Two-Dimensional Pattern-Coupled Sparse Bayesian Learning via Generalized Approximate Message Passing

Jun Fang, *Member, IEEE*, Lizao Zhang, and Hongbin Li, *Senior Member, IEEE*

***Abstract*— We consider the problem of recovering 2D block-sparse signals with unknown cluster patterns. The 2D block-sparse patterns arise naturally in many practical applications, such as foreground detection and inverse synthetic aperture radar imaging. To exploit the underlying block-sparse structure, we propose a 2D pattern-coupled hierarchical Gaussian prior model. The proposed pattern-coupled hierarchical Gaussian prior model imposes a soft coupling mechanism among neighboring coefficients through their shared hyperparameters. This coupling mechanism enables effective and automatic learning of the underlying irregular cluster patterns, without requiring any *a priori* knowledge of the block partition of sparse signals. We develop a computationally efficient Bayesian inference method, which integrates the generalized approximate message passing technique with the proposed prior model. Simulation results show that the proposed method offers competitive recovery performance for a range of 2D sparse signal recovery and image processing applications over the existing method, meanwhile achieving a significant reduction in the computational complexity.**

***Index Terms*— Pattern-coupled sparse Bayesian learning, block-sparse structure, expectation-maximization (EM), generalized approximate message passing (GAMP).**

I. Introduction

COMPRESSED sensing is a recently emerged technique for signal sampling and data acquisition which enables to recover sparse signals from undersampled linear measurements

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

where $\boldsymbol{A} \in \mathbb{R}^{M\times N}$ is a sampling matrix with $M \ll N$, $\boldsymbol{x}$ denotes an N-dimensional sparse signal, and $\boldsymbol{w}$ denotes the additive noise. The problem has been extensively studied and a variety of algorithms, e.g. the orthogonal matching pursuit (OMP) algorithm [1], the basis pursuit (BP) method [2], the iterative reweighted ℓ_1 and ℓ_2 algorithms [3], and the sparse Bayesian learning method [4], [5] were proposed. In many practical applications, in addition to the sparse structure, sparse signals may exhibit two-dimensional cluster patterns that can be utilized to enhance the recovery performance. For example, the target of interest in the synthetic aperture radar/inverse synthetic aperture radar (SAR/ISAR) images often demonstrates continuity in both the range and cross-range domains [6]. In video surveillance, the foreground image exhibits a cluster pattern since the foreground objects (humans, cars, text etc.) generally occupy a small continuous region of the scene [7]. Besides these, block-sparsity is also present in temporal observations of a time-varying block-sparse signal whose support varies slowly over time [8].

Analyses [9]–[11] show that exploiting the inherent block-sparse structure not only leads to relaxed conditions for exact reconstruction, but also helps improve the recovery performance considerably. A number of algorithms have been proposed for recovering block-sparse signals over the past few years, e.g., block-OMP [11], mixed ℓ_2/ℓ_1 norm-minimization [9], group LASSO [12], model-based CoSaMP [10], and block-sparse Bayesian learning [13], [14]. These algorithms, however, require *a priori* knowledge of the block partition (e.g. the number of blocks and location of each block) such that the coefficients in each block are grouped together and enforced to share a common sparsity pattern. In practice, the prior information about the block partition of sparse signals is often unavailable, especially for two-dimensional signals since the block partition of a two-dimensional signal involves not only the location but also the shape of each block. For example, foreground images have irregular and unpredictable cluster patterns which are very difficult to be estimated *a priori*. To address this difficulty, a few sophisticated Bayesian methods which do not need the knowledge of the block partition were developed. In [15], a "spike-and-slab" prior model was proposed, where by introducing dependencies among mixing weights, the prior model has the potential to encourage sparsity and promote a tree structure simultaneously. This "spike-and-slab" prior model was later extended to accommodate block-sparse signals [6], [16]. Nevertheless, for the "spike-and-slab" prior introduced in [15] and [16], the posterior distribution cannot be derived analytically, and a Markov chain Monte Carlo (MCMC) sampling method has to be employed for Bayesian inference. In [17] and [18], a graphical prior, also referred to as the

Manuscript received May 22, 2015; revised October 14, 2015 and February 17, 2016; accepted April 15, 2016. Date of publication April 20, 2016; date of current version May 4, 2016. This work was supported by the National Science Foundation of China under Grant U1530154, Grant 61428103, Grant 61522104, and the National Science Foundation under Grant ECCS-1408182. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Christopher Wyatt.

J. Fang and L. Zhang are with the National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: junfang@uestc.edu.cn; 15196637434@163.com).

H. Li is with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030 USA (e-mail: hongbin.li@stevens.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2016.2556582

"Boltzmann machine", is employed as a prior on the sparsity support in order to induce statistical dependencies between atoms. With such a prior, the maximum a posterior (MAP) estimator requires an exhaustive search over all possible sparsity patterns. To overcome the intractability of the combinatorial search, a greedy method [17] and a variational mean-field approximation method [18] were developed to approximate the MAP. In [19], to cope with the unknown cluster pattern, an expanded model is employed by assuming that the original sparse signal is a superposition of a number of overlapping blocks, and the coefficients in each block share the same sparsity pattern. Conventional block sparse Bayesian learning algorithms such as those in [14] can then be applied to the expanded model.

A. Contributions of This Work

Despite the above efforts, most existing block-sparse recovery methods have limited capability in handling irregular cluster patterns or incur a prohibitive computational complexity, especially for two-dimensional (2D) signals whose overall dimension is usually large. To address these issues, in this paper, we develop a computationally efficient Bayesian method for 2D block-sparse signal recovery. The proposed method is able to recover irregularly clustered sparse signals by exploiting the underlying block-sparse structure, while without requiring the knowledge of the block partition of sparse signals. More specifically, the contributions of this work are mainly from the following two aspects.

- Firstly, based on our previous work [20], we propose a generalized pattern-coupled hierarchical Gaussian prior model for 2D sparse signal recovery. The proposed pattern-coupled hierarchical Gaussian prior model offers a soft coupling mechanism between each sample of the sparse signal and its neighboring samples through their shared hyperparameters. This coupling mechanism enables effective and automatic learning of the underlying irregular cluster patterns, thus circumventing the need for the prior knowledge of block partition of 2D sparse signals, which is difficult to obtain in practice.
- Secondly, to reduce the computational complexity, we develop an efficient Bayesian inference method which integrates the generalized approximate message passing (GAMP) technique with the proposed prior model. The algorithm is developed within an expectation-maximization (EM) framework, using the GAMP to efficiently compute approximations of the posterior distributions of the hidden variables. The hyperparameters associated with the hierarchical Gaussian prior are learned by iteratively maximizing the Q-function which is calculated based on the posterior approximations obtained from the GAMP.

Simulation results show that the proposed method offers competitive recovery performance for 2D block-sparse signals as compared with existing methods, meanwhile achieving a significant reduction in computational complexity.

The rest of the paper is organized as follows. In Section II, we introduce a 2D pattern coupled hierarchical Gaussian framework to model the sparse prior and the pattern dependencies among the neighboring coefficients. In Section III, a GAMP-based EM algorithm is developed to obtain the maximum a posterior (MAP) estimate of the hyperparameters, along with the posterior distribution of the sparse signal. Simulation results are provided in Section IV, followed by concluding remarks in Section V.

II. Bayesian Model

We consider the problem of recovering a two-dimensional block-sparse signal $\boldsymbol{X} \in \mathbb{R}^{Q\times L}$ from compressed noisy measurements

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{X}) + \boldsymbol{w} \tag{2}$$

where $\boldsymbol{y} \in \mathbb{R}^M$ denotes the compressed measurement vector, $\boldsymbol{f}(\cdot)$ is a linear map: $\mathbb{R}^{Q\times L} \to \mathbb{R}^M$, with $M \ll N \triangleq QL$, and $\boldsymbol{w} \in \mathbb{R}^M$ is an additive multivariate Gaussian noise with zero mean and covariance matrix $\sigma^2\boldsymbol{I}$. Let $\boldsymbol{x} \triangleq \mathrm{vec}(\boldsymbol{X})$, the linear map $\boldsymbol{f}(\boldsymbol{X})$ can generally be expressed as

$$\boldsymbol{f}(\boldsymbol{X}) = \boldsymbol{A}\boldsymbol{x} \tag{3}$$

where $\boldsymbol{A} \in \mathbb{R}^{M\times N}$ denotes the measurement matrix. In the special case where $\boldsymbol{f}(\boldsymbol{X}) = \mathrm{vec}(\boldsymbol{B}\boldsymbol{X})$, then we have $\boldsymbol{A} \triangleq \boldsymbol{I} \otimes \boldsymbol{B}$, in which $\otimes$ stands for the Kronecker product. The above model (2) arises in image applications where signals are multi-dimensional in nature, or in the scenario where multiple snapshots of a time-varying sparse signal are available. In these applications, signals usually exhibit two-dimensional cluster patterns that can be utilized to improve the recovery accuracy. To leverage the underlying block-sparse structures, we introduce a 2D pattern-coupled Gaussian prior model which is a generalization of our previous work [20]. Before proceeding, we provide a brief review of the conventional hierarchical Gaussian prior model [4], and some of its extensions.

