Efficient Message Passing-Based Inference in the Multiple Measurement Vector Problem

Justin Ziniel and Philip Schniter
The Ohio State University, Dept. of E.C.E., Columbus, Ohio 43210; E-mail: {zinielj, schniter}@ece.osu.edu

***Abstract*—In this work, a Bayesian approximate message passing algorithm is proposed for solving the multiple measurement vector (MMV) problem in compressive sensing, in which a collection of sparse signal vectors that share a common support are recovered from undersampled noisy measurements. The algorithm, AMP-MMV, is capable of exploiting temporal correlations in the amplitudes of non-zero coefficients, and provides soft estimates of the signal vectors as well as the underlying support. Central to the proposed approach is an extension of recently developed approximate message passing (AMP) techniques to the amplitude-correlated MMV setting. Aided by these techniques, AMP-MMV offers a computational complexity that is linear in all problem dimensions. In order to allow for automatic parameter tuning, an expectation-maximization algorithm that complements AMP-MMV is described. Finally, a numerical study demonstrates the power of the proposed approach and its particular suitability for application to high-dimensional problems.**

I. Introduction

In this work we consider the *multiple measurement vector* (MMV) problem [1], in which, given T length-M measurement vectors, $\{\boldsymbol{y}^{(t)}\}_{t=1}^{T}$, the objective is to recover a collection of length-N sparse vectors $\{\boldsymbol{x}^{(t)}\}_{t=1}^{T}$, when $M < N$. Each measurement vector is obtained as

$$\boldsymbol{y}^{(t)} = \boldsymbol{A}\boldsymbol{x}^{(t)} + \boldsymbol{e}^{(t)}, \quad t = 1, \ldots, T, \tag{1}$$

where $\boldsymbol{A}$ is a known measurement matrix and $\boldsymbol{e}^{(t)}$ is corrupting additive noise. The unique feature of the MMV problem is the assumption of a common support for each signal vector $\boldsymbol{x}^{(t)}$.

Algorithms developed for the MMV problem are oftentimes intuitive extensions of single measurement vector (SMV) algorithms, and therefore share a similar taxonomy. Among the different techniques that have been proposed are mixed-norm minimization methods [2]–[5], greedy pursuit methods [2], [6], [7], and Bayesian methods [8]–[11]. Also of note are techniques that transform the MMV problem into a block-sparse SMV problem [11], [12]. Existing literature suggests that greedy pursuit techniques are outperformed by mixed-norm minimization approaches, which in turn are surpassed by Bayesian methods [2], [8], [11].

In this paper we provide a high-level overview of a recently proposed [13] algorithm, AMP-MMV, that leverages a novel *approximate message passing* (AMP) [14] framework to perform inference on a probabilistic signal model enforcing joint sparsity of the signal vectors. Ours joins a handful of MMV algorithms that also account for temporal correlations in the amplitudes of the non-zero coefficients (cf. [5], [11]), which we model as Gauss-Markov random processes. Incorporating this temporal correlation structure is crucial, not only because many real-world signals possess such structure, but because the performance of MMV algorithms is particularly sensitive to this structure [8], [11].

Since our probabilistic signal model relies on a set of hyperparameters that may not be known in practice, we describe a principled method of learning all of the hyperparameters from the data using an expectation-maximization (EM) algorithm [15]. Importantly, our EM algorithm makes use of information that has already been obtained in the process of executing AMP-MMV, making the EM procedure highly efficient.

Finally, we present results of a numerical study of AMP-MMV that includes a comparison against an oracle-aided support-aware Kalman smoother (SKS), as well as three state-of-the-art MMV algorithms. This study demonstrates that AMP-MMV performs well under a variety of challenging settings, and that it is especially suitable for application to high-dimensional problems.

II. Signal Model

As noted in Section I, we consider the linear measurement model (1), in which the signal $\boldsymbol{x}^{(t)} \in \mathbb{C}^N$ at timestep t is observed as $\boldsymbol{y}^{(t)} \in \mathbb{C}^M$ through the linear operator $\boldsymbol{A} \in \mathbb{C}^{M \times N}$. We assume $\boldsymbol{e}^{(t)} \sim \mathcal{CN}(\boldsymbol{0}, \sigma_e^2 \boldsymbol{I}_M)$ is circularly symmetric complex white Gaussian noise. We use $\mathcal{S} \triangleq \{n | x_n^{(t)} \neq 0\}$ to denote the indices of the time-invariant support of the signal, which is assumed to be suitably sparse, i.e., $|\mathcal{S}| \leq M$.

We decompose each coefficient $x_n^{(t)}$ as the product of two hidden variables:

$$x_n^{(t)} = s_n \cdot \theta_n^{(t)}, \tag{2}$$

where $s_n \in \{0, 1\}$ is a binary variable that indicates support set membership, and $\theta_n^{(t)} \in \mathbb{C}$ is a variable that provides the amplitude of coefficient $x_n^{(t)}$. When $s_n = 0$, $x_n^{(t)} = 0$ and $n \notin \mathcal{S}$, and when $s_n = 1$, $x_n^{(t)} = \theta_n^{(t)}$ and $n \in \mathcal{S}$. To model the sparsity of the signal, we treat each s_n as a Bernoulli random variable with $\Pr\{s_n = 1\} = \lambda_n < 1$.

