

# A Max-Product EM Algorithm for Reconstructing Markov-Tree Sparse Signals From Compressive Samples

Zhao Song and Aleksandar Dogandžić, *Senior Member, IEEE*

**Abstract**—We propose a Bayesian expectation-maximization (EM) algorithm for reconstructing Markov-tree sparse signals via belief propagation. The measurements follow an underdetermined linear model where the regression-coefficient vector is the sum of an unknown approximately sparse signal and a zero-mean white Gaussian noise with an unknown variance. The signal is composed of large- and small-magnitude components identified by binary state variables whose probabilistic dependence structure is described by a Markov tree. Gaussian priors are assigned to the signal coefficients given their state variables and the Jeffreys' noninformative prior is assigned to the noise variance. Our signal reconstruction scheme is based on an EM iteration that aims at maximizing the posterior distribution of the signal and its state variables given the noise variance. We construct the missing data for the EM iteration so that the complete-data posterior distribution corresponds to a hidden Markov tree (HMT) probabilistic graphical model that contains no loops and implement its maximization (M) step via a max-product algorithm. This EM algorithm estimates the vector of state variables *as well as* solves iteratively a linear system of equations to obtain the corresponding signal estimate. We select the noise variance so that the corresponding estimated signal and state variables obtained upon convergence of the EM iteration have the largest marginal posterior distribution. We compare the proposed and existing state-of-the-art reconstruction methods via signal and image reconstruction experiments.

**Index Terms**—Belief propagation, compressed sensing, expectation-maximization algorithms, hidden Markov models, signal reconstruction.

## I. INTRODUCTION

THE advent of compressive sampling (compressed sensing) in the past few years has sparked research activity in sparse signal reconstruction, whose main goal is to estimate the *sparsest*  $p \times 1$  signal coefficient vector  $\mathbf{s}$  from the  $N \times 1$  measurement vector  $\mathbf{y}$  satisfying the following underdetermined system of linear equations:

$$\mathbf{y} = H\mathbf{s} \quad (1)$$

where  $H$  is an  $N \times p$  sensing matrix and  $N \leq p$ .

Manuscript received October 09, 2012; revised April 01, 2013 and July 24, 2013; accepted July 29, 2013. Date of publication August 15, 2013; date of current version October 31, 2013. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Antonio De Maio. A portion of this work was presented at the SPIE Optics+Photonics Symposium, San Diego, CA, USA, August 2012.

The authors are with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011 USA (e-mail: zhaosong@iastate.edu; ald@iastate.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/TSP.2013.2277833

A tree dependency structure is exhibited by the wavelet coefficients of many natural images [1]–[7] (see also [3, Fig. 2]) as well as one-dimensional signals [1], [7], [8]. A probabilistic Markov tree structure has been introduced in [1] to model the statistical dependency between the state variables of wavelet coefficients. An approximate belief propagation algorithm has been first applied to compressive sampling by Baron, Sarvotham, and Baraniuk in [9], which employs sparse Rademacher sensing matrices for Bayesian signal reconstruction. Donoho, Maleki, and Montanari [10] simplified the sum-product algorithm by approximating messages with using a Gaussian distribution specified by two scalar parameters, leading to their approximate message passing approximate message passing (AMP) algorithm. Following the AMP framework, Schniter [11] proposed a turbo-AMP structured sparse signal recovery method based on loopy belief propagation and turbo equalization and applied it to reconstruct one-dimensional signals; [6] applied the turbo-AMP approach to reconstruct compressible images. A generalized approximate message passing (GAMP) algorithm that generalizes the AMP algorithm to arbitrary input and output channels and incorporates both max-sum and sum-product loopy belief propagation separately is proposed in [12]. However, the above references do not employ the exact form of the messages and also have the following limitations: [9] relies on sparsity of the sensing matrix, the methods in [9], [10], [12] apply to unstructured signals only, and the turbo-AMP approach in [6] and [11] needs sensing matrices to have approximately independent identically distributed (i.i.d.) elements, see [6, Section. III-C]. Indeed, turbo-AMP is sensitive to the presence of correlations among the elements of the sampling matrix and performs poorly if these correlations are sufficiently high and if norms of the columns or rows of the sampling matrix are sufficiently variable.

In [4] and [5], Marcov chain Monte Carlo (MCMC) and variational Bayesian (VB) schemes are used to reconstruct images that follow probabilistic Markov tree structure from linear measurements; however, [4] and [5] did not report large-scale examples: these schemes are computationally demanding and do not scale with increasing dimensionality of the reconstruction problem.

In this paper, we combine the hierarchical measurement model in [13] with a Markov tree prior on the binary state variables that identify the large- and small-magnitude signal coefficients and develop a Bayesian maximum *a posteriori* (MAP) expectation-maximization (EM) signal reconstruction scheme that aims at maximizing the posterior distribution of the signal and its state variables given the noise variance, where the maximization (M) step employs a max-product belief

propagation algorithm. Unlike the turbo-AMP scheme in [6] and [11], our reconstruction scheme *does not* require sensing matrices to have approximately i.i.d. elements and can handle correlations among these elements. Unlike the previous work, we *do not* approximate the message form in our belief propagation scheme. Indeed, the M step of our EM algorithm is exact because the expected complete-data posterior distribution that we maximize in the M step corresponds to the hidden Markov tree (HMT) graphical model that contains no loops. In [14], we proposed a similar EM algorithm for a random signal model [15] with a purely sparse vector of signal coefficients and a noninformative prior on this component given the binary state variables. We apply a grid search to select the noise variance so that the estimated signal and state variables have the largest marginal posterior distribution.

In Section II, we introduce our measurement and prior models. We assume that the Markov tree prior distribution is *known*. To reduce the number of tuning parameters for the tree prior, we further assume that these parameters do not change between Markov tree levels. This is in contrast to other approaches (e.g., [4]–[6]), which learn the Markov tree parameters from the measurements and allow their variation across the tree levels, see also the discussions in Sections V-B-2 and VI. Section III describes the proposed EM algorithm and establishes its properties; the implementation of the M step via the max-product algorithm is presented in Section III-A. The selection of the noise variance parameter is discussed in Section IV. Numerical simulations in Section V compare reconstruction performances of the proposed and existing methods.

We introduce the notation:  $I_n$  and  $\mathbf{0}_{n \times 1}$  denote the identity matrix of size  $n$  and the  $n \times 1$  vector of zeros, respectively; “ $T$ ”,  $\det(\cdot)$ , and  $\|\cdot\|_p$  are the transpose, determinant, and  $\ell_p$  norm, respectively;  $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \Sigma)$  denotes the probability density function (pdf) of a multivariate Gaussian random vector  $\mathbf{x}$  with mean  $\boldsymbol{\mu}$  and covariance matrix  $\Sigma$ ;  $\text{Inv}-\chi^2(\sigma^2|\nu, \sigma_0^2)$  denotes the pdf of a scaled inverse chi-square distribution with  $\nu$  degrees of freedom and a scale parameter  $\sigma_0^2$ , see [17, App. A];  $\mathbb{D}(p(\mathbf{x})||q(\mathbf{x}))$  denotes the Kullback-Leibler (KL) divergence from pdf  $p(\mathbf{x})$  to pdf  $q(\mathbf{x})$  [17, Sec. 2.8.2], [18, Sec. 8.5];  $|\mathcal{T}|$  is the cardinality of the set  $\mathcal{T}$ ;  $v(\cdot)$  is an invertible operator that transforms the two-dimensional matrix element indices into one-dimensional vector element indices. Finally,  $\rho_H$  denotes the largest singular value of a matrix  $H$ , also known as the spectral norm of  $H$ , and “ $\odot$ ” denotes the Hadamard (elementwise) product.

## II. MEASUREMENT AND PRIOR MODELS

We model an  $N \times 1$  real-valued measurement vector  $\mathbf{y}$  using the standard additive white Gaussian noise measurement model with the likelihood function given by the following pdf [3], [6]:

$$p_{\mathbf{y}|\mathbf{s}, \sigma^2}(\mathbf{y}|\mathbf{s}, \sigma^2) = \mathcal{N}(\mathbf{y}|H\mathbf{s}, \sigma^2 I_N) \quad (2)$$

where  $H$  is an  $N \times p$  real-valued sensing matrix with  $\text{rank}(H) = N$  satisfying the spectral norm condition

$$\rho_H = 1 \quad (3)$$

$\mathbf{s} = [s_1, s_2, \dots, s_p]^T$  is an unknown  $p \times 1$  real-valued signal coefficient vector, and  $\sigma^2$  is the unknown noise variance. We assume (3) without loss of generality because it is easily satisfied by appropriate scaling of the sensing matrix, measurements, and

noise variance,<sup>1</sup> provided that the spectral norm of the sensing matrix is easy to determine, see also footnote 2 for comments on the case where the spectral norm of the sensing matrix cannot be easily determined or estimated.

We adopt the Jeffreys' noninformative prior for the variance component  $\sigma^2$ :

$$p_{\sigma^2}(\sigma^2) \propto (\sigma^2)^{-1}. \quad (4)$$

Define the vector of binary state variables  $\mathbf{q} = [q_1, q_2, \dots, q_p]^T \in \{0, 1\}^p$  that determine if the magnitudes of the signal components  $s_i$ ,  $i = 1, 2, \dots, p$  are small ( $q_i = 0$ ) or large ( $q_i = 1$ ). Assume that  $s_i$  are conditionally independent given  $q_i$  and assign the following prior pdf to the signal coefficients:

$$p_{\mathbf{s}|\mathbf{q}, \sigma^2}(\mathbf{s}|\mathbf{q}, \sigma^2) = \mathcal{N}(\mathbf{s}|\mathbf{0}_{p \times 1}, \sigma^2 D(\mathbf{q})) \quad (5a)$$

where

$$D(\mathbf{q}) = \text{diag} \left\{ (\gamma^2)^{q_1} (\epsilon^2)^{1-q_1}, (\gamma^2)^{q_2} (\epsilon^2)^{1-q_2}, \dots, (\gamma^2)^{q_p} (\epsilon^2)^{1-q_p} \right\}. \quad (5b)$$

where  $\gamma^2$  and  $\epsilon^2$  are known positive constants and, typically,  $\gamma^2 \gg \epsilon^2$ . Hence, the large- and small-magnitude signal coefficients  $s_i$  corresponding to  $q_i = 1$  and  $q_i = 0$  are modeled as zero-mean Gaussian random variables with variances  $\gamma^2 \sigma^2$  and  $\epsilon^2 \sigma^2$ , respectively. Consequently,  $\gamma^2$  and  $\epsilon^2$  are relative variances (to the noise variance  $\sigma^2$ ) of the large- and small-magnitude signal coefficients. Equivalently,

We now introduce the Markov tree prior probability mass function (pmf) on the state variables  $q_i$  [1], [6]. To make this probability model easier to understand, we focus on the image reconstruction scenario where the elements of  $\mathbf{s}$  are the two-dimensional discrete wavelet transform (DWT) coefficients of the underlying image that we wish to reconstruct. Hence, we introduce two-dimensional signal element indices  $(i_1, i_2)$ . Recall that the conversion operator  $v(\cdot)$  is invertible; hence, there is a one-to-one correspondence between the corresponding one- and two-dimensional signal element indices. A parent wavelet coefficient with a two-dimensional position index  $(i_1, i_2)$  has four children in the finer wavelet decomposition level with two-dimensional indices  $(2i_1 - 1, 2i_2 - 1), (2i_1 - 1, 2i_2), (2i_1, 2i_2 - 1)$  and  $(2i_1, 2i_2)$ , see Fig. 1. The parent-child dependency assumption implies that, if a parent coefficient in a certain wavelet decomposition level has small (large) magnitude, then its children coefficients in the next finer wavelet decomposition level tend to have small (large) magnitude as well. Denote by  $\rho$  and  $\kappa$  the numbers of rows and columns of the image, and by  $L$  the number of wavelet decomposition levels (tree depth).

We set the prior pmf  $p_{\mathbf{q}}(\mathbf{q})$  as follows. In the first wavelet decomposition level ( $\ell = 1$ ), assign

$$p_{q_i}(1) = \Pr\{q_i = 1\} = \begin{cases} 1, & i \in \mathcal{A} \\ P_{\text{root}}, & i \in \mathcal{T}_{\text{root}} \end{cases} \quad (6a)$$

where

$$\mathcal{A} = v \left( \left\{ 1, 2, \dots, \frac{\rho}{2^L} \right\} \times \left\{ 1, 2, \dots, \frac{\kappa}{2^L} \right\} \right) \quad (6b)$$

<sup>1</sup>For a generic sensing matrix  $H'$  with  $\rho_{H'} \neq 1$ , data vector  $\mathbf{y}'$  and noise variance  $(\sigma^2)'$ , this scaling is performed as follows:  $H = H'/\rho_{H'}$ ,  $\mathbf{y} = \mathbf{y}'/\rho_{H'}$ , and  $\sigma^2 = (\sigma^2)'/\rho_{H'}^2$ , which guarantees that the new sensing matrix  $H$  satisfies (3).

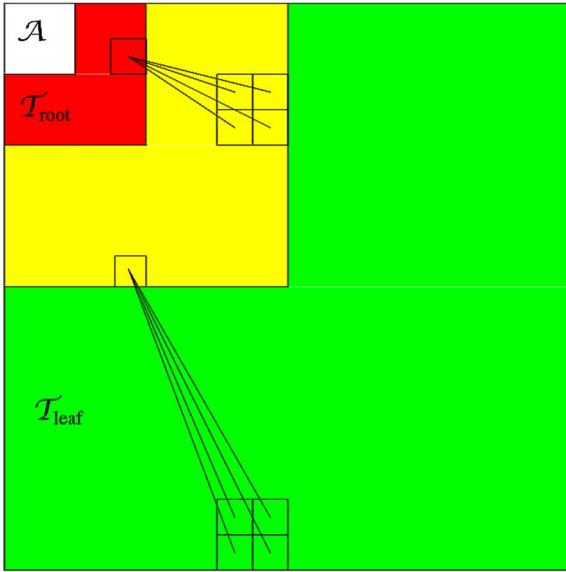


Fig. 1. Types of wavelet decomposition coefficients: approximation, root, and leaf, whose sets are denoted by  $\mathcal{A}$ ,  $\mathcal{T}_{\text{root}}$ , and  $\mathcal{T}_{\text{leaf}}$ , respectively.

$$\begin{aligned} \mathcal{T}_{\text{root}} = v &\left( \left\{ 1, 2, \dots, \frac{\rho}{2^{L-1}} \right\} \right. \\ &\times \left. \left\{ 1, 2, \dots, \frac{\kappa}{2^{L-1}} \right\} \right) \setminus \mathcal{A} \end{aligned} \quad (6c)$$

are the sets of indices of the approximation and root node coefficients and  $P_{\text{root}} \in (0, 1)$  is a known constant denoting the prior probability that a root node signal coefficient has large magnitude, see Fig. 1. In the levels  $\ell = 2, 3, \dots, L$ , assign

$$p_{q_i|q_{\pi(i)}}(1|q_{\pi(i)}) = \begin{cases} P_H, & q_{\pi(i)} = 1 \\ P_L, & q_{\pi(i)} = 0 \end{cases} \quad (6d)$$

where  $\pi(i)$  denotes the index of the parent of node  $i$ . Here,  $P_H \in (0, 1)$  and  $P_L \in (0, 1)$  are known constants denoting the probabilities that the signal coefficient  $s_i$  is large if the corresponding parent signal coefficient is large or small, respectively.