A. Review of Conventional Gaussian Prior Model

For ease of exposition, we consider the prior model for the two-dimensional signal $\boldsymbol{X}$ instead of its one-dimensional form $\boldsymbol{x}$. Let $x_{q,l}$ denote the (q,l)th entry of $\boldsymbol{X}$. In the conventional sparse Bayesian learning framework [4], a two-layer hierarchical Gaussian prior was employed to promote the sparsity of the solution. In the first layer, coefficients $\{x_{q,l}\}$ of $\boldsymbol{X}$ are assigned a Gaussian prior distribution

$$p(\boldsymbol{X}|\boldsymbol{\alpha}) = \prod_{q=1}^{Q}\prod_{l=1}^{L}\mathcal{N}(x_{q,l}|0,\alpha_{q,l}^{-1}) \tag{4}$$

where $\alpha_{q,l}$ is a non-negative hyperparameter controlling the sparsity of the coefficient $x_{q,l}$. The second layer specifies Gamma distributions as hyperpriors over the hyperparameters $\boldsymbol{\alpha} \triangleq \{\alpha_{q,l}\}$, i.e.

$$p(\boldsymbol{\alpha}) = \prod_{q=1}^{Q}\prod_{l=1}^{L}\mathrm{Gamma}(\alpha_{q,l}|a,b) \tag{5}$$

As discussed in [4], for properly chosen a and b, this hyperprior allows the posterior mean of $\alpha_{q,l}$ to become arbitrarily large. As a consequence, the associated coefficient $x_{q,l}$

will be driven to zero, thus yielding a sparse solution. This conventional hierarchical model, however, does not encourage structured-sparse solutions since the sparsity of each coefficient is determined by its own hyperparameter and the hyperparameters are independent of each other. In [13] and [14], the above hierarchical model was generalized to deal with block-sparse signals, in which a group of coefficients sharing the same sparsity pattern are assigned a multivariate Gaussian prior parameterized by a common hyperparameter. Nevertheless, this model requires the knowledge of the block partition to determine which coefficients should be grouped and assigned a common hyperparameter.

B. Proposed 2D Pattern-Coupled Hierarchical Model

To exploit the 2D block-sparse structure, we utilize the fact that the sparsity patterns of neighboring coefficients are statistically dependent. To capture the pattern dependencies among neighboring coefficients, the Gaussian prior for each coefficient $x_{q,l}$ not only involves its own hyperparameter $\alpha_{q,l}$, but also its immediate neighbor hyperparameters. Specifically, a prior over $\boldsymbol{X}$ is given by

$$p(\boldsymbol{X}|\boldsymbol{\alpha}) = \prod_{q=1}^{Q}\prod_{l=1}^{L}\mathcal{N}(x_{q,l}|0, \delta_{q,l}^{-1}) \tag{6}$$

where

$$\delta_{q,l} \triangleq \alpha_{q,l} + \beta \sum_{(i,j)\in N_{(q,l)}} \alpha_{i,j} \tag{7}$$

in which $N_{(q,l)}$ denotes the neighborhood of the grid point (q, l). Here we define $N_{(q,l)} \triangleq \{(q, l-1), (q, l+1), (q-1, l), (q+1, l)\}$.[1] Note that depending on different application scenarios, the definition of $N_{(q,l)}$ may be modified to capture different kinds of pattern dependencies. For example, to exploit the pattern dependencies among coefficients in rows, we can define $N_{(q,l)} \triangleq \{(q-1, l), (q+1, l)\}$. $\beta \in [0, 1]$ is a parameter indicating the pattern relevance between the coefficient $x_{q,l}$ and its neighboring coefficients. Clearly, this model is an extension of our previous prior model [20] to the two-dimensional case. When $\beta = 0$, the prior model (6) reduces to the conventional sparse Bayesian learning model. When $\beta > 0$, we see that the sparsity of each coefficient $x_{q,l}$ is not only controlled by the hyperparameter $\alpha_{q,l}$, but also by the neighboring hyperparameters $S_{\alpha_{q,l}} \triangleq \{\alpha_{i,j}|(i, j) \in N_{(q,l)}\}$. The coefficient $x_{q,l}$ will be driven to zero if $\alpha_{q,l}$ or any of its neighboring hyperparameters goes to infinity. In other words, suppose $\alpha_{q,l}$ approaches infinity, then not only its corresponding coefficient $x_{q,l}$ will be driven to zero, the neighboring coefficients $S_{x_{q,l}} \triangleq \{x_{i,j}|(i, j) \in N_{(q,l)}\}$ will decrease to zero as well. We see that the sparsity patterns of neighboring coefficients are related to each other through their shared hyperparameters. On the other hand, for any pair of neighboring coefficients, each of them has its own hyperparameters that are not shared by the other coefficient. Hence, no coefficients are pre-specified to share a common sparsity pattern, which enables the prior to provide flexibility to model any block-sparse structures.

Following [4], we use Gamma distributions as hyperpriors over the hyperparameters $\{\alpha_{q,l}\}$, i.e.

$$p(\boldsymbol{\alpha}) = \prod_{q=1}^{Q}\prod_{l=1}^{L}\text{Gamma}(\alpha_{q,l}|a, b) \tag{8}$$

where we set $a > 1$, and $b = 10^{-6}$. The choice of a will be elaborated later in our paper. Also, the noise variance $\sigma^2 \triangleq 1/\gamma$ is assumed unknown, and to estimate this parameter, we place a Gamma hyperprior over γ, i.e.

$$p(\gamma) = \text{Gamma}(\gamma|c, d) \tag{9}$$

where we set $c = 1$ and $d = 10^{-6}$.

III. Proposed Algorithm

We now proceed to perform Bayesian inference for the proposed pattern-coupled hierarchical model. The following model is considered since the linear map $\boldsymbol{f}(\boldsymbol{X})$ can be expressed as $\boldsymbol{f}(\boldsymbol{X}) = \boldsymbol{A}\boldsymbol{x}$

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w} \tag{10}$$

We first translate the prior for the two-dimensional signal $\boldsymbol{X}$ to a prior for its one-dimensional form $\boldsymbol{x}$. From (6), the prior over $\boldsymbol{x}$ can be expressed as

$$p(\boldsymbol{x}|\boldsymbol{\alpha}) = \prod_{n=1}^{N}\mathcal{N}(x_n|0, \eta_n^{-1}) \tag{11}$$

where

$$\eta_n \triangleq \alpha_n + \beta \sum_{i\in N_{(n)}} \alpha_i \tag{12}$$

in which $N_{(n)}$ denotes the neighbors of the point (q, l) on the two-dimensional grid, i.e. $N_{(n)} \triangleq \{(l-2)Q + q, lQ + q, (l-1)Q + q - 1, (l-1)Q + q + 1\}$.[2] The relation between n and (q, l) is given by $n = (l-1)Q + q$, that is, $l = \lceil n/Q \rceil$, and $q = n \mod Q$, in which $\lceil\cdot\rceil$ denotes the ceiling operator. Note that for notational convenience, we, with a slight abuse of notation, use x_n to denote the nth entry of $\boldsymbol{x}$ and α_n to denote the hyperparameter associated with the coefficient x_n. Also, let $\boldsymbol{\alpha} \triangleq \{\alpha_n\}$ since its exact meaning remains unaltered. From (8), we have

$$p(\boldsymbol{\alpha}) = \prod_{n=1}^{N}\text{Gamma}(\alpha_n|a, b) \tag{13}$$

An expectation-maximization (EM) algorithm can be developed for learning the sparse signal $\boldsymbol{x}$ as well as the hyperparameters $\{\boldsymbol{\alpha}, \gamma\}$. In the EM formulation, the signal $\boldsymbol{x}$ is treated as hidden variables, and we iteratively maximize a lower bound on the posterior probability $p(\boldsymbol{\alpha}, \gamma|\boldsymbol{y})$ (this lower bound is also referred to as the Q-function). Briefly speaking, the algorithm alternates between an E-step and a M-step. In the E-step, we need to compute the posterior distribution of $\boldsymbol{x}$ conditioned on

[1] Note that for edge grid points, they only have two or three immediate neighboring points, in which case the definition of $N_{(q,l)}$ changes accordingly.

