Pattern-Coupled Sparse Bayesian Learning for Recovery of Block-Sparse Signals

Jun Fang, *Member, IEEE*, Yanning Shen, Hongbin Li, *Senior Member, IEEE*, and Pu Wang, *Member, IEEE*

***Abstract*—We consider the problem of recovering block-sparse signals whose cluster patterns are unknown *a priori*. Block-sparse signals with nonzero coefficients occurring in clusters arise naturally in many practical scenarios. However, the knowledge of the block partition is usually unavailable in practice. In this paper, we develop a new sparse Bayesian learning method for recovery of block-sparse signals with unknown cluster patterns. A pattern-coupled hierarchical Gaussian prior is introduced to characterize the pattern dependencies among neighboring coefficients, where a set of hyperparameters are employed to control the sparsity of signal coefficients. The proposed hierarchical model is similar to that for the conventional sparse Bayesian learning. However, unlike the conventional sparse Bayesian learning framework in which each individual hyperparameter is associated independently with each coefficient, in this paper, the prior for each coefficient not only involves its own hyperparameter, but also its immediate neighbor hyperparameters. In doing this way, the sparsity patterns of neighboring coefficients are related to each other and the hierarchical model has the potential to encourage structured-sparse solutions. The hyperparameters are learned by maximizing their posterior probability. We exploit an expectation-maximization (EM) formulation to develop an iterative algorithm that treats the signal as hidden variables and iteratively maximizes a lower bound on the posterior probability. In the M-step, a simple suboptimal solution is employed to replace a gradient-based search to maximize the lower bound. Numerical results are provided to illustrate the effectiveness of the proposed algorithm.**

***Index Terms*—Block-sparse signal recovery, pattern-coupled hierarchical model, sparse Bayesian learning.**

Manuscript received November 04, 2013; revised April 08, 2014 and August 26, 2014; accepted November 19, 2014. Date of publication November 26, 2014; date of current version December 10, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. John McAllister. This work was supported in part by the National Science Foundation of China under Grants 61172114, 61201274, 61428103, the Fundamental Research Funds for the Central Universities (China) under Grant ZYGX2012J025, and the National Science Foundation under Grant ECCS-1408182.

J. Fang is 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).

Y. Shen is with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: shenx513@umn.edu).

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).

P. Wang is with the Schlumberger-Doll Research Center, Cambridge, MA 02139 USA (e-mail: pwang@ieee.org).

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/TSP.2014.2375133

I. Introduction

COMPRESSIVE sensing is a recently emerged technique of signal sampling and reconstruction, the main purpose of which is to recover sparse signals from much fewer linear measurements [1]–[3]

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

where $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is the sampling matrix with $m \ll n$, and $\boldsymbol{x}$ denotes the n-dimensional sparse signal with only K nonzero coefficients. Such a problem has been extensively studied and a variety of algorithms that provide consistent recovery performance guarantee were proposed, e.g., [1]–[6]. In practice, sparse signals usually have block-sparse structures that can be exploited to enhance the recovery performance. For example, the atomic decomposition of multi-band signals [7] or audio signals [8] usually results in a block-sparse structure in which the nonzero coefficients occur in clusters. This clustered sparse pattern is also exploited in applications such as gene expression analysis [9], and inverse synthetic aperture radar imaging [10]. Besides these, block-sparsity naturally arises in the setup of the multiple measurement vector problem.

A number of algorithms, e.g., block-OMP [11], mixed ℓ_2/ℓ_1 norm-minimization [12], group LASSO [13], StructOMP [14], model-based CoSaMP [15], and block-sparse Bayesian learning [16], [17][1] were proposed for recovery of block-sparse signals, and their recovery behaviors were analyzed in terms of the model-based restricted isometry property (RIP) [12], [15] and the mutual coherence [11]. Analyses suggested that exploiting the inherent structure of sparse signals helps improve the recovery performance considerably. These algorithms, albeit effective, require the knowledge of the cluster pattern (block partition) *a priori*. In practice, however, the prior information about the block partition of sparse signals is often unavailable. For example, we know that audio signals have structured sparse representations but the exact block pattern is unknown to us. To address this difficulty, many sophisticated Bayesian methods which do not need the knowledge of the block partition were developed. In [18], [19], a hierarchical Bayesian "spike-and-slab" prior model was introduced to encourage the sparseness and promote the cluster patterns simultaneously. Nevertheless, for both works [18], [19], the posterior distribution cannot be derived analytically, and a Markov chain Monte Carlo (MCMC)

[1]Although the algorithms were developed under the multiple measurement vector framework, they can be readily adapted to recovery of block-sparse signals.

sampling method has to be employed for Bayesian inference. In [20], [21], a graphical prior, also referred to as the "Boltzmann machine", was used to model the statistical dependencies between atoms. The Boltzmann machine is employed as a prior on the support of a sparse representation. However, the maximum a posterior (MAP) estimator with such a prior involves an exhaustive search over all possible sparsity patterns. To overcome the intractability of the combinatorial search, a greedy method [20] and a variational mean-field approximation method [21] were proposed to approximate the MAP. In [22], the support correlation across coefficients was modeled by a Markov chain defined by two transition probabilities, based on which an approximate message passing algorithm was developed for recovery of time-varying signals. Recently, Zhang and Rao generalized the block sparse Bayesian learning method to address recovery of block-sparse signals with unknown cluster patterns [23]. In their work, the components of the signal are partitioned into a number of overlapping blocks. The original data model is converted into an expanded model by adding redundant columns to the original measurement matrix and stacking all blocks to form an augmented vector. Conventional block sparse Bayesian learning algorithms such as [17] are then applied to the expended model. This overlapping structure provides flexibility in defining a block-sparse pattern. Besides the above Bayesian techniques, we note that a total variation approach, also referred to as the fused LASSO, was proposed in [9] to exploit the spatial amplitude smoothness, where the ℓ_1-norm of the successive differences of neighboring coefficients is used as a penalty term to encourage piecewise constant solutions.

In this paper, we develop a new sparse Bayesian learning method for recovery of block-sparse signals with unknown block partitioning structure. Similar to the conventional sparse Bayesian learning approach [24], [25], a Bayesian hierarchical Gaussian framework is employed to model the sparse prior, in which a set of hyperparameters are introduced to characterize the Gaussian prior and control the sparsity of the signal components. Conventional sparse learning approaches, however, assume independence between the elements of the sparse signal. Specifically, each individual hyperparameter is associated independently with each coefficient of the sparse signal. To encourage block-sparse patterns, in this paper, we propose a pattern coupled hierarchical Gaussian framework in which the sparsity of each coefficient is controlled not only by its own hyperparameter, but also by its neighbor hyperparameters. Such a prior has the potential to encourage clustered patterns and suppress "isolated coefficients" whose pattern is different from that of its neighboring coefficients. An iterative algorithm which exploits the expectation-maximization (EM) formulation is developed to learn the hyperparameters and to estimate the block-sparse signal. Our proposed algorithm not only admits a simple iterative procedure for Bayesian inference. It also demonstrates superiority over other existing methods for block-sparse signal recovery.

The rest of the paper is organized as follows. In Section II, we introduce a new pattern coupled hierarchical Gaussian framework to model the sparse prior and the pattern dependencies among the signal components. An iterative algorithm which resorts to the expectation-maximization (EM) formulation is developed in Section III to learn the hyperparameters and to estimate the block-sparse signal. Section IV extends the proposed Bayesian inference method to the scenario where the observation noise variance is unknown. Inspired by the new Bayesian method, an iterative reweighted ℓ_1 algorithm is also proposed in Section V for the recovery of block-sparse signals. Simulation results are provided in Section VI, followed by concluding remarks in Section VII.

II. Hierarchical Prior Model

We consider the problem of recovering a block-sparse signal $\boldsymbol{x} \in \mathbb{R}^n$ from noise-corrupted measurements

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w} \tag{2}$$

where $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ $(m < n)$ is the measurement matrix, and $\boldsymbol{w}$ is the additive multivariate Gaussian noise with zero mean and covariance matrix $\sigma^2\boldsymbol{I}$. The signal $\boldsymbol{x}$ has a block-sparse structure but the knowledge of the block partition is unavailable to us. This is usually the situation for many practical applications. In this case, the only information we can utilize is that the sparsity patterns of neighboring coefficients are statistically dependent. We, in the following, will propose a new hierarchical sparse Bayesian learning (SBL) model to capture the pattern dependencies among neighboring coefficients. Before we proceed, we first provide a brief review of the hierarchical models for the conventional SBL method [24] and the block-SBL methods [16], [17].

A. Review of Hierarchical Models for SBL and B-SBL

Sparse Bayesian learning was firstly introduced by Tipping in his pioneering work [24], where the regression and classification problem was considered. Later on in [25], [26], sparse Bayesian learning was introduced to solve the sparse recovery problem, and demonstrated superior performance for sparse signal recovery in a series of experiments [25], [26]. The success of sparse Bayesian learning has inspired a growing interest in Bayesian techniques for sparse signal recovery. In some recent works, e.g., [27], [28], Bayesian techniques using the approximate message passing (AMP) were proposed for compressed sensing, which achieves even improved performance over the sparse Bayesian learning method for some sparse signal recovery problems. In the conventional sparse Bayesian learning framework, $\boldsymbol{x}$ is assigned a Gaussian prior distribution

$$p(\boldsymbol{x}|\boldsymbol{\alpha}) = \prod_{i=1}^{n} p(x_i|\alpha_i) \tag{3}$$

where $p(x_i|\alpha_i) = \mathcal{N}(x_i|0, \alpha_i^{-1})$, and $\boldsymbol{\alpha} \triangleq \{\alpha_i\}$ are non-negative hyperparameters controlling the sparsity of the signal $\boldsymbol{x}$. Clearly, when α_i approaches infinity, the corresponding coefficient x_i becomes zero. By placing hyperpriors over $\{\alpha_i\}$, the hyperparameters $\{\alpha_i\}$ can be learned by maximizing their posterior probability. We see that in the above hierarchical Bayesian model, each hyperparameter is associated independently with each coefficient. The prior model assumes independence among coefficients and has no potential to encourage clustered sparse solutions.