The expected number of large-magnitude signal coefficients is

$$\mathbb{E} \left[ \sum_{i=1}^p q_i \right] = \frac{p}{4^L} \left( 1 + 3 \sum_{\ell=0}^{L-1} 4^\ell P_\ell \right) \quad (7a)$$

where  $P_\ell$  is the marginal probability that a state variable in the  $\ell$ th tree level is equal to one, computed recursively as follows:

$$P_\ell = P_{\ell-1} P_H + (1 - P_{\ell-1}) P_L \quad (7b)$$

initialized by  $P_0 = P_{\text{root}}$ .

Our wavelet tree structure consists of  $|\mathcal{T}_{\text{root}}|$  trees and spans all signal wavelet coefficients except the approximation coefficients; hence, the set of indices of the wavelet coefficients within the trees is

$$\mathcal{T} = v(\{1, 2, \dots, \rho\} \times \{1, 2, \dots, \kappa\}) \setminus \mathcal{A}. \quad (8a)$$

Define also the set of leaf variable node indices within the tree structure as

$$\begin{aligned} \mathcal{T}_{\text{leaf}} = v &\left( [\{1, 2, \dots, \rho\} \times \{1, 2, \dots, \kappa\}] \setminus \left[ \left\{ 1, 2, \dots, \frac{\rho}{2} \right\} \right. \right. \\ &\times \left. \left. \left\{ 1, 2, \dots, \frac{\kappa}{2} \right\} \right] \right) \end{aligned} \quad (8b)$$

see Fig. 1. We have 5 tuning parameters  $P_{\text{root}}$ ,  $P_H$ ,  $P_L$ ,  $\gamma^2$ , and  $\epsilon^2$ , each with a clear meaning. A fairly crude choice of these parameters is sufficient for achieving good reconstruction performance, see Section V.

The logarithm of the prior pmf  $p_{\mathbf{q}}(\mathbf{q})$  is

$$\begin{aligned} \ln p_{\mathbf{q}}(\mathbf{q}) = \text{const} &+ \left[ \sum_{i \in \mathcal{A}} \ln \mathbb{1}(q_i = 1) \right] \\ &+ \left[ \sum_{i \in \mathcal{T}_{\text{root}}} q_i \ln P_{\text{root}} + (1 - q_i) \ln(1 - P_{\text{root}}) \right] \\ &+ \left[ \sum_{i \in \mathcal{T} \setminus \mathcal{T}_{\text{root}}} q_i q_{\pi(i)} \ln P_H \right. \\ &\quad \left. + (1 - q_i) q_{\pi(i)} \ln(1 - P_H) \right. \\ &\quad \left. + q_i (1 - q_{\pi(i)}) \ln P_L \right. \\ &\quad \left. + (1 - q_i) (1 - q_{\pi(i)}) \ln(1 - P_L) \right] \end{aligned} \quad (9)$$

where const denotes the terms that are not functions of  $\mathbf{q}$ .

#### A. Bayesian Inference

Define the vectors of state variables and signal coefficients

$$\boldsymbol{\theta} = [\boldsymbol{\theta}_1^T \quad \boldsymbol{\theta}_2^T \quad \cdots \quad \boldsymbol{\theta}_p^T]^T, \quad \boldsymbol{\theta}_i = [q_i, s_i]^T. \quad (10)$$

The joint posterior distribution of  $\boldsymbol{\theta}$  and  $\sigma^2$  is

$$\begin{aligned} p_{\boldsymbol{\theta}, \sigma^2 | \mathbf{y}}(\boldsymbol{\theta}, \sigma^2 | \mathbf{y}) &\propto p_{\mathbf{y} | \mathbf{s}, \sigma^2}(\mathbf{y} | \mathbf{s}, \sigma^2) p_{\mathbf{s} | \mathbf{q}, \sigma^2}(\mathbf{s} | \mathbf{q}, \sigma^2) p_{\mathbf{q}}(\mathbf{q}) p_{\sigma^2}(\sigma^2) \\ &\propto (\sigma^2)^{-(p+N+2)/2} \left( \frac{\epsilon^2}{\gamma^2} \right)^{0.5 \sum_{i=1}^p q_i} p_{\mathbf{q}}(\mathbf{q}) \\ &\cdot \exp \left[ -0.5 \frac{\|\mathbf{y} - H\mathbf{s}\|_2^2}{\sigma^2} - 0.5 \frac{\mathbf{s}^T D^{-1}(\mathbf{q}) \mathbf{s}}{\sigma^2} \right] \end{aligned} \quad (11)$$

which implies

$$\begin{aligned} p_{\sigma^2 | \boldsymbol{\theta}, \mathbf{y}}(\sigma^2 | \boldsymbol{\theta}, \mathbf{y}) &= \text{Inv} - \chi^2 \left( \sigma^2 | p + N, \frac{\|\mathbf{y} - H\mathbf{s}\|_2^2 + \mathbf{s}^T D^{-1}(\mathbf{q}) \mathbf{s}}{p + N} \right) \\ & \end{aligned} \quad (12a)$$

$$\begin{aligned} p_{\boldsymbol{\theta} | \sigma^2, \mathbf{y}}(\boldsymbol{\theta} | \sigma^2, \mathbf{y}) &\propto \exp \left[ -0.5 \frac{\|\mathbf{y} - H\mathbf{s}\|_2^2 + \mathbf{s}^T D^{-1}(\mathbf{q}) \mathbf{s}}{\sigma^2} \right] \\ &\cdot \left( \frac{\epsilon^2}{\gamma^2} \right)^{0.5 \sum_{i=1}^p q_i} p_{\mathbf{q}}(\mathbf{q}). \end{aligned} \quad (12b)$$

We integrate the noise variance parameter from the joint posterior distribution as follows (see also [16, (5.5) on p. 126]):

$$p_{\theta|\mathbf{y}}(\theta|\mathbf{y}) = \frac{p_{\theta,\sigma^2|\mathbf{y}}(\theta,\sigma^2|\mathbf{y})}{p_{\sigma^2|\theta,\mathbf{y}}(\sigma^2|\theta,\mathbf{y})} \propto \frac{p_{\mathbf{q}}(\mathbf{q}) \left( \frac{\epsilon^2}{\gamma^2} \right)^{0.5 \sum_{i=1}^p q_i}}{\left[ \frac{\|\mathbf{y} - H\mathbf{s}\|_2^2 + \mathbf{s}^T D^{-1}(\mathbf{q})\mathbf{s}}{p+N} \right]^{\frac{p+N}{2}}}. \quad (13a)$$

For a fixed  $\mathbf{q}$ , (13a) is maximized with respect to  $\mathbf{s}$  at

$$\bar{\mathbf{s}}(\mathbf{q}) = D(\mathbf{q})H^T [I_N + HD(\mathbf{q})H^T]^{-1} \mathbf{y} \quad (13b)$$

which is the Bayesian linear-model minimum mean-square error (MMSE) estimator of  $\mathbf{s}$  for a given  $\mathbf{q}$  [19, Theorem 11.1]. As  $\epsilon^2$  decreases to zero,  $\bar{\mathbf{s}}(\mathbf{q})$  becomes more sparse (becoming exactly sparse for  $\epsilon^2 = 0$ ); as  $\epsilon^2$  increases,  $\bar{\mathbf{s}}(\mathbf{q})$  becomes less sparse.

Substituting (13b) into (13a) yields the *concentrated (profile) marginal posterior distribution*:

$$\max_{\mathbf{s}} p_{\theta|\mathbf{y}}(\theta|\mathbf{y}) \propto \frac{p_{\mathbf{q}}(\mathbf{q}) \left( \frac{\epsilon^2}{\gamma^2} \right)^{0.5 \sum_{i=1}^p q_i}}{\left\{ \frac{\mathbf{y}^T [I_N + HD(\mathbf{q})H^T]^{-1} \mathbf{y}}{p+N} \right\}^{\frac{p+N}{2}}} \quad (13c)$$

which is a function of the state variables  $\mathbf{q}$  only.

We wish to maximize (13a) with respect to  $\theta$ , but cannot perform this task directly. Consequently, we adopt an indirect approach: We first develop an EM algorithm for maximizing  $p_{\theta|\sigma^2,\mathbf{y}}(\theta|\sigma^2, \mathbf{y})$  in (12b) for a given  $\sigma^2$  (Section III) and then apply a grid search scheme for selecting the best noise variance parameter  $\sigma^2$  so that the estimated signal and state variables have the largest marginal posterior distribution (13a) (Section IV).

### III. AN EM ALGORITHM FOR MAXIMIZING $p_{\theta|\sigma^2,\mathbf{y}}(\theta|\sigma^2, \mathbf{y})$

Motivated by [13, Sec. V.A], we introduce the following hierarchical two-stage model:

$$p_{\mathbf{y}|\mathbf{z},\sigma^2}(\mathbf{y}|\mathbf{z},\sigma^2) = \mathcal{N}(\mathbf{y}|H\mathbf{z}, \sigma^2(I_N - HH^T)) \quad (14a)$$

$$p_{\mathbf{z}|\mathbf{s},\sigma^2}(\mathbf{z}|\mathbf{s},\sigma^2) = \mathcal{N}(\mathbf{z}|\mathbf{s}, \sigma^2 I_p) \quad (14b)$$

where  $\mathbf{z}$  is a  $p \times 1$  vector of *missing data*. Observe that the spectral norm condition (3) guarantees that the covariance matrix  $\sigma^2(I_N - HH^T)$  in (14a) is positive semidefinite.

Our EM algorithm for maximizing  $p_{\theta|\sigma^2,\mathbf{y}}(\theta|\sigma^2, \mathbf{y})$  in (12b) consists of iterating between the following E and M steps (see Appendix A):<sup>2</sup>

$$\begin{aligned} \text{E step : } \mathbf{z}^{(j)} &\triangleq \mathbb{E}_{\mathbf{z}|\sigma^2, \mathbf{y}, \mathbf{s}} [\mathbf{z}|\sigma^2, \mathbf{y}, \mathbf{s}^{(j)}] \\ &= [z_1^{(j)}, z_2^{(j)}, \dots, z_p^{(j)}]^T \\ &= \mathbf{s}^{(j)} + H^T (\mathbf{y} - H\mathbf{s}^{(j)}) \end{aligned} \quad (15)$$

<sup>2</sup>If the spectral norm of the sensing matrix  $H$  cannot be easily determined or estimated [and, therefore, (3) cannot be ensured], we can introduce an adaptive positive step size that multiplies the second summand in the E step (15); we also need to divide the first summand in (16a) by this quantity. Then, the step size adaptation can be performed along the lines of [20], with goal to ensure monotonicity of the EM iteration. Such a step size adaptation (which, in effect, estimates the spectral norm of  $H$ ) is typically completed within the first few EM iterations.

$$\begin{aligned} \text{M step : } \theta^{(j+1)} &= \arg \max_{\theta} \left\{ \ln [p_{\mathbf{q}}(\mathbf{q})] + 0.5 \ln \left( \frac{\epsilon^2}{\gamma^2} \right) \sum_{i=1}^p q_i \right. \\ &\quad \left. - 0.5 \frac{\|\mathbf{z}^{(j)} - \mathbf{s}\|_2^2 + \mathbf{s}^T D^{-1}(\mathbf{q})\mathbf{s}}{\sigma^2} \right\} \end{aligned} \quad (16a)$$

$$= \arg \max_{\theta} \ln p_{\theta|\sigma^2, \mathbf{z}}(\theta|\sigma^2, \mathbf{z}^{(j)}) \quad (16b)$$

where  $j$  denotes the iteration index. See, e.g., [17, Sec. 11.4], [21], and [22] for a general exposition on the EM algorithm and its properties and [16, Chapter 12.3] for its Bayesian version. To simplify the notation, we omit the dependence of the iterates  $\theta^{(j)}$  on  $\sigma^2$  in this section. Denote by  $\theta^{(+\infty)}$ ,  $\mathbf{s}^{(+\infty)}$ , and  $\mathbf{q}^{(+\infty)}$  the estimates of  $\theta$ ,  $\mathbf{s}$ , and  $\mathbf{q}$  obtained upon convergence of the above EM iteration.

For any two consecutive iterations  $j$  and  $j+1$ , this EM algorithm ensures that the objective posterior function *does not decrease*, i.e.,

$$p_{\theta|\sigma^2, \mathbf{y}}(\theta^{(j+1)}|\sigma^2, \mathbf{y}) \geq p_{\theta|\sigma^2, \mathbf{y}}(\theta^{(j)}|\sigma^2, \mathbf{y}) \quad (17)$$

see Appendix A. Monotonic convergence is also a key general property of the EM-type algorithms [22].

*Theorem 1:* The signal and binary state variable estimates  $\mathbf{s}^{(+\infty)}$  and  $\mathbf{q}^{(+\infty)}$  obtained upon convergence of the EM iteration (15), (16) satisfy

$$\mathbf{s}^{(+\infty)} = \bar{\mathbf{s}}(\mathbf{q}^{(+\infty)}). \quad (18)$$

Hence, this iteration provides an estimate  $\mathbf{q}^{(+\infty)}$  of the vector of state variables  $\mathbf{q}$  *as well as* finds the solution (13b) of the underlying linear system to obtain the corresponding signal estimate.

*Proof:* See Appendix A.  $\square$

Consequently, as  $\epsilon^2$  decreases to zero,  $\mathbf{s}^{(+\infty)}$  becomes more sparse; as  $\epsilon^2$  increases,  $\mathbf{s}^{(+\infty)}$  becomes less sparse.

Note that the M step in (16b) is equivalent to maximizing  $p_{\theta|\sigma^2, \mathbf{z}}(\theta|\sigma^2, \mathbf{z})$  for the missing data vector  $\mathbf{z} = \mathbf{z}^{(j)}$ . In the following section, we describe efficient maximization of  $p_{\theta|\sigma^2, \mathbf{z}}(\theta|\sigma^2, \mathbf{z})$ .

#### A. M Step: Maximizing $p_{\theta|\sigma^2, \mathbf{z}}(\theta|\sigma^2, \mathbf{z})$

Before we proceed, define

$$\hat{s}_i(0) = \frac{\epsilon^2}{1+\epsilon^2} z_i, \quad \hat{s}_i(1) = \frac{\gamma^2}{1+\gamma^2} z_i \quad (19)$$

where we omit the dependence of  $\hat{s}_i(0)$  and  $\hat{s}_i(1)$  on  $z_i$  to simplify the notation.