In order to model the temporal correlation of signal amplitudes, we treat the evolution of amplitudes over time as stationary first-order Gauss-Markov random processes. Specifically, we assume that $\theta_n^{(t)}$ evolves according to the following linear

Work supported in part by NSF grant CCF-1018368 and DARPA/ONR grant N66001-10-1-4090.

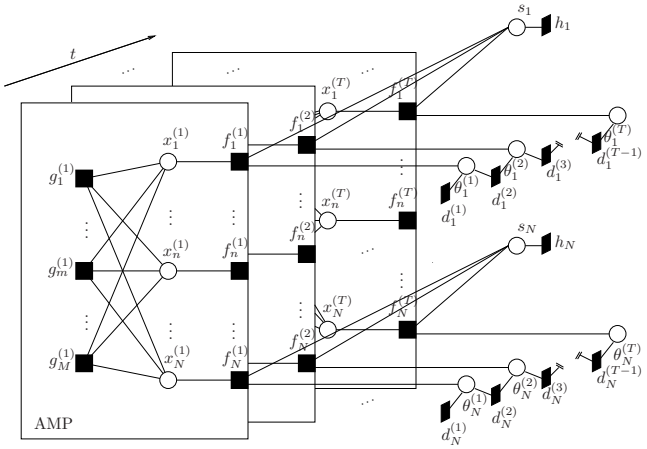

Fig. 1: Factor graph representation of $p(\bar{\boldsymbol{x}}, \bar{\boldsymbol{\theta}}, \boldsymbol{s}|\bar{\boldsymbol{y}})$ in (5).

dynamical system model:

$$\theta_n^{(t)} = (1-\alpha)(\theta_n^{(t-1)} - \zeta) + \alpha w_n^{(t)} + \zeta, \tag{3}$$

where $\zeta \in \mathbb{C}$ is the mean of the amplitude process, $w_n^{(t)} \sim \mathcal{CN}(0, \rho)$ is a circularly symmetric white Gaussian perturbation process, and $\alpha \in [0, 1]$ is a scalar that controls the correlation of $\theta_n^{(t)}$ across time. At one extreme, $\alpha = 0$, the random process is perfectly correlated ($\theta_n^{(t)} = \theta_n^{(t-1)}$), while at the other extreme, $\alpha = 1$, the amplitudes evolve independently over time.

Under our model, the prior distribution of any signal coefficient, $x_n^{(t)}$, is a Bernoulli-Gaussian distribution:

$$p(x_n^{(t)}) = (1-\lambda_n)\delta\big(x_n^{(t)}\big) + \lambda_n \mathcal{CN}\big(x_n^{(t)}; \zeta, \sigma^2\big), \tag{4}$$

where $\delta(\cdot)$ is the Dirac delta function and $\sigma^2 \triangleq \frac{\alpha\rho}{2-\alpha}$ is the steady-state variance of $\theta_n^{(t)}$.

III. The AMP-MMV Algorithm

The statistical structure of the signal model from Section II, which we will exploit, becomes apparent from a factorization of the posterior joint pdf of all random variables. If we define $\bar{\boldsymbol{y}}$ to be the collection of all measurement vectors, $\{\boldsymbol{y}^{(t)}\}_{t=1}^T$, and define $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{\theta}}$ similarly, then the posterior joint distribution factors as follows:

$$p(\bar{\boldsymbol{x}}, \bar{\boldsymbol{\theta}}, \boldsymbol{s}|\bar{\boldsymbol{y}}) \propto \prod_{t=1}^{T} \Bigg(\prod_{m=1}^{M} p(y_m^{(t)}|\boldsymbol{x}^{(t)}) \prod_{n=1}^{N} p(x_n^{(t)}|\theta_n^{(t)}, s_n) \times p(\theta_n^{(t)}|\theta_n^{(t-1)}) \Bigg) \prod_{n=1}^{N} p(s_n), \tag{5}$$

where $\propto$ indicates equality up to a normalizing constant, and $p(\theta_n^{(1)}|\theta_n^{(0)}) \triangleq p(\theta_n^{(1)})$. A convenient graphical representation of this decomposition is given by a *factor graph*, which is an undirected bipartite graph that connects the pdf "factors" of (5) with the variables that make up their arguments. The factor graph for the decomposition of (5) is shown in Fig. 1. For visual clarity, the $\{\theta_n^{(t)}\}_{t=1}^T$ and s_n variable nodes have been removed from the graph for the intermediate index n, but should in fact be present at every index $n = 1, \ldots, N$.

Factor	Distribution	Functional Form
$g_m^{(t)}(\boldsymbol{x}^{(t)})$	$p(y_m^{(t)}\|\boldsymbol{x}^{(t)})$	$\mathcal{CN}(\boldsymbol{a}_m^T\boldsymbol{x}^{(t)}, \sigma_e^2)$
$f_n^{(t)}(x_n^{(t)}, s_n, \theta_n^{(t)})$	$p(x_n^{(t)}\|s_n, \theta_n^{(t)})$	$\delta(x_n^{(t)} - s_n\theta_n^{(t)})$
$h_n(s_n)$	$p(s_n)$	$(1-\lambda_n)^{(1-s_n)}(\lambda_n)^{s_n}$
$d_n^{(1)}(\theta_n^{(1)})$	$p(\theta_n^{(1)})$	$\mathcal{CN}(\zeta, \sigma^2)$
$d_n^{(t)}(\theta_n^{(t)}, \theta_n^{(t-1)})$	$p(\theta_n^{(t)}\|\theta_n^{(t-1)})$	$\mathcal{CN}((1-\alpha)\theta_n^{(t-1)}+\alpha\zeta, \alpha^2\rho)$

TABLE I: The factors, underlying distributions, and functional forms associated with the signal model of Section II.