In [16], 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, i.e.,

$$p(\boldsymbol{x}_i|\alpha_i) = \mathcal{N}(0, \alpha_i^{-1}\boldsymbol{I}) \tag{4}$$

where $\boldsymbol{x}_i$ denotes the ith block of $\boldsymbol{x}$, α_i is the hyperparameter controlling the sparsity of $\boldsymbol{x}_i$. The model (4) was further improved in [17] to accommodate temporally correlated sources

$$p(\boldsymbol{x}_i|\alpha_i) = \mathcal{N}(0, \alpha_i^{-1}\boldsymbol{B}_i) \tag{5}$$

in which $\boldsymbol{B}_i$ is a positive definite matrix that captures the correlation structure of $\boldsymbol{x}_i$. Nevertheless, as can be seen, both models (4) and (5) require the block partition knowledge in order to assign a common hyperparameter to a group of coefficients that share the same sparsity pattern.

B. Proposed Pattern-Coupled Hierarchical Model

In the following, we propose a pattern-coupled hierarchical model to cope with block-sparse signals with unknown block-sparse structures. The new model utilizes the fact that the sparsity patterns of neighboring coefficients are statistically dependent. Specifically, in our model, the Gaussian prior for each coefficient not only involves its own hyperparameter, but also its immediate neighbor hyperparameters. More precisely, a prior over $\boldsymbol{x}$ is given by

$$p(\boldsymbol{x}|\boldsymbol{\alpha}) = \prod_{i=1}^{n} p(x_i|\alpha_i, \alpha_{i+1}, \alpha_{i-1}) \tag{6}$$

where

$$p(x_i|\alpha_i, \alpha_{i+1}, \alpha_{i-1}) = \mathcal{N}(x_i|0, (\alpha_i + \beta\alpha_{i+1} + \beta\alpha_{i-1})^{-1}) \tag{7}$$

and we assume $\alpha_0 = 0$ and $\alpha_{n+1} = 0$ for the end points x_1 and x_n, $0 \leq \beta \leq 1$ is a parameter indicating the pattern relevance between the coefficient x_i and its neighboring coefficients $\{x_{i+1}, x_{i-1}\}$. When $\beta = 0$, the prior distribution (6) reduces to the prior for the conventional sparse Bayesian learning. When $\beta > 0$, we see that the sparsity of x_i is not only controlled by the hyperparameter α_i, but also by the neighboring hyperparameters $\{\alpha_{i+1}, \alpha_{i-1}\}$. Suppose α_i approaches infinity, then its corresponding coefficient x_i will become zero. Meanwhile, since α_i is involved in the prior for the neighboring coefficients $\{x_{i+1}, x_{i-1}\}$, these two coefficients will decrease to zero as well. We see that the sparsity patterns of neighboring coefficients are coupled through their shared hyperparameters. On the other hand, the hyperparameters, during the learning process, are also related to each other through their commonly connected coefficients. Such a coupled hierarchical model has the potential to encourage structured-sparse solutions, while without imposing any pre-defined structures on the recovered signals. This property enables to learn the block-sparse structure in an automatic manner.

Following the conventional sparse Bayesian learning, we use Gamma distributions as hyperpriors over the hyperparameters $\{\alpha_i\}$, i.e.,

$$p(\boldsymbol{\alpha}) = \prod_{i=1}^{n} \text{Gamma}(\alpha_i|a, b) = \prod_{i=1}^{n} \Gamma(a)^{-1} b^a \alpha_i^a e^{-b\alpha_i} \tag{8}$$

where $\Gamma(a) = \int_0^\infty t^{a-1}e^{-t}dt$ is the Gamma function. As discussed in [24], this two-layer Gaussian-inverse Gamma hierarchical prior results in a learning process which tends to switch off most of the coefficients that are deemed to be irrelevant, and only keep very few relevant coefficients to explain the data. This mechanism is also called as "automatic relevance determination". In the conventional sparse Bayesian framework, very small values, e.g., 10^{-4}, are assigned to the two parameters a and b. In this paper, similar to the conventional sparse Bayesian learning, we assign a very small value, say 10^{-4}, to the parameter b. Nevertheless, the choice of a is different: we use a more favorable prior which sets a larger a (say, $a = 1$) in order to achieve the desired "pruning" effect for our proposed hierarchical Bayesian model. Clearly, the Gamma prior with a larger a encourages large values of the hyperparameters, and therefore promotes the sparseness of the solution since the larger the hyperparameter, the smaller the variance of the corresponding coefficient. We will elaborate on the choice of the parameter a and its impact on the recovery performance later in our paper.

C. Discussions

Below we provide some discussions to gain insight into our proposed hierarchical model. As stated earlier, when a hyperparameter α_i approaches infinity, both its corresponding coefficient x_i and its neighboring coefficients $\{x_{i+1}, x_{i-1}\}$ will become zero. This zero coupling effect enables to recover block-sparse signals in a more reliable way. Note that due to the zero coupling, sporadic recovery errors which misidentify an active (nonzero) coefficient (located in a nonzero block) as an isolated zero component are almost impossible to happen since a nonzero-to-zero misidentification usually means that one of the hyperparameters associated with this misidentified coefficient becomes infinity. In other words, nonzero-to-zero flip-over errors should occur in clusters. Since it is much less likely to make a sequence of consecutive errors than making a single error during the reconstruction, the zero coupling can help improve the reliability of block-sparse signal recovery. Also, the likelihood of misidentifying a zero component (in a zero block) as an active coefficient is considerably reduced as either one of its neighboring hyperparameters goes to infinity, this coefficient will be driven to zero.

Although our proposed hierarchical model tends to encourage structured-sparse solutions, it is also flexible to accommodate conventional sparse signals with isolated active coefficients. Suppose x_i is an isolated nonzero coefficient and its neighboring coefficients $\{x_{i\pm1}, x_{i\pm2}, x_{i\pm3}\}$ are all zeros. Exact recovery of the isolated coefficient along with its zero neighboring coefficients is still possible, in which case the estimated hyperparameters $\{\alpha_i, \alpha_{i\pm1}\}$ will have finite values, while $\{\alpha_{i\pm2}\}$ will become infinity such that the estimated

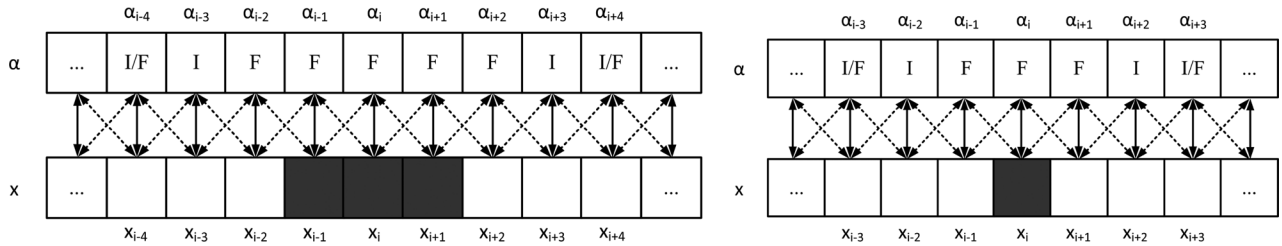


Fig. 1. Schematic diagrams to show the flexibility of the proposed hierarchical model in accommodating block-sparse patterns (left figure) as well as conventional sparse patterns (right figure), where 'F' indicates that the corresponding hyperparameter α_i has a finite value, 'I' indicates an infinite value, shadow and blank square-blocks represent nonzero and zero coefficients, respectively.

coefficients $\{x_{i\pm1}, x_{i\pm2}, x_{i\pm3}\}$ are zeros (See Fig. 1). This is exactly the case in our simulations. Experimental results also corroborate our claim: our proposed algorithm provides decent performance in exactly recovering conventional sparse signals.

In our model, each coefficient (say x_i) only has connections with its immediate (forward and backward) neighbor hyperparameters (besides its own), i.e., $\{\alpha_{i+1}, \alpha_{i-1}\}$. This connection pattern can be readily generalized to include multiple neighbor hyperparameters. Nevertheless, associating each coefficient with multiple neighbor hyperparameters could lead to an excessive coupling effect. Suppose each coefficient is connected to two (forward and backward) neighbor hyperparameters. In this case, the hyperparameter α_i has connections with the following five coefficients $\{x_i, x_{i\pm1}, x_{i\pm2}\}$. As a consequence, if a coefficient is misidentified, then this recovery error will affect the recovery of its neighboring four coefficients. In this sense, considering immediate neighbors seems render the most flexible framework to characterize block-sparse signals.

If the block partition is known *a priori*, the knowledge of the block partition (e.g., number of blocks and block sizes) can be utilized to facilitate the algorithm design [16], [17]. In this case, the number of hyperparameters that are used to control the sparsity of the coefficients can be significantly reduced because the coefficients in the same block share the same sparsity pattern. Nevertheless, for our work, the knowledge of block partition is unknown and therefore we still need to introduce n hyperparameters to control the sparsity of n coefficients.