Observe that

$$p_{\theta|\sigma^2, \mathbf{z}}(\theta|\sigma^2, \mathbf{z}) \propto p_{\theta_A|\sigma^2, \mathbf{z}}(\theta_A|\sigma^2, \mathbf{z}) p_{\theta_T|\sigma^2, \mathbf{z}}(\theta_T|\sigma^2, \mathbf{z}) \quad (20)$$

where  $\theta_A$  and  $\theta_T$  consist of  $\theta_i$ ,  $i \in \mathcal{A}$  and  $\theta_i$ ,  $i \in \mathcal{T}$ , respectively, and

$$p_{\theta_A|\sigma^2, \mathbf{z}}(\theta_A|\sigma^2, \mathbf{z}) \propto \prod_{i \in \mathcal{A}} \mathcal{N}(z_i|s_i, \sigma^2) \mathcal{N}(s_i|0, \gamma^2 \sigma^2) \mathbb{1}(q_i=1) \quad (21a)$$

$$p_{\theta_T|\sigma^2, \mathbf{z}}(\theta_T|\sigma^2, \mathbf{z}) \propto \left\{ \prod_{i \in \mathcal{T}} \mathcal{N}(z_i|s_i, \sigma^2) [\mathcal{N}(s_i|0, \gamma^2 \sigma^2)]^{q_i} \cdot [\mathcal{N}(s_i|0, \epsilon^2 \sigma^2)]^{1-q_i} \right\} p_{\mathbf{q}_T}(\mathbf{q}_T). \quad (21b)$$

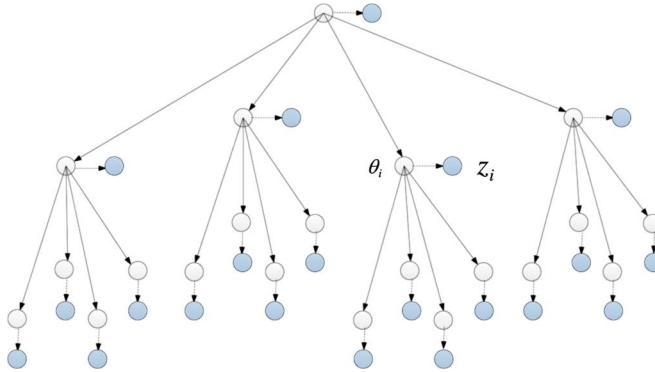


Fig. 2. A HMT, part of the probabilistic model (21b).

Here, (21a) follows from (6a) and (21b) corresponds to the HMT probabilistic model that contains no loops. Fig. 2 depicts an HMT that is a part of the probabilistic model (21b). Maximizing  $p_{\theta_A|\sigma^2, z}(\theta_A|\sigma^2, z^{(j)})$  in (21a) with respect to  $\theta_i$ ,  $i \in \mathcal{A}$  yields

$$\hat{\theta}_i = [1, \hat{s}_i(1)]^T, \quad i \in \mathcal{A} \quad (22)$$

where we have used the identity (B1a) in Appendix B.

We now apply the max-product belief propagation algorithm [23]–[25] to each tree in our wavelet tree structure, with the goal to find the mode of  $p_{\theta_T|\sigma^2, z}(\theta_T|\sigma^2, z)$ . We represent the HMT probabilistic model for  $p_{\theta_T|\sigma^2, z}(\theta_T|\sigma^2, z)$  via *potential functions* as [see (21b)]

$$p_{\theta_T|\sigma^2, z}(\theta_T|\sigma^2, z) \propto \left[ \prod_{i \in \mathcal{T} \setminus \mathcal{T}_{\text{root}}} \psi_i(\theta_i) \psi_{i, \pi(i)}(q_i, q_{\pi(i)}) \right] \cdot \left[ \prod_{i \in \mathcal{T}_{\text{root}}} \psi_i(\theta_i) \right] \quad (23)$$

where

$$\psi_i(\theta_i) = \mathcal{N}(z_i|s_i, \sigma^2) [\mathcal{N}(s_i|0, \gamma^2 \sigma^2)]^{q_i} [\mathcal{N}(s_i|0, \epsilon^2 \sigma^2)]^{1-q_i} \quad (24a)$$

for  $i \in \mathcal{T} \setminus \mathcal{T}_{\text{root}}$ ,

$$\psi_i(\theta_i) = \mathcal{N}(z_i|s_i, \sigma^2) [P_{\text{root}} \mathcal{N}(s_i|0, \gamma^2 \sigma^2)]^{q_i} \cdot [(1 - P_{\text{root}}) \mathcal{N}(s_i|0, \epsilon^2 \sigma^2)]^{1-q_i} \quad (24b)$$

for  $i \in \mathcal{T}_{\text{root}}$ , and

$$\psi_{i, \pi(i)}(q_i, q_{\pi(i)}) = [P_H^{q_i} (1 - P_H)^{1-q_i}]^{q_{\pi(i)}} \cdot [P_L^{q_i} (1 - P_L)^{1-q_i}]^{1-q_{\pi(i)}} \quad (24c)$$

for  $i \in \mathcal{T} \setminus \mathcal{T}_{\text{root}}$ .

Our algorithm for maximizing (23) consists of computing and passing upward and downward messages and calculating and maximizing beliefs.

1) *Computing and Passing Upward Messages:* We propagate the upward messages from the lowest decomposition level (i.e., the leaves) towards the root of the tree. Fig. 3(a) depicts the computation of the upward message from variable node  $\theta_i$  to its parent node  $\theta_{\pi(i)}$  wherein we also define a *child* of  $\theta_i$  as a variable node  $\theta_k$  with index  $k \in \text{ch}(i)$ , where  $\text{ch}(i)$  is the index set of the children of  $i$ : for  $i = v(i_1, i_2)$ ,  $\text{ch}(i) =$

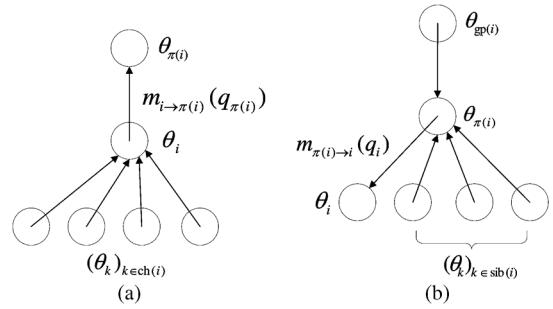


Fig. 3. Computing and passing (a) upward and (b) downward messages.

$\{v((2i_1 - 1, 2i_2 - 1), (2i_1 - 1, 2i_2), (2i_1, 2i_2 - 1), (2i_1, 2i_2))\}$ . Here, we use a circle and an edge with an arrow to denote a variable node and a message, respectively. The upward messages have the following general form [24]:

$$m_{i \rightarrow \pi(i)}(q_{\pi(i)}) = \alpha \max_{\theta_i} \left\{ \psi_i(\theta_i) \psi_{i, \pi(i)}(q_i, q_{\pi(i)}) \prod_{k \in \text{ch}(i)} m_{k \rightarrow i}(q_i) \right\} \quad (25)$$

where  $\alpha > 0$  denotes a normalizing constant used for computational stability [24]. For nodes with no children (corresponding to level  $L$ , i.e.,  $i \in \mathcal{T}_{\text{leaf}}$ ), we set the multiplicative term  $\prod_{k \in \text{ch}(i)} m_{k \rightarrow i}(\theta_i)$  in (25) to one.

In Appendix B-I, we show that the only two candidates for  $\theta_i$  in the maximization of (25) are  $[0, \hat{s}_i(0)]^T$  and  $[1, \hat{s}_i(1)]^T$ , see also (19).

Substituting these candidates into (25) and normalizing the messages yields (see Appendix B-I)

$$m_{i \rightarrow \pi(i)}(q_{\pi(i)}) = [\mu_i^u(0)]^{1-q_{\pi(i)}} [\mu_i^u(1)]^{q_{\pi(i)}} \quad (26a)$$

where  $[\mu_i^u(0), \mu_i^u(1)]^T = \boldsymbol{\mu}_i^u$ ,

$$\begin{aligned} \boldsymbol{\mu}_i^u &= \frac{[\max \{\boldsymbol{\nu}_{0,i}^u \odot \boldsymbol{\eta}_i^u\}, \max \{\boldsymbol{\nu}_{1,i}^u \odot \boldsymbol{\eta}_i^u\}]^T}{\max \{\boldsymbol{\nu}_{0,i}^u \odot \boldsymbol{\eta}_i^u\} + \max \{\boldsymbol{\nu}_{1,i}^u \odot \boldsymbol{\eta}_i^u\}} \\ &= \frac{[\exp (\ln (\max \{\boldsymbol{\nu}_{0,i}^u \odot \boldsymbol{\eta}_i^u\}) - \ln (\max \{\boldsymbol{\nu}_{1,i}^u \odot \boldsymbol{\eta}_i^u\}))], 1]^T}{1 + \exp (\ln (\max \{\boldsymbol{\nu}_{0,i}^u \odot \boldsymbol{\eta}_i^u\}) - \ln (\max \{\boldsymbol{\nu}_{1,i}^u \odot \boldsymbol{\eta}_i^u\}))} \end{aligned} \quad (26b)$$

$$\boldsymbol{\nu}_{0,i}^u = [1 - P_L, P_L]^T \odot \boldsymbol{\phi}(z_i) \quad (26c)$$

$$\boldsymbol{\nu}_{1,i}^u = [1 - P_H, P_H]^T \odot \boldsymbol{\phi}(z_i) \quad (26d)$$

$$\boldsymbol{\eta}_i^u = \begin{cases} \bigodot_{k \in \text{ch}(i)} \boldsymbol{\mu}_k^u, & i \in \mathcal{T} \setminus \mathcal{T}_{\text{leaf}} \\ [1, 1]^T, & i \in \mathcal{T}_{\text{leaf}} \end{cases} \quad (26e)$$

$$\boldsymbol{\phi}(z) = \left[ \frac{\exp (-0.5 \frac{z^2}{\sigma^2 + \sigma^2 \epsilon^2})}{\epsilon}, \frac{\exp (-0.5 \frac{z^2}{\sigma^2 + \sigma^2 \gamma^2})}{\gamma} \right]^T \quad (26f)$$

and  $\epsilon = \sqrt{\epsilon^2} > 0$  and  $\gamma = \sqrt{\gamma^2} > 0$ . A numerically stable implementation of (26b) that we employ is illustrated in the second expression in (26b). Similarly, the elementwise products in (26c)–(26e) are implemented as exponentiated sums of logarithms of the product terms.

2) *Computing and Passing Downward Messages:* Upon obtaining all the upward messages, we now compute the downward messages and propagate them from the root towards the

lowest level (i.e., the leaves). Fig. 3(b) depicts the computation of the downward message from the parent  $\boldsymbol{\theta}_{\pi(i)}$  to the variable node  $\boldsymbol{\theta}_i$ , which involves upward messages to  $\boldsymbol{\theta}_{\pi(i)}$  from its other children, i.e. the *siblings* of  $\boldsymbol{\theta}_i$ , marked as  $\boldsymbol{\theta}_k$ ,  $k \in \text{sib}(i)$ . This downward message also requires the message sent to  $\boldsymbol{\theta}_{\pi(i)}$  from its parent node, which is the *grandparent* of  $\boldsymbol{\theta}_i$ , denoted by  $\boldsymbol{\theta}_{\text{gp}(i)}$ . The downward messages have the general form [24]:

$$m_{\pi(i) \rightarrow i}(q_i) = \alpha \max_{\boldsymbol{\theta}_{\pi(i)}} \left\{ \psi_{\pi(i)}(\boldsymbol{\theta}_{\pi(i)}) \psi_{i, \pi(i)}(q_i, q_{\pi(i)}) \cdot m_{\text{gp}(i) \rightarrow \pi(i)}(q_{\pi(i)}) \prod_{k \in \text{sib}(i)} m_{k \rightarrow \pi(i)}(q_{\pi(i)}) \right\} \quad (27)$$

where  $\alpha > 0$  denotes a normalizing constant used for computational stability. For the variable nodes  $i$  in the second decomposition level that have no grandparents (i.e.,  $\pi(i) \in \mathcal{T}_{\text{root}}$ ), we set the multiplicative term  $m_{\text{gp}(i) \rightarrow \pi(i)}(q_{\pi(i)})$  in (27) to one.

In Appendix B-II, we show that the only two candidates for  $\boldsymbol{\theta}_{\pi(i)}$  in the maximization of (27) are  $[0, \hat{s}_{\pi(i)}(0)]^T$  and  $[1, \hat{s}_{\pi(i)}(1)]^T$ , see also (19). Substituting these candidates into (27) and normalizing the messages yields (see Appendix B-II)

$$m_{\pi(i) \rightarrow i}(q_i) = [\mu_i^d(0)]^{1-q_i} [\mu_i^d(1)]^{q_i} \quad (28a)$$

for  $\pi(i) \in \mathcal{T} \setminus \mathcal{T}_{\text{leaf}}$ , where  $[\mu_i^d(0), \mu_i^d(1)]^T = \boldsymbol{\mu}_i^d$  and

$$\begin{aligned} \boldsymbol{\mu}_i^d &= \frac{[\max\{\boldsymbol{\nu}_{0,i}^d \odot \boldsymbol{\eta}_i^d\}, \max\{\boldsymbol{\nu}_{1,i}^d \odot \boldsymbol{\eta}_i^d\}]^T}{\max\{\boldsymbol{\nu}_{0,i}^d \odot \boldsymbol{\eta}_i^d\} + \max\{\boldsymbol{\nu}_{1,i}^d \odot \boldsymbol{\eta}_i^d\}} \\ &= \frac{[\exp(\ln(\max\{\boldsymbol{\nu}_{0,i}^d \odot \boldsymbol{\eta}_i^d\}) - (\ln \max\{\boldsymbol{\nu}_{1,i}^d \odot \boldsymbol{\eta}_i^d\}))], 1]^T}{1 + \exp(\ln(\max\{\boldsymbol{\nu}_{0,i}^d \odot \boldsymbol{\eta}_i^d\}) - \ln(\max\{\boldsymbol{\nu}_{1,i}^d \odot \boldsymbol{\eta}_i^d\}))} \end{aligned} \quad (28b)$$

$$\boldsymbol{\nu}_{0,i}^d = [1 - P_L, 1 - P_H]^T \odot \boldsymbol{\phi}(z_{\pi(i)}) \odot \left[ \bigodot_{k \in \text{sib}(i)} \boldsymbol{\mu}_k^u \right] \quad (28c)$$

$$\boldsymbol{\nu}_{1,i}^d = [P_L, P_H]^T \odot \boldsymbol{\phi}(z_{\pi(i)}) \odot \left[ \bigodot_{k \in \text{sib}(i)} \boldsymbol{\mu}_k^u \right] \quad (28d)$$

$$\boldsymbol{\eta}_i^d = \begin{cases} [1 - P_{\text{root}}, P_{\text{root}}]^T, & \pi(i) \in \mathcal{T}_{\text{root}} \\ \boldsymbol{\mu}_{\pi(i)}^d, & \pi(i) \in (\mathcal{T} \setminus \mathcal{T}_{\text{root}}) \setminus \mathcal{T}_{\text{leaf}} \end{cases} \quad (28e)$$

A numerically stable implementation of (28b) that we employ is illustrated in the second expression in (28b).

