang, X. Yuan and Z.-Q. He are with the Center for Intelligent Networking and Communications and also with the National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: zhangmingchen@std.uestc.edu.cn; xjyuan@uestc.edu.cn); zhenqinghe@uestc.edu.cn.

superior recovery performance. It is known that message passing based CS algorithms are sensitive to the choice of measurement matrices, since the sum-product rule used in message calculation requires the independence of relevant messages. The convergence of the AMP is guaranteed when the elements of the measurement matrix A are independently and identically distributed Gaussian and the length of the signal N is large [19]. Turbo-CS relaxes the requirement on A such that A is allowed to consist of rows randomly selected from an orthogonal basis [23] or is right-rotationally invariant [22]. However, the performance of the message passing based algorithms may deteriorate severely when other measurement matrices are involved. It is therefore desirable to design a message passing based CS algorithm that is able to handle a wider range of applications.

In this paper, we propose a new message passing based CS algorithm for the reconstruction of block-sparse signals. A novel hierarchical Gaussian framework is deployed to model the sparse prior, in which the unknown signal components are modeled as independent Gaussian variables with zero mean and certain variances. Each variance is still regarded as a random variable and assigned a Bernoulli-Gamma prior with a support indicator. These support indicators, a.k.a. the state variables, are assigned as a Markov random field (MRF) to capture the block sparsity. Such a prior has the potential to encourage block-sparse patterns and suppress "isolated coefficients" whose pattern is different from that of its neighboring coefficients. Message passing is performed based on the hierarchical probability model and an iterative algorithm is accordingly developed to estimate the block-sparse signal. For messages difficult to compute, we give approximate calculation methods. The model hyperparameters are updated during the iteration. Since the state of the variances plays a crucial role in message propagation, we refer to our proposed algorithm as variance state propagation (VSP). Our numerical results show that VSP inherits the superior performance of the message passing based compressed sensing algorithms while maintaining the robustness to the choice of measurement matrices.

The rest of the paper is organized as follows. In Section II, we introduce the MRF-combined hierarchical probability model that characterizes the sparse prior and the pattern dependencies among the signal components. An iterative message passing algorithm is developed in Section III to estimate the blocksparse signal. Section III contains approximate methods for messages that are difficult to compute, and learning methods for model hyperparameters. Simulation results are provided in Section IV, followed by concluding remarks in Section V.

II. PROBABILITY MODEL

The goal of this work is to recover a block-sparse signal $x \in \mathbb{C}^N$ from the noise-corrupted measurements

$$y = Ax + w \tag{2}$$

where $\mathbf{A} \in \mathbb{C}^{M \times N}$ (M < N) is the measurement matrix, and $\mathbf{w} \in \mathbb{C}^{M}$ is a circularly symmetric complex Gaussian (CSCG) noise with zero mean and covariance matrix $\sigma^2 \mathbf{I}$. We use a hierarchical Gaussian prior model to characterize the block-

sparse structure of the unknown signal x in (2). Specifically, x is assigned a conditional Gaussian prior

$$p(\boldsymbol{x}|\boldsymbol{v}) = \prod_{i=1}^{N} p(x_i|v_i)$$
(3)

where $\boldsymbol{x} = [x_1, \ldots, x_N]^T$, $\boldsymbol{v} = [v_1, \ldots, v_N]^T$, and $p(x_i|v_i) = \mathcal{CN}(x_i; 0, v_i)$ is a CSCG distribution with zero mean and variance v_i . Note that each v_i is the variance of signal component x_i to control the sparsity. When v_i approaches zero, the corresponding component x_i becomes zero. In this work, each v_i is assigned a conditionally independent distribution given by

$$p(v_i|s_i) = \operatorname{Gamma}(v_i; a, b)\delta(s_i - 1) + \delta(v_i)\delta(s_i + 1), \quad (4)$$

where $s_i \in \{-1, 1\}$ is a hidden binary state; $\delta(\cdot)$ denotes the Dirac delta function; Gamma $(v_i; a, b)$ is the Gamma distribution defined as

$$\operatorname{Gamma}(v_i; a, b) = \begin{cases} \frac{b^a v_i^{a-1} e^{-bv_i}}{\Gamma(a)}, & v_i > 0\\ 0, & \text{otherwise} \end{cases}$$
(5)

with $\Gamma(a) = \int_0^\infty t^{a-1} e^{-t} dt$ being the Gamma function. We use the Gamma distribution to characterize the nonzero part of each nonnegative sparse random variable v_i . In addition, when there is no prior knowledge of the random variable, a and b can be set to a small value (e.g., 10^{-10}) to make the distribution noninformative. Let ρ be the fraction of nonzero elements in x, the distribution $p(s_i)$ is modeled as

$$p(s_i) = \begin{cases} \rho, & s_i = 1\\ 1 - \rho, & s_i = -1. \end{cases}$$
(6)

Then, the marginal distribution of v_i can be expressed as

$$p(v_i) = \sum_{s_i} p(v_i|s_i)p(s_i)$$

= $\rho \text{Gamma}(v_i; a, b) + (1 - \rho)\delta(v_i).$ (7)

Furthermore, we use a Markov random field (MRF) prior to describe the block-sparse structure of v. The hidden state variables can be modeled by the classic Ising model [26] as

$$p(s) \propto \exp\left(\sum_{i=1}^{N} \left(\frac{1}{2} \sum_{k \in \mathcal{D}_i} \beta s_k - \alpha\right) s_i\right)$$
 (8)

where $\mathcal{D}_i \subset \{1, \ldots, N\} \setminus i$ is the neighbors of index i; α and β are parameters of the MRF. A larger β implies a larger size of each block of non-zeros, and a larger α encourages a sparser x.

We proceed to perform Bayesian inference based on the proposed hierarchical model. From the Bayesian rule, the joint probability of p(y, x, v, s) can be decomposed as

$$p(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{v}, \boldsymbol{s}) = p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x}|\boldsymbol{v})p(\boldsymbol{v}|\boldsymbol{s})p(\boldsymbol{s})$$
$$= p(\boldsymbol{y}|\boldsymbol{x})\prod_{i}^{N}p(x_{i}|v_{i})p(v_{i}|s_{i})p(\boldsymbol{s}).$$
(9)

The dependencies of the random variables in the factorization (9) can be shown by a factor graph as depicted in Fig. 1(a), where circles represent variable nodes and squares represent



Fig. 1: (a) The factor graph characterizes the hierarchical probability model assumed in (9). (b) An example of the 4-connected MRF with dimension 3×3 .

factor nodes. The factor nodes $\{f_i\}, \{g_i\}, l$, and γ in Fig. 1(a) are defined as

$$f_i: p(v_i|s_i), \tag{10a}$$

$$p_i: p(x_i|v_i) = \mathcal{CN}(x_i; 0, v_i), \tag{10b}$$

$$l: p(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{CN}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}; \boldsymbol{0}, \sigma^2 \boldsymbol{I}), \qquad (10c)$$

$$\gamma: p(s) \propto \left(\prod_{i=1}^{N} \prod_{k \in \mathcal{D}_{i}} \varphi(s_{i}, s_{k})\right)^{\overline{2}} \prod_{i=1}^{N} \psi(s_{i})$$
 (10d)

where

$$\varphi(s_i, s_k) = \exp(\beta s_i s_k), \tag{11a}$$

$$\psi(s_i) = \exp(-\alpha s_i). \tag{11b}$$

The factor graph in Fig. 1(a) includes two modules, namely, the linear module that handles the linear constraint in (2) and the MRF module that handles the MRF prior of s in (8). We further see that, with γ in (10d), the MRF module can be expanded as a sub-factor graph. In this paper, we will mostly focus on the commonly used 4-connected MRF as illustrated in Fig. 1(b), though our algorithm can be readily applied to other forms of MRFs including the one-dimensional Markov chain.

Based on the above probability model, an optimal solution of \boldsymbol{x} can be found by solving $\max_{\boldsymbol{x}} p(\boldsymbol{x}|\boldsymbol{y})$. However, solving this problem is computationally infeasible even for moderate values of M and N. In this paper, we propose a low-complexity yet near-optimal message passing algorithm, termed variance state propagation, as detailed in what follows.