[2] For edge grid points which have only two or three neighboring points, the definition of $N_{(n)}$ changes accordingly.

the observed data and the hyperparameters estimated from the tth iteration, i.e.

$$p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha}^{(t)}, \gamma^{(t)}) \propto p(\boldsymbol{x}|\boldsymbol{\alpha}^{(t)}) p(\boldsymbol{y}|\boldsymbol{x}, \gamma^{(t)}) \tag{14}$$

It can be readily verified that the posterior $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha}^{(t)}, \gamma^{(t)})$ follows a Gaussian distribution with its mean and covariance matrix given respectively by

$$\boldsymbol{\mu} = \gamma^{(t)} \boldsymbol{\Phi} \boldsymbol{A}^T \boldsymbol{y} \tag{15}$$

$$\boldsymbol{\Phi} = (\gamma^{(t)} \boldsymbol{A}^T \boldsymbol{A} + \boldsymbol{D})^{-1} \tag{16}$$

where $\boldsymbol{D} \triangleq \text{diag}(\eta_1^{(t)}, \ldots, \eta_N^{(t)})$. The Q-function can then be computed. In the M-step, we maximize the Q-function with respect to the hyperparameters $\{\boldsymbol{\alpha}, \gamma\}$.

It can be seen that the EM algorithm, at each iteration, requires to update the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha}^{(t)}, \gamma^{(t)})$, which involves computing an $N \times N$ matrix inverse. Thus the EM-based algorithm has a computational complexity of $\mathcal{O}(N^3)$ flops, and therefore is not suitable for many real-world applications involving large dimensions. In the following, we will develop a computationally efficient algorithm by resorting to the generalized approximate message passing (GAMP) technique [21]. GAMP is a very-low-complexity Bayesian iterative technique recently developed [21] for obtaining an approximation of the marginal posterior distribution $p(x_n|\boldsymbol{y}, \boldsymbol{\alpha}^{(t)}, \gamma^{(t)}), \forall n$. It therefore can naturally be embedded within the EM framework to replace the computation of the true posterior distribution. From GAMP's point of view, the hyperparameters $\{\boldsymbol{\alpha}, \gamma\}$ are considered as known. The hyperparameters can be updated in the M-step based on the approximate posterior distribution of $\boldsymbol{x}$. We now proceed to derive the GAMP algorithm for the pattern-coupled Gaussian hierarchical prior model.

A. Pattern-Coupled Hierarchical Gaussian GAMP

GAMP was developed in a message passing-based framework. By using central-limit-theorem approximations, message passing between variable nodes and factor nodes can be greatly simplified, and the loopy belief propagation on the underlying factor graph can be efficiently performed. As noted in [22] and [21], the central-limit-theorem approximations become exact in the large-system limit under i.i.d. zero-mean sub-Gaussian $\boldsymbol{A}$.

For notational convenience, let $\boldsymbol{\theta} \triangleq \{\boldsymbol{\alpha}, \gamma\}$ denote the hyperparameters. Firstly, GAMP approximates the true marginal posterior distribution $p(x_n|\boldsymbol{y}, \boldsymbol{\theta})$ by

$$\hat{p}(x_n|\boldsymbol{y}, \hat{r}_n, \tau_n^r, \boldsymbol{\theta}) = \frac{p(x_n|\boldsymbol{\theta}) \mathcal{N}(x_n|\hat{r}_n, \tau_n^r)}{\int_x p(x_n|\boldsymbol{\theta}) \mathcal{N}(x_n|\hat{r}_n, \tau_n^r)} \tag{17}$$

where $\hat{r}_n$ and τ_n^r are quantities iteratively updated during the iterative process of the GAMP algorithm. Here, we have dropped their explicit dependence on the iteration number k for simplicity. Substituting (11) into (17), it can be easily verified that the approximate posterior $\hat{p}(x_n|\boldsymbol{y}, \hat{r}_n, \tau_n^r, \boldsymbol{\theta})$ follows a Gaussian distribution with its mean and variance given respectively as

$$\mu_n^x \triangleq \frac{\hat{r}_n}{1 + \eta_n \tau_n^r} \tag{18}$$

$$\phi_n^x \triangleq \frac{\tau_n^r}{1 + \eta_n \tau_n^r} \tag{19}$$

Another approximation is made to the noiseless output $z_m \triangleq \boldsymbol{a}_m^T \boldsymbol{x}$, where $\boldsymbol{a}_m^T$ denotes the mth row of $\boldsymbol{A}$. GAMP approximates the true marginal posterior $p(z_m|\boldsymbol{y}, \boldsymbol{\theta})$ by

$$\hat{p}(z_m|\boldsymbol{y}, \hat{p}_m, \tau_m^p, \boldsymbol{\theta}) = \frac{p(y_m|z_m, \boldsymbol{\theta}) \mathcal{N}(z_m|\hat{p}_m, \tau_m^p)}{\int_z p(y_m|z_m, \boldsymbol{\theta}) \mathcal{N}(z_m|\hat{p}_m, \tau_m^p)} \tag{20}$$

where $\hat{p}_m$ and τ_m^p are quantities iteratively updated during the iterative process of the GAMP algorithm. Again, here we dropped their explicit dependence on the iteration number k. Under the additive white Gaussian noise assumption, we have $p(y_m|z_m, \boldsymbol{\theta}) = \mathcal{N}(y_m|z_m, 1/\gamma)$. Thus $\hat{p}(z_m|\boldsymbol{y}, \hat{p}_m, \tau_m^p, \boldsymbol{\theta})$ also follows a Gaussian distribution with its mean and variance given by

$$\mu_m^z \triangleq \frac{\tau_m^p \gamma y_m + \hat{p}_m}{1 + \gamma \tau_m^p} \tag{21}$$

$$\phi_m^z \triangleq \frac{\tau_m^p}{1 + \gamma \tau_m^p} \tag{22}$$

With the above approximations, we can now define the following two scalar functions: $g_{\text{in}}(\cdot)$ and $g_{\text{out}}(\cdot)$ that are used in the GAMP algorithm. The input scalar function $g_{\text{in}}(\cdot)$ is simply defined as the posterior mean μ_n^x [21], i.e.

$$g_{\text{in}}(\hat{r}_n, \tau_n^r, \boldsymbol{\theta}) = \mu_n^x = \frac{\hat{r}_n}{1 + \eta_n \tau_n^r} \tag{23}$$

The scaled partial derivative of $\tau_n^r g_{\text{in}}(\hat{r}_n, \tau_n^r, \boldsymbol{\theta})$ with respect to $\hat{r}_n$ is the posterior variance ϕ_n^x, i.e.

$$\tau_n^r \frac{\partial}{\partial \hat{r}_n} g_{\text{in}}(\hat{r}_n, \tau_n^r, \boldsymbol{\theta}) = \phi_n^x = \frac{\tau_n^r}{1 + \eta_n \tau_n^r} \tag{24}$$

The output scalar function $g_{\text{out}}(\cdot)$ is related to the posterior mean μ_m^z as follows

$$\begin{aligned} g_{\text{out}}(\hat{p}_m, \tau_m^p, \boldsymbol{\theta}) &= \frac{1}{\tau_m^p}(\mu_m^z - \hat{p}_m) \\ &= \frac{1}{\tau_m^p}\left(\frac{\tau_m^p \gamma y_m + \hat{p}_m}{1 + \gamma \tau_m^p} - \hat{p}_m\right) \end{aligned} \tag{25}$$

The partial derivative of $g_{\text{out}}(\hat{p}_m, \tau_m^p, \boldsymbol{\theta})$ is related to the posterior variance ϕ_m^z in the following way

$$\tau_m^p \frac{\partial}{\partial \hat{p}_m} g_{\text{out}}(\hat{p}_m, \tau_m^p, \boldsymbol{\theta}) = \frac{\phi_m^z - \tau_m^p}{\tau_m^p} = \frac{-\gamma \tau_m^p}{1 + \gamma \tau_m^p} \tag{26}$$

Given the above definitions of $g_{\text{in}}(\cdot)$ and $g_{\text{out}}(\cdot)$, the GAMP algorithm tailored to the considered sparse signal estimation problem can now be summarized as follows (details of the derivation of the GAMP algorithm can be found in [21]), in which a_{mn} denotes the (m, n)th entry of $\boldsymbol{A}$, $\mu_n^x(k)$ and $\phi_n^x(k)$ denote the posterior mean and variance of x_n at iteration k, respectively.