The above upward and downward messages have discrete representations, which is practically important and is a consequence of the fact that we use a Gaussian prior on the signal coefficients, see (5). Indeed, in contrast with the existing message passing algorithms for compressive sampling [6], [9]–[11], our max-product scheme employs *exact* messages.

3) *Maximizing Beliefs*: Upon computing and passing all the upward and downward messages, we maximize the beliefs, which have the following general form [24]:

$$b(\boldsymbol{\theta}_i) = \alpha \psi_i(\boldsymbol{\theta}_i) m_{\pi(i) \rightarrow i}(q_i) \prod_{k \in \text{ch}(i)} m_{k \rightarrow i}(q_i) \quad (29)$$

for each  $i \in \mathcal{T}$ , where  $\alpha > 0$  is a normalizing constant. [In (29), we set  $m_{\pi(i) \rightarrow i}(q_i) = 1$  if  $i \in \mathcal{T}_{\text{root}}$  and  $\prod_{k \in \text{ch}(i)} m_{k \rightarrow i}(q_i) = 1$  if  $i \in \mathcal{T}_{\text{leaf}}$ .] We then use these beliefs to obtain the mode

$$\widehat{\boldsymbol{\theta}}_{\mathcal{T}} = \arg \max_{\boldsymbol{\theta}_{\mathcal{T}}} p_{\boldsymbol{\theta}_{\mathcal{T}} | \sigma^2, \mathbf{z}}(\boldsymbol{\theta}_{\mathcal{T}} | \sigma^2, \mathbf{z}) \quad (30)$$

where the elements of  $\widehat{\boldsymbol{\theta}}_{\mathcal{T}}$  are [see (19)]

$$\begin{aligned} \widehat{\boldsymbol{\theta}}_i &= [\widehat{q}_i, \widehat{s}_i(\widehat{q}_i)]^T = \arg \max_{\boldsymbol{\theta}_i} b(\boldsymbol{\theta}_i) \\ &= \begin{cases} [1, \widehat{s}_i(1)]^T, & \beta_i(1) \geq \beta_i(0) \\ [0, \widehat{s}_i(0)]^T, & \text{otherwise} \end{cases}, \quad i \in \mathcal{T} \end{aligned} \quad (31a)$$

and

$$\begin{aligned} \boldsymbol{\beta}_i &= [\beta_i(0), \beta_i(1)]^T \\ &= \begin{cases} \alpha_1 [1 - P_{\text{root}}, P_{\text{root}}]^T \odot \boldsymbol{\phi}(z_i) \odot \boldsymbol{\eta}_i^u, & i \in \mathcal{T}_{\text{root}} \\ \alpha_1 \boldsymbol{\phi}(z_i) \odot \boldsymbol{\mu}_i^d \odot \boldsymbol{\eta}_i^u, & i \in \mathcal{T} \setminus \mathcal{T}_{\text{root}} \end{cases}. \end{aligned} \quad (31b)$$

Here,  $\alpha_1 > 0$  is a normalizing constant. The detailed derivation for the forms of  $\widehat{\boldsymbol{\theta}}_i$  and  $\boldsymbol{\beta}_i$  in (31) is provided in Appendix B-III.

#### IV. SELECTING $\sigma^2$ VIA GRID SEARCH

We can integrate  $\sigma^2$  out, yielding the marginal posterior of  $\boldsymbol{\theta}$  in (13a), and derive an ‘outer’ EM iteration for maximizing  $p_{\boldsymbol{\theta} | \mathbf{y}}(\boldsymbol{\theta} | \mathbf{y})$ . Even though it guarantees monotonic increase of the marginal posterior  $p_{\boldsymbol{\theta} | \mathbf{y}}(\boldsymbol{\theta} | \mathbf{y})$ , the ‘outer’ EM iteration does not work well in practice because it gets stuck in an undesirable local maximum of  $p_{\boldsymbol{\theta} | \mathbf{y}}(\boldsymbol{\theta} | \mathbf{y})$ . To find a better (generally local) maximum of  $p_{\boldsymbol{\theta} | \mathbf{y}}(\boldsymbol{\theta} | \mathbf{y})$ , we apply a grid search over  $\sigma^2$  as follows.

We apply the EM algorithm in Section III using a range of values of the regularization parameter  $\sigma^2$ . We traverse the grid of  $K$  values of  $\sigma^2$  sequentially and use the signal estimate from the previous grid point to initialize the signal estimation at the current grid point (as depicted in Fig. 4): in particular, we move from a larger  $\sigma^2$  (say  $\sigma_{\text{old}}^2$ ) to the next smaller  $\sigma_{\text{new}}^2$  ( $< \sigma_{\text{old}}^2$ ) and use  $\mathbf{s}^{(+\infty)}(\sigma_{\text{old}}^2)$  (obtained upon convergence of the EM iteration in Section III for  $\sigma^2 = \sigma_{\text{old}}^2$ ) to initialize the EM iteration at  $\sigma_{\text{new}}^2$ . The largest  $\sigma^2$  on the grid and the initial signal estimate at this grid point are selected as

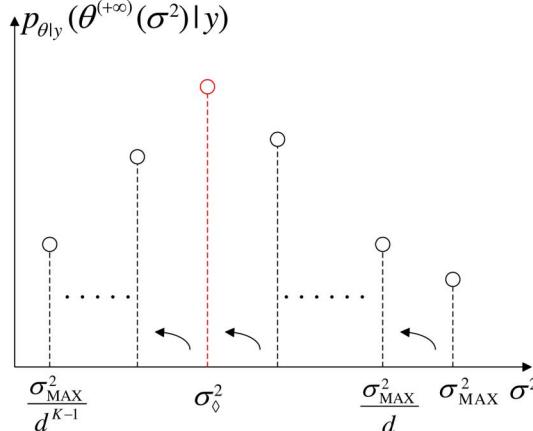
$$\sigma_{\text{MAX}}^2 = \frac{\|\mathbf{y}\|_2^2}{p + N}, \quad \boldsymbol{\theta}^{(0)}(\sigma_{\text{MAX}}^2) = \mathbf{0}_{2p \times 1}. \quad (32a)$$

The consecutive grid points  $\sigma_{\text{new}}^2$  and  $\sigma_{\text{old}}^2$  satisfy

$$\sigma_{\text{new}}^2 = \frac{\sigma_{\text{old}}^2}{d} \quad (32b)$$

where  $d > 1$  is a constant determining the search resolution. Finally, we select the  $\sigma^2$  from the above grid of candidates that yields the largest marginal posterior distribution (13a):

$$\sigma_{\diamond}^2 = \arg \max_{\sigma^2 \in \{\sigma_{\text{MAX}}^2, \sigma_{\text{MAX}}^2 / d, \dots, \sigma_{\text{MAX}}^2 / d^{K-1}\}} p_{\boldsymbol{\theta} | \mathbf{y}}(\boldsymbol{\theta}^{(+\infty)}(\sigma^2) | \mathbf{y}) \quad (33)$$

Fig. 4. Grid search for selecting  $\sigma^2$ .

and the final estimates of  $\theta$  and  $s$  as  $\theta^{(+\infty)}(\sigma_\diamond^2)$  and  $s^{(+\infty)}(\sigma_\diamond^2)$ , respectively, see Fig. 4.

## V. NUMERICAL EXAMPLES

We compare the reconstruction performances of the following methods: (a) our proposed max-product EM (MP-EM) algorithm in Section III with the variance parameter  $\sigma^2$  selected via grid search using the marginal-posterior based criterion in Section IV, search resolution  $d = 2$ , and zero initial signal estimate:

$$\mathbf{s}^{(0)} = \mathbf{0}_{p \times 1} \quad (34)$$

with Matlab implementations available at <http://home.eng.iastate.edu/~ald/MPEM.html>; (b) our MP-EM algorithm in Section III with  $\sigma^2$  tuned manually for good performance (labeled MP-EM<sub>OPT</sub>) with  $d = 2$  and zero  $\mathbf{s}^{(0)}$  in (34), used as a benchmark; (c) the Gaussian-mixture version of the turbo-AMP approach [6] with a Matlab implementation in [26] and the tuning hyperparameters chosen as the default values<sup>3</sup> in this implementation; (d) the fixed-point continuation active set (FPC<sub>AS</sub>) algorithm [27] that aims at minimizing the Lagrangian cost function

$$0.5\|\mathbf{y} - H\mathbf{s}\|_2^2 + \tau\|\mathbf{s}\|_1 \quad (35a)$$

with the regularization parameter  $\tau$  computed as

$$\tau = 10^a \|\mathbf{H}^T \mathbf{y}\|_\infty \quad (35b)$$

where  $a$  is a tuning parameter chosen manually to achieve good reconstruction performance; (e) the Barzilai-Borwein version of the gradient-projection for sparse reconstruction (GPSR) method with debiasing in [28, Sec. III.B] with the convergence threshold  $\text{tolP} = 10^{-5}$  and tuning parameter  $a$  in (35b) chosen manually to achieve good reconstruction performance; (f) the normalized iterative hard thresholding (NIHT) scheme [29] initialized by the zero  $\mathbf{s}^{(0)}$  in (34); (g) the model-based iterative hard thresholding (MB-IHT) algorithm [7] using a greedy tree approximation [30], initialized by the zero  $\mathbf{s}^{(0)}$  in

<sup>3</sup>These default values were designed for a set of approximately sparse wavelet coefficients of natural images, see [6], which differ from the simulated signals in Section V-A.

(34). (h) the VB tree-structured compressive sensing [5] with a Matlab implementation in [31] and the tuning hyperparameters chosen as the default values in this implementation;<sup>4</sup> For the MP-EM, NIHT, and MB-IHT iterations, we use the following convergence criterion:

$$\frac{\|\mathbf{s}^{(j+1)} - \mathbf{s}^{(j)}\|_2^2}{p} < \delta \quad (36)$$

where  $\delta > 0$  is the convergence threshold selected in the following examples so that the performances of the above methods do not change significantly by further decreasing  $\delta$ .

For MP-EM, we set the tuning constants in all following examples as<sup>5</sup>

$$\gamma^2 = 1000, \quad \epsilon^2 = 0.1, \quad P_{\text{root}} = P_H = 0.2, \quad P_L = 10^{-5} \quad (37)$$

which leads to  $\frac{\mathbb{E}[\sum_{i=1}^p q_i]}{p} = 0.0108$ .

The sensing matrix  $H$  has the following structure:

$$H = \frac{1}{\rho_\Phi} \Phi \Psi \quad (38)$$

where  $\Phi$  is the  $N \times p$  sampling matrix and  $\Psi$  is the  $p \times p$  orthogonal transform matrix (satisfying  $\Psi \Psi^T = I_p$ ). Note that  $H$  in (38) satisfies the spectral norm condition (3). We set the tree depth  $L = 4$ .

### A. Small-Scale Structured Sparse Signal Reconstruction

We generated the binary state variables  $\mathbf{q}$  of length  $p = 1024$  using the Markov tree model in Section II. Conditional on  $q_i, s_i$  are generated according to (5a). Here, the matrix-to-vector conversion operator  $v(\cdot)$  corresponds to simple columnwise conversion, except for VB whose implementation [31] requires the use of Matlab's `wavedec2` function for this purpose. The sampling matrix  $\Phi$  in (38) have been simulated using

- (i) a *white Gaussian matrix* whose entries are i.i.d. standard Gaussian random variables,
- (ii) a *row-correlated Gaussian matrix* with i.i.d. zero-mean Gaussian columns (indexed by  $k = 1, 2, \dots, p$ ) having covariance matrix whose  $(i, j)$ th element is

$$\text{cov}(\Phi_{i,k}, \Phi_{j,k}) = r^{|i-j|}, \quad i, j = 1, 2, \dots, N \quad (39a)$$

useful, e.g., in modeling time-series data [32, Sec. 5], and

- (iii) a *column-correlated Gaussian matrix* with i.i.d. zero-mean Gaussian rows (indexed by  $k = 1, 2, \dots, N$ ) having covariance matrices whose  $(i, j)$ th element is

$$\text{cov}(\Phi_{k,i}, \Phi_{k,j}) = c^{|i-j|}, \quad i, j = 1, 2, \dots, p. \quad (39b)$$

The general column correlation model (iii) for the design (sensing) matrices is analyzed in [32]–[34], see also [35]–[37], which employ this correlation structure. Correlations among

<sup>4</sup>We scaled the sensing matrix  $H$  by  $\sqrt{p/\text{tr}(HH^T)}$  prior to applying the VB method, which helped improve its performance compared with using the unscaled  $H$ . This scaling is applied in the turbo-AMP implementation [26].

<sup>5</sup>The selections of  $\gamma^2$  and  $\epsilon^2$  in (37) enforce a purely sparse signal model because  $\gamma^2 \gg \epsilon^2$ . When selecting  $P_{\text{root}}$ ,  $P_H$ , and  $P_L$ , we suggest to use (7a) and check that the expected number of large-magnitude signal coefficients is roughly of the order of the signal sparsity level that we expect. For example, the selections in (37) lead to the normalized expected number of large-magnitude signal coefficients  $\mathbb{E}[\sum_{i=1}^p q_i]/p = 0.0108$ .

columns of the design matrices occur e.g., in genomic applications [38, Sec. 18.4] and spatially correlated designs are relevant to functional magnetic resonance imaging (fMRI) [39].

The transform matrix  $\Psi$  in (38) is chosen to be identity:  $\Psi = I_p$ . Hence, in this example, the sampling and sensing matrices  $\Phi$  and  $H$  are the same up to a proportionality constant.

We simulate the observation vectors  $\mathbf{y}$  using the measurement and prior models in (2), (5), and (6) and following model parameters:

$$\begin{aligned} \epsilon_*^2 &= 1, \quad \sigma_*^2 = 10^{-6}, \quad (P_{\text{root}})_* = (P_H)_* = 0.5, \\ (P_L)_* &= 10^{-4}, \quad \gamma_*^2 \in \{10^3, 10^4, 10^5\} \end{aligned} \quad (40)$$

where the subscripts  $*$  emphasize that these selections are the true model parameters employed to simulate the measurements and are generally *different* from the tuning constants (37) employed by the MP-EM method. Here, our goal is to show the performance of the MP-EM method in the case where there is a mismatch between the tuning parameters and corresponding true model parameters. The choices  $(P_H)_*$ ,  $(P_{\text{root}})_*$ , and  $(P_L)_*$  in (40) correspond to the normalized expected number of large-magnitude signal coefficients  $\frac{\mathbb{E}[\sum_{i=1}^p q_i]}{p} = 0.0919$ , computed using (7a).

We vary the values of  $\gamma_*^2$  to test the performances of various methods at different signal-to-noise ratios (SNRs).

Our performance metric is the *average* normalized mean-square error (NMSE) of an estimate  $\tilde{\mathbf{s}}$  of the signal coefficient vector (used also in e.g., [40]):

$$\text{NMSE}\{\tilde{\mathbf{s}}\} = \mathbb{E}_{\Phi, \mathbf{s}, \mathbf{y}} \left[ \frac{\|\tilde{\mathbf{s}} - \mathbf{s}\|_2^2}{\|\mathbf{s}\|_2^2} \right] \quad (41)$$

computed using 500 Monte Carlo trials, where *averaging* is performed over the random Gaussian sampling matrices  $\Phi$ , signal  $\mathbf{s}$ , and measurements  $\mathbf{y}$ .