III. VARIANCE STATE PROPAGATION ALGORITHM

A. Sum-Product Message Passing

We will basically follow the sum-product rule for message passing over the factor graph in Fig. 1(a). We start from the output messages of the linear module. In Fig. 1(a), suppose that variable node v_i receives a message $\nu_{g_i \rightarrow v_i}$ from the factor node g_i . The message from v_i to f_i is still given by $\nu_{g_i \rightarrow v_i}$. Then the message from f_i to s_i is a Bernoulli distribution given by

$$\nu_{f_i \to s_i} \propto \int_{v_i} p(v_i | s_i) \nu_{g_i \to v_i}$$

$$= \pi_{f_i \to s_i} \delta(s_i - 1) + (1 - \pi_{f_i \to s_i}) \delta(s_i + 1)$$
(12a)
(12b)

where

$$\pi_{f_i \to s_i} = \frac{\int_{v_i} \nu_{g_i \to v_i} \operatorname{Gamma}(v_i; a, b)}{\int_{v_i} \nu_{g_i \to v_i} \operatorname{Gamma}(v_i; a, b) + \int_{v_i} \nu_{g_i \to v_i} \delta(v_i)}.$$
(13)

With the inputs $\{\nu_{f_i \to s_i}\}$, we are now ready to describe the messages involved in the MRF. For simplicity, we give the details of the messages passed in the 4-connected MRF as shown in Fig. 1(b). The left, right, top, bottom neighbors to node s_i are indexed by i_1, i_r, i_t, i_b , respectively, i.e., $\mathcal{D}_i = \{i_1, i_r, i_t, i_b\}$. The left, right, top, and bottom input messages of each variable s_i are represented as Bernoulli distributions $\nu_i^{l}, \nu_i^{r}, \nu_i^{t}$, and ν_i^{b} , respectively. The input message of s_i from the left is given by

$$\nu_{i}^{l} \propto \int_{s_{i_{l}}} \nu_{f_{i_{l}} \rightarrow s_{i_{l}}} \prod_{k \in \{l, t, b\}} \nu_{i_{l}}^{k} \psi(s_{i_{l}}) \varphi(s_{i}, s_{i_{l}})$$
$$= \lambda_{i}^{l} \delta(s_{i} - 1) + (1 - \lambda_{i}^{l}) \delta(s_{i} + 1)$$
(14)

where λ_i^l is shown in (15) (at the top of the next page). The messages from the right, the top, and the bottom have similar representations. The output message of the MRF for each s_i can be calculated as

$$\nu_{s_i \to f_i} = \pi_{s_i \to f_i} \delta(s_i - 1) + (1 - \pi_{s_i \to f_i}) \delta(s_i + 1) \quad (16)$$

where

$$\pi_{s_i \to f_i} = \frac{e^{-\alpha} \prod_{k \in \{\mathsf{l},\mathsf{r},\mathsf{t},\mathsf{b}\}} \lambda_i^k}{e^{-\alpha} \prod_{k \in \{\mathsf{l},\mathsf{r},\mathsf{t},\mathsf{b}\}} \lambda_i^k + e^\alpha \prod_{k \in \{\mathsf{l},\mathsf{r},\mathsf{t},\mathsf{b}\}} (1 - \lambda_i^k)}.$$
(17)

Then, the message from f_i to v_i is a Bernoulli-Gamma distribution given by

$$\nu_{f_i \to v_i} \propto \int_{s_i} p(v_i | s_i) \nu_{s_i \to f_i}$$

$$= \pi_{s_i \to f_i} \operatorname{Gamma}(v_i; a, b) + (1 - \pi_{s_i \to f_i}) \delta(v_i).$$
(18b)

With $\nu_{v_i \to q_i} = \nu_{f_i \to v_i}$, the message from g_i to x_i is given by

$$\nu_{g_i \to x_i} \propto \int_{v_i} p(x_i | v_i) \nu_{v_i \to g_i}.$$
(19)

$$\lambda_{i}^{\mathsf{l}} = \frac{\pi_{f_{i_{\mathsf{l}}} \to s_{i_{\mathsf{l}}}} \prod_{k \in \{\mathsf{l},\mathsf{t},\mathsf{b}\}} \lambda_{i_{\mathsf{l}}}^{k} e^{-\alpha+\beta} + (1 - \pi_{f_{i_{\mathsf{l}}} \to s_{i_{\mathsf{l}}}}) \prod_{k \in \{\mathsf{l},\mathsf{t},\mathsf{b}\}} (1 - \lambda_{i_{\mathsf{l}}}^{k}) e^{\alpha-\beta}}{(e^{\beta} + e^{-\beta}) \left(\pi_{f_{i_{\mathsf{l}}} \to s_{i_{\mathsf{l}}}} e^{-\alpha} \prod_{k \in \{\mathsf{l},\mathsf{t},\mathsf{b}\}} \lambda_{i_{\mathsf{l}}}^{k} + (1 - \pi_{f_{i_{\mathsf{l}}} \to s_{i_{\mathsf{l}}}}) e^{\alpha} \prod_{k \in \{\mathsf{l},\mathsf{t},\mathsf{b}\}} (1 - \lambda_{i_{\mathsf{l}}}^{k}) \right)}$$
(15)

The message from x_i to l is $\nu_{x_i \to l} = \nu_{g_i \to x_i}$. We now consider the message from factor node l back to variable node x_i . From the sum-product rule, $\nu_{l \to x_i}$ can be expressed as

$$\nu_{l \to x_{i}} \propto \int_{\boldsymbol{x}_{\setminus i}} p(\boldsymbol{y}|\boldsymbol{x}) \prod_{i' \neq i} \nu_{x_{i'} \to l}$$
$$= \int_{\boldsymbol{x}_{\setminus i}} p(\boldsymbol{y}|\boldsymbol{x}) \prod_{i' \neq i} \int_{v_{i'}} p(x_{i'}|v_{i'}) \nu_{v_{i'} \to g_{i'}}$$
(20)

for $i' \neq i$, where $x_{\setminus i}$ denotes all the entries of x except the *i*-th entry. Clearly, $\nu_{x_i \to g_i} = \nu_{l \to x_i}, \forall i$. Then, the messages $\nu_{g_i \to v_i}$ and $\nu_{v_i \to f_i}$ can be computed as

$$\nu_{v_i \to f_i} = \nu_{g_i \to v_i} \propto \int_{x_i} \nu_{x_i \to g_i} p(x_i | v_i)$$
(21)

where $\nu_{x_i \to g_i} = \nu_{l \to x_i}$. The above messages are calculated iteratively until convergence.

B. Update of $\nu_{g_i \rightarrow v_i}$

The algorithm in Subsection III-A is a straightforward application of the sum-product rule for message passing. This algorithm, however, is difficult to implement due to the high computational complexity involved in evaluating the integrals in (20) and (21). To reduce complexity, we propose to approximate the outputs of the linear module $\{\nu_{g_i \to v_i}\}$ as follows. By substituting $\nu_{x_i \to g_i} = \nu_{l \to x_i}$ in (20), the message $\nu_{g_i \to v_i}$ in (21) can be written as

$$\nu_{g_{i} \to v_{i}} \propto \int_{x_{i}} p(x_{i}|v_{i}) \int_{\boldsymbol{x}_{\setminus i}} p(\boldsymbol{y}|\boldsymbol{x}) \int_{\boldsymbol{v}_{\setminus i}} \prod_{i' \neq i} \left(\nu_{v_{i'} \to g_{i'}} p(x_{i'}|v_{i'}) \right)$$

$$\propto \int_{\boldsymbol{v}_{\setminus i}} \prod_{i' \neq i} \nu_{v_{i'} \to g_{i'}} \left(\int_{\boldsymbol{x}} p(\boldsymbol{y}|\boldsymbol{x}) p(\boldsymbol{x}|\boldsymbol{v}) \right)$$

$$\propto \int_{\boldsymbol{v}_{\setminus i}} p(\boldsymbol{y}|\boldsymbol{v}) \prod_{i' \neq i} \nu_{v_{i'} \to g_{i'}}.$$
(22)

The integral in (22) is difficult to solve. To simplify the message calculation, we propose to replace the output of the linear module for node v_i by the mean $\mu_{g_i \to v_i} = \mathbb{E}_{\nu_{g_i \to v_i}}[v_i]$, where the expectation \mathbb{E} is taken over the distribution $\nu_{g_i \to v_i}$. Similar ideas for message replacements and approximations have been previously employed, e.g., in denoising-based turbo compressed sensing [27] in which a denoiser is used for message approximation when the probability model of a node is incomplete or unavailable. While $\mathbb{E}_{\nu_{g_i \to v_i}}[v_i]$ is still difficult to evaluate, we further approximate $\mu_{g_i \to v_i}$ by

$$\mu_{g_i \to v_i} = \operatorname*{arg\,max}_{v_i} p(\boldsymbol{y}|\boldsymbol{v})|_{\{v_{i'} = \mu_{v_{i'} \to g_{i'}}, \forall i' \neq i\}}, \qquad (23)$$

where $\mu_{v_i \to g_i}$ is the input mean of v_i for the linear module, i.e.

$$\mu_{v_i \to g_i} = \mathbb{E}_{\nu_{v_i \to g_i}}[v_i], \forall i.$$
(24)