The factor nodes in Fig. 1 have all been assigned alphabetic labels for clarity. The correspondence between these factor labels, the underlying distributions, and the functional form of the distributions is presented in Table I.

Our approach to performing inference on the factor graph of Fig. 1 is based on belief propagation [16], which, in cycle-free graphs, is an instance of the sum-product algorithm. When the factor graph contains cycles, the same rules that define the sum-product algorithm can still be applied, however convergence to the correct posterior marginal distributions is no longer guaranteed. Despite this, loopy belief propagation has been successfully applied to many problems, including Markov random field inference, LDPC decoding, and compressive sensing [13]. In what follows, we use $\nu_{a\to b}(\cdot)$ to denote a message that is passed from node a to a connected node b.

A. Message Scheduling

Since the factor graph of Fig. 1 contains many cycles, there are a number of valid ways to schedule, or sequence, the messages that are exchanged in the graph. In this work, we make use of an intuitive decomposition of message scheduling into four distinct phases, which could be ordered in a number of different ways to enable, amongst others, causal filtering and non-causal smoothing [13]. We label each phase using the mnemonics **(into)**, **(within)**, **(out)**, and **(across)**.

To aid our discussion, Fig. 2 summarizes each of the four phases. Arrows indicate the direction that messages are moving, and only those nodes and edges participating in a particular phase are shown in that phase. For the **(across)** phase we show messages being passed forward in time, and omit the backwards pass. The figure also introduces the notation that we adopt for the different variables that serve to parameterize the messages. Certain variables, e.g., $\overrightarrow{\eta}_n^{(t)}$ and $\overleftarrow{\eta}_n^{(t)}$, are accented with directional arrows to differentiate messages moving in opposite directions along the same edge.

In phase **(into)**, messages are passed from the s_n and $\theta_n^{(t)}$ variable nodes *into* frame t. Loosely speaking, these messages convey current beliefs about the values of $\boldsymbol{s}$ and $\boldsymbol{\theta}^{(t)}$. In phase **(within)**, messages are exchanged *within* frame t, producing an estimate of $\boldsymbol{x}^{(t)}$ using the current beliefs about $\boldsymbol{s}$ and $\boldsymbol{\theta}^{(t)}$ together with the available measurements $\boldsymbol{y}^{(t)}$. In phase **(out)**, the estimate of $\boldsymbol{x}^{(t)}$ is used to refine the beliefs about $\boldsymbol{s}$ and $\boldsymbol{\theta}^{(t)}$ by passing messages *out* of frame t. Finally, in phase **(across)**, messages are sent from $\theta_n^{(t)}$ to either $\theta_n^{(t+1)}$ or $\theta_n^{(t-1)}$, thus conveying information *across* time about temporal correlation in the signal amplitudes. Upon choosing an ordering of these phases, messages are exchanged until

either convergence occurs, or a maximum number of allowable iterations is reached.

B. Implementing the Message Passes

Most of the messages can be derived by applying the rules of the sum-product algorithm. In this sub-section, we focus on a handful of messages in the **(within)** phase whose implementation requires a departure from these rules.

Inspection of Fig. 2(b) reveals a dense interconnection between the $\{x_n^{(t)}\}$ and $\{g_m^{(t)}\}$ nodes. Applying standard sum-product rules would result in an algorithm that required the evaluation of multi-dimensional integrals that grew exponentially in number in both N and M, which is clearly computationally infeasible for meaningful problem sizes. Instead, we turn to a recently developed framework known as *approximate message passing* (AMP).

A complete description of AMP is beyond the scope of this work, and we refer the interested reader to [14], [17]. For the purposes of this discussion, we simply note that AMP is an efficient means of performing inference on the factor graph in Fig. 2(b), given generic signal priors, and is specified by steps (A4) - (A8) in Table II. A recent theoretical analysis of AMP [18] shows that in the large-system limit (i.e., M, $N \to \infty$ with M/N fixed), the behavior of AMP is governed by a state evolution whose fixed points, when unique, correspond to minimum mean square error (MMSE) signal estimates.

From AMP's viewpoint, $\nu_{f_n^{(t)} \to x_n^{(t)}}(\cdot)$ is the "prior distribution" for $x_n^{(t)}$, which takes the Bernoulli-Gaussian form

$$\nu_{f_n^{(t)} \to x_n^{(t)}}(x_n^{(t)}) = (1-\overleftarrow{\pi}_n^{(t)})\delta(x_n^{(t)}) + \overleftarrow{\pi}_n^{(t)} \mathcal{CN}(x_n^{(t)}; \overleftarrow{\xi}_n^{(t)}, \overleftarrow{\psi}_n^{(t)}). \tag{6}$$

This "prior" determines the AMP soft-thresholding functions defined in (D1) - (D4) of Table II. The derivation of these thresholding functions closely follows those outlined in [19], which considered the special case of a zero-mean Bernoulli-Gaussian prior.

