SUPPORT AGNOSTIC BAYESIAN RECOVERY OF JOINTLY SPARSE SIGNALS

Mudassir Masood* *and Tareq Y. Al-Naffouri**§

* King Abdullah Univ. of Science & Technology
Department of Electrical Engineering
Thuwal 23955-6900, Makkah, Saudi Arabia

§ King Fahd Univ. of Petroleum & Minerals
Department of Electrical Engineering
Dhahran 31261, Saudi Arabia

ABSTRACT

A matching pursuit method using a Bayesian approach is introduced for recovering a set of sparse signals with common support from a set of their measurements. This method performs Bayesian estimates of joint-sparse signals even when the distribution of active elements is not known. It utilizes only the *a priori* statistics of noise and the sparsity rate of the signal, which are estimated without user intervention. The method utilizes a greedy approach to determine the approximate MMSE estimate of the joint-sparse signals. Simulation results demonstrate the superiority of the proposed estimator.

1. INTRODUCTION

The problem addressed by compressed sensing and sparse recovery algorithms is to recover an unknown sparse vector $\mathbf{x} \in \mathbb{C}^{N\times 1}$ from single measurement vector (SMV) $\mathbf{y} \in \mathbb{C}^{M\times 1}$ where $M < N$. A closely related problem is to jointly recover multiple unknown sparse vectors having same support from multiple measurement vectors (MMV). This problem could be viewed as recovering an unknown row-sparse matrix $\mathbf{X} \in \mathbb{C}^{N\times L}$ from an observation matrix $\mathbf{Y} \in \mathbb{C}^{M\times L}$. Some of the applications where multiple observations could be utilized include sparse channel equalization [1], massive MIMO [2,3], blind source separation [4], imaging of brain using magnetoencephalography (MEG) and electroencephalography (EEG) [5], respiratory movement tracking [6] and multivariate regression [7].

Several algorithms have been proposed taking into account the case of multiple measurement vectors. Some of the recent works include that done by Rao *et al* [8, 9] while others include simultaneous OMP (S-OMP) [10], MMV-OMP (M-OMP) and MMV-focal underdetermined system solver (M-FOCUSS) [11] to name a few. Another algorithm called the ReMBo algorithm [12] follows a fundamentally different strategy. It proposes to model the MMV problem as an SMV problem. To do so, the algorithm linearly combines the measurement vectors and then solve the resulting SMV problem using a predetermined algorithm. Another class of algorithms exploit the properties of the unknown sparse signals such as correlation and structure. For example, orthogonal subspace matching pursuit (OSMP) and subspace-augmented multiple signal classification (SA-MUSIC) algorithms proposed by Lee *et al* [13] utilize some of the inherent properties of the unknown signals for recovery.

Most of these algorithms belong to the category of convex relaxation algorithms which are agnostic to support distribution[1] and hence demonstrate robust performance. Algorithms considering the problem of Bayesian MMV support recovery are not as common. Notable exceptions are M-SBL and AR-SBL based on sparse Bayesian learning algorithm (SBL). These algorithms assume Gaussian prior over the non-zero elements of the unknown sparse signals.

The focus of the present paper is on developing a Bayesian approach for sparse signal recovery using multiple observations. Specifically, we pursue a Bayesian approach similar to that proposed in [14, 15] that combines the advantages of the two approaches summarized above. On the one hand, the approach is Bayesian, acknowledging the noise statistics and the signal sparsity rate, while on the other hand, the approach is agnostic to the signal support statistics (making it especially useful when these statistics are unknown or non-Gaussian). Specifically, our approach provides the following advantages: (i) it provides Bayesian estimates of the sparse signals even when the signal support prior is non-Gaussian or unknown. (ii) it is agnostic to the support distribution and so the parameters of this distribution whether Gaussian or not, need not be estimated. This is particularly useful when the signal support priors are not i.i.d. (iii) it features the capability of sensing supports by different sensing matrices which unlike other contemporary algorithms allows us to capture the signal information in a much better manner. (iv) it utilizes the prior Gaussian statistics of the additive noise and the sparsity rate of the signals and is able to estimate these in a robust manner from the data. (v) it enjoys low complexity thanks to its greedy approach and the order-recursive update of its metrics.

This work was supported by SABIC through an internally funded project from DSR, KFUPM (Project No. SB101006).

[1] In the paper we use the term support distribution to refer to the distribution of the active elements of the unknown signal $\mathbf{x}$.