It is interesting to compare (23) with (22). We may treat each $\nu_{v_{i'} \to q_{i'}}$ as the prior of $v_{i'}$, for $i' \neq i$. Then $\nu_{q_i \to v_i}$ in (22) can

be regarded as the likelihood of v_i given \boldsymbol{y} , and $\mathbb{E}_{\nu_{g_i \to v_i}}[v_i]$ is the corresponding mean. In contrast, (23) only requires the prior mean $\mu_{v_{i'} \to g_{i'}}$ of each $v_{i'}$ rather than the whole distribution $\nu_{v_{i'} \to g_{i'}}$. Thus, $\mu_{g_i \to v_i}$ in (23) can be treated as the maximum likelihood of v_i given \boldsymbol{y} and $\mu_{v_{i'} \to g_{i'}}$, $i' \neq i$. From the estimation theory, it is known that $\mu_{g_i \to v_i}$ given by (23) is always inferior to $\mu_{g_i \to v_i}$ calculated based on (22), provided that the prior distributions $\{\nu_{v'_i \to g'_i}\}$ are accurate. We next present two methods to solve $\mu_{g_i \to v_i}$ in (23). For notational convenience, denote

$$\mathcal{S} = \{ v_i = \mu_{v_i \to g_i}, \forall i \}$$
(25a)

4

$$\mathcal{S}_{\backslash i} = \{ v_{i'} = \mu_{v_{i'} \to g_{i'}}, \forall i' \neq i \}.$$
(25b)

C. Gradient Method for Solving (23)

For the problem in (23), a straightforward solution is to find a stationary point of $p(\boldsymbol{y}|\boldsymbol{v})|_{\mathcal{S}_{\setminus i}}$ with respect to v_i via gradient descent (GD). We note that $p(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{v}) \propto p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x}|\boldsymbol{v})$ is a complex Gaussian distribution with the mean \boldsymbol{m} and the covariance $\boldsymbol{\Phi}$ given by

$$\boldsymbol{m} = \sigma^{-2} \boldsymbol{\Phi} \boldsymbol{A}^H \boldsymbol{y} \tag{26}$$

$$\boldsymbol{\Phi} = \left(\sigma^{-2}\boldsymbol{A}^{H}\boldsymbol{A} + \boldsymbol{D}^{-1}\right)^{-1}$$
(27)

where D is a diagonal matrix with the *i*-th diagonal element equal to v_i . Then, p(y|v) can be expressed as

$$p(\boldsymbol{y}|\boldsymbol{v}) = \int_{\boldsymbol{x}} p(\boldsymbol{y}|\boldsymbol{x}) p(\boldsymbol{x}|\boldsymbol{v})$$

= $\frac{\exp\left(\boldsymbol{m}^{H} \boldsymbol{\Phi}^{-1} \boldsymbol{m} - \sigma^{-2} \boldsymbol{y}^{H} \boldsymbol{y}\right) |\boldsymbol{\Phi}|}{(\pi\sigma^{2})^{M} |\boldsymbol{D}|} \int_{\boldsymbol{x}} \mathcal{CN}(\boldsymbol{x}; \boldsymbol{m}, \boldsymbol{\Phi})$
= $\frac{\exp\left(-\sigma^{-2} \boldsymbol{y}^{H} \boldsymbol{y}\right)}{(\pi\sigma^{2})^{M}} \cdot \frac{\exp\left(\boldsymbol{m}^{H} \boldsymbol{\Phi}^{-1} \boldsymbol{m}\right) |\boldsymbol{\Phi}|}{\prod_{i=1}^{N} v_{i}}.$ (28)

The first term in (28) is independent of v. Thus, problem (23) can be equivalently written as

$$\mu_{g_i \to v_i} = \arg\min_{v_i} \chi(\boldsymbol{y}, \boldsymbol{v})|_{\mathcal{S}_{\backslash i}},$$
(29)

where

$$\chi(\boldsymbol{y}, \boldsymbol{v}) \triangleq -\ln \frac{\exp\left(\boldsymbol{m}^{H} \boldsymbol{\Phi}^{-1} \boldsymbol{m}\right) |\boldsymbol{\Phi}|}{\prod_{i=1}^{N} v_{i}}$$
$$= -\boldsymbol{m}^{H} \boldsymbol{\Phi}^{-1} \boldsymbol{m} - \ln |\boldsymbol{\Phi}| + \sum_{i=1}^{N} \ln v_{i}.$$
(30)

It is difficult to obtain an analytical solution to problem (29). We propose to use the gradient descent to find a stationary point of (29), with the update rule given by

$$\mu_{g_i \to v_i} = \mu_{v_i \to g_i} - \epsilon_i \frac{\partial \chi}{\partial v_i} |_{v_i = \mu_{v_i \to g_i}}$$
(31)

where ϵ_i is an appropriate step size that can be selected from the backtracking line search to satisfy

$$\chi(\boldsymbol{y}, \boldsymbol{v})|_{\mathcal{S}_{\backslash i}, v_i = \mu_{q_i \to v_i}} \le \chi(\boldsymbol{y}, \boldsymbol{v})|_{\mathcal{S}}.$$
(32)

The closed-form expression of the partial derivative $\frac{\partial \chi}{\partial v_i}$ is given by

$$\frac{\partial \chi}{\partial v_i} = -\left(\frac{\boldsymbol{y}^H \boldsymbol{A} \boldsymbol{u}_i}{\sigma^2 v_i}\right)^2 - \operatorname{Tr}\left[\boldsymbol{E}_i \boldsymbol{\Phi}\right] + \frac{1}{v_i}, \quad (33)$$

where u_i is the *i*-th column of Φ , and

$$\boldsymbol{E}_{i} \triangleq \begin{bmatrix} \boldsymbol{0} & & \\ & \frac{1}{v_{i}^{2}} & \\ & & \boldsymbol{0} \end{bmatrix}$$
(34)

is a diagonal matrix with only one nonzero element $\frac{1}{v_i^2}$ in the *i*-th diagonal position. The detailed derivation of (33) is given in Appendix A. We can update all the entries of v based on (31) in a sequential manner. However, when N is large, using backtracking to calculate ϵ_i for every single v_i sequentially imposes a heavy computational burden. We consider to use a common step size for all $\{v_i\}$, i.e.

$$\boldsymbol{\mu}^{(\text{new})} = \boldsymbol{\mu}^{(\text{old})} - \epsilon \nabla \chi(\boldsymbol{y}, \boldsymbol{\mu}^{(\text{old})})$$
(35)

where $\boldsymbol{\mu}^{(\text{old})} \triangleq [\mu_{v_1 \to g_1}, \dots, \mu_{v_N \to g_N}]^T$, the gradient $\nabla \chi(\boldsymbol{y}, \boldsymbol{v})$ is defined by

$$\nabla \chi(\boldsymbol{y}, \boldsymbol{v}) \triangleq \left[\frac{\partial \chi}{\partial v_1}, \dots, \frac{\partial \chi}{\partial v_N}\right]^T,$$
 (36)

and ϵ is the common step size obtained from the backtracking line search rule satisfying

$$\chi(\boldsymbol{y}, \boldsymbol{\mu}^{(\text{new})}) \le \chi(\boldsymbol{y}, \boldsymbol{\mu}^{(\text{old})}).$$
(37)

Thus, by letting $[\mu_{g_1 \to v_1}, \ldots, \mu_{g_N \to v_N}] = \mu^{(\text{new})}$, we obtain an update of $\{\mu_{g_i \to v_i}\}$. The above approximate solution to (22) is referred to as the GD-based solver, as summarized in Algorithm 1. We note that iteration is introduced in Algorithm 1 to find a stationary point of (23).

D. ELBO-Based Method for Solving (23)

The gradient descent method described above, though conceptually simple, suffers from slow convergence and high complexity. This inspires us to develop an alternative solution to (23) with improved performance. Let q(x) be a distribution function of x, and define

$$\eta(v_i) \triangleq \ln p(\boldsymbol{y}|\boldsymbol{v})|_{\mathcal{S}_{\backslash i}}$$

= $\mathcal{L}(v_i, q(\boldsymbol{x})) + D_{\mathrm{KL}}(q(\boldsymbol{x})||p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{v})|_{\mathcal{S}_{\backslash i}})$ (38)

where $\mathcal{L}(v_i, q(\boldsymbol{x}))$ is the evidence lower bound (ELBO) defined by

$$\mathcal{L}(v_i, q(\boldsymbol{x})) \triangleq \int_{\boldsymbol{x}} q(\boldsymbol{x}) \ln \frac{p(\boldsymbol{y}, \boldsymbol{x} | \boldsymbol{v})|_{\mathcal{S}_{\backslash i}}}{q(\boldsymbol{x})}, \quad (39)$$

and

$$D_{\mathrm{KL}}(q(\boldsymbol{x})||p(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{v})|_{\mathcal{S}_{\backslash i}}) \\ \triangleq -\int_{\boldsymbol{x}} q(\boldsymbol{x}) \ln \frac{p(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{v})|_{\mathcal{S}_{\backslash i}}}{q(\boldsymbol{x})} \quad (40)$$

is the Kullback-Leibler divergence between $q(\boldsymbol{x})$ and $p(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{v})|_{\mathcal{S}_{\backslash i}}$. Equation (38) holds for any choice of $q(\boldsymbol{x})$ and v_i . Since $D_{\mathrm{KL}}(q(\boldsymbol{x})||p(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{v})|_{\mathcal{S}_{\backslash i}}) \geq 0$, $\mathcal{L}(v_i,q(\boldsymbol{x}))$ is

indeed a lower bound of $\eta(v_i)$. Thus, to approximately solve (23), we turn to maximize $\mathcal{L}(v_i, q(\boldsymbol{x}))$ as

$$\mu_{g_i \to v_i} = \operatorname*{arg\,max}_{v_i} \mathcal{L}(v_i, q(\boldsymbol{x})) \tag{41}$$

where $q(\boldsymbol{x})$ is chosen as

$$q(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{v})|_{\mathcal{S}}.$$
(42)

The following proposition ensures that the choice of q(x) in (42) yields a good approximate solution to (23).