We select the convergence threshold in (36) to  $\delta = 10^{-10}$ . For MP-EM and MP-EM<sub>OPT</sub>, we set the grid length  $K = 16$ . The tuning parameters for MP-EM are given in (37).

The NIHT and MB-IHT methods require knowledge of the signal sparsity level (i.e., an upper bound on the number of nonzero coefficients); in this example, we set the signal sparsity level for these methods to the exact number of large-magnitude signal coefficients  $\sum_{i=1}^p q_i$ . For GPSR and FPC<sub>AS</sub>, we vary  $a$  in (35b) within the set  $\{-1, -2, -3, -4, -5, -6, -7, -8, -9\}$  and, for each  $N/p$  and each of the two methods, we use the optimal  $a$  that achieves the smallest NMSE.

The turbo-AMP implementation in [26] requires a function input  $\text{xRange}$  that corresponds to the range of the input signal  $\Psi \mathbf{s}$ . In this example, we set the value of this tuning constant to six standard deviations of the signal coefficients in  $\mathbf{s}$ :

$$\text{xRange} = 6\sigma_* \sqrt{\frac{\mathbb{E}[\sum_{i=1}^p q_i]}{p} \gamma_*^2 + \left(1 - \frac{\mathbb{E}[\sum_{i=1}^p q_i]}{p}\right) \epsilon_*^2} \quad (42)$$

where  $\sigma_* = \sqrt{\sigma_*^2}$ ; turbo-AMP with this selection performs well compared with other choices of  $\text{xRange}$  that we tested. Selecting too small or too large  $\text{xRange}$  would lead to deteriorated performance of turbo-AMP. Turbo-AMP is particularly sensitive to underestimation of this quantity and less sensitive to selecting larger values than optimal.

*1) White and Row-Correlated Sensing Matrices:* Fig. 5 shows the NMSEs of different methods as functions of the subsampling factor  $N/p$  for the three choices of  $\gamma_*^2$  in (40), corresponding to relatively low, medium, and high SNRs, and white and row-correlated sensing matrices with correlation parameter  $r = 0.2$  in (39a). Here, a larger value of the high-signal relative variance  $\gamma_*^2$  implies a relatively higher SNR. Indeed, for each method, the signal with higher SNR can be reconstructed with a smaller NMSE than the signal with lower SNR: Compare Figs. 5(a), 5(c), and 5(e) as well as Figs. 5(b), 5(d), and 5(f).

For white Gaussian sampling matrices, the methods that employ the probabilistic tree structure of the signal coefficients (turbo-AMP, MP-EM, MP-EM<sub>OPT</sub>, and VB) clearly outperform all other approaches, see Figs. 5(a), 5(c), and 5(e). For row-correlated Gaussian sampling matrices, MP-EM and MP-EM<sub>OPT</sub> achieve the best overall performances, followed by the VB method; turbo-AMP is sensitive to introducing correlation among elements of the sampling matrix  $\Phi$  and performs poorly for smaller  $N/p$ , see Figs. 5(b), 5(d), and 5(f).

For white Gaussian sampling matrices, we observe the following:

- at low SNR, MP-EM and MP-EM<sub>OPT</sub> outperform other approaches when  $N/p > 0.275$ , see Fig. 5(a);
- at medium and high SNRs, turbo-AMP achieves the best overall performance, followed by MP-EM<sub>OPT</sub> and MP-EM, see Figs. 5(c) and 5(e).

The NIHT method performs relatively poorly for smaller  $N/p$ , but improves as  $N/p$  increases. For sufficiently high  $N/p$ , NIHT achieves smaller NMSEs than other methods that do not exploit the probabilistic tree structure.

In Fig. 5, the NMSEs of MP-EM are close to those of MP-EM<sub>OPT</sub>, which implies that the marginal-posterior based criterion in Section IV selects the noise variance parameter well in this example.

The performance of turbo-AMP deteriorates with introduction of correlation among elements of the sampling matrix  $\Phi$ : The NMSEs of turbo-AMP for some subsampling factors are more than an order of magnitude larger for row-correlated sampling matrices than for white sampling matrices. In contrast, the NMSEs for all the other methods increase only slightly when we introduce sampling matrix correlation (39a), compare the left and right-hand sides of Fig. 5. Increasing this correlation by increasing  $r$  to 0.3 in (39a) results in further performance deterioration of turbo-AMP (i.e., turbo-AMP has very high NMSEs for all  $N/p$  in this case), whereas the competing methods continue to perform well.

The VB method performs well under both white and row-correlated sensing matrix scenarios and turbo-AMP has a superior reconstruction performance under the white sensing matrix scenario. These good performances are likely facilitated by the fact that VB and turbo-AMP *learn* the Markov tree parameters from the measurements.

*2) Column-Correlated Sensing Matrices:* Fig. 6(a) shows the NMSEs of different methods as functions of the subsampling factor  $N/p$  for column-correlated sampling matrices having the correlation constant  $c = 0.2$  in (39b) under the medium SNR scenario. Here, MP-EM and MP-EM<sub>OPT</sub> have the smallest NMSEs over nearly the entire range of  $N/p$  considered. Fig. 6(b) shows the NMSEs as functions of  $c$  for  $N/p$  fixed at 0.4. Here, only turbo-AMP is very sensitive to

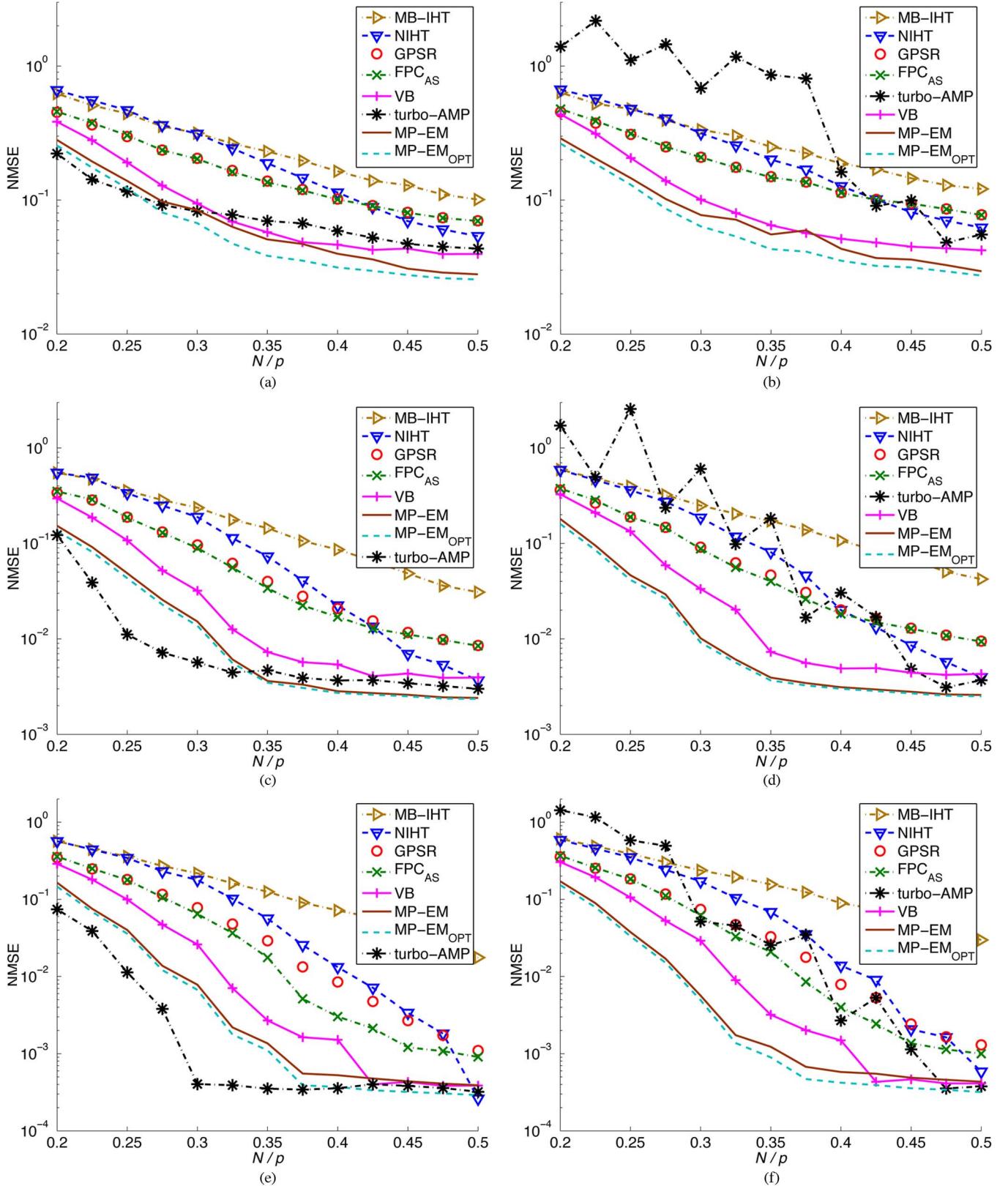


Fig. 5. NMSEs as functions of the subsampling factor  $N/p$  for (a)-(b) low SNR with  $\gamma_*^2 = 10^3$ , (c)-(d) medium SNR with  $\gamma_*^2 = 10^4$ , and (e)-(f) high SNR with  $\gamma_*^2 = 10^5$  using [left: (a), (c), (e)] white and [right: (b), (d), (f)] row-correlated sensing matrices with correlation parameter  $r = 0.2$ , respectively.

the presence of correlations among the elements of the sampling matrix, whereas all other methods vary only slightly as functions of  $c$ .

We also observe numerical instability of turbo-AMP when correlated Gaussian sampling matrices are employed, which is exhibited by the oscillatory behavior of its NMSEs in the right

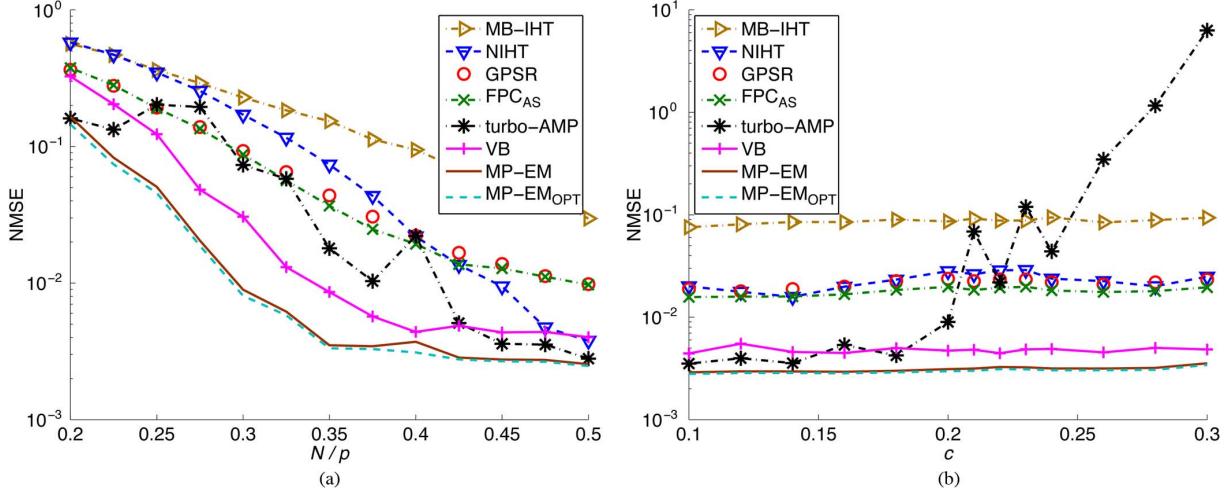


Fig. 6. Column-correlated sensing matrices: NMSEs as functions of (a) the subsampling factor  $N/p$  for correlation parameter  $c = 0.2$  and (b)  $c$  for  $N/p = 0.4$  under the medium SNR scenario with  $\gamma_*^2 = 10^4$ .

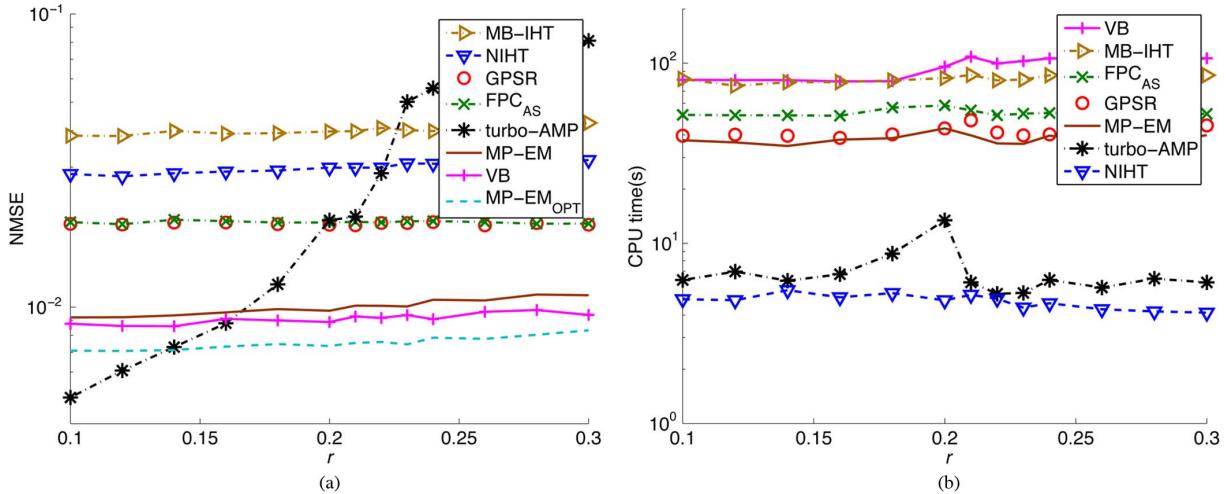


Fig. 7. (a) NMSEs and (b) CPU times as functions of the correlation parameter  $r$  for the  $128 \times 128$  ‘Cameraman’ image when  $N/p = 0.3$ .

side of Fig. 5 and in Fig. 6 [demanding more averaging than the 500 Monte Carlo trials that we employ to estimate (41)].

We simulated sampling matrices  $\Phi$  that have variable column norms or row norms, which led to deteriorating performances of turbo-AMP in both cases, whereas the competing methods perform well. The fact that turbo-AMP has been derived assuming Gaussian sensing matrices with i.i.d. elements explains its poor performance for sensing matrices that deviate sufficiently from this assumption.

The MB-IHT method, which employs a greedy tree approximation and deterministic tree structure, achieves quite a poor NMSE performance in Figs. 5 and 6. A relatively poor performance of MB-COSAMP (which employs the same deterministic tree structure) has also been reported in [6, Sec. IV.B].

### B. Image Reconstruction

We reconstruct  $128 \times 128$  and  $256 \times 256$  test images from noiseless compressive samples ( $\sigma_*^2 = 0$ ). Here, the matrix-to-vector conversion operator  $v(\cdot)$  is based on the columnwise conversion for  $128 \times 128$  images, and Matlab wavelet decomposition function `wavedec2` with Haar wavelet for  $256 \times 256$  images, which has also been used in [4] and [6]. Before taking

the wavelet transform, we subtract the mean of original image to ensure that  $\Psi s$  has zero mean.