We have now derived an efficient algorithm to obtain approximate posterior distributions for the

Algorithm 1 GAMP Algorithm

Initialization: given $\boldsymbol{\theta}^{(t)}$; set $k=0$, $\hat{s}_m^{(-1)}=0, \forall m \in \{1,\ldots,M\}$; $\{\mu_n^x(k)\}_{n=1}^N$ and $\{\phi_n^x(k)\}_{n=1}^N$ are initialized as the mean and variance of the prior distribution. Repeat the following steps until $\sum_n |\mu_n^x(k+1)-\mu_n^x(k)|^2 \le \epsilon$, where ϵ is a pre-specified error tolerance.

Step 1. $\forall m \in \{1,\ldots,M\}$:

$$\hat{z}_m(k) = \sum_n a_{mn}\mu_n^x(k)$$
$$\tau_m^p(k) = \sum_n a_{mn}^2\phi_n^x(k)$$
$$\hat{p}_m(k) = \hat{z}_m(k) - \tau_m^p(k)\hat{s}_m(k-1)$$

Step 2. $\forall m \in \{1,\ldots,M\}$:

$$\hat{s}_m(k) = g_{\text{out}}(\hat{p}_m(k), \tau_m^p(k), \boldsymbol{\theta}^{(t)})$$
$$\tau_m^s(k) = -\frac{\partial}{\partial \hat{p}_m} g_{\text{out}}(\hat{p}_m(k), \tau_m^p(k), \boldsymbol{\theta}^{(t)})$$

Step 3. $\forall n \in \{1,\ldots,N\}$:

$$\tau_n^r(k) = \left(\sum_m a_{mn}^2 \tau_m^s(k)\right)^{-1}$$
$$\hat{r}_n(k) = \mu_n^x(k) + \tau_n^r(k)\sum_m a_{mn}\hat{s}_m(k)$$

Step 4. $\forall n \in \{1,\ldots,N\}$:

$$\mu_n^x(k+1) = g_{\text{in}}(\hat{r}_n(k), \tau_n^r(k), \boldsymbol{\theta}^{(t)})$$
$$\phi_n^x(k+1) = \tau_n^r(k)\frac{\partial}{\partial \hat{r}_n} g_{\text{in}}(\hat{r}_n(k), \tau_n^r(k), \boldsymbol{\theta}^{(t)})$$

Output: $\{\hat{r}_n(k_0), \tau_n^r(k_0)\}$, $\{\hat{p}_m(k_0), \tau_m^p(k_0)\}$, and $\{\mu_n^x(k_0+1), \phi_n^x(k_0+1)\}$, where k_0 stands for the last iteration.

variables $\boldsymbol{x}$ and $\boldsymbol{z} \triangleq \boldsymbol{A}\boldsymbol{x}$. We see that the GAMP algorithm no longer needs to compute the aforementioned matrix inverse. The dominating operations in each iteration is the simple matrix-vector multiplications, which scale as $\mathcal{O}(MN)$. Thus the computational complexity is significantly reduced. In the following, we discuss how to update the hyperparameters via the EM.

B. Hyperparameter Learning via EM

In the EM framework, the hyperparameters $\{\boldsymbol{\alpha}, \gamma\}$ are estimated by treating $\boldsymbol{x}$ as hidden variables and iteratively maximizing the Q-function, i.e.

$$\begin{aligned}\{\boldsymbol{\alpha}^{(t+1)}, \gamma^{(t+1)}\} &= \arg\max_{\boldsymbol{\alpha},\gamma} Q(\boldsymbol{\alpha}, \gamma\,|\boldsymbol{\alpha}^{(t)}, \gamma^{(t)}) \\ &= \arg\max_{\boldsymbol{\alpha},\gamma} E_{\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\alpha}^{(t)},\gamma^{(t)}}[\log p(\boldsymbol{\alpha}, \gamma\,|\boldsymbol{x}, \boldsymbol{y})]\end{aligned} \tag{27}$$

where the operator $E_{\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\alpha}^{(t)},\gamma^{(t)}}[\cdot]$ denotes the expectation with respect to the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha}^{(t)}, \gamma^{(t)})$. The details of the derivation are included in Appendix. In summary, the hyperparameter α_n can be updated according to the following sub-optimal but effective rule

$$\alpha_n^{(t+1)} = \frac{a-1}{0.5\omega_n + b} \quad \forall n \tag{28}$$

Algorithm 2 PCSBL-GAMP Algorithm

1. Initialization: given $\boldsymbol{\alpha}^{(0)}$ and $\gamma^{(0)}$.
2. For $t \ge 0$: given $\boldsymbol{\alpha}^{(t)}$ and $\gamma^{(t)}$, call the GAMP algorithm. Based on the outputs of the GAMP algorithm, update the hyperparameters $\boldsymbol{\alpha}^{(t+1)}$ and $\gamma^{(t+1)}$ according to (28) and (31).
3. Continue the above iteration until the difference between two consecutive estimates of $\boldsymbol{x}$ is negligible, or a maximum number of iterations is reached.

where

$$\omega_n \triangleq \langle x_n^2\rangle + \beta \sum_{i\in N_{(n)}} \langle x_i^2\rangle \tag{29}$$

Here $\langle x_n^2\rangle$ denotes the expectation with respect to the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha}^{(t)}, \gamma^{(t)})$. Here we use $\hat{p}(x_n|\boldsymbol{y}, \hat{r}_n(k_0), \tau_n^r(k_0), \boldsymbol{\theta}^{(t)})$, i.e. the approximate posterior distribution of x_n obtained from the GAMP algorithm, to replace the true posterior distribution $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha}^{(t)}, \gamma^{(t)})$ in computing the expectation. Since $\hat{p}(x_n|\boldsymbol{y}, \hat{r}_n(k_0), \tau_n^r(k_0), \boldsymbol{\theta}^{(t)})$ follows a Gaussian distribution with its mean and variance given by (18)–(19), we have

$$\langle x_n^2\rangle = \frac{(\hat{r}_n(k_0))^2}{(1+\eta_n^{(t)}\tau_n^r(k_0))^2} + \frac{\tau_n^r(k_0)}{1+\eta_n^{(t)}\tau_n^r(k_0)} \tag{30}$$

The update rule for γ is given by

$$\gamma^{(t+1)} = \frac{M+2c-2}{2d+\sum_m \langle (y_m - z_m)^2\rangle} \tag{31}$$

where $\langle\cdot\rangle$ denotes the expectation with respect to $p(z_m|\boldsymbol{y}, \hat{p}_m(k_0), \tau_m^p(k_0), \boldsymbol{\theta}^{(t)})$, i.e. the approximate posterior distribution of z_m. Recalling that the approximate posterior of z_m follows a Gaussian distribution with its mean and variance given by (21)–(22), we have

$$\langle (y_m - z_m)^2\rangle = (y_m - \mu_m^z)^2 + \phi_m^z \tag{32}$$

where μ_m^z and ϕ_m^z are given by (21)–(22), with $\{\hat{p}_m, \tau_m^p\}$ replaced by $\{\hat{p}_m(k_0), \tau_m^p(k_0)\}$, and γ replaced by $\gamma^{(t)}$.

So far we have completed the development of our GAMP-based pattern-coupled sparse Bayesian learning algorithm. For clarify, we now summarize our proposed PCSBL-GAMP algorithm in Algorithm 2.

C. Discussions

Notice that the update rule (28) resembles that of the conventional sparse Bayesian learning work [4] except that ω_n is equal to $\langle x_n^2\rangle$ for the conventional sparse Bayesian learning method, while for our case, ω_n is a weighted summation of $\langle x_n^2\rangle$ and $\langle x_i^2\rangle$ for $i \in N_{(n)}$. Numerical results show that this update rule, although sub-optimal, guarantees an exact recovery in the noiseless case and provides superior recovery performance. The reason can be explained as follows. If we examine the update rule (28) more closely, we can see that the Bayesian Occam's razor which contributes to the success of the conventional sparse Bayesian learning

method also works for our proposed algorithm. Specifically, note that in the E-step, when computing the posterior mean and covariance matrix, a large hyperparameter α_n not only suppresses the posterior mean and variance of its associated coefficient x_n, but also the posterior means and variances associated with its neighboring coefficients $\{x_i | i \in N_n\}$ (c.f. (15)-(16)). As a result, the value of ω_n becomes smaller, which in turn leads to a larger hyperparameter α_n according to the update formula (28). This negative feedback mechanism keeps decreasing most of the entries of $\hat{\boldsymbol{x}}$ until they reach machine precision and become negligible (as small as 10^{-8}), while leaving only a few prominent nonzero entries survived to fit the data. The process eventually leads to an exact block-sparse solution, and thus no pruning operation is needed.