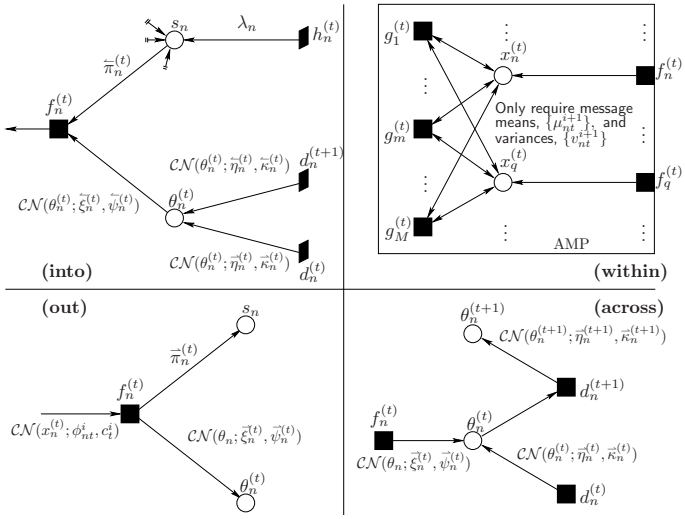

Fig. 2: A summary of the four message passing phases, including message notation and form.

% Define soft-thresholding functions:

$$F_{nt}(\phi;c) \triangleq (1+\gamma_{nt}(\phi;c))^{-1}\Big(\frac{\overleftarrow{\psi}_n^{(t)}\phi + \overleftarrow{\xi}_n^{(t)} c}{\overleftarrow{\psi}_n^{(t)} + c}\Big) \tag{D1}$$

$$G_{nt}(\phi;c) \triangleq (1+\gamma_{nt}(\phi;c))^{-1}\Big(\frac{\overleftarrow{\psi}_n^{(t)} c}{\overleftarrow{\psi}_n^{(t)} + c}\Big) + \gamma_{nt}(\phi;c)|F_n(\phi;c)|^2 \tag{D2}$$

$$F'_{nt}(\phi;c) \triangleq \tfrac{\partial}{\partial\phi}F_{nt}(\phi,c) = \tfrac{1}{c}G_{nt}(\phi;c) \tag{D3}$$

$$\gamma_{nt}(\phi;c) \triangleq \Big(\frac{1-\overleftarrow{\pi}_n^{(t)}}{\overleftarrow{\pi}_n^{(t)}}\Big)\Big(\frac{\overleftarrow{\psi}_n^{(t)}+c}{c}\Big) \times \exp\Big(-\Big[\frac{\overleftarrow{\psi}_n^{(t)}|\phi|^2 + \overleftarrow{\xi}_n^{(t)*}c\phi + \overleftarrow{\xi}_n^{(t)}c\phi^* - c|\overleftarrow{\xi}_n^{(t)}|^2}{c(\overleftarrow{\psi}_n^{(t)}+c)}\Big]\Big) \tag{D4}$$

% Begin passing messages . . .

for $t = 1, \ldots, T$, $\forall n$:

% Execute the **(into)** phase . . .

$$\overleftarrow{\pi}_n^{(t)} = \frac{\lambda_n \cdot \Pi_{t' \neq t} \overrightarrow{\pi}_n^{(t')}}{(1-\lambda_n)\cdot\Pi_{t'\neq t}(1-\overrightarrow{\pi}_n^{(t')}) + \lambda_n \cdot \Pi_{t'\neq t}\overrightarrow{\pi}_n^{(t')}} \tag{A1}$$

$$\overleftarrow{\psi}_n^{(t)} = \frac{\overrightarrow{\kappa}_n^{(t)}\cdot\overleftarrow{\kappa}_n^{(t)}}{\overrightarrow{\kappa}_n^{(t)} - \overleftarrow{\kappa}_n^{(t)}} \tag{A2}$$

$$\overleftarrow{\xi}_n^{(t)} = \overleftarrow{\psi}_n^{(t)} \cdot \Big(\frac{\overrightarrow{\eta}_n^{(t)}}{\overrightarrow{\kappa}_n^{(t)}} + \frac{\overleftarrow{\eta}_n^{(t)}}{\overleftarrow{\kappa}_n^{(t)}}\Big) \tag{A3}$$

% Initialize AMP-related variables . . .

$\forall m : z_{mt}^1 = y_m^{(t)}, \forall n : \mu_{nt}^1 = 0$, and $c_t^1 = 100 \cdot \sum_{n=1}^N \psi_n^{(t)}$

% Execute the **(within)** phase using AMP . . .

for $i = 1, \ldots, I, \forall n, m$:

$$\phi_{nt}^i = \sum_{m=1}^M A_{mn}^* z_{mt}^i + \mu_{nt}^i \tag{A4}$$

$$\mu_{nt}^{i+1} = F_{nt}(\phi_{nt}^i; c_t^i) \tag{A5}$$

$$v_{nt}^{i+1} = G_{nt}(\phi_{nt}^i; c_t^i) \tag{A6}$$

$$c_t^{i+1} = \sigma_e^2 + \tfrac{1}{M}\sum_{n=1}^N v_{nt}^{i+1} \tag{A7}$$

$$z_{mt}^{i+1} = y_m^{(t)} - \boldsymbol{a}_m^T \boldsymbol{\mu}_t^{i+1} + \tfrac{z_{mt}^i}{M}\sum_{n=1}^N F'_{nt}(\phi_{nt}^i; c_t^i) \tag{A8}$$

end

$$\hat{x}_n^{(t)} = \mu_{nt}^{I+1} \quad \text{\% Store current estimate of } x_n^{(t)} \tag{A9}$$