III. Proposed Bayesian Inference Algorithm: Known Noise Variance

We now proceed to develop a sparse Bayesian learning method for block-sparse signal recovery. For ease of exposition, we first assume that the noise variance σ^2 is known *a priori*. Extension of the Bayesian inference to the case of unknown noise variance will be discussed in the next section. Based on the above hierarchical model, the posterior distribution of $\boldsymbol{x}$ can be computed as

$$p(\boldsymbol{x}|\boldsymbol{\alpha}, \boldsymbol{y}) \propto p(\boldsymbol{x}|\boldsymbol{\alpha})p(\boldsymbol{y}|\boldsymbol{x}) \tag{9}$$

where $\boldsymbol{\alpha} \triangleq \{\alpha_i\}$, $p(\boldsymbol{x}|\boldsymbol{\alpha})$ is given by (6), and

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{(\sqrt{2\pi\sigma^2})^m} \exp\left(-\frac{\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2}{2\sigma^2}\right) \tag{10}$$

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

$$\begin{aligned} \boldsymbol{\mu} &= \sigma^{-2}\boldsymbol{\Phi}\boldsymbol{A}^T\boldsymbol{y} \\ \boldsymbol{\Phi} &= (\sigma^{-2}\boldsymbol{A}^T\boldsymbol{A} + \boldsymbol{D})^{-1} \end{aligned} \tag{11}$$

where $\boldsymbol{D}$ is a diagonal matrix with its ith diagonal element equal to $(\alpha_i + \beta\alpha_{i+1} + \beta\alpha_{i-1})$, i.e.,

$$\boldsymbol{D} \triangleq \mathrm{diag}(\alpha_1 + \beta\alpha_2 + \beta\alpha_0, \ldots, \alpha_n + \beta\alpha_{n-1} + \beta\alpha_{n+1}) \tag{12}$$

Given a set of estimated hyperparameters $\{\alpha_i\}$, the maximum a posterior (MAP) estimate of $\boldsymbol{x}$ is the mean of its posterior distribution, i.e.,

$$\hat{\boldsymbol{x}}_{\mathrm{MAP}} = \boldsymbol{\mu} = (\boldsymbol{A}^T\boldsymbol{A} + \sigma^2\boldsymbol{D})^{-1}\boldsymbol{A}^T\boldsymbol{y} \tag{13}$$

Our problem therefore reduces to estimating the set of hyperparameters $\{\alpha_i\}$. With hyperpriors placed over α_i, learning the hyperparameters becomes a search for their posterior mode, i.e., maximization of the posterior probability $p(\boldsymbol{\alpha}|\boldsymbol{y})$. A strategy to maximize the posterior probability is to exploit the expectation-maximization (EM) formulation [24], treating the signal $\boldsymbol{x}$ as hidden variables and iteratively maximizing a lower bound of the posterior probability $p(\boldsymbol{\alpha}|\boldsymbol{y})$, i.e., $E_{\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\alpha}}[\log p(\boldsymbol{\alpha}|\boldsymbol{x})]$, the expected value of the complete log-posterior of $\boldsymbol{\alpha}$, where the operator $E_{\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\alpha}}[\cdot]$ denotes the expectation with respect to the distribution $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha})$. Specifically, the algorithm produces a sequence of estimates $\boldsymbol{\alpha}^{(t)}$, $t = 1, 2, 3, \ldots$, by applying two alternating steps, namely, the E-step and the M-step [29].

1) E-Step: Given the current estimates of the hyperparameters $\boldsymbol{\alpha}^{(t)}$ and the observed data $\boldsymbol{y}$, the E-step requires computing the expected value (with respect to the missing variables $\boldsymbol{x}$) of the complete log-posterior of $\boldsymbol{\alpha}$, which is also referred to as the Q-function; we have

$$\begin{aligned} Q(\boldsymbol{\alpha}|\boldsymbol{\alpha}^{(t)}) &= E_{\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\alpha}^{(t)}}[\log p(\boldsymbol{\alpha}|\boldsymbol{x})] \\ &= \int p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha}^{(t)}) \log p(\boldsymbol{\alpha}|\boldsymbol{x}) d\boldsymbol{x} \\ &= \int p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha}^{(t)}) \log[p(\boldsymbol{\alpha})p(\boldsymbol{x}|\boldsymbol{\alpha})] d\boldsymbol{x} + c \\ &= \log p(\boldsymbol{\alpha}) + \int p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha}^{(t)}) \log p(\boldsymbol{x}|\boldsymbol{\alpha}) d\boldsymbol{x} + c \end{aligned} \tag{14}$$

where c is a constant independent of $\boldsymbol{\alpha}$. Substituting (6) into (14), and ignoring the term independent of $\boldsymbol{\alpha}$, the Q-function can be re-expressed as

$$Q(\boldsymbol{\alpha}|\boldsymbol{\alpha}^{(t)}) = \log p(\boldsymbol{\alpha}) + \frac{1}{2}\sum_{i=1}^{n}\Bigg(\log(\alpha_i + \beta\alpha_{i+1} + \beta\alpha_{i-1}) - (\alpha_i + \beta\alpha_{i+1} + \beta\alpha_{i-1})\int p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha}^{(t)})x_i^2 d\boldsymbol{x}\Bigg) \quad (15)$$

Since the posterior $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha}^{(t)})$ is a multivariate Gaussian distribution with its mean and covariance matrix given by (11), we have

$$\int p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha}^{(t)})x_i^2 d\boldsymbol{x} = E_{\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\alpha}^{(t)}}\left[x_i^2\right] = \hat{\mu}_i^2 + \hat{\phi}_{i,i} \quad (16)$$

where $\hat{\mu}_i$ denotes the ith entry of $\hat{\boldsymbol{\mu}}$, $\hat{\phi}_{i,i}$ denotes the ith diagonal element of the covariance matrix $\hat{\boldsymbol{\Phi}}$, $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Phi}}$ are computed according to (11), with $\boldsymbol{\alpha}$ replaced by the current estimate $\boldsymbol{\alpha}^{(t)}$. With the specified prior (8), the Q-function can eventually be written as

$$Q(\boldsymbol{\alpha}|\boldsymbol{\alpha}^{(t)}) = \sum_{i=1}^{n}\Bigg(a\log\alpha_i - b\alpha_i + \frac{1}{2}\log(\alpha_i + \beta\alpha_{i+1} + \beta\alpha_{i-1}) - \frac{1}{2}(\alpha_i + \beta\alpha_{i+1} + \beta\alpha_{i-1})(\hat{\mu}_i^2 + \hat{\phi}_{i,i})\Bigg) \quad (17)$$

2) M-Step: In the M-step, a new estimate of $\boldsymbol{\alpha}$ is obtained by maximizing the Q-function, i.e.,

$$\boldsymbol{\alpha}^{(t+1)} = \arg\max_{\boldsymbol{\alpha}} Q(\boldsymbol{\alpha}|\boldsymbol{\alpha}^{(t)}) \quad (18)$$

For the conventional sparse Bayesian learning, maximization of the Q-function can be decoupled into a number of separate optimizations in which each hyperparameter α_i is updated independently. This, however, is not the case for the problem being considered here. We see that the hyperparameters in the Q-function (17) are entangled with each other due to the logarithm term $\log(\alpha_i + \beta\alpha_{i+1} + \beta\alpha_{i-1})$. In this case, an analytical solution to the optimization (18) 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, such a gradient-based search, albeit effective, does not provide any insight into the learning process. Moreover, gradient-based methods involve higher computational complexity as compared with an analytical update rule. To overcome the drawbacks of gradient-based methods, we consider an alternative strategy which aims at finding a simple, analytical sub-optimal solution of (18). Such an analytical sub-optimal solution can be obtained by examining the optimality condition of (18). Suppose $\boldsymbol{\alpha}^*$ is the optimal solution of (18), then the first derivative of the Q-function with respect to $\boldsymbol{\alpha}$ equals to zero at the optimal point, i.e.,

$$\left.\frac{\partial Q(\boldsymbol{\alpha}|\boldsymbol{\alpha}^{(t)})}{\partial\boldsymbol{\alpha}}\right|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}^*} = \mathbf{0} \quad (19)$$

To examine this optimality condition more thoroughly, we compute the first derivative of the Q-function with respect to each individual hyperparameter:

$$\frac{\partial Q(\boldsymbol{\alpha}|\boldsymbol{\alpha}^{(t)})}{\partial\alpha_i} = \frac{a}{\alpha_i} - b - \frac{1}{2}\omega_i + \frac{1}{2}(\nu_i + \beta\nu_{i+1} + \beta\nu_{i-1}) \quad \forall i = 1, \ldots, n \quad (20)$$

where $\nu_0 = 0$, $\nu_{n+1} = 0$, and for $i = 1, \ldots, n$, we have

$$\omega_i \triangleq (\hat{\mu}_i^2 + \hat{\phi}_{i,i}) + \beta(\hat{\mu}_{i+1}^2 + \hat{\phi}_{i+1,i+1}) + \beta(\hat{\mu}_{i-1}^2 + \hat{\phi}_{i-1,i-1}) \quad (21)$$

$$\nu_i \triangleq \frac{1}{\alpha_i + \beta\alpha_{i+1} + \beta\alpha_{i-1}} \quad (22)$$

Note that for notational convenience, we allow the subscript indices of the notations $\hat{\mu}_i$ and $\hat{\phi}_{i,i}$ in (21) equal to 0 and $n+1$. Although these notations $\{\hat{\mu}_0, \hat{\phi}_{0,0}, \hat{\mu}_{n+1}, \hat{\phi}_{n+1,n+1}\}$ do not have any meaning, they can be used to simplify our expression. Clearly, they should all be set equal to zero, i.e., $\hat{\mu}_0 = \hat{\mu}_{n+1} = \hat{\phi}_{0,0} = \hat{\phi}_{n+1,n+1} = 0$. Recalling the optimality condition, we therefore have

$$\frac{a}{\alpha_i^*} + \frac{1}{2}(\nu_i^* + \beta\nu_{i+1}^* + \beta\nu_{i-1}^*) = b + \frac{1}{2}\omega_i \quad \forall i = 1, \ldots, n \quad (23)$$

where $\nu_0^* = 0$, $\nu_{n+1}^* = 0$, and

$$\nu_i^* \triangleq \frac{1}{\alpha_i^* + \beta\alpha_{i+1}^* + \beta\alpha_{i-1}^*} \qquad \forall i = 1, \ldots, n$$