IV. Simulation Results

We now carry out experiments to illustrate the performance of our proposed algorithm, also referred to as PCSBL-GAMP algorithm, and its comparison with other existing methods. The performance of the proposed algorithm[3] will be examined using both synthetic and real data. The parameters a, b, c, d for our proposed algorithm are set equal to $a = 1.5$, $b = 10^{-6}$, $c = 1$, and $d = 10^{-6}$ throughout our experiments. The relevance parameter β is set equal to $\beta = 1$ in our experiments. In fact, our empirical results suggest that its choice is not very critical to the recovery performance as long as β is set into the region $\beta \in [0.1, 1]$.

A. Synthetic Data

We first evaluate the recovery performance of the PCSBL-GAMP method using the synthetic data. In our simulations, we generate a one-dimensional block-sparse signal in a way similar to [20]. Suppose the N-dimensional sparse signal contains K nonzero coefficients (K is also denoted as the sparsity level) which are partitioned into T blocks with random sizes and random locations. The nonzero coefficients of the sparse signal $\boldsymbol{x}$ and the measurement matrix $\boldsymbol{A} \in \mathbb{R}^{M \times N}$ are randomly generated with each entry independently drawn from a normal distribution, and then the sparse signal $\boldsymbol{x}$ and columns of $\boldsymbol{A}$ are normalized to unit norm. Fig. 1 depicts the success rate of the proposed PCSBL-GAMP method vs. the ratio M/N, where we set $N = 200$, $K = 40$ and $T = 6$. The success rate is computed as the ratio of the number of successful trials to the total number of independent runs. A trial is considered successful if the normalized squared recovery error $\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2 / \|\boldsymbol{x}\|^2$ is no greater than 10^{-6}, where $\hat{\boldsymbol{x}}$ denotes the estimate of the sparse signal. The success rates of the EM-based PC-SBL method [20] (referred to as the PCSBL-EM), the conventional SBL [4], and the basis pursuit (BP) method [2], [23] are also included for comparison. From Fig. 1, we see that the PCSBL-GAMP method achieves almost the same performance as that of the PCSBL-EM method, and presents a significant performance advantage over the SBL and BP methods due to exploiting the underlying

[3]Matlab codes for our algorithm are available at http://www.junfang-uestc.net/codes/PCSBL-GAMP.rar.

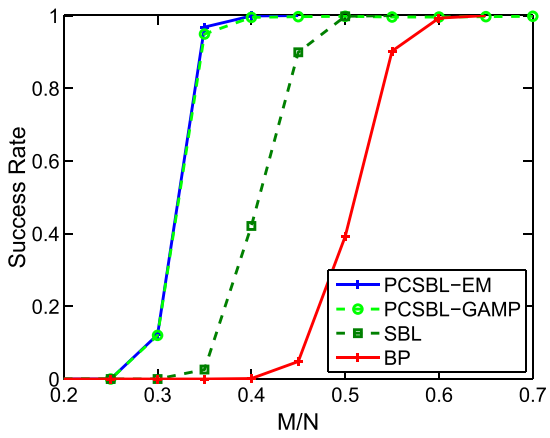

Fig. 1. Success rates vs. the ratio M/N.

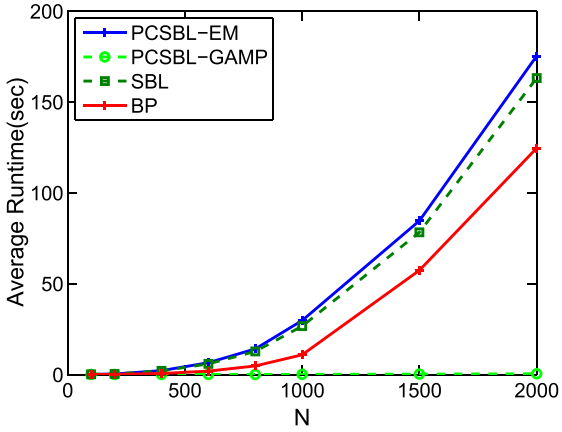

Fig. 2. Average run times vs. N.

block-sparse structures. The average run times of respective algorithms as a function of the signal dimension N is plotted in Fig. 2, where we set $M = 0.4N$. Results are averaged over 100 independent runs. We see that the PCSBL-GAMP requires much less run time than the PCSBL-EM, particularly when the signal dimension N is large. Also, it can be observed that the average run time of the PCSBL-EM grows rapidly with an increasing N, whereas the average run time of the PCSBL-GAMP increases very slowly. This observation coincides with our computational complexity analysis.

We also carry out experiments using patches of letters "C" and "S" (16×16 pixels) with black background, where most of the pixels on the patches are zeros and the nonzero coefficients exhibit irregular block patterns. We compare our method with other block-sparse signal recovery algorithms, namely, the cluster-structured MCMC algorithm (Cluss-MCMC) [16], the Boltzman machine-based greedy pursuit algorithm (BM-MAP-OMP) [17], and the block sparse Bayesian learning method (BSBL) method [14], [19]. Note that although the BM-MAP-OMP and the BSBL algorithms are developed for one-dimensional sparse signal recovery, we extend their methods to the two-dimensional scenario. In our simulations, model parameters used by the competing algorithms are adjusted to achieve the best performance.

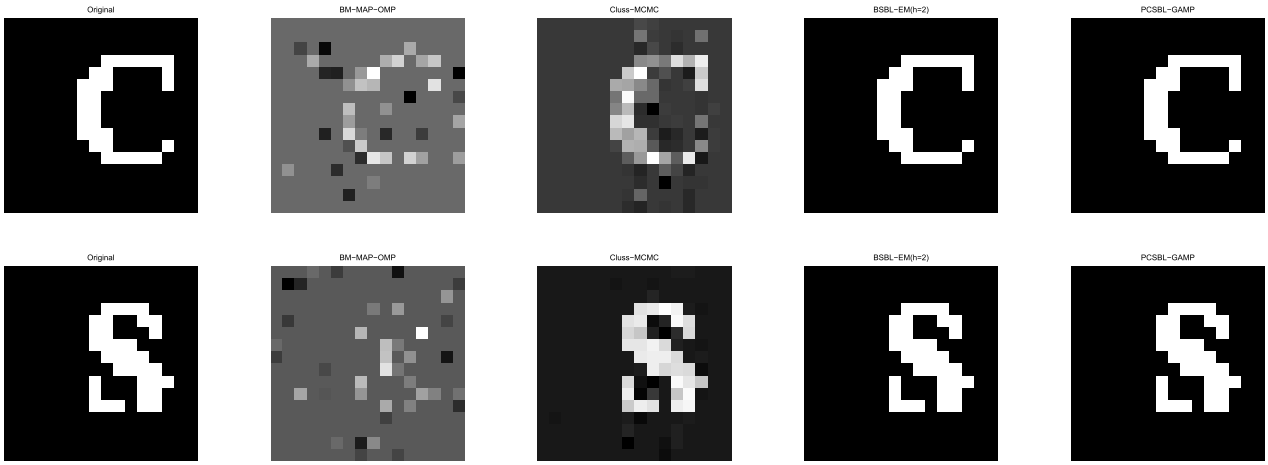

Fig. 3. Original patches of letters and patches reconstructed by respective algorithms for a noiseless case.

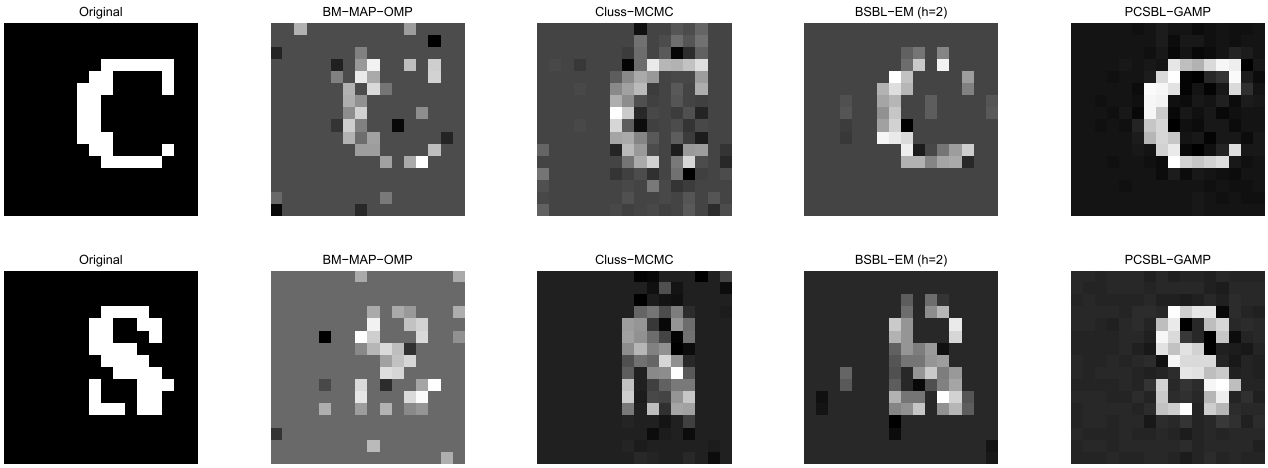

Fig. 4. Original patches of letters and patches reconstructed by respective algorithms, SNR=20dB.