% Execute the **(out)** phase . . .

$$\overrightarrow{\pi}_n^{(t)} = \Big(1 + \Big(\frac{\overleftarrow{\pi}_n^{(t)}}{1-\overleftarrow{\pi}_n^{(t)}}\Big)\gamma_{nt}(\phi_{nt}^I; c_t^{I+1})\Big)^{-1} \tag{A10}$$

$$(\overrightarrow{\xi}_n^{(t)}, \overrightarrow{\psi}_n^{(t)}) = \text{taylor\_approx}(\overleftarrow{\pi}_n^{(t)}, \phi_{nt}^I, c_t^I) \tag{A11}$$

% Execute the **(across)** phase from $\theta_n^{(t)}$ to $\theta_n^{(t+1)}$ . . .

$$\overrightarrow{\eta}_n^{(t+1)} = (1-\alpha)\Big(\frac{\overrightarrow{\kappa}_n^{(t)}\overrightarrow{\psi}_n^{(t)}}{\overrightarrow{\kappa}_n^{(t)} + \overrightarrow{\psi}_n^{(t)}}\Big)\Big(\frac{\overrightarrow{\eta}_n^{(t)}}{\overrightarrow{\kappa}_n^{(t)}} + \frac{\overrightarrow{\xi}_n^{(t)}}{\overrightarrow{\psi}_n^{(t)}}\Big) + \alpha\zeta \tag{A12}$$

$$\overrightarrow{\kappa}_n^{(t+1)} = (1-\alpha)^2\Big(\frac{\overrightarrow{\kappa}_n^{(t)}\overrightarrow{\psi}_n^{(t)}}{\overrightarrow{\kappa}_n^{(t)} + \overrightarrow{\psi}_n^{(t)}}\Big) + \alpha^2\rho \tag{A13}$$

end

TABLE II: Message update equations for a "serial" configuration of the four message passing phases.

For convenience, we summarize the message update equations in Table II, where we provide a pseudocode implementation of AMP-MMV. Inspection of the pseudocode reveals that the overall per-iteration complexity is linear in all problem dimensions, that is, $\mathcal{O}(TNM)$ flops, reflecting the substantial complexity reduction that comes from AMP.

IV. Estimating the Model Parameters

In order to learn the model prior parameters $\{\lambda_n\}_{n=1}^N$, ζ, α, ρ, and σ_e^2, we develop an expectation-maximization (EM) algorithm [15] that couples with the message passing procedure described in Section III-A to provide a means of learning all of the model parameters while simultaneously estimating the signal $\bar{\boldsymbol{x}}$ and its support $\boldsymbol{s}$.

The EM algorithm is an appealing choice for performing parameter estimation for several reasons. First and foremost, the EM algorithm is a well-studied and principled means of parameter estimation, offering provable convergence to a local maximum of the likelihood function [15]. Second, it is an iterative algorithm, and thus pairs naturally with concurrent iterations of AMP-MMV. Finally, the expectation step of the EM algorithm relies on quantities that have already been

% Define key quantities obtained from AMP-MMV at iteration k:

$$\mathrm{E}[s_n|\bar{\boldsymbol{y}}] = \frac{\lambda_n \prod_{t=1}^T \overrightarrow{\pi}_n^{(t)}}{\lambda_n \prod_{t=1}^T \overrightarrow{\pi}_n^{(t)} + (1-\lambda_n)\prod_{t=1}^T (1-\overrightarrow{\pi}_n^{(t)})} \quad \text{(Q1)}$$

$$\tilde{v}_n^{(t)} \triangleq \mathrm{var}\{\theta_n^{(t)}|\bar{\boldsymbol{y}}\} = \left(\frac{1}{\overrightarrow{\kappa}_n^{(t)}} + \frac{1}{\overrightarrow{\psi}_n^{(t)}} + \frac{1}{\overleftarrow{\kappa}_n^{(t)}}\right)^{-1} \quad \text{(Q2)}$$

$$\tilde{\mu}_n^{(t)} \triangleq \mathrm{E}[\theta_n^{(t)}|\bar{\boldsymbol{y}}] = \tilde{v}_n^{(t)} \cdot \left(\frac{\overrightarrow{\eta}_n^{(t)}}{\overrightarrow{\kappa}_n^{(t)}} + \frac{\overrightarrow{\xi}_n^{(t)}}{\overrightarrow{\psi}_n^{(t)}} + \frac{\overleftarrow{\eta}_n^{(t)}}{\overleftarrow{\kappa}_n^{(t)}}\right) \quad \text{(Q3)}$$

$v_n^{(t)} \triangleq \mathrm{var}\{x_n^{(t)}|\bar{\boldsymbol{y}}\}$ % See (A6) of Table II

$\mu_n^{(t)} \triangleq \mathrm{E}[x_n^{(t)}|\bar{\boldsymbol{y}}]$ % See (A5) of Table II