For turbo-AMP, we set the function input `xRange` to 255, which is the difference between the minimum and maximum possible image values in this example.<sup>6</sup> Observe that the turbo-AMP implementation in [26] needs additional prior information about the signal range, which is not required by other methods.

*1) Medium Scale With Row-Correlated Gaussian Sampling Matrices:* We reconstruct the  $128 \times 128$  ‘Cameraman’ image (cropped from the original  $256 \times 256$  image in Fig. 9(b), as was also done in [26], [31] and corresponding papers [5], [6]) from compressive samples generated using row-correlated Gaussian sampling matrices with covariances between the elements described by (39a). Our performance metric is the NMSE in (41) computed using 10 Monte Carlo trials, where the averaging is performed only over the random Gaussian sampling matrices  $\Phi$ .

In this example, the convergence threshold in (36) is set to  $\delta = 0.01$ . For MP-EM and MP-EM<sub>OPT</sub>, we set the grid length  $K = 16$ . The tuning parameters for MP-EM are given in (37).

<sup>6</sup>The authors thank Dr. Subhojit Som from Microsoft Inc. for the correspondence with regard to setting this parameter.

We set the sparsity level  $r$  for NIHT as  $2000N/p$  and  $2500N/p$  for MB-IHT, tuned for good NMSE performance.

Fig. 7 shows the NMSEs and CPU times of different methods reconstructing the  $128 \times 128$  ‘Cameraman’ image as functions of the correlation parameter  $r$  in (39a) with  $N/p = 0.3$ . Since MP-EM and MP-EM<sub>OPT</sub> have the same runtime, we report only that of MP-EM in Fig. 7(b). Turbo-AMP has the smallest NMSE when  $N/p \leq 0.12$ . However, its NMSE increases sharply as  $r$  becomes larger: turbo-AMP has the largest NMSE when  $N/p > 0.22$ . In contrast, the NMSEs for all the other methods keep nearly constants as we increase  $r$ . The MP-EM, MP-EM<sub>OPT</sub>, and VB methods have smaller NMSEs than GPSR, FPC<sub>AS</sub>, NIHT, and MB-IHT for all the correlation coefficients  $r$  considered. The VB approach performs slightly better than MP-EM, but is slower than MP-EM and MP-EM<sub>OPT</sub>. In terms of CPU time, NIHT is the fastest among all the methods compared and turbo-AMP requires 0.3 s to 8.6 s more than NIHT, both of which are faster than the remaining methods.<sup>7</sup> The VB scheme consumes the largest amount of CPU time among all the methods for all the correlation coefficient  $r$  considered; MP-EM and MP-EM<sub>OPT</sub> are faster than GPSR, FPC<sub>AS</sub>, MB-IHT, and VB.

As before, the good performance of VB is likely facilitated by the fact that it learns the Markov tree parameters from the measurements.

Fig. 8 shows the reconstructed  $128 \times 128$  ‘Cameraman’ image by the best four methods for  $N/p = 0.3$  and  $r = 0.2$  using one realization of the sampling matrix  $\Phi$ . In Fig. 8, we also report the peak signal-to-noise ratios (PSNRs) of these methods, where the PSNR of an estimated signal  $\tilde{\mathbf{s}}$  is defined as [41, eq. (3.7)]:

$$\text{PSNR}(\text{dB}) = 10 \log_{10} \left\{ \frac{[(\Psi \mathbf{s})_{\text{MAX}} - (\Psi \mathbf{s})_{\text{MIN}}]^2}{\|\tilde{\mathbf{s}} - \mathbf{s}\|_2^2/p} \right\}. \quad (43)$$

*2) Large Scale With Structurally Random Sampling Matrices:* We now reconstruct several  $256 \times 256$  test images shown in Fig. 9 from compressive samples. The sampling matrix  $\Phi$  is generated from structurally random compressive samples [42] and the transform matrix  $\Psi$  in (38) is the  $p \times p$  orthogonal inverse Haar wavelet transform matrix, which implies that the sensing matrix  $H$  has orthonormal rows:  $HH^T = I_N$  and, consequently,  $\rho_\Phi = \rho_H = 1$ . Our performance metric in this example is the PSNR, see (43).

In this example, the convergence threshold in (36) is set to  $\delta = 0.1$ . For MP-EM and MP-EM<sub>OPT</sub>, we set the grid length  $K = 12$ . The tuning parameters for MP-EM are the same as before and given in (37).

We set the signal sparsity levels for NIHT and MB-IHT to  $10000 N/p$  and  $15000 N/p$ , respectively, tuned for good PSNR performance. For FPC<sub>AS</sub> and GPSR, we set the regularization parameter  $a = -3$  [see (35b)], which yields generally the best PSNR performance for these two methods.

We do not include the VB method in this example because its implementation [31] cannot be applied to reconstruct the large-scale images in Fig. 9.

Fig. 10 shows the PSNRs and CPU times of different methods reconstructing the  $256 \times 256$  ‘Cameraman’ image, as functions

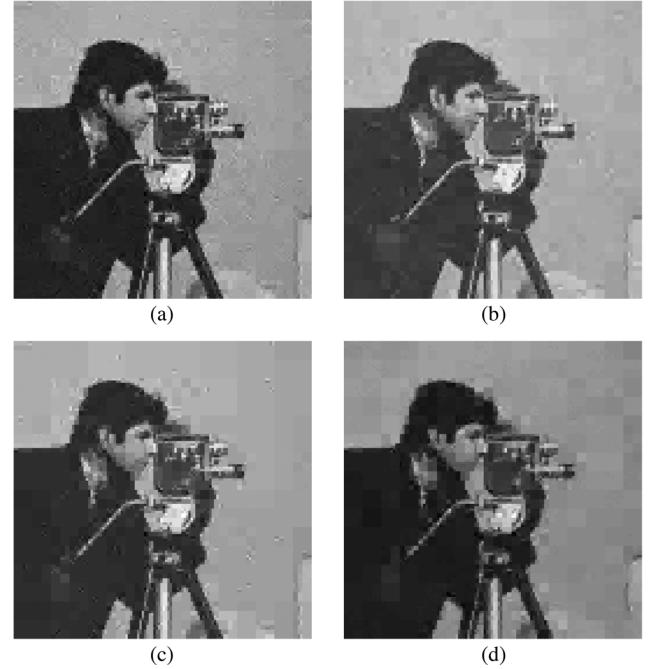


Fig. 8. The  $128 \times 128$  ‘Cameraman’ image reconstructed by various methods for  $r = 0.2$  and  $N/p = 0.3$ . (a) MP-EM<sub>OPT</sub> (PSNR 26.3 dB); (b) VB (PSNR 25.8 dB); (c) MP-EM (PSNR 24.9 dB); (d) turbo-AMP (PSNR 23.8 dB).

TABLE I  
PSNRs FOR  $N/p = 0.35$

	NIHT	MB-IHT	FPC <sub>AS</sub>	GPSR	turbo-AMP	MP-EM	MP-EM <sub>OPT</sub>
Lena	24.3	24.8	25.3	25.5	<b>29.2</b>	27.8	27.9
Cameraman	26.0	26.0	26.8	26.8	<b>30.6</b>	29.9	30.1
House	29.8	29.7	30.5	30.5	<b>33.4</b>	32.6	33.1
Boat	22.5	22.9	23.7	24.0	<b>27.1</b>	26.1	26.1
Einstein	26.9	27.4	27.4	27.7	<b>30.4</b>	30.0	30.0
Peppers	25.8	26.2	26.1	26.2	<b>30.2</b>	29.2	29.3
Couple	28.8	29.1	30.3	30.2	<b>33.6</b>	32.6	32.7

of the subsampling factor  $N/p$ . Turbo-AMP has the highest PSNRs for all  $N/p$ . The performances of MP-EM and MP-EM<sub>OPT</sub> are close to that of turbo-AMP: the PSNRs of MP-EM<sub>OPT</sub> are 0.4 dB to 0.7 dB less than those of turbo-AMP. Moreover, the PSNR improvement for MP-EM against its other closest competitors varies between 2.1 dB to 3.2 dB. In terms of CPU time, NIHT is the fastest among all the methods compared; turbo-AMP is the second fastest and takes around 4 s for each  $N/p$ . The MP-EM method requires 3.3 s to 6.5 s more than turbo-AMP, but is clearly faster than GPSR, FPC<sub>AS</sub>, and MB-IHT for nearly all measurement points. As before, MP-EM and MP-EM<sub>OPT</sub> have the same runtime and we report only that of MP-EM in Fig. 10(b).

Table I shows the PSNRs of the compared methods for different images and  $N/p$  equal to 0.35. The MP-EM, MP-EM<sub>OPT</sub>, and turbo-AMP methods clearly outperform the other methods for every image. In Table I, turbo-AMP is better than MP-EM<sub>OPT</sub> and MP-EM for all the images: The improvement in terms of PSNR varies between 0.3 dB and 1.4 dB.

In Fig. 10 and Table I, MB-IHT achieves a fair performance and consumes the largest amount of CPU time. Turbo-AMP performs well for all  $N/p$  and images and outperforms all competitors, which is likely because

<sup>7</sup>Regarding the reported CPU time, note that the turbo-AMP code does not use Matlab only, but combines Matlab and JAVA codes.



Fig. 9. The  $256 \times 256$  test images. (a) Lena; (b) cameraman; (c) house; (d) boat; (e) Einstein; (f) peppers; (g) couple.

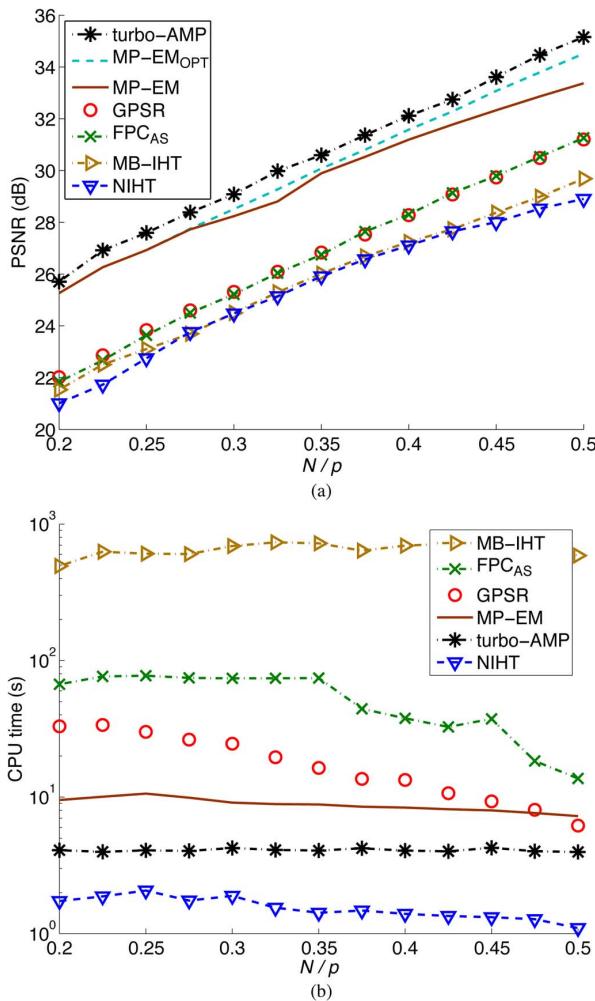


Fig. 10. (a) PSNRs and (b) CPU times as functions of the subsampling factor  $N/p$  for the  $256 \times 256$  ‘Cameraman’ image.

- it uses a more general prior on the binary state variables (than our MP-EM method), which allows the tree probability parameters  $P_H$ ,  $P_L$ ,  $\gamma^2$ , and  $\epsilon^2$  to vary between the signal decomposition levels, and
- learns the tree probability parameters from the measurements.

In contrast, our MP-EM method employs the crude choices of the tree and other tuning parameters in (37).

Fig. 11 shows the reconstructed  $256 \times 256$  ‘Cameraman’ image by the best four methods for  $N/p = 0.35$ : In this case, the turbo-AMP algorithm achieves the best reconstructed image quality compared with other methods, followed closely

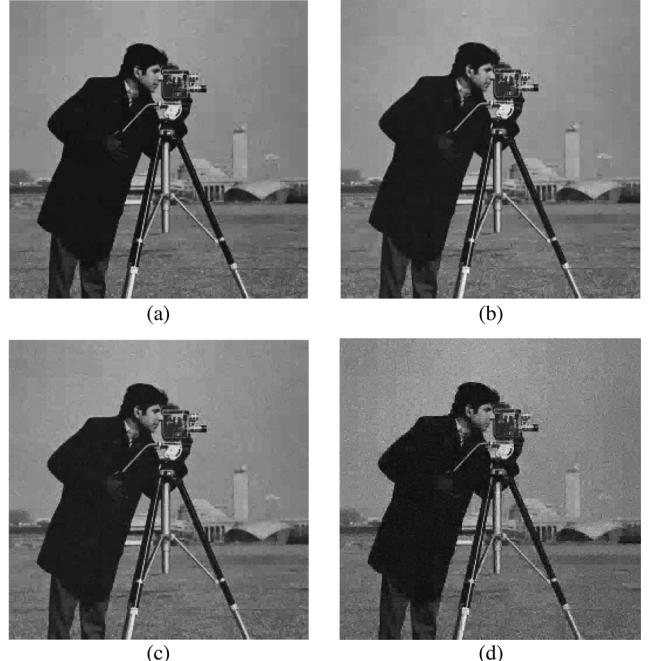


Fig. 11. The ‘Cameraman’ image reconstructed by various method for  $N/p = 0.35$ . (a) turbo-AMP (PSNR = 30.59 dB); (b) MP-EM<sub>OPT</sub> (PSNR = 30.08 dB); (c) MP-EM (PSNR = 29.89 dB); and (d) GPSR (PSNR 26.83 dB).

by MP-EM and MP-EM<sub>OPT</sub>; the reconstructions of all other methods are clearly inferior to these schemes.

## VI. CONCLUDING REMARKS

We presented a Bayesian EM algorithm for reconstructing approximately sparse signal from compressive samples using a Markov tree prior for the signal coefficients. We employed the max-product belief propagation algorithm to implement the M step of the proposed EM iteration. Compared with the existing message passing algorithms in the compressive sampling area, our method does not approximate the message form. The simulation results show that our algorithm often outperforms existing algorithms for simulated signals and standard test images with different sampling operators and can successfully reconstruct signals collected by sampling matrices with correlated elements and variable norms of rows and columns.

Our future work will include the convergence analysis of the MP-EM algorithm, incorporating other measurement models, using a more general prior distribution for the binary state variables, and designing schemes for learning the probabilistic Markov tree parameters from the measurements.