Proposition 1. With $\mu_{g_i \to v_i}$ given by (41), the following inequality holds:

$$p(\boldsymbol{y}|\boldsymbol{v})|_{\mathcal{S}_{\backslash i}, v_i = \mu_{q_i \to v_i}} \ge p(\boldsymbol{y}|\boldsymbol{v})|_{\mathcal{S}}.$$
(43)

The proof of Proposition 1 is given in Appendix B. The following Proposition gives the solution of (41).

Proposition 2. The solution of (41) is given by

$$\mu_{g_i \to v_i} = |m_i|^2 + \phi_{i,i}, \tag{44}$$

where m_i denotes the *i*-th entry of m in (26), and $\phi_{i,i}$ denotes the *i*-th diagonal element of the covariance matrix Φ in (27).

The proof of Proposition 2 is given in Appendix C. In this way, we obtain an update of $\mu_{g_i \rightarrow v_i}$ for each *i*. The above approximate solution to (22) is referred to as the ELBO-based solver as summarized in Algorithm 2. Similarly to the GD-based solver, iteration is introduced to ensure a better performance.

E. Update of $\pi_{f_i \to s_i}$

Recall from (23) that we replace the message $\nu_{g_i \to v_i}$ by the mean $\mu_{g_i \to v_i}$. As a consequence, $\pi_{f_i \to s_i}$ in (12) cannot be calculated by using (13) since $\nu_{g_i \to v_i}$ is not available. To carry out message passing from f_i to s_i , we need to find a new approach to update $\pi_{f_i \to s_i}$.

In (24) we notice that $\mu_{v_i \to g_i}$ is set to the mean of $v_i \sim \nu_{v_i \to g_i}$, and $\nu_{v_i \to g_i}$ is a Bernoulli-Gamma distribution shown in (18b). Thus, $\mu_{v_i \to g_i}$ is calculated by using

$$\mu_{v_i \to g_i} = \mathbb{E}_{\nu_{v_i \to g_i}}[v_i] = \frac{a}{b} \pi_{s_i \to f_i},\tag{45}$$

where $\frac{a}{b}$ is the mean of the Gamma distribution Gamma($v_i; a, b$). Equation (45) shows that under the probability model specified in Section II, $\mu_{v_i \to g_i}$ is simply the product of $\pi_{s_i \to f_i}$ and the mean $\frac{a}{b}$ of Gamma($v_i; a, b$). Inspired by this, we propose a *moment matching* method that mimics the relationship between $\pi_{s_i \to f_i}$ and $\mu_{v_i \to g_i}$ to establish a map between $\mu_{g_i \to v_i}$ and $\pi_{f_i \to s_i}$, i.e.

$$\mu_{g_i \to v_i} = \frac{a}{b} \pi_{f_i \to s_i}.$$
(46)

Here, for $\mu_{g_i \to v_i}$ given by the linear module, $\pi_{s_i \to f_i}$ may be greater than 1 and therefore is not necessarily a valid probability. To avoid this, we need to limit $\pi_{s_i \to f_i}$ to the range of [0, 1]. That is, given $\mu_{g_i \to v_i}$ and the parameters $\{a, b\}$ of Gamma $(\cdot; a, b), \pi_{f_i \to s_i}$ can be approximated as

$$\pi_{f_i \to s_i} = \min\left(\frac{\mu_{g_i \to v_i}}{a/b}, 1\right),\tag{47}$$

Input: $y, A, \{\mu_{v_i \to g_i}\}, T_{in}, \sigma^2$ 1 t = 0;2 $\hat{\mu} = [\mu_{v_1 \to g_1}, \dots, \mu_{v_N \to g_N}]^T;$ 3 while $t < T_{in}$ do 4 Compute $\nabla \chi(y, \hat{\mu})$ based on (33); 5 Find ϵ that satisfies $\chi(y, \hat{\mu} + \epsilon \nabla \chi(y, \hat{\mu})) \le \chi(y, \hat{\mu})$ using the backtracking line search; 6 $\hat{\mu} = \hat{\mu} - \epsilon \nabla \chi(y, \hat{\mu});$ 7 t = t + 1;Output: $[\mu_{g_1 \to v_1}, \dots, \mu_{g_N \to v_N}]^T = \hat{\mu}$

Algorithm 2: ELBO-based solver for (23)Input: $y, A, \{\mu_{v_i \to g_i}\}, T_{in}, \sigma^2$ 1 t = 0;2 $\hat{\mu}^{(0)} = [\mu_{v_1 \to g_1}, \dots, \mu_{v_N \to g_N}]^T;$ 3 while $t < T_{in}$ do4 $D = \operatorname{diag}(\hat{\mu});$ 56 $\hat{\mu}_{g_i \to v_i} = [\hat{\mu}_{g_1 \to v_1}, \dots, \hat{\mu}_{g_N \to v_N}]^T$ with each $\hat{\mu}_{g_i \to v_i} = |m_i|^2 + \phi_{i,i};$ 7Cutput: $[\mu_{g_1 \to v_1}, \dots, \mu_{g_N \to v_N}]^T = \hat{\mu}$

where $\pi_{f_i \to s_i}$ is cropped to 1 to ensure that it is a valid probability. Then $\nu_{f_i \to s_i}$ in (12) can be constructed by using $\pi_{f_i \to s_i}$ in (47). Subsequent message passing can therefore proceed.

F. Parameter Tuning

In this subsection, we discuss the choice of model parameters. The noise variance σ^2 is considered as a *priori* known in the previous sections. In practice, σ^2 can be learned under the expectation-maximization (EM) framework in a similar way as in the SBL [5].

Parameters α and β of the MRF can also be learned by the EM algorithm. However, we find in numerical experiments that the algorithm performs well when α and β are fixed to empirical values, and learning α and β by EM yields a marginal gain.

The update of the two parameters a and b of the check function f_i in (4) is more crucial. At the beginning of the iteration, since there is not much prior information of the sparse signal x, we set a and b to very small values (e.g., $a = b = 10^{-10}$) to ensure that the Gamma distribution part in f_i is noninformative. As the iteration proceeds, we update a and b to make the calculation of the messages more accurate. In (45) and (47), we notice that when calculating $\mu_{v_i \to q_i}$ and $\pi_{f_i \to s_i}$, only a/b (the mean of Gamma $(v_i; a, b)$) is used. So in the update of f_i , we fix a and only update b to adjust the mean of $Gamma(v_i; a, b)$. In iteration, each v_i receives an update $\mu_{q_i \to v_i}$ that is an estimate of v_i . From the probability model, we see that $\{v_i\}$ are drawn from (7). Thus, with high probability, v_i s with the largest values of $\mu_{q_i \rightarrow v_i}$ are nonzeros drawn from the Gamma distribution part $Gamma(v_i; a, b)$ in f_i . Based on this observation, we propose to update $\frac{a}{b}$ as follows. Let $\mu_1 \geq \mu_2 \geq \ldots \geq \mu_N$ be the reordered sequence of

Algorithm 3: VSP algorithm

Input: $\boldsymbol{y}, \boldsymbol{A}, \boldsymbol{a}, \boldsymbol{b}, T_{\text{out}}, T_{\text{in}}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho}, \boldsymbol{\vartheta}, \sigma^2$ 1 $t = 0, \mu_{v_i \to q_i} = 0, \forall i;$ while $t < T_{out}$ do 2 Call Algorithm 1 or 2 to update $\{\mu_{q_i \to v_i}\}$; 3 if t < T - 1 then 4 Compute κ based on (48); 5 $\forall i \in \{1, \ldots, N\}$, compute $\pi_{f_i \to s_i}$ based on (49); 6 $\forall i \in \{1, \ldots, N\}$, compute $\pi_{s_i \to f_i}$ based on (17); 7 $\forall i \in \{1, \ldots, N\}$, compute $\mu_{v_i \to g_i}$ based on (50); 8 9 t = t + 1;