% EM update equations:

$$\lambda^{k+1} = \tfrac{1}{N}\textstyle\sum_{n=1}^N \mathrm{E}[s_n|\bar{\boldsymbol{y}}] \quad \text{(E1)}$$

$$\zeta^{k+1} = \left(\tfrac{N(T-1)}{\rho^k} + \tfrac{N}{(\sigma^2)^k}\right)^{-1}\left(\tfrac{1}{(\sigma^2)^k}\textstyle\sum_{n=1}^N \tilde{\mu}_n^{(1)} + \sum_{t=2}^T \sum_{n=1}^N \tfrac{1}{\alpha^k \rho^k}\left(\tilde{\mu}_n^{(t)} - (1-\alpha^k)\tilde{\mu}_n^{(t-1)}\right)\right) \quad \text{(E2)}$$

$$\alpha^{k+1} = \tfrac{1}{4N(T-1)}\left(\mathfrak{b} - \sqrt{\mathfrak{b}^2 + 8N(T-1)\mathfrak{c}}\right) \quad \text{(E3)}$$

where:

$$\mathfrak{b} \triangleq \tfrac{2}{\rho^k}\textstyle\sum_{t=2}^T \sum_{n=1}^N \mathfrak{Re}\{\mathrm{E}[\theta_n^{(t)*}\theta_n^{(t-1)}|\bar{\boldsymbol{y}}]\} - \mathfrak{Re}\{(\tilde{\mu}_n^{(t)} - \tilde{\mu}_n^{(t-1)})^*\zeta^k\} - \tilde{v}_n^{(t-1)} - |\tilde{\mu}_n^{(t-1)}|^2$$

$$\mathfrak{c} \triangleq \tfrac{2}{\rho^k}\textstyle\sum_{t=2}^T \sum_{n=1}^N \tilde{v}_n^{(t)} + |\tilde{\mu}_n^{(t)}|^2 + \tilde{v}_n^{(t-1)} + |\tilde{\mu}_n^{(t-1)}|^2 - 2\mathfrak{Re}\{\mathrm{E}[\theta_n^{(t)*}\theta_n^{(t-1)}|\bar{\boldsymbol{y}}]\}$$

$$\rho^{k+1} = \tfrac{1}{(\alpha^k)^2 N(T-1)}\textstyle\sum_{t=2}^T \sum_{n=1}^N \tilde{v}_n^{(t)} + |\tilde{\mu}_n^{(t)}|^2 + (\alpha^k)^2|\zeta^k|^2 - 2(1-\alpha^k)\mathfrak{Re}\{\mathrm{E}[\theta_n^{(t)*}\theta_n^{(t-1)}|\bar{\boldsymbol{y}}]\} - 2\alpha^k\mathfrak{Re}\{\tilde{\mu}_n^{(t)*}\zeta^k\} + 2\alpha^k(1-\alpha^k)\mathfrak{Re}\{\tilde{\mu}_n^{(t-1)*}\zeta^k\} + (1-\alpha^k)(\tilde{v}_n^{(t-1)} + |\tilde{\mu}_n^{(t-1)}|^2) \quad \text{(E4)}$$

$$\sigma_e^{2\ k+1} = \tfrac{1}{TM}\left(\textstyle\sum_{t=1}^T \|\boldsymbol{y}^{(t)} - \boldsymbol{A}\boldsymbol{\mu}^{(t)}\|^2 + \mathbf{1}_N^T \boldsymbol{v}^{(t)}\right) \quad \text{(E5)}$$

TABLE III: EM algorithm update equations for the signal model parameters of Section II.

computed in the process of executing AMP-MMV, and so the EM procedure is highly efficient.

We let $\Gamma \triangleq \{\lambda, \zeta, \alpha, \rho, \sigma_e^2\}$ denote the set of all model parameters, and let Γ^k denote the set of parameter estimates at the k^{th} EM iteration. Here we have assumed that the binary support indicator variables share a common activity probability, λ, i.e., $\Pr\{s_n = 1\} = \lambda\ \forall n$. For all parameters except σ_e^2, we use $\boldsymbol{s}$ and $\bar{\boldsymbol{\theta}}$ as the so-called "missing" data of the EM algorithm, while for σ_e^2 we use $\bar{\boldsymbol{x}}$.

After an initial iteration of AMP-MMV, approximate marginal posterior distributions are available for each of the underlying random variables, e.g., $p(s_n|\bar{\boldsymbol{y}})$, along with pairwise joint posterior distributions, e.g., $p(\theta_n^{(t)}, \theta_n^{(t-1)}|\bar{\boldsymbol{y}})$. With these distributions, it is possible to perform the iterative expectation and maximization steps required to maximize $p(\bar{\boldsymbol{y}}|\Gamma)$ in closed-form. We adopt a Gauss-Seidel scheme, performing coordinate-wise maximization, e.g.,

$$\lambda^{k+1} = \underset{\lambda}{\mathrm{argmax}}\, \mathrm{E}_{\boldsymbol{s},\bar{\boldsymbol{\theta}}|\bar{\boldsymbol{y}}}\left[\log p(\bar{\boldsymbol{y}}, \boldsymbol{s}, \bar{\boldsymbol{\theta}})\big|\bar{\boldsymbol{y}}, \lambda, \Gamma^k \backslash \{\lambda^k\}\right],$$

where k is the iteration index common to both AMP-MMV and the EM algorithm. In Table III we provide the EM parameter update equations for our model.

V. Numerical Study

In this section we present a limited summary of an extensive numerical study that was conducted [13] to explore the performance characteristics and tradeoffs of AMP-MMV. MATLAB® code was written[1] to implement the algorithm described in Section III, along with the EM parameter estimation procedure of Section IV.

[1]Code available at ece.osu.edu/~schniter/turboAMPmmv.