APPENDIX A  
DERIVATION OF THE MP-EM ITERATION AND PROOFS OF ITS  
MONOTONICITY AND THEOREM 1

We first determine the complete-data posterior distribution and the distribution of the missing data  $\mathbf{z}$  given the observed data  $\mathbf{y}$  and parameters  $\boldsymbol{\theta}$  and  $\sigma^2$ . We then use these distributions to derive the EM iteration in Section III following the standard approach outlined in, e.g., [16, Sec. 12.3]. Finally, we prove the monotonicity of the MP-EM iteration in (17) and Theorem 1.

Consider the hierarchical two-stage model in (14). The complete-data posterior distribution for known  $\sigma^2$  is

$$\begin{aligned} & p_{\boldsymbol{\theta}, \mathbf{z} | \sigma^2, \mathbf{y}}(\boldsymbol{\theta}, \mathbf{z} | \sigma^2, \mathbf{y}) \\ & \propto p_{\mathbf{y} | \mathbf{z}, \sigma^2}(\mathbf{y} | \mathbf{z}, \sigma^2) p_{\mathbf{z} | \mathbf{s}}(\mathbf{z} | \mathbf{s}) p_{\mathbf{s} | \mathbf{q}, \sigma^2}(\mathbf{s} | \mathbf{q}, \sigma^2) p_{\mathbf{q}}(\mathbf{q}) \\ & \propto \frac{\exp \left\{ -0.5(\mathbf{y} - H\mathbf{z})^T [C(\sigma^2)]^{-1} (\mathbf{y} - H\mathbf{z}) \right\}}{\sqrt{\det [C(\sigma^2)]}} \\ & \cdot \left( \frac{\epsilon^2}{\gamma^2} \right)^{0.5 \sum_{i=1}^p q_i} \\ & \cdot p_{\mathbf{q}}(\mathbf{q}) \exp \left[ -0.5 \|\mathbf{z} - \mathbf{s}\|_2^2 / \sigma^2 - 0.5 \mathbf{s}^T D^{-1}(\mathbf{q}) \mathbf{s} / \sigma^2 \right] \end{aligned} \quad (\text{A1a})$$

where  $C(\sigma^2) = \sigma^2(I_N - HH^T)$ . Consequently, the distribution of the missing data  $\mathbf{z}$  given the observed data  $\mathbf{y}$  and parameters  $\boldsymbol{\theta}$  and  $\sigma^2$  is

$$\begin{aligned} & p_{\mathbf{z} | \sigma^2, \mathbf{y}, \boldsymbol{\theta}}(\mathbf{z} | \sigma^2, \mathbf{y}, \boldsymbol{\theta}) \\ & = p_{\mathbf{z} | \sigma^2, \mathbf{y}, \mathbf{s}}(\mathbf{z} | \sigma^2, \mathbf{y}, \mathbf{s}) \\ & = \mathcal{N}(\mathbf{z} | \mathbb{E}_{\mathbf{z} | \sigma^2, \mathbf{y}, \mathbf{s}}(\mathbf{z} | \sigma^2, \mathbf{y}, \mathbf{s}), \text{cov}_{\mathbf{z} | \sigma^2, \mathbf{y}, \mathbf{s}}(\mathbf{z} | \sigma^2, \mathbf{y}, \mathbf{s})) \end{aligned} \quad (\text{A1b})$$

where

$$\begin{aligned} \mathbb{E}_{\mathbf{z} | \sigma^2, \mathbf{y}, \mathbf{s}}(\mathbf{z} | \sigma^2, \mathbf{y}, \mathbf{s}) & = \left\{ H^T [C(\sigma^2)]^{-1} H + I_p / \sigma^2 \right\}^{-1} \\ & \cdot \left\{ H^T [C(\sigma^2)]^{-1} \mathbf{y} + \mathbf{s} / \sigma^2 \right\} \end{aligned} \quad (\text{A1c})$$

$$\text{cov}_{\mathbf{z} | \sigma^2, \mathbf{y}, \mathbf{s}}(\mathbf{z} | \sigma^2, \mathbf{y}, \mathbf{s}) = \left\{ H^T [C(\sigma^2)]^{-1} H + I_p / \sigma^2 \right\}^{-1}. \quad (\text{A1d})$$

By using the matrix inversion lemma [43, eq. (2.22), p. 424] and the following identity [43, p. 425]:

$$(R + STU)^{-1} ST = R^{-1} S (T^{-1} + UR^{-1} S)^{-1} \quad (\text{A2})$$

we simplify the conditional mean of the missing data in (A1c) to the familiar backprojection form:

$$\mathbb{E}_{\mathbf{z} | \sigma^2, \mathbf{y}, \mathbf{s}}[\mathbf{z} | \sigma^2, \mathbf{y}, \mathbf{s}] = \mathbf{s} + H^T(\mathbf{y} - H\mathbf{s}) \quad (\text{A3})$$

We now derive the EM iteration in Section III by noting that the objective function  $\ln p_{\boldsymbol{\theta} | \sigma^2, \mathbf{y}}(\boldsymbol{\theta} | \sigma^2, \mathbf{y})$  that we aim to maximize satisfies the following property [see e.g., [16, eq. (12.4)]:

$$\ln p_{\boldsymbol{\theta} | \sigma^2, \mathbf{y}}(\boldsymbol{\theta} | \sigma^2, \mathbf{y}) = \mathcal{Q}(\boldsymbol{\theta} | \boldsymbol{\theta}^{(j)}) - \mathcal{H}(\boldsymbol{\theta} | \boldsymbol{\theta}^{(j)}) \quad (\text{A4a})$$

where

$$\mathcal{Q}(\boldsymbol{\theta} | \boldsymbol{\theta}^{(j)}) \triangleq \mathbb{E}_{\mathbf{z} | \sigma^2, \mathbf{y}, \boldsymbol{\theta}} \left[ \ln p_{\boldsymbol{\theta}, \mathbf{z} | \sigma^2, \mathbf{y}}(\boldsymbol{\theta}, \mathbf{z} | \sigma^2, \mathbf{y}) | \sigma^2, \mathbf{y}, \boldsymbol{\theta}^{(j)} \right] \quad (\text{A4b})$$

$$\mathcal{H}(\boldsymbol{\theta} | \boldsymbol{\theta}^{(j)}) \triangleq \mathbb{E}_{\mathbf{z} | \sigma^2, \mathbf{y}, \boldsymbol{\theta}} \left[ \ln p_{\mathbf{z} | \sigma^2, \mathbf{y}, \boldsymbol{\theta}}(\mathbf{z} | \sigma^2, \mathbf{y}, \boldsymbol{\theta}) | \sigma^2, \mathbf{y}, \boldsymbol{\theta}^{(j)} \right] \quad (\text{A4c})$$

are the expected complete-data log-posterior distribution and negative entropy of the conditional missing data pdf. The expected complete-data log-posterior  $\mathcal{Q}(\boldsymbol{\theta} | \boldsymbol{\theta}^{(j)})$  follows easily by taking the logarithm of the complete-data posterior distribution (A1a), ignoring constant terms (not functions of  $\boldsymbol{\theta}$ ), and computing the conditional expectation with respect to the missing data given the observed data and parameters from the  $j$ th iteration. Now, the M step requires maximization of  $\mathcal{Q}(\boldsymbol{\theta} | \boldsymbol{\theta}^{(j)})$  with respect to  $\boldsymbol{\theta}$ :

$$\boldsymbol{\theta}^{(j+1)} = \arg \max_{\boldsymbol{\theta}} \mathcal{Q}(\boldsymbol{\theta} | \boldsymbol{\theta}^{(j)}) \quad (\text{A5a})$$

and (16a) follows.

The monotonicity of the MP-EM iteration in (17) follows from

$$\begin{aligned} & \ln p_{\boldsymbol{\theta} | \sigma^2, \mathbf{y}}(\boldsymbol{\theta}^{(j+1)} | \sigma^2, \mathbf{y}) - \ln p_{\boldsymbol{\theta} | \sigma^2, \mathbf{y}}(\boldsymbol{\theta}^{(j)} | \sigma^2, \mathbf{y}) \\ & = \mathbb{D}(p_{\mathbf{z} | \sigma^2, \mathbf{y}, \boldsymbol{\theta}}(\mathbf{z} | \sigma^2, \mathbf{y}, \boldsymbol{\theta}^{(j)}) \| p_{\mathbf{z} | \sigma^2, \mathbf{y}, \boldsymbol{\theta}}(\mathbf{z} | \sigma^2, \mathbf{y}, \boldsymbol{\theta}^{(j+1)}) ) \\ & + \mathcal{Q}(\boldsymbol{\theta}^{(j+1)} | \boldsymbol{\theta}^{(j)}) - \mathcal{Q}(\boldsymbol{\theta}^{(j)} | \boldsymbol{\theta}^{(j)}) \geq 0 \end{aligned} \quad (\text{A5b})$$

by the nonnegativity of KL divergence [17, Theorem 2.8.1], [18, Theorem 8.6.1] and the fact that  $\mathcal{Q}(\boldsymbol{\theta}^{(j+1)} | \boldsymbol{\theta}^{(j)}) - \mathcal{Q}(\boldsymbol{\theta}^{(j)} | \boldsymbol{\theta}^{(j)}) \geq 0$  because  $\mathcal{Q}(\boldsymbol{\theta} | \boldsymbol{\theta}^{(j)})$  is maximized at  $\boldsymbol{\theta}^{(j+1)}$ . Here, (A5b) follows by using the identity  $\mathcal{H}(\boldsymbol{\theta} | \boldsymbol{\theta}) - \mathcal{H}(\boldsymbol{\theta}' | \boldsymbol{\theta}) = \mathbb{D}(p_{\mathbf{z} | \sigma^2, \mathbf{y}, \boldsymbol{\theta}}(\mathbf{z} | \sigma^2, \mathbf{y}, \boldsymbol{\theta}) \| p_{\mathbf{z} | \sigma^2, \mathbf{y}, \boldsymbol{\theta}}(\mathbf{z} | \sigma^2, \mathbf{y}, \boldsymbol{\theta}'))$ .

*Proof of Theorem 1:* For a given  $\mathbf{q}$ , (16a) is a quadratic function of  $\mathbf{s}$  that is easy to maximize with respect to  $\mathbf{s}$  [see also (10)]:

$$\arg \max_{\mathbf{s}} \mathcal{Q}(\boldsymbol{\theta} | \boldsymbol{\theta}^{(j)}) = [D^{-1}(\mathbf{q}) + I_p]^{-1} \mathbf{z}^{(j)}. \quad (\text{A6})$$

Therefore, the estimates of  $\mathbf{s}$  and  $\mathbf{q}$  obtained upon convergence of the EM iteration in Section III to its fixed point satisfy:

$$\begin{aligned} \mathbf{s}^{(+\infty)} & = \left[ D^{-1}(\mathbf{q}^{(+\infty)}) + I_p \right]^{-1} \mathbf{z}^{(+\infty)} \\ & = \left[ D^{-1}(\mathbf{q}^{(+\infty)}) + I_p \right]^{-1} \\ & \cdot \left[ \mathbf{s}^{(+\infty)} + H^T(\mathbf{y} - H\mathbf{s}^{(+\infty)}) \right] \end{aligned} \quad (\text{A7})$$

where the second equality follows by using (15). Solving (A7) for  $\mathbf{s}^{(+\infty)}$  yields

$$\mathbf{s}^{(+\infty)} = \left[ D^{-1}(\mathbf{q}^{(+\infty)}) + H^T H \right]^{-1} H^T \mathbf{y} \quad (\text{A8})$$

and (18) follows.  $\square$

APPENDIX B  
DERIVATION OF THE MESSAGES AND BELIEFS IN SECTION III-A

Before we proceed, note the following useful identities:

$$\arg \max_{s_i} \mathcal{N}(z_i | s_i, \sigma^2) \mathcal{N}(s_i | 0, \tau^2) = \frac{\tau^2 z_i}{\sigma^2 + \tau^2} \quad (\text{B1a})$$

$$\max_{s_i} \mathcal{N}(z_i | s_i, \sigma^2) \mathcal{N}(s_i | 0, \tau^2) = \frac{\exp \left( -0.5 \frac{z_i^2}{\sigma^2 + \tau^2} \right)}{\sqrt{2\pi\sigma^2} \sqrt{2\pi\tau^2}}. \quad (\text{B1b})$$

#### A. Upward Messages

We can use induction to simplify the multiplicative term  $\prod_{k \in \text{ch}(i)} m_{k \rightarrow i}(q_i)$  in (25) as follows:

$$\prod_{k \in \text{ch}(i)} m_{k \rightarrow i}(q_i) = \left[ \prod_{k \in \text{ch}(i)} \mu_k^u(0) \right]^{1-q_i} \left[ \prod_{k \in \text{ch}(i)} \mu_k^u(1) \right]^{q_i} \quad (\text{B2})$$

see also Fig. 3(a). For leaf nodes  $i$ , (B2) becomes an empty product and is therefore equal to one by convention.

Substituting (B2) into (25) yields

$$\begin{aligned} m_{i \rightarrow \pi(i)}(q_{\pi(i)}) &= \alpha \max_{\boldsymbol{\theta}_i} \\ &\left\{ \mathcal{N}(z_i | s_i, \sigma^2) [\mathcal{N}(s_i | 0, \gamma^2 \sigma^2)]^{q_i} \right. \\ &\cdot [\mathcal{N}(s_i | 0, \epsilon^2 \sigma^2)]^{1-q_i} \left[ \prod_{k \in \text{ch}(i)} \mu_k^u(0) \right]^{1-q_i} \left[ \prod_{k \in \text{ch}(i)} \mu_k^u(1) \right]^{q_i} \\ &\cdot [P_L^{q_i} (1 - P_L)^{1-q_i}]^{1-q_{\pi(i)}} [P_H^{q_i} (1 - P_H)^{1-q_i}]^{q_{\pi(i)}} \left. \right\}. \end{aligned} \quad (\text{B3})$$

For  $q_{\pi(i)} = 0$ , we have

$$\begin{aligned} m_{i \rightarrow \pi(i)}(0) &= \alpha_1 \max \\ &\left\{ (1 - P_L) \left[ \prod_{k \in \text{ch}(i)} \mu_k^u(0) \right] \exp \left( -0.5 \frac{z_i^2}{\sigma^2 + \sigma^2 \epsilon^2} \right) / \epsilon, \right. \\ &\left. P_L \left[ \prod_{k \in \text{ch}(i)} \mu_k^u(1) \right] \exp \left( -0.5 \frac{z_i^2}{\sigma^2 + \sigma^2 \gamma^2} \right) / \gamma \right\} \end{aligned} \quad (\text{B4a})$$

and, for  $q_{\pi(i)} = 1$ , we have

$$\begin{aligned} m_{i \rightarrow \pi(i)}(1) &= \alpha_1 \max \\ &\left\{ (1 - P_H) \left[ \prod_{k \in \text{ch}(i)} \mu_k^u(0) \right] \exp \left( -0.5 \frac{z_i^2}{\sigma^2 + \sigma^2 \epsilon^2} \right) / \epsilon, \right. \\ &\left. P_H \left[ \prod_{k \in \text{ch}(i)} \mu_k^u(1) \right] \exp \left( -0.5 \frac{z_i^2}{\sigma^2 + \sigma^2 \gamma^2} \right) / \gamma \right\} \end{aligned} \quad (\text{B4b})$$

where we have used (B1b) with  $\tau^2 = \sigma^2 \epsilon^2$  and  $\tau^2 = \sigma^2 \gamma^2$  and  $\alpha_1 > 0$  is an appropriate normalizing constant. It follows from (B1b) that the only two candidates of  $\boldsymbol{\theta}_i$  to maximize (B3) are  $[0, \hat{s}_i(0)]^T$  and  $[1, \hat{s}_i(1)]^T$ .