10 Compute the posterior mean m of x in (26) based on $\{\mu_{\alpha} \rightarrow \mu_{\gamma}\}$:

$(P^{*}g_{i} \rightarrow v_{i}),$				
Output:	$\hat{m{x}}=m{m}$			

 $\mu_{g_1 \to v_1}, \ldots, \mu_{g_N \to v_N}$. We assume that those v_i s corresponding to the K' largest entries, i.e., $\{\mu_1, \ldots, \mu_{K'}\}$, are drawn from Gamma $(v_i; a, b)$. Then, by approximating the statistical mean by a sample mean, we obtain

$$\frac{a}{b} = \kappa \triangleq \frac{1}{K'} \sum_{h=1}^{K'} \mu_h.$$
(48)

For the choice of K', it is found in the experiments that the best performance is achieved when $K' = \lfloor \vartheta K \rceil$ with coefficient $\vartheta \in [1, 2]$, where $\lfloor u \rceil$ is a function which returns the nearest integer to $u \in \mathbb{R}$, and K is the number of nonzero elements in \boldsymbol{x} . Based on the above discussion, the calculations of $\pi_{f_i \to s_i}$ and $\mu_{v_i \to q_i}$ are changed to

$$\pi_{f_i \to s_i} = \min\left(\frac{\mu_{g_i \to v_i}}{\kappa}, 1\right) \tag{49}$$

$$\mu_{v_i \to g_i} = \kappa \pi_{s_i \to f_i}.$$
(50)

G. Overall Algorithm

The overall VSP algorithm is summarized in Algorithm 3. The input argument y is the noise-corrupted measurement; A is the measurement matrix; $\mu^{(0)}$ is the initial estimate of v; a, bare the shape and the rate parameter of the Gamma distribution; $T_{\rm out}$ is the number of the outer iteration; $T_{\rm in}$ is the number of iterations in the GD-based and ELBO-based solvers; α and β are the parameters of the Markov random field; ρ is the proportion of nonzero elements in x; ϑ is the coefficient; σ^2 is the variance of the Gaussian noise. In the outer iteration of VSP, the algorithm first calls the GD-based solver or the ELBObased solver to calculate an estimate $[\mu_{g_1 \rightarrow v_1}, \dots, \mu_{g_N \rightarrow v_N}]^T$ of v. The estimate $[\mu_{g_1 \to v_1}, \dots, \mu_{g_N \to v_N}]^T$ is then passed into the Markov random field for further processing to encourage the block sparsity of $\{\mu_{q_i \rightarrow v_i}\}$ (Lines 6 to 9 of Algorithm 3). The output $[\mu_{v_1 \to g_1}, \ldots, \mu_{v_N \to g_N}]^T$ is taken as the initial value of the GD-based/ELBO-based solver in the next iteration (Line 10 of Algorithm 3). In a sense, the role of the Markov random field in VSP is to iteratively adjust the initial variances of the GDbased/ELBO-based solver according to the block-sparse prior of x. The final estimate of x is given by $\{\mu_{q_i \to v_i}\}$ via (26). Since the state of the variances (i.e., $\{s_i\}$) plays a crutial role in message propagation, we refer to our proposed algorithm as variance state propagation.

Algorithm	Complexity ¹	Number of Blocks ²	Size of Blocks ²	Location of Blocks ²	Sparsity ²	Robustness ³	Description
OMP [5]	$\mathcal{O}(KMN)$	-	_	_	1	Yes	Naive greedy algorithms that require a large number of measurements for reliable recovery.
CoSaMP [6]	$\mathcal{O}(TMN)$	-	-	_	1	Yes	
SBL [8]	$O(TM^3)$	_	-	-	×	Yes	A Bayesian-based algorithm that repeatedly calculates the posterior distribution of the signal and updates the parameters of the model prior.
IHT [28]	$\mathcal{O}(TMN)$	-	_	_	1	Yes	A representative iterative thresholding algorithm.
Block-OMP [13]	_	✓	1	1	1	Yes	Improved greedy algorithms specified for
Block-CoSaMP [13]	-	1	1	1	1	Yes	block-sparse signal recovery. It requires
Struct-OMP [29]	-	\checkmark	×	×	1	Yes	strong prior knowledge.
CluSS-MCMC [15]	$\mathcal{O}(TN^2)$	×	×	x	x	Yes	A Bayesian based algorithm that adopts a Bernoulli-Gaussian hierarchical model as the prior. Since the posterior distribution can not be derived analytically, MCMC sampling is employed in inference.
PC-SBL [18]	$\mathcal{O}(TM^3)$	×	X	×	x	Yes	An improved algorithm specified for block- sparse signals based on the SBL framework. A new pattern-coupled Gaussian probability model is employed as the prior.
PCSBL-GAMP [30]	$\mathcal{O}(TMN)$	×	×	×	x	No	A modified PC-SBL algorithm which re- duces the complexity of PC-SBL by using message passing, but at the same time in- creases the sensitivity to the measurement matrix.
STCS [23]	$\mathcal{O}(TM^3)$	X	×	×	×	No	Message passing algorithms that have superior performance when the measurement matrix satisfies certain conditions, but have no performance
CGAMP [25]	$\mathcal{O}(TMN)$	×	×	×	×	No	guarantee when the conditions are not met.
VSP	$\mathcal{O}(TM^3)$	×	×	×	×	Yes	A new message-passing based algorithm that has near-optimal performance and is robust to the choice of the measurement matrix.

TABLE I: Properties of VSP and Other Popular Compressed Sensing Algorithms

¹ "Complexity" here is evaluated for a general measurement matrix. The complexities of some algorithms (such as STCS) can be reduced when the measurement matrix takes a special structure. "T" stands for the total number of iterations. Particularly, $T = T_{in}T_{out}$ for VSP.

² "\" denotes "necessary" for the corresponding algorithm. "\" denotes "unnecessary" for the corresponding algorithm. "-" denotes no consideration for the corresponding algorithm.

³ "Robustness" here means the robustness of the algorithm to the choice of the measurement matrix.

H. Further Discussions

The total complexity of the VSP algorithm consists of the implementation of Algorithm 1 (or Algorithm 2) and the message passing steps of Algorithm 3. The complexities of both Algorithms 1 and 2 are dominated by the calculation of Φ in (27). According to the Woodbury matrix identity, this $N \times N$ matrix inversion can be converted to an $M \times M$ matrix inversion, which requires $\mathcal{O}(M^3)$ flops per iteration. Consequently, the complexities of Algorithms 1 and 2 are both $\mathcal{O}(T_{\rm in}M^3)$. The calculation of the messages in steps 6–10 of Algorithm 3 requires the complexity of $\mathcal{O}(N)$. Therefore, by considering the outer iteration, the total complexity of the VSP algorithm is $\mathcal{O}(T_{\rm out}T_{\rm in}M^3)$.

Table I shows the complexity and the required prior information of VSP and some other popular compressed sensing algorithms. Compared with other methods, VSP has a significant advantage in its near-optimal performance and robustness to the measurement matrix, while its computational complexity is acceptable. VSP introduces variance variables in the probability model, resulting in a much lower correlation between messages passed on the factor graph. VSP thus inherits the superior performance of the message passing-based algorithms and maintains a good robustness to the measurement matrix. Further, by using the Markov random field to model the states of the variance variables, VSP well exploits the prior knowledge of block sparsity. The performance of VSP is examined in the next section.

IV. NUMERICAL RESULTS

We now carry out simulations to illustrate the performance of our proposed VSP algorithm. We first test the performance of the VSP algorithm using the GD-based solver (VSP-GD) and the VSP algorithm using the ELBO-based solver (VSP-ELBO) under a relatively simple environment, and then compare VSP with the other existing block-sparse signal recovery algorithms in several practical applications.



Fig. 2: NMSE performances of VSP-GD and VSP-ELBO under different K'. N = 50, K = 10, L = 1, M = 25, and $T_{out} = 2$. The NMSE of SBL is also provided as benchmark. (a) NMSEs of VSP-GD versus the SNR under different K'. For the GD–based solver, $T_{in} = 7000$. (b) NMSEs of VSP-ELBO versus the SNR under different K'. For the ELBO–based solver $T_{in} = 30$.



Fig. 3: NMSE performances of VSP-GD and VSP-ELBO under different T_{in} respectively. N = 50, K = 10, L = 1, M = 25, K' = 2K, and $T_{out} = 2$. (a) NMSEs of VSP-GD versus the SNR under different T_{in} . (b) NMSEs of VSP-ELBO versus the SNR under different T_{in} . The NMSE of SBL is also provided as a benchmark.