For the BSBL algorithms, the block size parameter h is set equal to 2. Fig. 3 and 4 depict the original patches and the reconstructed patches under a noiseless case and a noisy case respectively, where we set $M = 80$. For the noisy case, white Gaussian noise is added to the patches. The signal-to-noise ratio (SNR) is set to 20dB, with SNR $\triangleq 10\log(\|\boldsymbol{Ax}\|_2^2/M\sigma^2)$. When there is no noise, we see that the proposed method is able to achieve an exact reconstruction of the original patches with a moderate number of measurements. For the noisy case, it can be observed that our proposed PCSBL-GAMP method provides the best visual quality with recognizable letters, whereas the letters reconstructed by other algorithms have considerably lower quality, particularly for the Cluss-MCMC and the BM-MAP-OMP methods. This result also implies that our proposed method is flexible to accommodate any irregular cluster patterns.

B. Satellite Image Recovery

In this subsection, we carry out experiments on a non-negative 256×256 satellite image.[4] The image is sparse in the spatial domain, with only 6678 (approximately 10.2% of total pixels) nonzero pixels. In our experiments, compressive measurements are corrupted by additive i.i.d. Gaussian noise, i.e. $\boldsymbol{y} = \boldsymbol{Ax} + \boldsymbol{w}$, where the image is represented as a one-dimensional vector $\boldsymbol{x}$. The sensing matrix $\boldsymbol{A}$ is chosen to be the same as that used in [24], i.e. $\boldsymbol{A} = \boldsymbol{\Phi\Psi S}$, where $\boldsymbol{\Phi} \in \{0,1\}^{M\times N}$ and its rows are randomly selected from the $N \times N$ identity matrix, $\boldsymbol{\Psi} \in \{-1,1\}^{N\times N}$ is a Hadamard transform matrix, $\boldsymbol{S} \in \mathbb{R}^{N\times N}$ is a diagonal matrix with its entries randomly chosen from $\{-1,1\}$. Sensing using such a measurement matrix can be executed using a fast binary algorithm, which makes the hardware implementation simple. Note that the BM-MAP-OMP, the BSBL and the Cluss-MCMC methods were not included in this experiment due to their prohibitive computational complexity when the signal dimension is large. Instead, we compare our method with some other computationally efficient GAMP-based methods, namely, the EM-NNGM-GAMP method [24] and the EM-GM-AMP method [25]. These two methods have demonstrated state-of-the-art recovery performance in a series of experiments. The spectral projected gradient (SPG) method (referred to as SPGL1) which was developed in [26] to

[4]Image data is available at http://sourceforge.net/projects/gampmatlab.

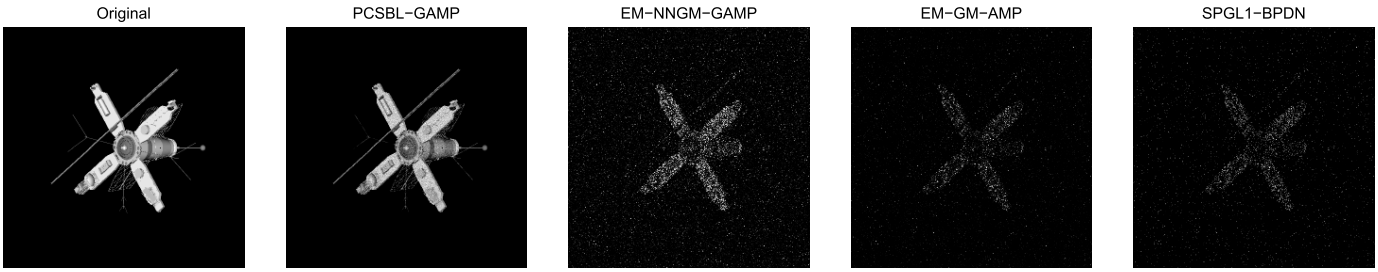

Fig. 5. Original satellite image and images reconstructed by respective algorithms.

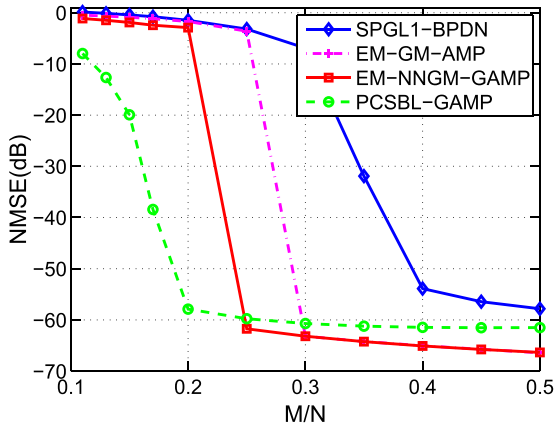

Fig. 6. Satellite image: NMSEs vs. the ratio M/N.

efficiently solve the basis pursuit or basis pursuit denoising optimizations is also included for comparison. Among these methods, only the EM-NNGM-GAMP algorithm exploits the non-negativity of the satellite image. Therefore the signals recovered by these algorithms, except the EM-NNGM-GAMP, may contain negative coefficients. These negative coefficients are manually set to zero in our simulations. Fig. 5 shows the original satellite image and the images reconstructed by respective algorithms, where we set $M = 0.15N$ and SNR = 60dB. It can be seen that the PCSBL-GAMP offers a significantly better image quality as compared with other methods. Fig. 6 plots the normalized mean square errors (NMSEs) of respective algorithms vs. the ratio M/N. Results are averaged over 100 independent trails, with the sensing matrix $\boldsymbol{A}$ randomly generated for each trial. The recovered negative coefficients are kept unaltered in calculating the NMSEs. We see that our proposed PCSBL-GAMP method outperforms other algorithms by a big margin for a small ratio M/N (e.g. $M/N \leq 0.25$), where data acquisition is practically appealing due to high compression rates.

C. 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 innovations are sparse in the spatial image domain. By exploiting this sparsity, the sparse foreground innovations within a scene can be reconstructed using compressed measurements, which relieves the communications burden placed on data transmission [7]. Specifically, the idea is to reconstruct the foreground image from the difference between the compressed measurements of the background image and the compressed measurements of the test image [7]

$$\boldsymbol{y}_f = \boldsymbol{y}_t - \boldsymbol{y}_b = \boldsymbol{A}(\boldsymbol{x}_t - \boldsymbol{x}_b) = \boldsymbol{A}\boldsymbol{x}_f$$

where $\boldsymbol{x}_t$ and $\boldsymbol{x}_b$ represent the test and the background images, respectively; $\boldsymbol{y}_t$ and $\boldsymbol{y}_b$ denote the compressed measurements of the test and background images, respectively; and $\boldsymbol{x}_f$ is the foreground image to be recovered. In our experiments, we use the Convoy2 data set that was used in [27]. The Convoy2 data set was collected on the Spesutie island, consisting of a video sequence with 260 frames and one background frame recorded by a single static camera. The video sequence has a dynamic sparse foreground as vehicles enter and exit the filed of view over time. We first choose the 40th frame of the Convoy2 data set as a test image, which is shown in Fig. 7. The background image and the foreground image are also included in Fig. 7. The foreground image is regarded as the groundtruth image. This foreground image, however, does not have a pure background since $\boldsymbol{x}_f = \boldsymbol{x}_t - \boldsymbol{x}_b$ is not an exactly sparse signal and contains many small nonzero components. In our experiments, the original images of 480×381 pixels are resized to 120×96 pixels. For the resized foreground image, we have a total number of 923 coefficients whose magnitudes are greater than 10^{-2}, thus the percentage of nonzero coefficients is $923/(120 \times 96) = 8.01\%$. Again, the BM-MAP-OMP, BSBL, and the Cluss-MCMC methods are not included due to their prohibitive computational complexity. Here we compare our method with the EM-GM-AMP method [25] and the EM-BG-AMP method [28]. Fig. 8 depicts images reconstructed by respective algorithms, where we use only $M = 0.1N$ measurements. The measurement matrix $\boldsymbol{A}$ is randomly generated with each entry independently drawn from a normal distribution. We see that our proposed PCSBL-GAMP method provides the finest image quality with a clear appearance of the vehicle, whereas the object silhouettes recovered by other methods are hardly recognizable. In our next experiments, frames from the 10th to 60th are used as test images. For each test image, we use $M = 0.1N$ measurements to recover the difference (foreground) image. Fig. 9 shows the NMSEs of respective algorithms vs. the frame number, where the NMSEs are obtained by averaging over 100 independent runs, with the measurement matrix $\boldsymbol{A}$ randomly generated for each run. From Fig. 9, we observe

Fig. 7. From left to right: the test image $\boldsymbol{x}_t$ (40th frame of the Convoy2 data set), the background image $\boldsymbol{x}_b$, the foreground image $\boldsymbol{x}_f$.