For comparison to AMP-MMV, we tested two other Bayesian algorithms for the MMV problem, MSBL [8] and T-MSBL[2] [11], along with a recently proposed greedy algorithm designed specifically for highly correlated signals, subspace-augmented MUSIC[3] (SA-MUSIC) [7]. We also implemented an oracle-aided support-aware Kalman smoother (SKS), which provides a lower bound on the achievable MSE of any algorithm.

Three performance metrics were considered throughout our tests. The first metric, which we refer to as the time-averaged normalized MSE (TNMSE), is defined as $\mathrm{TNMSE}(\bar{\boldsymbol{x}}, \hat{\bar{\boldsymbol{x}}}) \triangleq \frac{1}{T}\sum_{t=1}^T \|\boldsymbol{x}^{(t)} - \hat{\boldsymbol{x}}^{(t)}\|_2^2 / \|\boldsymbol{x}^{(t)}\|_2^2$, where $\hat{\boldsymbol{x}}^{(t)}$ is an estimate of $\boldsymbol{x}^{(t)}$. The second metric is the normalized support error rate (NSER), defined as $\mathrm{NSER}(\mathcal{S}, \hat{\mathcal{S}}) \triangleq \left(|\mathcal{S} \setminus \hat{\mathcal{S}}| + |\hat{\mathcal{S}} \setminus \mathcal{S}|\right)/|\mathcal{S}|$, where $\mathcal{S}$ and $\hat{\mathcal{S}}$ are the sets of true and estimated supports, respectively. The third and final metric is runtime, which is an important metric given the prevalence of high-dimensional datasets.

In Fig. 3, we plot our three metrics as a function of the measurements-to-active-coefficients ratio, M/K, where $K \triangleq |\mathcal{S}|$. Simulation details are provided in plot titles. For AMP-MMV, two traces appear on the NSER plot, with the ◯ marker corresponding to a K-aware support estimation method used by both MSBL and T-MSBL, and the △ marker corresponding to a K-agnostic support estimate obtained from AMP-MMV's posteriors $p(s_n|\bar{\boldsymbol{y}})$. We see that, when $M/K \geq 2$, the TNMSE performance of both AMP-MMV and T-MSBL is almost identical to that of the oracle-aided SKS. However, when $M/K < 2$, every algorithm's support estimation performance (NSER) degrades, and the TNMSE consequently grows. Indeed, when $M/K < 1.50$, all of the algorithms perform poorly compared to the SKS, although T-MSBL performs the best of the four. We also note the superior NSER performance of AMP-MMV over much of the range. From the runtime plot we see the tremendous efficiency of AMP-MMV. Over the region in which AMP-MMV is performing well, we see that its runtime is more than one order-of-magnitude faster than SA-MUSIC, and two orders-of-magnitude faster than either T-MSBL or MSBL.

A key consideration of our method is ensuring that it is suitable for high-dimensional problems. Our complexity analysis indicated that a single iteration of AMP-MMV could be completed in $\mathcal{O}(TNM)$ flops. However, to verify that it scales well with problem size, we performed an experiment in which the signal dimension, N, was swept logarithmically over the range $[100, 10000]$, and M was scaled proportionally such that $N/M = 3$.

The results of this experiment are provided in Fig. 4. Several features of these plots are of interest. First, we observe that the performance of every algorithm improves noticeably as problem dimensions grow from $N = 100$ to $N = 1000$, with AMP-MMV and T-MSBL converging in TNMSE performance to the SKS bound. The second observation that we point out

[2]Code available at dsp.ucsd.edu/~zhilin/Software.html.

[3]Code obtained through personal correspondence with authors.

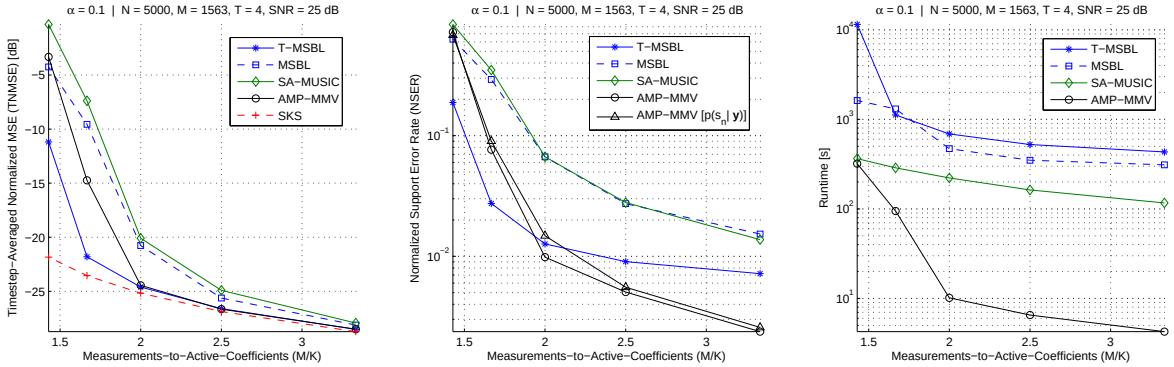

Fig. 3: A plot of the NSER, TNMSE (in dB), and runtime of T-MSBL, MSBL, AMP-MMV, and the SKS versus M/K. Correlation coefficient $1-\alpha = 0.90$.