Since all hyperparameters $\{\alpha_i\}$ and β are non-negative, we have

$$\frac{1}{\alpha_i^*} > \nu_i^* > 0 \qquad \forall i = 1, \ldots, n$$
$$\frac{1}{\beta\alpha_{i+1}^*} > \nu_i^* > 0 \qquad \forall i = 1, \ldots, n-1$$
$$\frac{1}{\beta\alpha_{i-1}^*} > \nu_i^* > 0 \qquad \forall i = 2, \ldots, n$$

Hence the term on the left-hand side of (23) is lower and upper bounded respectively by

$$\frac{a + c_0}{\alpha_i^*} \geq \frac{a}{\alpha_i^*} + \frac{1}{2}(\nu_i^* + \beta\nu_{i+1}^* + \beta\nu_{i-1}^*) > \frac{a}{\alpha_i^*} \quad (24)$$

where $c_0 = 1.5$ for $i = 2, \ldots, n-1$, and $c_0 = 1$ for $i = \{1, n\}$. Combining (23)–(24), we arrive at

$$\alpha_i^* \in \left[\frac{a}{0.5\omega_i + b}, \frac{a + c_0}{0.5\omega_i + b}\right] \quad \forall i = 1, \ldots, n \quad (25)$$

A sub-optimal solution to (18) can therefore be simply chosen as

$$\hat{\alpha}_i = \frac{a}{0.5\omega_i + b} \qquad \forall i = 1, \ldots, n \quad (26)$$

where $a > 0$ and $b = 10^{-4}$. We see that the solution (26) provides a simple rule for the hyperparameter update. Also, notice that the update rule (26) resembles that of the conventional sparse Bayesian learning work [24], [25] except that the parameter ω_i is equal to $\hat{\mu}_i^2 + \hat{\phi}_{i,i}$ for the conventional sparse Bayesian

learning method, while for our case, ω_i is a weighted summation of $\hat{\mu}_j^2 + \hat{\phi}_{j,j}$ for $j = i-1,\ i, i+1$.

For clarity, we now summarize the proposed iterative algorithm as follows.

1) At iteration t ($t = 0, 1, \ldots$): Given a set of hyperparameters $\boldsymbol{\alpha}^{(t)} = \{\alpha_i^{(t)}\}$, compute the mean $\hat{\boldsymbol{\mu}}$ and covariance matrix $\hat{\boldsymbol{\Phi}}$ of the posterior distribution $p(\boldsymbol{x}|\boldsymbol{\alpha}^{(t)}, \boldsymbol{y})$ according to (11), and compute the MAP estimate $\hat{\boldsymbol{x}}^{(t)}$ according to (13).
2) Update the hyperparameters $\boldsymbol{\alpha}^{(t+1)}$ according to (26), where ω_i is given by (21).
3) Continue the above iteration until $\|\hat{\boldsymbol{x}}^{(t+1)} - \hat{\boldsymbol{x}}^{(t)}\|_2 \leq \epsilon$, where ϵ is a prescribed tolerance value.

Remark 1: Although the above algorithm employs a sub-optimal solution (26) to update the hyperparameters in the M-step, numerical results show that the sub-optimal update rule is quite effective and presents recovery performance similar to using a gradient-based search method. This is because the sub-optimal solution (26) provides a reasonable estimate of the optimal solution when the parameter a is set away from zero. In addition, although a theoretical analysis of the convergence behavior is unavailable, the proposed algorithm demonstrates a fast convergence rate and is guaranteed to converge to the true solution (in the noiseless case) with overwhelming probability given a decent number of measurements. Numerical results also suggest that the choice of a is not very critical to the recovery performance: the proposed algorithm provides stable recovery performance as long as a is set in a reasonable region $a \in [0.5, 2]$.

Remark 2: The update rule (26) not only admits a simple analytical form which is computationally efficient, it also provides an insight into the proposed algorithm. The Bayesian Occam's razor which contributes to the success of the conventional sparse Bayesian learning method also works here to automatically select an appropriate simple model. To see this, note that in the E-step, when computing the posterior mean and covariance matrix, a large hyperparameter α_i tends to suppress the values of the corresponding components $\{\mu_j, \phi_j\}$ for $j = i-1,\ i, i+1$ (cf. (11)). As a result, the value of ω_i becomes small, which in turn leads to a larger hyperparameter α_i (cf. (26)). This negative feedback mechanism keeps decreasing most of the entries in $\hat{\boldsymbol{x}}$ until they become negligible, while leaving only a few prominent nonzero entries survived to explain the data. The process eventually leads to a block-sparse solution. Nevertheless, when the noise variance is relatively large, due to the coupling effect, recovery errors (e.g., zeros misidentified as nonzero values) tend to spread across coefficients instead of occurring in a sporadic manner. In this case, some coefficients are reduced to small values (of order 10^{-3}) but will not keep decreasing and reach machine precision. To deal with this issue, in the hyperparameter update stage, we can simply set the hyperparameters which are greater than a certain value (e.g., 10^3) equal to a sufficiently large value (e.g., 10^8). Specifically, the update rule (26) for the hyperparameters is replaced by

$$\hat{\alpha}_i^{(t+1)} = \begin{cases} \frac{a}{0.5\omega_i + b} & \text{if } \hat{\alpha}_i^{(t+1)} < \tau \\ 10^8 & \text{if } \hat{\alpha}_i^{(t+1)} \geq \tau \end{cases} \quad \forall i = 1, \ldots, n \qquad (27)$$

where τ is a tuning parameter introduced to our proposed algorithm. It is generally not difficult to set τ. Our simulation result shows that stable recovery performance can always be achieved when τ is set in the range $[0.5 \times 10^3, 5 \times 10^3]$. The above update rule in fact amounts to pruning those coefficients that are small during iterations. In our simulations, unless otherwise specified, the original update rule (26) is used for the hyperparameter update.

Remark 3: The parameter β is introduced to quantify the pattern relevance among neighboring coefficients. Its choice seems not very critical to the recovery performance, as demonstrated by our simulation results. There are indeed scenarios[2] where the choice β makes a difference to the recovery performance. Nevertheless, our experiments suggest that for signals which have a block-sparse structure, choosing a nonzero $\beta \in (0, 1]$ can always result in a performance improvement as compared with setting $\beta = 0$. For the case where the block-sparse pattern is not very clear, a safe choice for β could be $\beta = 0.1$, which imposes a very mild coupling effect among neighboring coefficients. Also, we conjecture that by placing an appropriate prior over β, the parameter β may be learned from the data as well, and hopefully, the algorithm will eventually have the ability to automatically learn whether or not a sparse signal has a block-sparse structure. This will be an interesting direction worthy of our future study.

Remark 4: For our proposed algorithm, the main computational task at each iteration is to calculate the covariance matrix $\boldsymbol{\Phi}$, which involves computing the inverse of an $n \times n$ matrix. By using the Woodbury identity, this $n \times n$ matrix inversion can be converted to an $m \times m$ matrix inversion. Hence the total number of floating-point operations required at each iteration are of order $\mathcal{O}(m^3)$, and of order $\mathcal{O}(Lm^3)$ for the proposed algorithm, where L denotes the number of required iterations for convergence.

IV. Proposed Bayesian Inference Algorithm: Unknown Noise Variance

In the previous section, for simplicity of exposition, we assume that the noise variance σ^2 is known *a priori*. This assumption, however, may not hold valid in practice. In this section, we discuss how to extend our previously developed Bayesian inference method to the scenario where the noise variance σ^2 is unknown.

For notational convenience, define

$$\gamma \triangleq \sigma^{-2}.$$

Following the conventional sparse Bayesian learning framework [24], we place a Gamma hyperprior over γ:

$$p(\gamma) = \text{Gamma}(\gamma|c, d) = \Gamma(c)^{-1} d^c \gamma^c e^{-d\gamma} \qquad (28)$$

where the parameters c and d are set to small values, e.g., $c = d = 10^{-4}$. As we already derived in the previous section, given the hyperparameters $\boldsymbol{\alpha}$ and the noise variance σ^2, the posterior $p(\boldsymbol{x}|\boldsymbol{\alpha}, \gamma, \boldsymbol{y})$ follows a Gaussian distribution with its mean and covariance matrix given by (11). The MAP estimate of $\boldsymbol{x}$

[2]Our algorithm can be adapted to recover the time-varying sparse signals. In such a scenario our experimental results suggest that the choice of β makes a difference. The result, however, is not included because it is beyond the scope of this paper.

is equivalent to the posterior mean. Our problem therefore becomes jointly estimating the hyperparameters $\boldsymbol{\alpha}$ and the noise variance σ^2 (or equivalently γ). Again, these parameters can be learned via maximizing their posterior probability $p(\boldsymbol{\alpha}, \gamma|\boldsymbol{y})$. The alternating EM steps are briefly discussed below.

1) E-Step: In the E-step, given the current estimates of the parameters $\{\boldsymbol{\alpha}^{(t)}, \gamma^{(t)}\}$ and the observed data $\boldsymbol{y}$, we compute the expected value (with respect to the missing variables $\boldsymbol{x}$) of the complete log-posterior of $\{\boldsymbol{\alpha}, \gamma\}$, that is, $E_{\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\alpha}^{(t)},\gamma^{(t)}}[\log p(\boldsymbol{\alpha}, \gamma|\boldsymbol{x}, \boldsymbol{y})]$, where the operator $E_{\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\alpha}^{(t)},\gamma^{(t)}}[\cdot]$ denotes the expectation with respect to the distribution $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha}^{(t)}, \gamma^{(t)})$. Since

$$p(\boldsymbol{\alpha}, \gamma|\boldsymbol{x}, \boldsymbol{y}) \propto p(\boldsymbol{\alpha})p(\boldsymbol{x}|\boldsymbol{\alpha})p(\gamma)p(\boldsymbol{y}|\boldsymbol{x}, \gamma) \tag{29}$$

the Q-function can be expressed as a summation of two terms

$$\begin{aligned} Q(\boldsymbol{\alpha}, \gamma|\boldsymbol{\alpha}^{(t)}, \gamma^{(t)}) = {} & E_{\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\alpha}^{(t)},\gamma^{(t)}}[\log p(\boldsymbol{\alpha})p(\boldsymbol{x}|\boldsymbol{\alpha})] \\ & + E_{\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\alpha}^{(t)},\gamma^{(t)}}[\log p(\gamma)p(\boldsymbol{y}|\boldsymbol{x}, \gamma)] \end{aligned} \tag{30}$$

where the first term has exactly the same form as the Q-function (14) obtained in the previous section, except with the known noise variance σ^2 replaced by the current estimate $(\sigma^{(t)})^2 = 1/\gamma^{(t)}$, and the second term is a function of the variable γ.