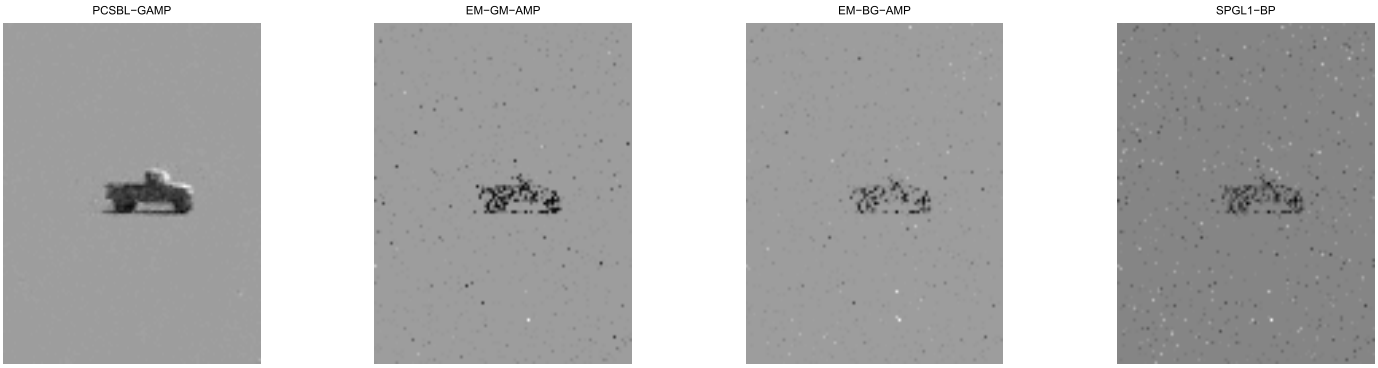

Fig. 8. Foreground images reconstructed by respective algorithms.

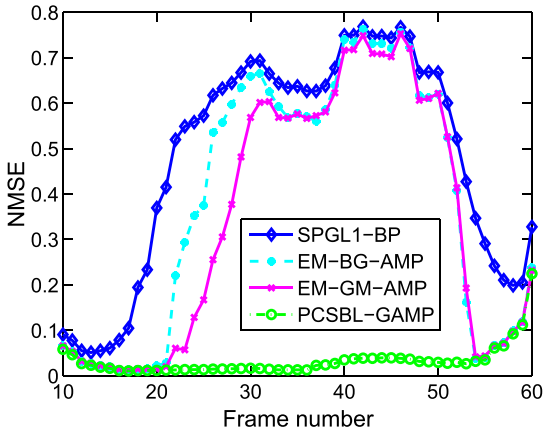

Fig. 9. NMSEs vs. the frame number.

that our proposed method presents a significant performance advantage over other methods.

V. Conclusions

We developed a computationally efficient pattern-coupled sparse Bayesian learning method for recovery of two-dimensional block-sparse signals whose cluster patterns are unknown *a priori*. A two-dimensional pattern-coupled hierarchical Gaussian prior model is proposed to characterize and exploit the pattern dependencies among neighboring coefficients. The proposed pattern-coupled hierarchical model is effective and flexible to capture any underlying block-sparse structures, without requiring the prior knowledge of the block partition. An efficient Bayesian inference method was developed by integrating the generalized approximate message passing (GAMP) technique into the proposed algorithm. Specifically, the algorithm was developed within an expectation-maximization (EM) framework, where the GAMP is employed to efficiently compute an approximation of the marginal posterior distribution of hidden variables. Simulation results show that our proposed algorithm presents a substantial performance advantage over other existing state-of-the-art methods in image recovery, meanwhile achieving a significant reduction in computational complexity.

Appendix
Hyperparameter Learning via EM

It can be readily verified that the Q-function can be decomposed into a summation of two terms

$$\begin{aligned} Q(\boldsymbol{\alpha}, \gamma\,|\boldsymbol{\alpha}^{(t)}, \gamma^{(t)}) &= E_{\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\alpha}^{(t)},\gamma^{(t)}}[\log p(\boldsymbol{\alpha})p(\boldsymbol{x}|\boldsymbol{\alpha})] \\ &\quad + E_{\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\alpha}^{(t)},\gamma^{(t)}}[\log p(\gamma)p(\boldsymbol{y}|\boldsymbol{x},\gamma)] \\ &\triangleq Q(\boldsymbol{\alpha}|\boldsymbol{\alpha}^{(t)}, \gamma^{(t)}) + Q(\gamma\,|\boldsymbol{\alpha}^{(t)}, \gamma^{(t)}) \end{aligned} \quad (33)$$

Hence $\boldsymbol{\alpha}$ and γ can be estimated independently.

We first examine the update of $\boldsymbol{\alpha}$, i.e.

$$\boldsymbol{\alpha}^{(t+1)} = \arg\max_{\boldsymbol{\alpha}} Q(\boldsymbol{\alpha}|\boldsymbol{\alpha}^{(t)}, \gamma^{(t)}) \quad (34)$$

Recalling (11)–(13), $Q(\boldsymbol{\alpha}|\boldsymbol{\alpha}^{(t)}, \gamma^{(t)})$ can be expressed as

$$Q(\boldsymbol{\alpha}|\boldsymbol{\alpha}^{(t)}, \gamma^{(t)}) \propto \sum_{n=1}^{N}\left((a-1)\log\alpha_n - b\alpha_n + \frac{1}{2}\log\eta_n - \frac{1}{2}\eta_n\langle x_n^2\rangle\right) \quad (35)$$

where $\langle x_n^2\rangle$ denotes the expectation with respect to the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha}^{(t)}, \gamma^{(t)})$. We see that in the Q-function (35), hyperparameters are entangled with each other due to the logarithm term $\log\eta_n$ (note that η_n, defined in (12), is a function of $\boldsymbol{\alpha}$). In this case, an analytical solution to the optimization (34) is difficult to obtain. Gradient descent methods can certainly be used to search for the optimal solution. Nevertheless, for gradient descent methods, there is no explicit formula for the hyperparameter update. Also, gradient-based methods involve higher computational complexity as compared with an analytical update rule. Here we consider an alternative strategy which aims at finding a simple, analytical sub-optimal solution of (34). Such an analytical sub-optimal solution can be obtained by examining the optimality condition of (34). Suppose $\boldsymbol{\alpha}^* = \{\alpha_n^*\}$ is the optimal solution of (34). Then the first derivative of the Q-function with respect to $\boldsymbol{\alpha}$ equals to zero at the optimal point, i.e.

$$\frac{\partial Q(\boldsymbol{\alpha}|\boldsymbol{\theta}^{(t)})}{\partial\alpha_n}|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}^*} = \frac{a-1}{\alpha_n^*} - b + \frac{1}{2}\nu_n^* - \frac{1}{2}\omega_n = 0 \quad (36)$$

where

$$\nu_n^* \triangleq \frac{1}{\eta_n^*} + \beta\sum_{i\in N_{(n)}}\frac{1}{\eta_i^*} \quad (37)$$

$$\omega_n \triangleq \langle x_n^2\rangle + \beta\sum_{i\in N_{(n)}}\langle x_i^2\rangle \quad (38)$$

Since all hyperparameters $\{\alpha_n\}$ and β are non-negative, it can be easily verified $(1/\alpha_n^*) > (1/\eta_n^*) > 0$, and $(1/\beta\alpha_n^*) > (1/\eta_i^*) > 0$ for $i \in N_{(n)}$. Therefore ν_n^* is bounded by

$$\frac{5}{\alpha_n^*} > \nu_n^* > 0 \quad (39)$$

Consequently we have

$$\frac{a+1.5}{\alpha_n^*} > \frac{a-1}{\alpha_n^*} + \frac{1}{2}\nu_n^* > \frac{a-1}{\alpha_n^*} \quad (40)$$