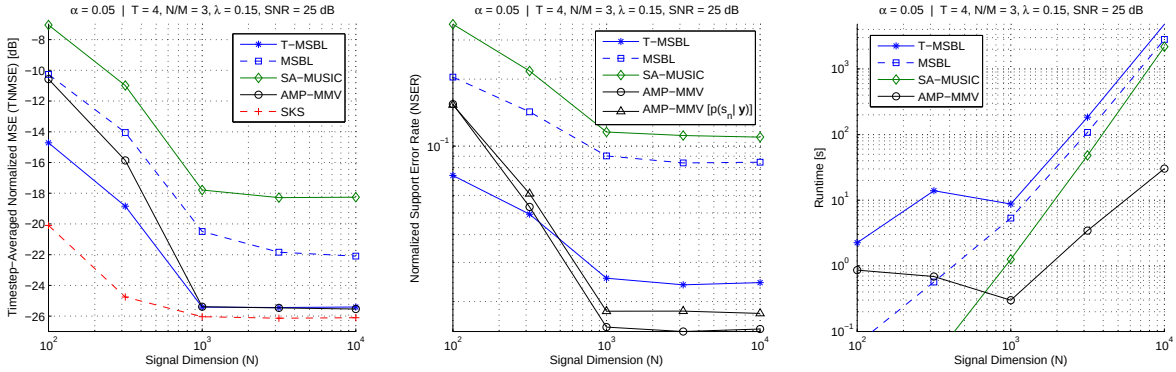

Fig. 4: A plot of the NSER, TNMSE (in dB), and runtime of T-MSBL, MSBL, AMP-MMV, and the SKS versus signal dimension, N. Correlation coefficient $1-\alpha = 0.95$.

is that AMP-MMV is extremely fast. Indeed, a problem with $NT = 40000$ unknowns can be solved accurately in just under 30 seconds. Finally, we note that AMP-MMV scales with increasing problem dimensions more favorably than the other methods; at $N = 10000$, AMP-MMV runs two orders-of-magnitude faster than the other techniques.

REFERENCES

[1] B. D. Rao and K. Kreutz-Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors," in *IEEE Digital Signal Process. Workshop*, (Bryce Canyon, UT), June 1998.

[2] S. F. Cotter, B. D. Rao, K. Engan, and K. Kreutz-Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors," *IEEE Trans. Signal Process.*, vol. 53, pp. 2477–2488, Jul. 2005.

[3] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation. Part II: Convex relaxation," *Signal Processing*, vol. 86, pp. 589–602, Apr. 2006.

[4] M. M. Hyder and K. Mahata, "A robust algorithm for joint sparse recovery," *IEEE Signal Process. Lett.*, vol. 16, pp. 1091–1094, Dec. 2009.

[5] Z. Zhang and B. D. Rao, "Iterative reweighted algorithms for sparse signal recovery with temporally correlated source vectors," in *IEEE Int'l Conf. Acoust., Speech & Signal Process. (ICASSP)*, (Prague, Czech Republic), pp. 3932–3935, May 2011.

[6] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit," *Signal Processing*, vol. 86, pp. 572–588, Apr. 2006.

[7] K. Lee, Y. Bresler, and M. Junge, "Subspace methods for joint sparse recovery." arXiv:1004.3071, 2010.

[8] D. P. Wipf and B. D. Rao, "An empirical Bayesian strategy for solving the simultaneous sparse approximation problem," *IEEE Trans. Signal Process.*, vol. 55, pp. 3704–3716, Jul. 2007.

[9] G. Tzagkarakis, D. Milioris, and P. Tsakalides, "Multiple-measurement Bayesian compressed sensing using GSM priors for DOA estimation," in *IEEE Int'l Conf. Acoust., Speech & Signal Process. (ICASSP)*, (Dallas, TX), Mar. 2010.

[10] S. Shedthikere and A. Chockalingam, "Bayesian framework and message passing for joint support and signal recovery of approximately sparse signals," in *IEEE Int'l Conf. Acoust., Speech & Signal Process. (ICASSP)*, (Prague, Czech Republic), pp. 4032–4035, May 2011.

[11] Z. Zhang and B. D. Rao, "Sparse signal recovery with temporally correlated source vectors using Sparse Bayesian Learning," *IEEE J. Selected Topics Signal Process.*, vol. 5, pp. 912–926, Sept. 2011.

[12] Y. Eldar and M. Mishali, "Robust recovery of signals from a structured union of subspaces," *IEEE Trans. Inform. Theory*, vol. 55, pp. 5302–5316, Nov. 2009.

[13] J. Ziniel and P. Schniter, "Efficient high-dimensional inference in the multiple measurement vector problem." arXiv:1111.5272 [cs.IT], Nov. 2011.

[14] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing," in *Proc. Nat'l Acad. Sciences*, vol. 106, pp. 18914–18919, Nov. 2009.

[15] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc., B*, vol. 39, pp. 1–38, 1977.

[16] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*. San Mateo, CA: Morgan Kaufman, 1988.

[17] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing: I. Motivation and construction," in *Proc. Inform. Theory Workshop*, Jan. 2010.

[18] M. Bayati and A. Montanari, "The dynamics of message passing on dense graphs, with applications to compressed sensing," *IEEE Trans. Inform. Theory*, vol. 57, pp. 764–785, Feb. 2011.

[19] P. Schniter, "Turbo reconstruction of structured sparse signals," in *Conf. on Inform. Sciences & Syst. (CISS)*, pp. 1 –6, Mar. 2010.