2) M-Step: We observe that in the Q-function (30), the parameters $\boldsymbol{\alpha}$ and γ to be learned are separated from each other. This allows the estimation of $\boldsymbol{\alpha}$ and γ to be decoupled into the following two independent problems:

$$\boldsymbol{\alpha}^{(t+1)} = \arg\max_{\boldsymbol{\alpha}} E_{\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\alpha}^{(t)},\gamma^{(t)}}[\log p(\boldsymbol{\alpha})p(\boldsymbol{x}|\boldsymbol{\alpha})] \tag{31}$$

$$\gamma^{(t+1)} = \arg\max_{\gamma} E_{\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\alpha}^{(t)},\gamma^{(t)}}[\log p(\gamma)p(\boldsymbol{y}|\boldsymbol{x}, \gamma)] \tag{32}$$

The first optimization problem (31) has been thoroughly studied in the previous section, where we provided a simple analytical form (26) for the hyperparameter update. We now discuss the estimation of the parameter γ. Recalling (28), we have

$$\begin{aligned} & E_{\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\alpha}^{(t)},\gamma^{(t)}}[\log p(\gamma)p(\boldsymbol{y}|\boldsymbol{x}, \gamma)] \\ & = \frac{m}{2}\log\gamma - \frac{\gamma}{2}E_{\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\alpha}^{(t)},\gamma^{(t)}}\left[\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2\right] + c\log\gamma - d\gamma \end{aligned} \tag{33}$$

Computing the first derivative of (33) with respect to γ and setting it equal to zero, we get

$$\frac{1}{\gamma} = \frac{\chi + 2d}{m + 2c} \tag{34}$$

where

$$\chi \triangleq E_{\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\alpha}^{(t)},\gamma^{(t)}}\left[\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2\right]$$

Note that the posterior $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha}^{(t)}, \gamma^{(t)})$ follows a Gaussian distribution with mean $\hat{\boldsymbol{\mu}}$ and covariance matrix $\hat{\boldsymbol{\Phi}}$, where $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Phi}}$ are computed via (11) with γ (i.e., σ^2) and $\boldsymbol{\alpha}$ replaced by the current estimates $\{\gamma^{(t)}, \boldsymbol{\alpha}^{(t)}\}$. Hence χ can be computed as

$$\begin{aligned} \chi = {} & \boldsymbol{y}^T\boldsymbol{y} - 2E[\boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{y}] + E[\boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x}] \\ = {} & \boldsymbol{y}^T\boldsymbol{y} - 2\hat{\boldsymbol{\mu}}^T\boldsymbol{A}^T\boldsymbol{y} + \hat{\boldsymbol{\mu}}^T\boldsymbol{A}^T\boldsymbol{A}\hat{\boldsymbol{\mu}} + \mathrm{tr}\left(\hat{\boldsymbol{\Phi}}\boldsymbol{A}^T\boldsymbol{A}\right) \\ \overset{(a)}{=} {} & \|\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{\mu}}\|_2^2 + (\gamma^{(t)})^{-1}\sum_{i=1}^{n}\rho_i \end{aligned} \tag{35}$$

where the last equality (a) follows from

$$\begin{aligned} \mathrm{tr}\left(\hat{\boldsymbol{\Phi}}\boldsymbol{A}^T\boldsymbol{A}\right) & = \mathrm{tr}\left(\hat{\boldsymbol{\Phi}}\boldsymbol{A}^T\boldsymbol{A} + (\gamma^{(t)})^{-1}\hat{\boldsymbol{\Phi}}\hat{\boldsymbol{D}} - (\gamma^{(t)})^{-1}\hat{\boldsymbol{\Phi}}\hat{\boldsymbol{D}}\right) \\ & = (\gamma^{(t)})^{-1}\mathrm{tr}\left(\hat{\boldsymbol{\Phi}}(\gamma^{(t)}\boldsymbol{A}^T\boldsymbol{A} + \hat{\boldsymbol{D}}) - \hat{\boldsymbol{\Phi}}\hat{\boldsymbol{D}}\right) \\ & = (\gamma^{(t)})^{-1}\mathrm{tr}\left(\boldsymbol{I} - \hat{\boldsymbol{\Phi}}\hat{\boldsymbol{D}}\right) \\ & = (\gamma^{(t)})^{-1}\sum_{i=1}^{n}\rho_i \end{aligned} \tag{36}$$

in which $\hat{\boldsymbol{D}}$ is given by (12) with $\boldsymbol{\alpha}$ replaced by the current estimate $\boldsymbol{\alpha}^{(t)}$, and

$$\rho_i \triangleq 1 - \hat{\phi}_{i,i}(\alpha_i^{(t)} + \beta\alpha_{i-1}^{(t)} + \beta\alpha_{i+1}^{(t)}) \qquad \forall i \tag{37}$$

Note that $\alpha_0^{(t)}$ and $\alpha_{n+1}^{(t)}$ are set to zero when computing ρ_1 and ρ_n. Substituting (35) back into (34), a new estimate of γ, i.e., the optimal solution to (32), is given by

$$\frac{1}{\gamma^{(t+1)}} = \frac{\|\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{\mu}}\|_2^2 + (\gamma^{(t)})^{-1}\sum_i \rho_i + 2d}{m + 2c} \tag{38}$$

The above update formula has a similar form as that for the conventional sparse Bayesian learning (cf. [24, Equation (50)]). The only difference lies in that $\{\rho_i\}$ are computed differently: for the conventional sparse Bayesian learning method, ρ_i is computed as $\rho_i = 1 - \hat{\phi}_{i,i}\alpha_i^{(t)}$, while ρ_i is given by (37) for our algorithm.

The sparse Bayesian learning algorithm with unknown noise variance is now summarized as follows.

1) At iteration t ($t = 0, 1, \ldots$): given the current estimates of $\boldsymbol{\alpha}^{(t)}$ and $\gamma^{(t)}$, compute the mean $\hat{\boldsymbol{\mu}}$ and the covariance matrix $\hat{\boldsymbol{\Phi}}$ of the posterior distribution $p(\boldsymbol{x}|\boldsymbol{\alpha}^{(t)}, \gamma^{(t)}, \boldsymbol{y})$ via (11), and calculate the MAP estimate $\hat{\boldsymbol{x}}^{(t)}$ according to (13).
2) Compute a new estimate of $\boldsymbol{\alpha}$, denoted as $\boldsymbol{\alpha}^{(t+1)}$, according to (26), where ω_i is given by (21); update γ via (38), which yields a new estimate of γ, denoted as $\gamma^{(t+1)}$.
3) Continue the above iteration until $\|\hat{\boldsymbol{x}}^{(t+1)} - \hat{\boldsymbol{x}}^{(t)}\|_2 \leq \epsilon$, where ϵ is a prescribed tolerance value.

Remark 1: When the noise variance is unknown, both the choice of a and the initial estimate of the noise variance have an impact on the recovery performance. In this case, our experiments suggest that $a = 0.5$ is a safe and robust choice which ensures that the proposed algorithm delivers a stable recovery performance irrespective of the initial estimate of the noise variance.

V. A Modified Iterative Reweighted Algorithm

Sparse Bayesian learning algorithms have a close connection with the reweighted ℓ_1 or ℓ_2 methods. In fact, a dual-form analysis [30] reveals that sparse Bayesian learning can be considered as a non-separable reweighted strategy solving a non-separable penalty function. Inspired by this insight, we here propose a reweighted ℓ_1 method for the recovery of block-sparse signals when the block structure of the sparse signal is unknown.

Conventional reweighted ℓ_1 methods iteratively minimize the following weighted ℓ_1 function (for simplicity, we consider the noise-free case):

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \sum_{i=1}^{n} w_i^{(t)} |x_i| \\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y} \end{aligned} \tag{39}$$

where the weighting parameters are given by $w_i^{(t)} = 1/(|x_i^{(t-1)}| + \epsilon), \forall i$, and ϵ is a pre-specified positive parameter. In a series of experiments [31], the above iterative reweighted algorithm outperforms the conventional ℓ_1-minimization method by a considerable margin. For the above weighted ℓ_1-minimization problem, the weight can in fact be interpreted as a sparse-encouraging hyperparameter associated with each coefficient, and a larger weight has the potential to be more sparsity-encouraging. The fascinating idea of the iterative reweighted algorithm is that the weights are updated based on the previous estimate of the solution. Specifically, previous estimate serves as prior knowledge of the sparse signal. If the estimate of a coefficient is small, then we expect that the coefficient is more likely being sparse, thus a large weight which is more sparse-encouraging is assigned, and vice versa. As a result, the value of the coefficient whose estimate is small tends to be smaller (until become negligible) in the next estimate. This explains why iterative reweighted algorithms usually yield sparser solutions than the conventional ℓ_1-minimization method.