A. GD Solver vs. ELBO Solver

In this subsection we compare the performance of VSP-GD and VSP-ELBO under different parameter settings. To evaluate the recovery performance, we introduce the normalized mean square error (NMSE) metric, which is calculated by averaging normalized squared errors $||\hat{x} - x||_2^2/||x||_2^2$ over independent trials, where \hat{x} denotes an estimate of x. In our experiments, the block-sparse signals are generated in a similar way as in [18]. The sparse signal $x \in \mathbb{C}^N$ contains K nonzero coefficients partitioned into L blocks, and the location and the size of each block are randomly assigned. The block sizes $\{B_l\}_{l=1}^L$ are determined as follows: we randomly generate L positive random variables $\{r_l\}_{l=1}^L$ with their sum equal to one. Then we set $B_l = \lceil K\tau_l \rceil$ for the first L-1 blocks and $B_L = K - \sum_{l=1}^{L-1} B_l$ for the last block, where $\lceil x \rceil$ denotes the ceiling operator that gives the smallest integer no smaller than x. Similarly, we partition the N-dimensional vector into L super-blocks using the same set of values $\{r_l\}_{l=1}^L$, and place each of the L nonzero blocks into one unique super-block with a randomly generated starting position, where the starting position is carefully selected to prevent the nonzero block from going beyond the super-block. The nonzero coefficients and the elements of the measurement matrix $\{x_k\}_{k=1}^K$ are independently drawn from the standard complex Gaussian (SCG) distribution [31] with zero mean and unit variance. The given results are averaged by 200 independent trails.

We first test the impact of the choice of K' on the performance of VSP-GD and VSP-ELBO. In (48) we notice that when updating parameters of the check function $f_i = p(v_i|s_i)$, the sparsity K is needed to determine K'. In practice, accurate knowledge of K is a pretty strong prior that may be unavailable.



Fig. 4: NMSEs of the respective algorithms under Gaussian measurement matrices. (a) NMSEs versus the SNR. N = 200, K = 30, L = 1, and M = 75. (b) NMSEs versus the number of measurements. N = 200, K = 30, L = 1, and SNR = 20 dB.



Fig. 5: NMSEs of the respective algorithms under cropped-Hermitian measurement matrices. (a) NMSEs versus the SNR. N = 100, K = 20, L = 2, and M = 60. (b) NMSEs versus the number of measurements. N = 100, K = 20, L = 2, and SNR = 20 dB.

In many cases, we may only know the approximate number of nonzero elements in x. Therefore, we hope that the VSP algorithm is not sensitive to the value of K'. In our simulation, we test the sensitivity of VSP to K' by fixing K and adjusting ϑ . The NMSEs of VSP-GD and VSP-ELBO versus the signalnoise-ratio (SNR) are presented in Fig. 2. The SNR is defined as $20 \log\{||Ax||_2/\sigma\}$ in dB, where A is the measurement matrix in (2), and σ is the standard deviation of the complex Gaussian noise. Fig. 2(a) and Fig. 2(b) show the NMSE curves for VSP-GD and VSP-ELBO at different values of K', respectively. The NMSE curve of SBL is also provided for comparision. We see that when K' ranges from K to 2K, both VSP-GD and VSP-ELBO exhibit significant performance gains over the baseline SBL, and more importantly, the performance gains are generally not sensitive to the value of K'. In Fig. 2(a) we observe that, when the SNR is less than 10 dB, the NMSE performance of VSP-GD is relatively insensitive to K'. When the SNR is greater than 10 dB, a larger K' brings a slight gain. In Fig. 2(b) we also observe that a larger K' (1.6 $K \sim 2K$) makes the performance of VSP-ELBO slightly better. Comparing Fig. 2(a) and Fig. 2(b), we find that as the SNR increases, the gap between VSP-GD and SBL narrows, whereas VSP-ELBO does not suffer from this problem.

We now examine the impact of T_{in} for VSP. We choose K' = 2K and $T_{out} = 2$. The block-sparse signals and the measurement matrices are generated in the same way as in the previous experiment. Fig. 3 shows the NMSE of VSP-GD versus SNR with different T_{in} . The NMSE of SBL versus SNR is also provided for comparision. In Fig. 3(a), we observe that VSP-GD needs a large T_{in} ($T_{in} = 7000$) to ensure good performance. In contrast, VSP-ELBO achieves a significant NMSE gain over SBL at a relatively small T_{in} value ($T_{in} = 10$ in Fig. 3(b)).

Through the above two sets of experiments, we find that VSP-ELBO is superior to VSP-GD in terms of both recov-



Fig. 6: NMSEs of respective algorithms under concatenated-exponential-Gaussian measurement matrices. (a) NMSEs versus the SNR. N = 300, K = 50, L = 3, and M = 120. (b) NMSEs versus the number of measurements. N = 300, K = 50, L = 3, and SNR = 20 dB.

ery performance and computational complexity. In subsequent experiments for comparision with the other existing methods, "VSP" always refers to "VSP-ELBO" with $T_{out} = 2$, $T_{in} = 30$, and K' = 2K.

B. Synthetic Data

In this subsection we evaluate the recovery performance of the VSP for synthetic block-sparse signals. We consider three different measurement matrix structures to test the robustness of the VSP algorithm. Here, the block-sparse signals are generated in the same manner as described in Section IV-A. The existing algorithm for sparse signal recovery, including the orthogonal matching pursuit (OMP) [5], conventional sparse Bayesian learning (SBL) [8], pattern-coupled sparse Bayesian learning (PC-SBL), clustered Gaussian approximate message passing (CGAMP) [25], and structured turbo compressed sensing (STCS) [23] are taken into account for comparison.

The NMSEs of the respective algorithms versus the SNR and the number of measurements are depicted in Fig. 4, Fig. 5, and Fig. 6. The SNR is defined in the same way as in the preceding subsection. We also plot a genie bound as a benchmark, which is obtained by a linear minimum mean-square error (LMMSE) estimator with perfectly known non-zero positions of x. The results are averaged by 500 realizations.

Fig. 4 is obtained under complex Gaussian measurement matrices, namely, the elements of A are independently drawn from the SCG distribution. In Fig. 4(a) we observe that the NMSEs of STCS and CGAMP almost coincide with the genie bound predicted by the LMMSE. This is consistent with our expectation since A here is a right-rotationally invariant (RRI) matrix. The excellent performance of the message passing algorithms under this scenario has been previously confirmed in [23], [25]. From Fig. 4(a), we observe that although the performance (in NMSE) of VSP is not as good as CGAMP and STCS at low SNR, the NMSE of VSP can asymptotically approach the genie bound as the SNR increases. At the same time, VSP performs better than OMP, SBL, and PC- SBL throughout the entire observation range. In Fig. 4(b), we note that the NMSE of CGAMP almost coincides with the genie bound. STCS has a similar performance with CGAMP when a large number of measurements is avaiable, but exhibits instability when the number of measurements is less than 90. VSP performs significantly better than SBL and PC-SBL in the entire observation range, and its gap from the genie bound is always kept small (within 1 dB).

Fig. 5 is obtained under cropped-Hermitian measurement matrices, namely, the measurement matrix $A \in \mathbb{C}^{M \times N}$ in each independent trial is generated in the following manner. First we generate a square matrix $A_1 \in \mathbb{C}^{N \times N}$ from the SCG distribution and accordingly form a Hermitian matrix $A_2 = A_1 \times A_1^H$. The measurement matrix A consists of the first M rows of A_2 . Under this setting A is not a RRI matrix. In Fig. 5(a) we observe that, the performance of STCS deteriorates seriously compared to that in Fig. 4(a). The corresponding NMSE can not approach the genie bound any more. CGAMP performs even worse than STCS. The NMSE of CGAMP is not given in Fig. 5, since otherwise it will make the other curves indistinguishable. It is seen that the SBL-based compressed sensing algorithms still work well under this measurement matrix, and the proposed VSP is clearly the best among them. As the SNR increases, the NMSE curve of the VSP gradually approaches the genie bound. When the SNR is 30 dB, the NMSE gap between VSP and genie bound is within 1dB. In Fig. 5(b), we observe that the NMSE curve of VSP decreases smoothly as the number of measurements increases. In the entire observation range, VSP outperforms the other algorithms by a substantial margin.

Fig. 6 is obtained under concatenated-exponential-Gaussian measurement matrices: In each independent trial, the measurement matrix $A \in \mathbb{C}^{M \times N}$ is a concatenation of two matrices $A_3 \in \mathbb{C}^{M \times N/2}$ and $A_4 \in \mathbb{C}^{M \times N/2}$, i.e., $A = [A_3 A_4]$. Each element in A_3 is randomly drawn from the SCG distribution, and the real part and the imaginary part of each element in A_4 are randomly drawn from an exponential distribution with the rate = 3. A measurement matrix with such unevenly distributed



Fig. 7: Original image of handwritten digit "0" and the reconstructed images by SBL, PCSBL-GAMP, and VSP under the Gaussian measurement matrix. SNR = 10 dB.