#### B. Downward Messages

Based on the results in Section III-A-1 and Appendix B-I, we simplify the product of upward messages sent from the siblings of node  $i$  in (27) as follows [see (26a)]:

$$\prod_{k \in \text{sib}(i)} m_{k \rightarrow \pi(i)}(q_{\pi(i)}) = \left[ \prod_{k \in \text{sib}(i)} \mu_k^u(0) \right]^{1-q_{\pi(i)}} \cdot \left[ \prod_{k \in \text{sib}(i)} \mu_k^u(1) \right]^{q_{\pi(i)}} \quad (\text{B5})$$

see also Fig. 3(b). Note here that for nodes  $\pi(i) \in \mathcal{T}_{\text{root}}$ , we set the message  $m_{\text{gp}(i) \rightarrow \pi(i)}(q_{\pi(i)})$  to one by convention.

Substituting (B5) into (27) yields

$$\begin{aligned} m_{\pi(i) \rightarrow i}(q_i) &= \alpha \max_{\boldsymbol{\theta}_{\pi(i)}} \left\{ \mathcal{N}(z_{\pi(i)}; s_{\pi(i)}, \sigma^2) \left[ \mu_{\pi(i)}^d(1) \right]^{q_{\pi(i)}} \right. \\ &\cdot \left[ \mu_{\pi(i)}^d(0) \right]^{1-q_{\pi(i)}} \left[ \prod_{k \in \text{sib}(i)} \mu_k^u(0) \right]^{1-q_{\pi(i)}} \\ &\cdot \left[ \prod_{k \in \text{sib}(i)} \mu_k^u(1) \right]^{q_{\pi(i)}} \\ &\cdot [P_H^{q_i} (1 - P_H)^{1-q_i}]^{q_{\pi(i)}} [P_L^{q_i} (1 - P_L)^{1-q_i}]^{1-q_{\pi(i)}} \\ &\cdot \left. [\mathcal{N}(s_{\pi(i)}; 0, \gamma^2 \sigma^2)]^{q_{\pi(i)}} [\mathcal{N}(s_{\pi(i)}; 0, \epsilon^2 \sigma^2)]^{1-q_{\pi(i)}} \right\}. \end{aligned} \quad (\text{B6})$$

For  $q_i = 0$ , we have

$$\begin{aligned} m_{\pi(i) \rightarrow i}(0) &= \alpha_1 \max \\ &\left\{ \mu_{\pi(i)}^d(0) (1 - P_L) \left[ \prod_{k \in \text{sib}(i)} \mu_k^u(0) \right] \right. \\ &\cdot \exp \left( -0.5 \frac{z_{\pi(i)}^2}{\sigma^2 + \sigma^2 \epsilon^2} \right) / \epsilon, \mu_{\pi(i)}^d(1) (1 - P_H) \\ &\cdot \left. \left[ \prod_{k \in \text{sib}(i)} \mu_k^u(1) \right] \exp \left( -0.5 \frac{z_{\pi(i)}^2}{\sigma^2 + \sigma^2 \gamma^2} \right) / \gamma \right\} \end{aligned} \quad (\text{B7a})$$

and for  $q_i = 1$ , we have

$$\begin{aligned} m_{\pi(i) \rightarrow i}(1) &= \alpha_1 \max \\ &\left\{ \mu_{\pi(i)}^d(0) P_L \left[ \prod_{k \in \text{sib}(i)} \mu_k^u(0) \right] \exp \left( -0.5 \frac{z_{\pi(i)}^2}{\sigma^2 + \sigma^2 \epsilon^2} \right) / \epsilon, \right. \\ &\left. \mu_{\pi(i)}^d(1) P_H \left[ \prod_{k \in \text{sib}(i)} \mu_k^u(1) \right] \exp \left( -0.5 \frac{z_{\pi(i)}^2}{\sigma^2 + \sigma^2 \gamma^2} \right) / \gamma \right\} \end{aligned} \quad (\text{B7b})$$

where we have used (B1b) with  $\tau^2 = \sigma^2 \epsilon^2$  and  $\tau^2 = \sigma^2 \gamma^2$  and  $\alpha_1 > 0$  is an appropriate normalizing constant. The only two candidates to maximize (B6) are  $[0, \hat{s}_{\pi(i)}(0)]^T$  and  $[1, \hat{s}_{\pi(i)}(1)]^T$ .

#### C. Beliefs

Define the vector  $\boldsymbol{\beta}_i = [\beta_i(0), \beta_i(1)]^T$  as

$$\beta_i(0) = \max_{s_i} b([0, s_i]^T), \quad \beta_i(1) = \max_{s_i} b([1, s_i]^T) \quad (\text{B8})$$

where  $b(\boldsymbol{\theta}_i)$  are the beliefs defined in (29).

1) *Beliefs for the Root Nodes*: For root nodes  $i \in \mathcal{T}_{\text{root}}$ , the beliefs  $b(\boldsymbol{\theta}_i)$  in (29) become

$$\begin{aligned} b(\boldsymbol{\theta}_i) &= \alpha \mathcal{N}(z_i; s_i, \sigma^2) [P_{\text{root}} \mathcal{N}(s_i; 0, \gamma^2 \sigma^2)]^{q_i} [(1 - P_{\text{root}}) \\ &\cdot \mathcal{N}(s_i; 0, \epsilon^2 \sigma^2)]^{1-q_i} \left[ \prod_{k \in \text{ch}(i)} \mu_k^u(0) \right]^{1-q_i} \left[ \prod_{k \in \text{ch}(i)} \mu_k^u(1) \right]^{q_i}. \end{aligned} \quad (\text{B9})$$

and (B8) simplify to [see (B1b)]  $[\beta_i(0), \beta_i(1)]^T = \alpha_1 [1 - P_{\text{root}}, P_{\text{root}}]^T \odot \boldsymbol{\phi}(z_i) \odot \boldsymbol{\eta}_i^u$ .

2) *Beliefs for the Non-Root Nodes:* For  $i \in (\mathcal{T} \setminus \mathcal{T}_{\text{root}})$ , the beliefs  $b(\theta_i)$  in (29) become

$$\begin{aligned} b(\theta_i) &= \alpha \mathcal{N}(z_i; s_i, \sigma^2) [\mu_i^d(0)]^{1-q_i} [\mu_i^d(1)]^{q_i} \\ &\quad \cdot [\mathcal{N}(s_i; 0, \gamma^2 \sigma^2)]^{q_i} \cdot [\mathcal{N}(s_i; 0, \epsilon^2 \sigma^2)]^{1-q_i} \\ &\quad \cdot \left[ \prod_{k \in \text{ch}(i)} \mu_k^u(0) \right]^{1-q_i} \left[ \prod_{k \in \text{ch}(i)} \mu_k^u(1) \right]^{q_i} \end{aligned} \quad (\text{B10})$$

and (B8) simplify to [see (B1b)]  $[\beta_i(0), \beta_i(1)]^T = \alpha_1 \phi(z_i) \odot \mu_i^d \odot \eta_i^u$ .

Consequently, the mode  $\hat{\theta}_i$  is computed as

$$\hat{\theta}_i = \arg \max_{\theta_i} b(\theta_i) = \begin{cases} [1, \hat{s}_i(1)], & \beta_i(1) \geq \beta_i(0) \\ [0, \hat{s}_i(0)], & \text{otherwise.} \end{cases} \quad (\text{B11})$$

which follows from (B1a).

Note that the normalizing constants  $\alpha$  and  $\alpha_1$  in the above upward and downward messages and beliefs have been set so that  $m_{i \rightarrow \pi(i)}(0) + m_{i \rightarrow \pi(i)}(1) = 1$ ,  $m_{\pi(i) \rightarrow i}(0) + m_{\pi(i) \rightarrow i}(1) = 1$ , and  $\beta_i(0) + \beta_i(1) = 1$  respectively.

## REFERENCES

- [1] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk, "Wavelet-based statistical signal processing using hidden Markov models," *IEEE Trans. Signal Process.*, vol. 46, no. 4, pp. 886–902, Apr. 1998.
- [2] J. K. Romberg, H. Choi, and R. G. Baraniuk, "Bayesian tree-structured image modeling using wavelet-domain hidden markov models," *IEEE Trans. Image Process.*, vol. 10, no. 7, pp. 1056–1068, 2001.
- [3] V. Cevher, P. Indyk, L. Carin, and R. G. Baraniuk, "Sparse signal recovery and acquisition with graphical models," *IEEE Signal Process. Mag.*, vol. 27, no. 6, pp. 92–103, Nov. 2010.
- [4] 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.
- [5] L. He, H. Chen, and L. Carin, "Tree-structured compressive sensing with variational Bayesian analysis," *IEEE Signal Process. Lett.*, vol. 17, no. 3, pp. 233–236, 2010.
- [6] S. Som and P. Schniter, "Compressive imaging using approximate message passing and a Markov-tree prior," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3439–3448, 2012.
- [7] 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.
- [8] Z. Lu, D. Y. Kim, and W. Pearlman, "Wavelet compression of ECG signals by the set partitioning in hierarchical trees algorithm," *IEEE Trans. Biomed. Eng.*, vol. 47, no. 7, pp. 849–856, 2000.
- [9] D. Baron, S. Sarvotham, and R. G. Baraniuk, "Bayesian compressive sensing via belief propagation," *IEEE Trans. Signal Process.*, vol. 58, no. 1, pp. 269–280, 2010.
- [10] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proc. Nat. Acad. Sci.*, vol. 106, no. 45, pp. 18 914–18 919, 2009.
- [11] P. Schniter, "Turbo reconstruction of structured sparse signals," in *Proc. Conf. Inf. Sci. Syst.*, Princeton, NJ, USA, Mar. 2010, pp. 1–6.
- [12] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," ArXiv e-prints, Aug. 2012, arXiv: 1010.5141 [cs.IT] [Online]. Available: <http://arxiv.org/abs/1010.5141>
- [13] M. A. T. Figueiredo and R. D. Nowak, "An EM algorithm for wavelet based image restoration," *IEEE Trans. Image Process.*, vol. 12, pp. 906–916, 2003.
- [14] Z. Song and A. Dogandžić, "A Bayesian max-product EM algorithm for reconstructing structured sparse signals," in *Proc. Conf. Inf. Sci. Syst.*, Princeton, NJ, USA, Mar. 2012, pp. 1–6.
- [15] K. Qiu and A. Dogandžić, "Sparse signal reconstruction via ECME hard thresholding," *IEEE Trans. Signal Process.*, vol. 60, pp. 4551–4569, Sep. 2012.
- [16] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*, 2nd ed. London, U.K.: Chapman & Hall, 2004.
- [17] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.
- [18] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Hoboken, NJ, USA: Wiley, 2006.
- [19] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1993.
- [20] A. Dogandžić, R. Gu, and K. Qiu, "Mask iterative hard thresholding algorithms for sparse image reconstruction of objects with known contour," in *Proc. Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, Nov. 2011, pp. 2111–2116.
- [21] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc. Ser. B*, vol. 39, no. 1, pp. 1–38, 1977, with discussion.
- [22] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*, 2nd ed. New York, NY, USA: Wiley, 2008.
- [23] D. Koller and N. Friedman, *Probabilistic Graphical Models*. Cambridge, MA, USA: MIT Press, 2009.
- [24] Y. Weiss and W. Freeman, "On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 736–744, 2001.
- [25] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA, USA: Morgan Kaufmann, 1988.
- [26] Compressive Imaging Using Turbo AMP. ver. 1.1, Feb. 2013 [Online]. Available: <http://www2.ece.ohio-state.edu/~schniter/turboAMPimaging/>
- [27] Z. Wen, W. Yin, D. Goldfarb, and Y. Zhang, "A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization, and continuation," *SIAM J. Sci. Comput.*, vol. 32, no. 4, pp. 1832–1857, 2010.
- [28] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 586–597, 2007.
- [29] T. Blumensath and M. E. Davies, "Normalized iterative hard thresholding: Guaranteed stability and performance," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 298–309, 2010.
- [30] R. G. Baraniuk, "Optimal tree approximation with wavelets," in *Proc. SPIE Wavelet Appl. Signal Image Process. VII*, Denver, CO, USA, 1999, pp. 196–207.
- [31] Bayesian Compressive Sensing, Aug. 2009 [Online]. Available: <http://people.ee.duke.edu/~lcarin/BCS.html>
- [32] G. Raskutti, M. J. Wainwright, and B. Yu, "Restricted eigenvalue properties for correlated Gaussian designs," *J. Mach. Learn. Res.*, vol. 11, pp. 2241–2259, Aug. 2010.
- [33] M. J. Wainwright, "Sharp thresholds for high-dimensional and noisy sparsity recovery using 1-constrained quadratic programming (Lasso)," *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 2183–2202, 2009.
- [34] 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, 2009.
- [35] F. R. Bach, "Consistency of the group Lasso and multiple kernel learning," *J. Mach. Learn. Res.*, vol. 9, pp. 1179–1225, 2008.
- [36] G. Obozinski, M. J. Wainwright, and M. I. Jordan, "Support union recovery in high-dimensional multivariate regression," *Ann. Statist.*, vol. 39, no. 1, pp. 1–47, 2011.
- [37] N. Nguyen and T. Tran, "Robust Lasso with missing and grossly corrupted observations," *IEEE Trans. Inf. Theory*, vol. 59, no. 4, pp. 2036–2058, 2013.
- [38] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. New York, NY, USA: Springer, 2009.
- [39] G. Varoquaux, A. Gramfort, and B. Thirion, "Small-sample brain mapping: Sparse recovery on spatially correlated designs with randomization and clustering," in *Proc. 29th Int. Conf. Mach. Learn.*, Edinburgh, Scotland, Jun.–July 2012.
- [40] P. Schniter, L. C. Potter, and J. Ziniel, "Fast Bayesian matching pursuit," in *Proc. Worksh. Inf. Theory Appl. (ITA)*, La Jolla, CA, USA, Jan. 2008, pp. 326–333.
- [41] J.-L. Starck, F. Murtagh, and J. M. Fadili, *Sparse Image and Signal Processing: Wavelets, Curvelets, Morphological Diversity*. New York, NY, USA: Cambridge Univ. Press, 2010.
- [42] T. Do, L. Gan, N. Nguyen, and T. Tran, "Fast and efficient compressive sensing using structurally random matrices," *IEEE Trans. Signal Process.*, vol. 60, no. 1, pp. 139–154, 2012.
- [43] D. A. Harville, *Matrix Algebra From a Statistician's Perspective*. New York, NY, USA: Springer-Verlag, 1997.

**Zhao Song**, photograph and biography not available at the time of publication.

**Aleksandar Dogandžić**, photograph and biography not available at the time of publication.