As discussed in our previous section, the basic idea of our proposed sparse Bayesian learning method is to establish a coupling mechanism such that the sparsity patterns of neighboring coefficients are somehow related to each other. Inspired by this, we modify the weight update rule of the reweighted ℓ_1 algorithm as follows

$$w_i^{(t)} = \frac{1}{|x_i^{(t-1)}| + \beta|x_{i+1}^{(t-1)}| + \beta|x_{i-1}^{(t-1)}| + \epsilon} \qquad \forall i \tag{40}$$

We see that, in the above update rule, each weight is not only a function of its corresponding coefficient, but also a function of its neighboring coefficients. Since the amplitude of the weight reflects how much we want to encourage a coefficient to become a sparse component, it means that the neighboring coefficients will also have an impact on the sparsity pattern of each coefficient. In this way, a coupling mechanism among neighboring coefficients is established, and the modified reweighted ℓ_1-minimization algorithm has the potential to encourage block-sparse solutions. Experiments show that the proposed modified reweighted ℓ_1 method yields considerably improved results over the conventional reweighted ℓ_1 method in recovering block-sparse signals. It also serves as a good reference method for comparison with the proposed Bayesian sparse learning approach.

VI. Simulation Results

We now carry out experiments to illustrate the performance of our proposed algorithm, also referred to as the pattern-coupled sparse Bayesian learning (PC-SBL) 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 and b for our proposed algorithm are set equal to $a = 0.5$ and $b = 10^{-4}$ throughout our experiments.

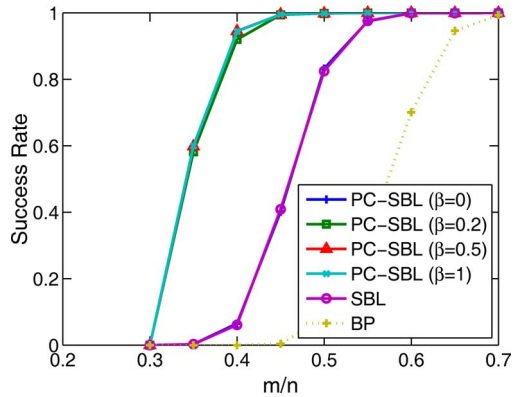


Fig. 2. Success rates of the proposed algorithm vs. the ratio m/n for different choices of β.

A. Synthetic Data

Let us first consider the synthetic data case. In our simulations, we generate the block-sparse signal in a similar way to [23]. Suppose the n-dimensional sparse signal contains K nonzero coefficients (K is also denoted as the sparsity level) which are partitioned into L blocks with random sizes and random locations. Specifically, the block sizes $\{B_l\}_{l=1}^{L}$ can be determined as follows: we randomly generate L positive random variables $\{r_l\}_{l=1}^{L}$ with their sum equal to one, then we can simply set $B_l = \lceil Kr_l \rceil$ 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 can 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 each super-block with a randomly generated starting position (the starting position, however, is selected such that the nonzero block will not go beyond the super-block). Also, in our experiments, 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.

We introduce the following metrics to evaluate the recovery performance of respective algorithms, namely, the normalized mean squared error (NMSE) and the success rate. The NMSE is calculated by averaging normalized squared errors $\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2/\|\boldsymbol{x}\|_2^2$ of 10^3 independent runs, where $\hat{\boldsymbol{x}}$ denotes the estimate of the true signal $\boldsymbol{x}$. 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 error is no greater than 10^{-6}. Also, to examine the ability to identify the true support of sparse signals, a new

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

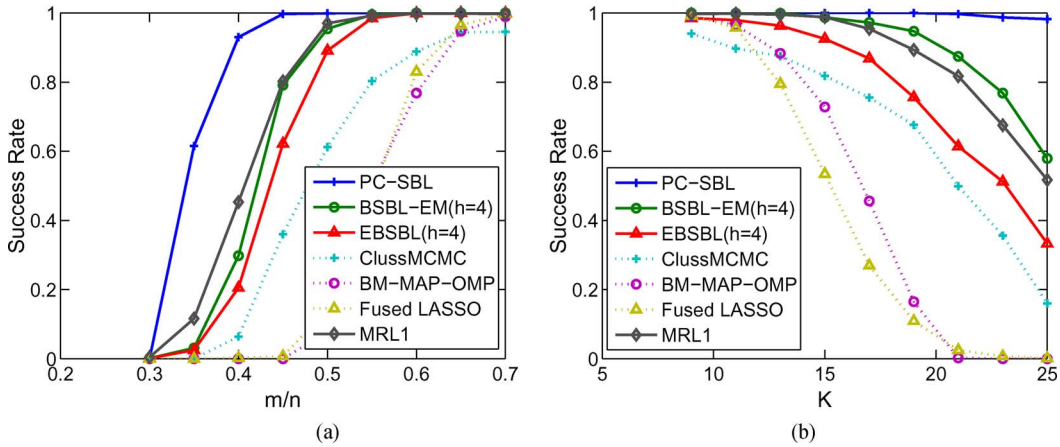


Fig. 3. Success rates of respective algorithms. (a) Success rates vs. m/n, $n = 100$, $K = 25$, and $L = 4$. (b) Success rates vs. K, $m = 40$, $n = 100$, and $L = 3$.

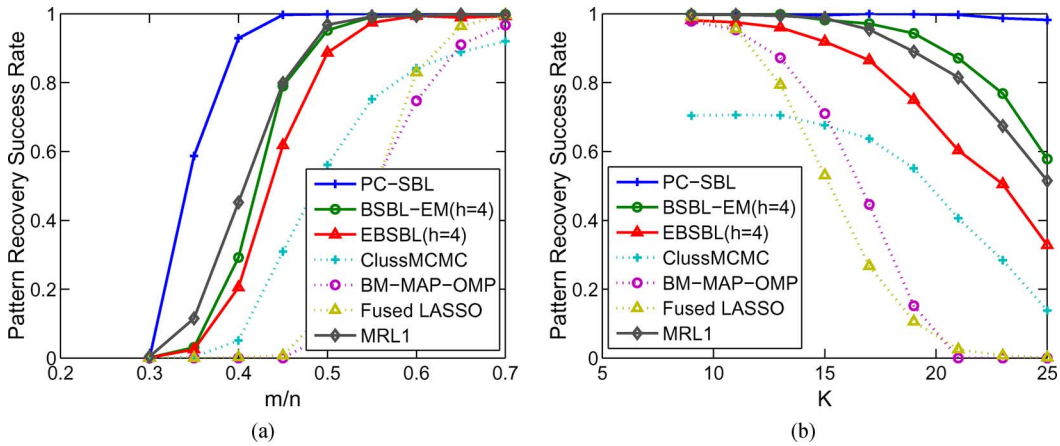


Fig. 4. Pattern recovery success rates of respective algorithms. (a) Pattern recovery success rates vs. m/n, $n = 100$, $K = 25$, and $L = 4$. (b) Pattern recovery success rates vs. K, $m = 40$, $n = 100$, and $L = 3$.

metric named as "pattern recovery success rate" is introduced. The pattern recovery success rate, again, is defined as the ratio of the number of successful trials to the total number of independent runs. A trial, however, is considered successful only if the support of the block-sparse signal is exactly recovered. Note that for all algorithms, inactive coefficients will not be exact zeros due to machine precision and the precision specified by the stopping criterion. A coefficient whose magnitude is less than 10^{-5} is deemed as a zero coefficient (this assumption is only adopted here to calculate the pattern recovery success rates). In our simulations, both the success rate and the pattern recovery success rate are used to measure the recovery performance for the noiseless case.

We first examine the recovery performance of our proposed algorithm (PC-SBL) under different choices of β. As indicated earlier in our paper, β ($0 \leq \beta \leq 1$) is a parameter quantifying the dependencies among neighboring coefficients. Fig. 2 depicts the success rates vs. the ratio m/n for different choices of β, where we set $n = 100$, $K = 25$, and $L = 4$. Results (in Fig. 2 and the following figures) are averaged over 1000 independent runs, with the measurement matrix and the sparse signal randomly generated for each run. The performance of the conventional sparse Bayesian learning method (denoted as "SBL") [24] and the basis pursuit method (denoted as "BP") [1], [2] is also included for our comparison. We see that when $\beta = 0$, our proposed algorithm performs the same as the SBL. This is an expected result since in the case of $\beta = 0$, our proposed algorithm is simplified as the SBL. Nevertheless, when $\beta > 0$, our proposed algorithm achieves a significant performance improvement (as compared with the SBL and BP) through exploiting the underlying block-sparse structure, even without knowing the exact locations and sizes of zero and nonzero blocks. We also observe that our proposed algorithm is not very sensitive to the choice of β as long as $\beta > 0$: it achieves similar success rates for different positive values of β. For simplicity, we set $\beta = 1$ throughout our following experiments.

Next, we compare our proposed algorithm with some other recently developed algorithms for block-sparse signal recovery, namely, the expanded block sparse Bayesian learning method (EBSBL) [23], the Boltzman machine-based greedy pursuit algorithm (BM-MAP-OMP) [20], the cluster-structured MCMC algorithm (CluSS-MCMC) [19], and the fused LASSO method (also referred to as the total-variation approach) [9]. The modified iterative reweighted ℓ_1 method (denoted as MRL1) proposed in Section V is also examined in our simulations. Note that all these algorithms were developed without the knowledge of the block-sparse structure. The block sparse Bayesian learning method (denoted as BSBL) developed in [17], [23] is

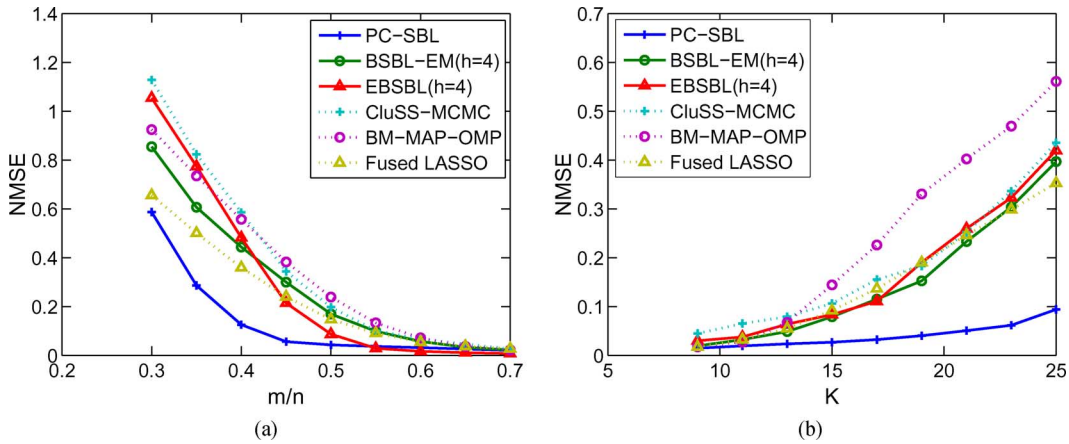


Fig. 5. Normalized MSEs of respective algorithms. (a) Normalized MSEs vs. m/n, $n = 100$, $K = 25$, and $L = 4$. (b) Normalized MSEs vs. the sparsity level K, $m = 40$, $n = 100$, and $L = 3$.