Fig. 8: Original image of handwritten digit "3" and the reconstructed images by SBL, PCSBL-GAMP, and VSP under the concatenated-exponential measurement matrix. SNR = 10 dB.

energy is very unfriendly to message passing based algorithms. In Fig. 6, we see that the STCS does not work well under this circumstance. The performance of CGAMP is omitted for the same reason as in Fig. 5. We observe that VSP again surpasses the other algorithms in terms of both recovery ability and the amount of measurements required.

C. Handwritten Digits Image Recovery

We now test the proposed VSP algorithm on twodimensional block-sparse signals. We carry out experiments on two handwritten digit images (28×28 pixels) drawn from the MNIST data set [32]. Digit "0" with 176 (22.45% of total pixels) nonzero pixels and digit "3" with 200 (25.51% of total pixels) nonzero pixels are the 2nd and 8th samples in the MNIST training set, respectively. The gray values of each image are normalized to a range of [0, 1]. Most of the pixels in the image are zeros and the nonzero coefficients exhibit irregular block patterns. The compressed measurements are corrupted by an additive i.i.d. Gaussian noise, i.e., y = Ax + w, where the image is represented as a one-dimensional vector x. Here we compare the recovery performance of the proposed VSP with those of SBL and PCSBL-GAMP. The PCSBL-GAMP algorithm is a generalization of the PC-SBL for two-dimensional block-sparse signals and uses Gaussian approximate message passing techniques to reduce computational complexity. In our simulations we set M = 400. The SNR is set to 10 dB. Fig. 7 and Fig. 8 depict the original images and the reconstructed images under two different settings of the measurement matrix, respectively. Fig. 7 is obtained under a Gaussian measurement matrix, in which the elements are randomly drawn from a normal distribution. It can be observed that the proposed VSP algorithm provides the best visual quality with recognizable digit. The digit reconstructed by the SBL has a poor quality and can not even distinguish the boundary of the digit. The PC-SBL gives a clear boundary, but does not eliminate the

noise well. The VSP not only recovers the boundary sharply, but also significantly suppresses the noise. Fig. 8 is obtained under a concatenated-exponential measurement matrix that is a concatenation of two matrices A_5 and A_6 . Elements of A_5 and A_6 are randomly drawn from two exponential distributions with the rates = 3 and = 1, respectively. In Fig. 8, it is seen that the PCSBL-GAMP totally fails due to the sensitivity of the GAMP algorithm to the measurement matrix structure. We see that in both cases, the proposed VSP offers a clearly better image recovery quality as compared with the other methods.

D. Background Subtraction

Background subtraction, also known as foreground detection, is a technique used to automatically detect and track moving objects in videos from static cameras. Usually, the foreground interests are sparse in the spatial image domain. By exploiting this sparsity, the sparse foreground interests within a scene can be reconstructed by using compressed measurements, which improves the real-time performance of signal processing. Specifically, the idea is to reconstruct the foreground image from the noisy corrupted difference between the compressed measurements of the background image and the compressed measurements of the test image

$$\mathbf{y}_f = \mathbf{y}_t - \mathbf{y}_b + \mathbf{w} = \mathbf{A}(\mathbf{x}_t - \mathbf{x}_b) + \mathbf{w} = \mathbf{A}\mathbf{x}_f + \mathbf{w}$$
 (51)

where \mathbf{x}_t and \mathbf{x}_b represent the test and background images, respectively; \mathbf{y}_t and \mathbf{y}_b denote the compressed measurements of the test and background images, respectively; \mathbf{w} is the additive Gaussian noise; and \mathbf{x}_f is the foreground image to be recovered. In our experiments, we use the UCSD background subtraction data set [33]. The data set consists of 18 video sequences collected by static cameras. We choose the 70-th frame and the 86-th frame of the "rain" subset as the background image \mathbf{x}_b and the test image \mathbf{x}_f , respectively. The background image, the test image, and the foreground image are shown in the top line



Fig. 9: Top line from left to right: the background image x_b (the 70-th frame of the UCSD-rain background subtraction data set), the test image x_t (the 86-th frame of the UCSD-rain background subtraction date set), the foreground image x_f . Bottom line from left to right: foreground images reconstructed by SBL, PCSBL-GAMP, VSP-SBL under concatenated-exponential measurement matrix. SNR = 10 dB.

of Fig. 9. The foreground image is regarded as the groundtruth image. This foreground image, however, does not have a pure background since $\mathbf{x}_{f} = \mathbf{x}_{t} - \mathbf{x}_{b}$ is not an exactly sparse signal and contains many small nonzero components. In our experiments, the original images of 228×308 pixels are resized to 114×154 pixels. For the resized foreground image, we have a total number of 3294 coefficients (18.76% of total pixels) whose magnitudes are greater than 10^{-2} . Images reconstructed by the SBL, the PC-SBL, and the VSP are depicted in the bottom line of Fig. 9, where M = 8000 and SNR = 10 dB. The measurement matrix A is randomly generated with each entry independently drawn from a normal distribution. We see that our proposed PCSBL-GAMP method provides the best image quality with a clear appearance of the vehicle, whereas the object silhouettes recovered by the other methods are seriously disturbed by noise.

V. CONCLUSION

In this paper, we developed a new sparse Bayesian learning method for recovery of block-sparse signals. A novel hierarchical Gaussian prior was proposed to characterize the blocksparse patterns of the unknown signals. The core idea of our algorithm is to iteratively update the variances in the prior Gaussian distributions. A Markov random field is combined to model the state variables of the variances of the independent Gaussian distributions. The proposed MRF-combined hierarchical model is effective and flexible to cope with various kinds of block-sparse structures. Our algorithm was developed based on the message passing principle, where for messages that are difficult to calculate, we have designed reasonable methods to achieve approximate calculations. In addition, hyperparameters can be updated within the iterative process. Simulation results show that our proposed algorithm demonstrates a superior performance over the other existing popular methods for blocksparse signal recovery.

APPENDIX A GRADIENT CALCULATION

The partial derivative of $\chi(\boldsymbol{y}, \boldsymbol{v})$ in (30) w.r.t. v_i is given by

$$\frac{\partial \chi}{\partial v_i} = -\frac{\partial \boldsymbol{m}^H \boldsymbol{\Phi}^{-1} \boldsymbol{m}}{\partial v_i} - \frac{\partial \ln |\boldsymbol{\Phi}|}{\partial v_i} + \frac{\partial \ln v_i}{v_i}.$$
 (52)

Furthermore,

$$\frac{\partial \boldsymbol{m}^{H} \boldsymbol{\Phi}^{-1} \boldsymbol{m}}{\partial v_{i}} = \sigma^{-4} \boldsymbol{y}^{H} \boldsymbol{A} \frac{\partial \boldsymbol{\Phi}}{\partial v_{i}} \boldsymbol{A}^{H} \boldsymbol{y}$$
(53)

and

$$\frac{\partial \mathbf{\Phi}}{\partial v_i} = \frac{\partial \left(\mathbf{D}^{-1} + \sigma^{-2} \mathbf{A}^H \mathbf{A} \right)^{-1}}{\partial v_i}$$
$$= -\mathbf{\Phi} \frac{\partial \left(\mathbf{D}^{-1} + \sigma^{-2} \mathbf{A}^H \mathbf{A} \right)}{\partial v_i} \mathbf{\Phi}$$
$$= \mathbf{\Phi} \mathbf{E}_i \mathbf{\Phi}. \tag{54}$$

 $\frac{\partial \ln |\Phi|}{\partial v_i}$ can be calculated by

$$\frac{\partial \ln |\mathbf{\Phi}|}{\partial v_i} = \frac{1}{|\mathbf{\Phi}|} \cdot \frac{\partial |\mathbf{\Phi}|}{\partial v_i}$$
$$= \operatorname{Tr} \left[\mathbf{\Phi}^{-1} \frac{\partial \mathbf{\Phi}}{\partial v_i} \right].$$
(55)

Then, by plugging (53)–(55) into (52), we obtain (33).