Combining (36) and (40), we reach that α_n^* is within the range

$$\alpha_n^* \in \left[\frac{a-1}{0.5\omega_n + b}, \frac{a+1.5}{0.5\omega_n + b}\right] \quad \forall n \quad (41)$$

A sub-optimal solution to (34) can therefore simply be chosen as

$$\alpha_n^{(t+1)} = \frac{a-1}{0.5\omega_n + b} \quad \forall n \quad (42)$$

We now discuss the update of the hyperparameter γ, the inverse of the noise variance. Since the GAMP algorithm also provides an approximate posterior distribution for the noiseless output $\boldsymbol{z} = \boldsymbol{A}\boldsymbol{x}$, we can simply treat $\boldsymbol{z}$ as hidden variables when learning the noise variance, i.e.

$$\gamma^{(t+1)} = \arg\max_{\gamma} E_{\boldsymbol{z}|\boldsymbol{y},\boldsymbol{\alpha}^{(t)},\gamma^{(t)}}[\log p(\gamma)p(\boldsymbol{y}|\boldsymbol{z},\gamma)] \triangleq \arg\max_{\gamma} Q(\gamma|\boldsymbol{\alpha}^{(t)}, \gamma^{(t)}) \quad (43)$$

Taking the partial derivative of the Q-function with respect to γ gives

$$\frac{\partial Q(\gamma|\boldsymbol{\alpha}^{(t)}, \gamma^{(t)})}{\partial\gamma} = \frac{c-1}{\gamma} - d + \frac{M}{2\gamma} - \frac{1}{2}\sum_{m=1}^{M}\langle(y_m - z_m)^2\rangle \quad (44)$$

where $\langle\cdot\rangle$ denotes the expectation with respect to $p(z_m|\boldsymbol{y}, \hat{p}_m(k_0), \tau_m^p(k_0), \boldsymbol{\theta}^{(t)})$, i.e. the approximate posterior distribution of z_m. Setting the derivative equal to zero, we obtain the update rule for γ as

$$\gamma^{(t+1)} = \frac{M + 2c - 2}{2d + \sum_m\langle(y_m - z_m)^2\rangle}. \quad (45)$$

REFERENCES

[1] 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.

[2] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1999.

[3] D. Wipf and S. Nagarajan, "Iterative reweighted ℓ_1 and ℓ_2 methods for finding sparse solutions," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 317–329, Apr. 2010.

[4] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, Sep. 2001.

[5] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2346–2356, Jun. 2008.

[6] L. Wang, L. Zhao, G. Bi, C. Wan, and L. Yang, "Enhanced ISAR imaging by exploiting the continuity of the target scene," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 9, pp. 5736–5750, Sep. 2014.

[7] V. Cevher, A. Sankaranarayanan, M. F. Duarte, D. Reddy, R. G. Baraniuk, and R. Chellappa, "Compressive sensing for background subtraction," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Marseille, France, Oct. 2008, pp. 155–168.

[8] N. Vaswani and W. Lu, "Modified-CS: Modifying compressive sensing for problems with partially known support," *IEEE Trans. Signal Process.*, vol. 58, no. 9, pp. 4595–4607, Sep. 2010.

[9] 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.

[10] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1982–2001, Apr. 2010.

[11] Y. C. Eldar, P. Kuppinger, and H. Bölcskei, "Block-sparse signals: Uncertainty relations and efficient recovery," *IEEE Trans. Inf. Theory*, vol. 58, no. 6, pp. 3042–3054, Jun. 2010.

[12] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. Roy. Statist. Soc. B (Statist. Methodol.)*, vol. 68, no. 1, pp. 49–67, 2006.

[13] D. P. Wipf and B. D. Rao, "An empirical Bayesian strategy for solving the simultaneous sparse approximation problem," *IEEE Trans. Signal Process.*, vol. 55, no. 7, pp. 3704–3716, Jul. 2007.

[14] 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, Sep. 2011.

[15] L. He and L. Carin, "Exploiting structure in wavelet-based Bayesian compressive sensing," *IEEE Trans. Signal Process.*, vol. 57, no. 9, pp. 3488–3497, Sep. 2009.

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

[17] 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.

[18] A. Drémeau, C. Herzet, and L. Daudet, "Boltzmann machine and mean-field approximation for structured sparse decompositions," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3425–3438, Jul. 2012.

[19] 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.

[20] 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.
[21] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Saint Petersburg, Russia, Jul./Aug. 2011, pp. 2168–2172.
[22] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing: I. Motivation and construction," in *Proc. IEEE Inf. Theory Workshop*, Cairo, Egypt, Jan. 2010, pp. 1–5.
[23] E. Candés and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.
[24] J. P. Vila and P. Schniter, "An empirical-Bayes approach to recovering linearly constrained non-negative sparse signals," *IEEE Trans. Signal Process.*, vol. 62, no. 18, pp. 4689–4703, Sep. 2014.
[25] 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.
[26] E. van den Berg and M. P. Friedlander, "Probing the Pareto frontier for basis pursuit solutions," *SIAM J. Sci. Comput.*, vol. 31, no. 2, pp. 890–912, 2008.
[27] G. Warnell, D. Reddy, and R. Chellappa, "Adaptive rate compressive sensing for background subtraction," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Kyoto, Japan, Mar. 2012, pp. 1477–1480.
[28] J. P. Vila and P. Schniter, "Expectation-maximization Bernoulli–Gaussian approximate message passing," in *Proc. 45th Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, USA, Nov. 2011, pp. 799–803.

Jun Fang (M'08) received the B.S. and M.S. degrees from Xidian University, Xi'an, China, in 1998 and 2001, respectively, and the Ph.D. degree from the National University of Singapore, Singapore, in 2006, all in electrical engineering.

In 2006, he was a Post-Doctoral Research Associate with the Department of Electrical and Computer Engineering, Duke University. From 2007 to 2010, he was a Research Associate with the Department of Electrical and Computer Engineering, Stevens Institute of Technology. Since 2011, he has been with the University of Electronic of Science and Technology of China. His current research interests include sparse theory and compressed sensing, and Bayesian inference for data analysis.

Dr. Fang received the IEEE Jack Neubauer Memorial Award in 2013 for the best systems paper published in the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He is an Associate Technical Editor of the *IEEE Communications Magazine*, and an Associate Editor of the IEEE SIGNAL PROCESSING LETTERS.

Lizao Zhang received the B.S. degree from the University of Electronic Science and Technology of China, Chengdu, in 2013. He is currently pursuing the master's degree with the University of Electronic of Science and Technology of China. In 2013, he joined the National Key Laboratory on Communications, University of Electronic of Science and Technology of China. His current research interests include compressed sensing, sparse theory, and statistical learning.

Hongbin Li (M'99–SM'08) received the B.S. and M.S. degrees from the University of Electronic Science and Technology of China, in 1991 and 1994, respectively, and the Ph.D. degree from the University of Florida, Gainesville, FL, in 1999, all in electrical engineering.

From 1996 to 1999, he was a Research Assistant with the Department of Electrical and Computer Engineering, University of Florida. Since 1999, he has been with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, where he is a Professor. He was a Summer Visiting Faculty Member with the Air Force Research Laboratory in the summers of 2003, 2004, and 2009. His general research interests include statistical signal processing, wireless communications, and radars.

Dr. Li is a member of Tau Beta Pi and Phi Kappa Phi. He received the IEEE Jack Neubauer Memorial Award in 2013 for the best systems paper published in the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the Outstanding Paper Award from the IEEE AFICON Conference in 2011, the Harvey N. Davis Teaching Award in 2003 and the Jess H. Davis Memorial Award for excellence in research in 2001 from the Stevens Institute of Technology, and the Sigma Xi Graduate Research Award from the University of Florida in 1999. He has been a member of the IEEE SPS Signal Processing Theory and Methods (2011 to now) Technical Committee (TC) and the IEEE SPS Sensor Array and Multichannel TC (2006 to 2012). He has been an Associate Editor of *Signal Processing* (Elsevier) (since 2013), the IEEE TRANSACTIONS ON SIGNAL PROCESSING (2006 to 2009 and since 2014), the IEEE SIGNAL PROCESSING LETTERS (2005 to 2006), and the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS (2003 to 2006), and a Guest Editor of the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING and the *EURASIP Journal on Applied Signal Processing*. He has been involved in various conference organization activities, including serving as a General Co-Chair for the Seventh IEEE Sensor Array and Multichannel Signal Processing Workshop, Hoboken, NJ, in 2012.