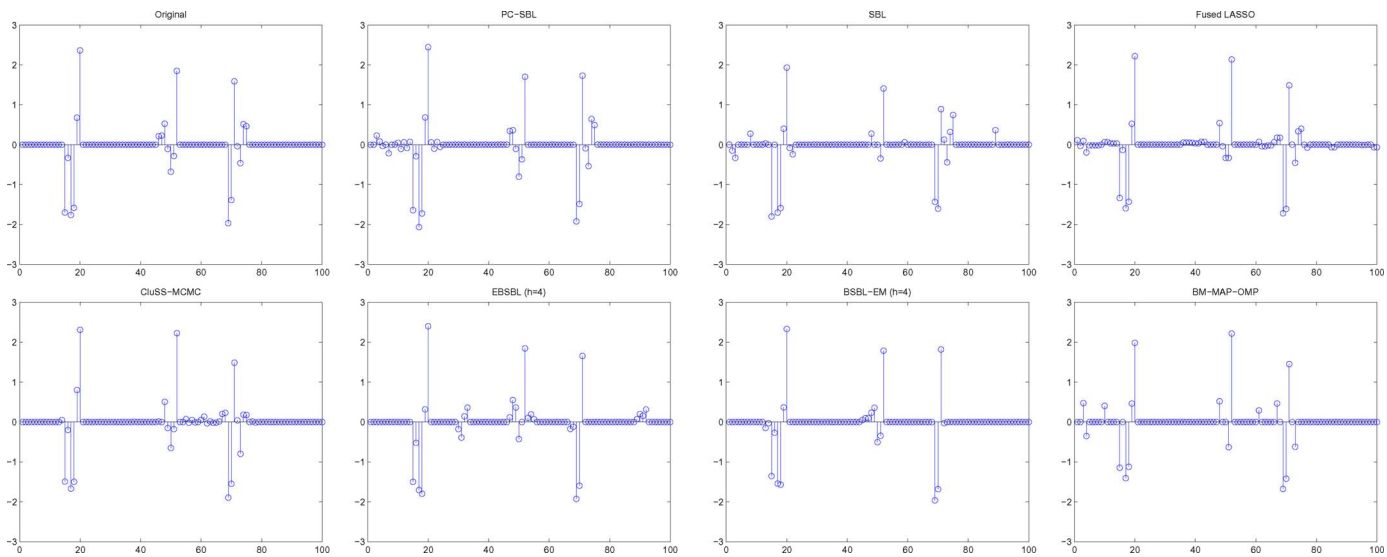


Fig. 6. The original synthetic signal and the signals reconstructed by respective algorithms.

included for comparison as well. Although the BSBL algorithm need to know the block partition, it still provides quite decent performance if the presumed block size, denoted by h, is properly selected. In our simulations, model parameters used by the competing algorithms are adjusted to achieve the best performance. For the EBSBL and the BSBL algorithms, the parameter h is set equal to 4. The fused LASSO method requires to set two values to constrain the ℓ_1-norm of the estimated coefficients and the total variation. These two values are selected using the knowledge of the true solution. For the CluSS-MCMC and the BM-MAP-OMP, we follow the suggestions and rules provided in [19], [20] to choose the parameters for their respective priors. Fig. 3 plots the success rates of respective algorithms as a function of the ratio m/n and the sparsity level K, respectively. Simulation results show that our proposed algorithm achieves highest success rates among all algorithms and outperforms other methods by a considerable margin. We also noticed that the modified reweighted ℓ_1 method (MRL1), although not as good as the proposed PC-SBL, still delivers acceptable performance which is comparable to the BSBL and better than the BM-MAP-OMP and the CluSS-MCMC. This is not surprising since the main ideas behind the MRL1 and the PC-SBL are quite similar. Next, we examine the ability of respective algorithms to identify the true support of sparse signals. Fig. 4 depicts the pattern recovery success rates as a function of the ratio m/n and the sparsity level K respectively. It can be observed that the behavior of respective algorithms in Fig. 4 resembles that in Fig. 3, where each algorithm achieves a pattern recovery success rate that is close to the previously obtained success rate. This is not surprising since if a trial is deemed successful according to the NMSE criterion, then with a high probability it will also meet the successful pattern recovery condition.

We now consider the noisy case where the measurements are contaminated by additive noise. The observation noise is assumed multivariate Gaussian with zero mean and covariance matrix $\sigma^2\boldsymbol{I}$. Also, in our simulations, the noise variance is assumed unknown (except for the BM-MAP-OMP). For the noisy case, our algorithm uses (27) to update the hyperparameters in order to remove the small coefficients and yield a sparser solution, where τ in (27) is set to 10^3. Nevertheless, experimental results suggest both update rules (26) and (27) render similar performance in terms of NMSE. The NMSEs of respective al-

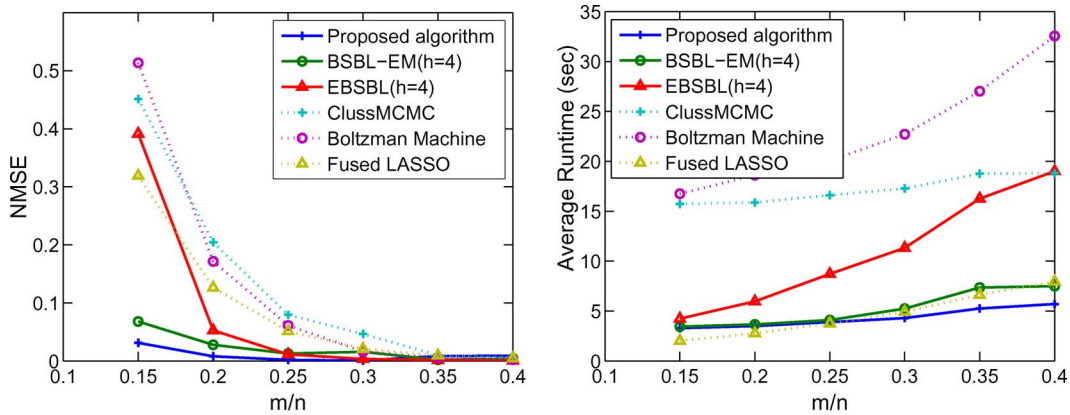


Fig. 7. Normalized MSEs and average run times of respective algorithms. (a) Normalized MSEs vs. m/n. (b) Average running times vs. m/n.

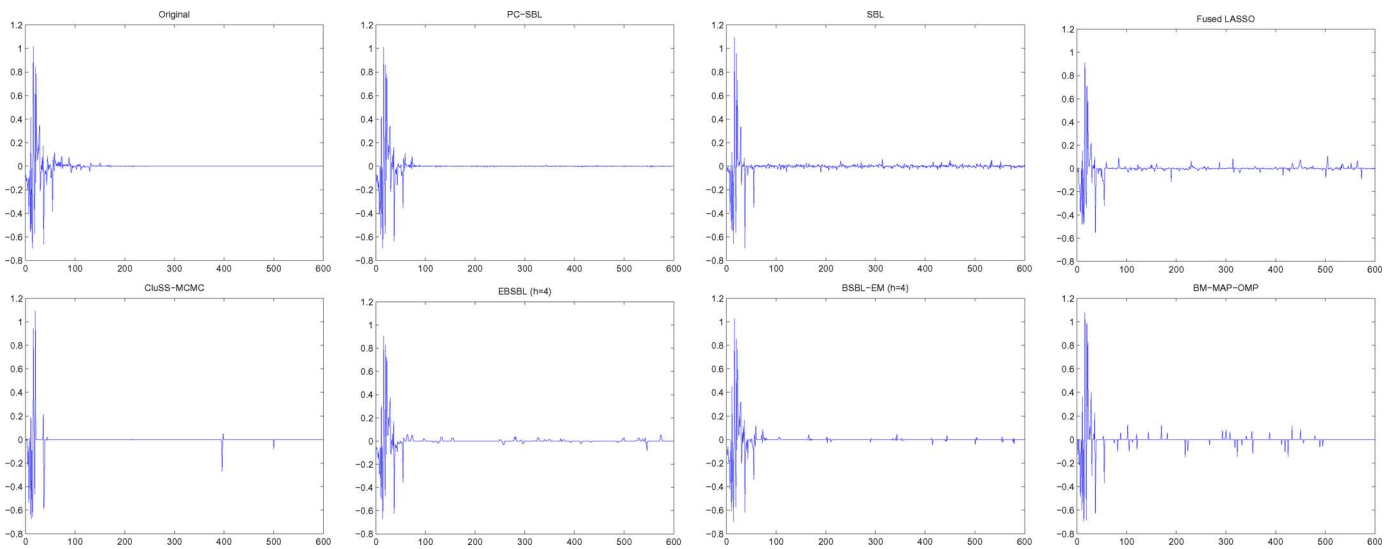


Fig. 8. The true DCT coefficients of a segment of audio and the coefficients reconstructed by respective algorithms.

gorithms as a function of the ratio m/n and the sparsity level K are plotted in Fig. 5, where the white Gaussian noise is added such that the signal-to-noise ratio (SNR), which is defined as $\mathrm{SNR(dB)} \triangleq 20\log_{10}(\|\boldsymbol{Ax}\|_2/\|\boldsymbol{w}\|_2)$[32], [33], is equal to 20 dB for each iteration. We see that our proposed algorithm yields a lower estimation error than other methods in the presence of additive Gaussian noise. Fig. 6 shows one realization of the randomly generated sparse signal and the signals reconstructed by respective algorithms. It can be observed that our proposed algorithm provides the most accurate estimate of the original signal, particularly those significant coefficients.