APPENDIX B PROOF OF PROPOSITION 1

From (41), we obtain

$$\mathcal{L}(\mu_{g_i \to v_i}, q(\boldsymbol{x})) \ge \mathcal{L}(\mu_{v_i \to g_i}, q(\boldsymbol{x})),$$
(56)

where $q(\boldsymbol{x})$ is given by (42). Since $D_{\mathrm{KL}}(p(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{v})|_{\mathcal{S}_{\backslash i},v_i=\mu_{g_i}\to v_i}||p(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{v})|_{\mathcal{S}})) \geq 0$, we obtain

$$\eta(\mu_{g_i \to v_i}) = \mathcal{L}(\mu_{g_i \to v_i}, q(\boldsymbol{x})) + D_{\mathrm{KL}}(q(\boldsymbol{x}) || p(\boldsymbol{x} | \boldsymbol{y}, \boldsymbol{v})|_{\mathcal{S}_{\backslash i}, v_i = \mu_{g_i \to v_i}}))$$
(57a)

$$\geq \mathcal{L}(\mu_{v_i \to g_i}, q(\boldsymbol{x})) + D_{\mathrm{KL}}(q(\boldsymbol{x}) || q(\boldsymbol{x}))$$
(57b)

$$=\eta(\mu_{v_i\to g_i}).\tag{57c}$$

By noting the monotonicity of the logarithm function, we arrive at (43). \Box

APPENDIX C

PROOF OF PROPOSITION 2

With $q(\boldsymbol{x})$ in (42), we obtain

$$\mathcal{L}(v_i, q(\boldsymbol{x})) = \int_{\boldsymbol{x}} q(\boldsymbol{x}) \ln \frac{p(\boldsymbol{y}, \boldsymbol{x} | \boldsymbol{v})|_{\mathcal{S} \setminus i}}{q(\boldsymbol{x})}$$

=
$$\int_{\boldsymbol{x}} p(\boldsymbol{x} | \boldsymbol{y}, \boldsymbol{v})|_{\mathcal{S}} \ln p(\boldsymbol{x} | \boldsymbol{v})|_{\mathcal{S} \setminus i} - \int_{\boldsymbol{x}} q(\boldsymbol{x}) \ln q(\boldsymbol{x})$$

+
$$\int_{\boldsymbol{x}} p(\boldsymbol{x} | \boldsymbol{y}, \boldsymbol{v})|_{\mathcal{S}} \ln p(\boldsymbol{y} | \boldsymbol{x})$$

=
$$Q(v_i) + C_1,$$
(58)

where C_1 is a constant independent of v_i and

$$Q(v_i) = \int_{\boldsymbol{x}} p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{v})|_{\mathcal{S}} \ln p(\boldsymbol{x}|\boldsymbol{v})|_{\mathcal{S}\setminus i}.$$
 (59)

Thus (41) can be recast as

$$\mu_{g_i \to v_i} = \operatorname*{arg\,max}_{v_i} Q(v_i). \tag{60}$$

Plugging $p(\boldsymbol{x}|\boldsymbol{v})$ in (10b) into (59) leads to

$$Q(v_i) = -\ln v_i - \frac{1}{x_i} \int_{\boldsymbol{x}} p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{v})|_{\mathcal{S}} |x_i|^2 + C_2 \qquad (61)$$

where C_2 is another constant independent of v_i . Further we notice

$$\int_{\boldsymbol{x}} p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{v})|_{\mathcal{S}} |x_i|^2 = |m_i|^2 + \phi_{i,i}$$
(62)

where m_i denotes the *i*-th entry of m in (26), and $\phi_{i,i}$ denotes the *i*-th diagonal element of the covariance matrix Φ in (27). Taking the derivative of (61) with respect to v_i and setting the result to zero, we obtain (44).

REFERENCES

- M. Lustig, D. Donoho, J. Santos, and J. Pauly, "Compressed sensing MRI," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 72–82, Mar. 2008.
- [2] U. Gamper, P. Boesiger, and S. Kozerke, "Compressed sensing in dynamic MRI," Magn. Reson. Med., vol. 59, no. 2, pp. 365–373, 2008.
- [3] W. U. Bajwa, J. Haupt, A. M. Sayeed, and R. Nowak, "Compressed channel sensing: A new approach to estimating sparse multipath channels," *Proc. IEEE*, vol. 98, no. 6, pp. 1058–1076, Jun. 2010.
- [4] C. R. Berger, S. Zhou, J. C. Preisig, and P. Willett, "Sparse channel estimation for multicarrier underwater acoustic communication: From subspace methods to compressed sensing," in OCEANS 2009-EUROPE. IEEE, May 2009, pp. 1–8.
- [5] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.
- [6] D. Needell and J. A. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Appl. Comput. Harmon. Anal.*, vol. 26, no. 3, pp. 301–321, 2009.

- [7] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Rev.*, vol. 43, no. 1, pp. 129–159, Jan. 2001.
- [8] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," J. Mach. Learn. Res., vol. 1, no. Jun, pp. 211–244, 2001.
- [9] A. Liu, V. K. N. Lau, and W. Dai, "Exploiting burst-sparsity in massive MIMO with partial channel support information," *IEEE Trans. Wireless Commun.*, vol. 15, no. 11, pp. 7820–7830, Nov. 2016.
- [10] Y. Han, J. Lee, and D. J. Love, "Compressed sensing-aided downlink channel training for FDD massive MIMO systems," *IEEE Trans. Commun.*, vol. 65, no. 7, pp. 2852–2862, Jul. 2017.
- [11] Y. C. Eldar and M. Mishali, "Robust recovery of signals from a structured union of subspaces," *IEEE Trans. Inf. Theory*, vol. 55, no. 11, pp. 5302– 5316, Nov. 2009.
- [12] J. M. Kim, O. K. Lee, and J. C. Ye, "Compressive MUSIC: Revisiting the link between compressive sensing and array signal processing," *IEEE Trans. Inf. Theory*, vol. 58, no. 1, pp. 278–301, Jan. 2012.
- [13] Y. C. Eldar, P. Kuppinger, and H. Bolcskei, "Block-sparse signals: Uncertainty relations and efficient recovery," *IEEE Trans. Signal Process.*, vol. 58, no. 6, pp. 3042–3054, Jun. 2010.
- [14] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. Royal Stat. Soc. B*, vol. 68, no. 1, pp. 49–67, Feb. 2006.
- [15] L. Yu, H. Sun, J. Barbot, and G. Zheng, "Bayesian compressive sensing for cluster structured sparse signals," *Signal Process.*, vol. 92, no. 1, pp. 259–269, Jan. 2012.
- [16] T. Peleg, Y. C. Eldar, and M. Elad, "Exploiting statistical dependencies in sparse representations for signal recovery," *IEEE Trans. Signal Process.*, vol. 60, no. 5, pp. 2286–2303, May 2012.
- [17] Z. Zhang and B. D. Rao, "Extension of SBL algorithms for the recovery of block sparse signals with intra-block correlation," *IEEE Trans. Signal Process.*, vol. 61, no. 8, pp. 2009–2015, Apr. 2013.
- [18] 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, Jan. 2015.
- [19] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 106, no. 45, pp. 18914–18919, Oct. 2009.
- [20] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," in 2011 Proc. IEEE Int. Symp. Inform. Thy. IEEE, 2011, pp. 2168–2172.
- [21] J. P. Vila and P. Schniter, "Expectation-maximization Gaussian-mixture approximate message passing," *IEEE Trans. Signal Process.*, vol. 61, no. 19, pp. 4658–4672, Oct. 2013.
- [22] J. Ma, X. Yuan, and L. Ping, "Turbo compressed sensing with partial dft sensing matrix," *IEEE Signal Process. Lett.*, vol. 22, no. 2, pp. 158–161, 2014.
- [23] L. Chen, A. Liu, and X. Yuan, "Structured turbo compressed sensing for massive MIMO channel estimation using a Markov prior," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4635–4639, May 2018.
- [24] L. Chena, A. Liuc, X. Kuai, and X. Yuan, "Structured turbo compressed sensing for downlink massive MIMO-OFDM channel estimation," *IEEE Trans. Wireless Commun.*, pp. 1–1, 2019.
- [25] Z.-Q. He, X. Yuan, and L. Chen, "Super-resolution channel estimation for massive MIMO via clustered sparse Bayesian learning," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 6156–6160, Jun. 2019.
- [26] S. Som and P. Schniter, "Approximate message passing for recovery of sparse signals with Markov-random-field support structure," in *Int'l Conf. Mach. Learn*, 2011.
- [27] Z. Xue, J. Ma, and X. Yuan, "Denoising-based turbo compressed sensing," *IEEE Access*, vol. 5, pp. 7193–7204, 2017.
- [28] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Appl. Comput. Harmon. Anal.*, vol. 27, no. 3, pp. 265– 274, 2009.
- [29] J. Huang, T. Zhang, and D. Metaxas, "Learning with structured sparsity," J. Mach. Learn. Res., vol. 12, no. Nov, pp. 3371–3412, 2011.
- [30] J. Fang, L. Zhang, and H. Li, "Two-dimensional pattern-coupled sparse Bayesian learning via generalized approximate message passing," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2920–2930, Jun. 2016.
- [31] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, 2005.
- [32] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278– 2324, 1998.
- [33] V. Mahadevan and N. Vasconcelos, "Spatiotemporal saliency in dynamic scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 171– 177, Jan. 2010.