B. Audio Data

In this subsection, we carry out experiments on real world audio signals. Audio signals have cluster-sparse structures in certain basis, such as discrete cosine transform (DCT) basis. Hence audio signals are suitable data sets for evaluating the effectiveness of a variety of block-sparse signal recovery algorithms. We consider a clean piano signal[4] in our simulations. The audio signal is divided into a number of short-time segments, with each consisting of $n = 600$ data samples. Fig. 7 depicts the NMSEs and the average run times of respective algorithms as a function of the ratio m/n. Results are averaged over 300 independent trials. For each trial, we randomly select a short-time segment and compress the segmented signal using a randomly generated measurement matrix $\boldsymbol{Q} \in \mathbb{R}^{m\times 600}$. The sensing matrix $\boldsymbol{A}$ can therefore be expressed as $\boldsymbol{A} = \boldsymbol{Q\Psi}$, where $\boldsymbol{\Psi} \in \mathbb{R}^{n\times n}$ represents the DCT basis in which audio signals have sparse representations. The short-time segment is then reconstructed by respective algorithms. We see that our proposed algorithm offers the best performance among all algorithms and presents a significant performance advantage over other algorithms (except the B-SBL) for a small ratio m/n, where data acquisition is more beneficial due to high compression rates. Meanwhile, it can be observed that our algorithm is one of the most computationally efficient algorithms which consumes similar run times as the B-SBL and the fused-LASSO methods. The reconstruction accuracy of respective algorithms can also be observed from the reconstructed DCT coefficients. We provide the true DCT coefficients of one randomly selected segment of audio, and the coefficients reconstructed by respective

[4]Available at http://homepage.univie.ac.at/monika.doerfler/StrucAudio.html.

algorithms. Results are depicted in Fig. 8. It can be seen that our proposed algorithm provides reconstructed coefficients that are closest to the groundtruth.

VII. Conclusions

We developed a new Bayesian method for recovery of block-sparse signals whose block-sparse structures are entirely unknown. A pattern-coupled hierarchical Gaussian prior model was introduced to characterize both the sparseness of the coefficients and the statistical dependencies between neighboring coefficients of the signal. The prior model, similar to the conventional sparse Bayesian learning model, employs a set of hyperparameters to control the sparsity of the signal coefficients. Nevertheless, in our framework, the sparsity of each coefficient not only depends on its corresponding hyperparameter, but also depends on its neighbor hyperparameters. Such a prior has the potential to encourage clustered patterns and suppress isolated coefficients whose patterns are different from their respective neighbors. The hyperparameters, along with the sparse signal, can be estimated by maximizing their posterior probability, where an iterative algorithm was developed by exploiting the expectation-maximization (EM) formulation. Numerical results show that our proposed algorithm achieves a significant performance improvement as compared with the conventional sparse Bayesian learning method through exploiting the underlying block-sparse structure, even without requiring the exact location and size of each block. It also demonstrates superiority over other existing methods and provides state-of-the-art performance for block-sparse signal recovery.

References

[1] 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, 1998.

[2] E. Candés and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, no. 12, pp. 4203–4215, Dec. 2005.

[3] D. L. Donoho, "Compressive sensing," *IEEE Trans. Inf. Theory*, vol. 52, pp. 1289–1306, 2006.

[4] 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.

[5] M. J. Wainwright, "Information-theoretic limits on sparsity recovery in the high-dimensional and noisy setting," *IEEE Trans. Inf. Theory*, vol. 55, no. 12, pp. 5728–5741, Dec. 2009.

[6] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Trans. Inf. Theory*, no. 5, pp. 2230–2249, May 2009.

[7] M. Mishali and Y. C. Eldar, "Blind multi-band signal reconstruction: compressed sensing for analog signals," *IEEE Trans. Signal Process.*, vol. 57, no. 3, pp. 993–1009, Mar. 2009.

[8] R. Gribonval and E. Bacry, "Harmonic decomposition of audio signals with matching pursuit," *IEEE Trans. Signal Process.*, vol. 51, no. 1, pp. 101–111, Jan. 2003.

[9] R. Tibshirani, M. Saunders, S. Robert, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused LASSO," *J. R. Statist. Soc. B*, pp. 91–108, 2005.

[10] J. Lv, L. Huang, Y. Shi, and X. Fu, "Inverse synthetic aperture radar imaging via modified smooth L_0 norm," *IEEE Antennas Wireless Propog. Lett.*, vol. 13, pp. 1235–1238, 2014.

[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] 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.

[13] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. R. Statist. Soc. B*, vol. 68, pp. 49–67, 2006.

[14] J. Huang, T. Zhang, and D. Metaxas, "Learning with structured sparsity," *J. Mach. Learn. Res.*, vol. 12, pp. 3371–3412, 2011.

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

[16] 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.

[17] 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.

[18] 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.

[19] L. Yu, H. Sun, J. P. Barbot, and G. Zheng, "Bayesian compressive sensing for cluster structured sparse signals," *Signal Process.*, vol. 92, pp. 259–269, 2012.

[20] 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.

[21] A. Drámeau, 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.

[22] J. Ziniel and P. Schniter, "Dynamic compressive sensing of time-varying signals via approximate message passing," *IEEE Trans. Signal Process.*, no. 21, pp. 5270–5284, Nov. 2013.

[23] 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.

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

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

[26] D. P. Wipf, "Bayesian methods for finding sparse representations," Ph.D. dissertation, Univ. of California, San Diego, CA, USA, 2006.

[27] J. P. Vila and P. Schniter, "Expectation-maximization Gaussian-mixture approximate message passing," *IEEE Trans. Signal Process.*, no. 19, pp. 4658–4672, Oct. 2013.

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

[29] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2007.

[30] 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.

[31] E. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted l_1 minimization," *J. Fourier Anal. Appl.*, vol. 14, pp. 877–905, Dec. 2008.

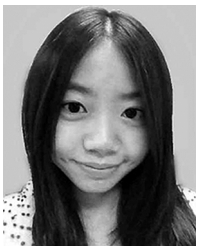
[32] L. Huang, H. C. So, and C. Qian, "Volumn-based method for spectrum sensing," *Digit. Signal Process.*, pp. 48–56, May 2014.

[33] C. Qian, L. Huang, W. J. Zeng, and H. C. So, "Direction-of-arrival estimation for coherent signals without knowledge of source number," *IEEE Sensor J.*, no. 9, pp. 3267–3273, Sep. 2014.

Jun Fang (M'08) received the B.S. and M.S. degrees from the 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.

During 2006, he was a postdoctoral research associate in the Department of Electrical and Computer Engineering, Duke University. From January 2007 to December 2010, he was a research associate with the Department of Electrical and Computer Engineering, Stevens Institute of Technology. Since January 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, and the Outstanding Paper Award from the IEEE AFICON Conference in 2011. He is an Associate Technical Editor for the *IEEE Communications Magazine*, and an Associate Editor for the IEEE Signal Processing Letters.

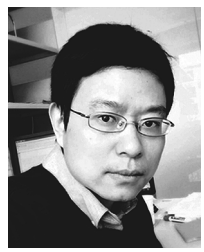


Yanning Shen received her B.Sc. and M.Sc degrees in Electrical Engineering from the University of Electronic Science and Technology of China, Chengdu, in 2011 and 2014, respectively. Since August 2014, she has been working towards her Ph.D. degree with the Department of Electrical and Computer Engineering, University of Minnesota. Her research interests include sparse signal processing and machine 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 July 1996 to May 1999, he was a Research Assistant in the Department of Electrical and Computer Engineering at the University of Florida. Since July 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 at 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 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 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 for *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), as well as a Guest Editor for 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 7th IEEE Sensor Array and Multichannel Signal Processing (SAM) Workshop, Hoboken, NJ, June 17–20, 2012. Dr. Li is a member of Tau Beta Pi and Phi Kappa Phi.



Pu Wang (S'05–M'12) received the B.Eng. and M.Eng. degrees from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2003 and 2006, respectively, and the Ph.D. degree from the Stevens Institute of Technology, Hoboken, NJ, USA, in 2011, all in electrical engineering.

He was an intern at the Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, in summer 2010, and a Research Assistant Professor at the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, from May 2011 to September 2012. Since October 2012, he has been with the Schlumberger-Doll Research, Cambridge, MA, as a research scientist. His current research interests include statistical signal processing, acoustics and ultrasonics in geophysical prospecting, Bayesian inference, time-frequency analysis, and sparse signal recovery. He currently holds four U.S. patents (applications).

Dr. Wang is the recipient of the IEEE Jack Neubauer Memorial Award in 2013 for the best systems paper published in the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and the Outstanding Paper Award from the IEEE AFRICON Conference in 2011. He received the Outstanding Doctoral Thesis in EE Award in 2011, the Edward Peskin Award in 2011, the Francis T. Boesch Award in 2008, and the Outstanding Research Assistant Award in 2007 from the Stevens Institute of Technology, the Excellent Master Thesis in Sichuan Province (of China) Award in 2007, and the Excellent Master Thesis Award from UESTC in 2006. He was an organization committee member of the Seventh IEEE Sensor Array and Multichannel Signal Processing (SAM) Workshop, Hoboken, New Jersey, June 2012, and the Fifth IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), Saint Martin, December 2013.