2. BAYESIAN SETUP FOR SUPPORT AGNOSTIC JOINTLY SPARSE SIGNAL RECONSTRUCTION

In this paper we will consider the estimation of a row-sparse matrix, $\mathbf{X} \in \mathbb{C}^{N \times L}$, from multiple observation vectors represented as a matrix $\mathbf{Y} \in \mathbb{C}^{M \times L}$, obeying the linear regression,

$$\mathbf{Y} = \mathbf{\Phi}\mathbf{X} + \mathbf{W}. \tag{1}$$

Here $\mathbf{\Phi} \in \mathbb{C}^{M \times N}$ is a known regression/sensing matrix[2] and $\mathbf{W}$ is a matrix representing a collection of additive white Gaussian noise vectors following $\mathcal{CN}(\mathbf{0}, \sigma^2_{\mathbf{w}}\mathbf{I}_M)$. To assist in algorithm development in the following, we will represent the matrices $\mathbf{X}, \mathbf{Y}$ and $\mathbf{W}$ as collection of column vectors $\mathbf{X} = [\mathbf{x}_1\, \mathbf{x}_2\, \ldots\, \mathbf{x}_L]$, $\mathbf{Y} = [\mathbf{y}_1\, \mathbf{y}_2\, \ldots\, \mathbf{y}_L]$ and $\mathbf{W} = [\mathbf{w}_1\, \mathbf{w}_2\, \ldots\, \mathbf{w}_L]$ respectively, wherever needed. The formulation is valid for $L = 1$ (SMV) as well as $L > 1$ (MMV) case. We shall assume that each unknown sparse vector $\mathbf{x}_i$ is modeled as $\mathbf{x}_i = \mathbf{x}_{A_i} \circ \mathbf{x}_B$ where $\mathbf{x}_{A_i}$ and $\mathbf{x}_{A_j}$ are independent and $\circ$ indicates element-by-element multiplication. The vector $\mathbf{x}_{A_i}$ models the support distribution and consists of elements that are drawn from some unknown distribution[3] and $\mathbf{x}_B$ is a binary vector whose entries are drawn i.i.d. from a Bernoulli distribution with success probability λ. The sparsity of vectors $\mathbf{x}_i$'s is controlled by λ and, therefore, we call it the sparsity rate.

We pursue an MMSE estimate of $\mathbf{X}$ given $\mathbf{Y}$ as follows

$$\hat{\mathbf{X}}_{\text{mmse}} \triangleq \mathbb{E}[\mathbf{X}|\mathbf{Y}] = \sum_{\mathcal{S}} p(\mathcal{S}|\mathbf{Y})\mathbb{E}[\mathbf{X}|\mathbf{Y}, \mathcal{S}], \tag{2}$$

where the sum is executed over all possible 2^N support sets. Given the support $\mathcal{S}$, (1) becomes, $\mathbf{Y} = \mathbf{\Phi}_{\mathcal{S}}\mathbf{X}_{\mathcal{S}} + \mathbf{W}$, where $\mathbf{\Phi}_{\mathcal{S}}$ is a matrix formed by selecting columns of $\mathbf{\Phi}$ indexed by support $\mathcal{S}$. On the contrary, $\mathbf{X}_{\mathcal{S}}$ is formed by selecting rows of $\mathbf{X}$ indexed by $\mathcal{S}$. Let us see how the sum (2) can be evaluated. Since the distribution of the support of $\mathbf{X}$ is unknown or possibly non-Gaussian, computation of $\mathbb{E}[\mathbf{X}|\mathbf{Y}, \mathcal{S}]$ in (2) is difficult or even impossible. Thus the best we could do is to replace it with the best linear unbiased estimator (BLUE)[4]

$$\mathbb{E}[\mathbf{X}|\mathbf{Y}, \mathcal{S}] \leftarrow (\mathbf{\Phi}^{\mathsf{H}}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{S}})^{-1}\mathbf{\Phi}^{\mathsf{H}}_{\mathcal{S}}\mathbf{Y}. \tag{3}$$

Now, the posterior in (2) can be written using Bayes rule as

$$p(\mathcal{S}|\mathbf{Y}) = p(\mathbf{Y}|\mathcal{S})p(\mathcal{S})/p(\mathbf{Y}) \tag{4}$$

The factor $p(\mathbf{Y})$ is a normalizing factor common to all posteriors and hence can be ignored. Since the elements in each vector $\mathbf{x}_i$ are activated according to a Bernoulli distribution with success probability λ, we have

$$p(\mathcal{S}) = \lambda^{|\mathcal{S}|}(1-\lambda)^{N-|\mathcal{S}|}. \tag{5}$$

To evaluate the likelihood $p(\mathbf{Y}|\mathcal{S})$, we represent it in terms of individual vectors i.e., $p(\mathbf{Y}|\mathcal{S}) = p(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_L|\mathcal{S})$ and noting that these vectors given the support $\mathcal{S}$ are independent, we have $p(\mathbf{Y}|\mathcal{S}) = p(\mathbf{y}_1|\mathcal{S})p(\mathbf{y}_2|\mathcal{S})\ldots p(\mathbf{y}_L|\mathcal{S})$. If $\mathbf{X}_{\mathcal{S}}$ is Gaussian, $p(\mathbf{Y}|\mathcal{S})$ would also be Gaussian and that is easy to evaluate. On the other hand, when the distribution of $\mathbf{X}$ is unknown or even when it is known but non-Gaussian, determining $p(\mathbf{Y}|\mathcal{S})$ is in general very difficult. To go around this, we note that each measurement vector $\mathbf{y}_i$ is formed by a vector in the subspace spanned by the columns of $\mathbf{\Phi}_{\mathcal{S}}$ plus a Gaussian noise vector, $\mathbf{w}_i$. This motivates us to eliminate the non-Gaussian component by projecting $\mathbf{y}_i$ onto the orthogonal complement space of $\mathbf{\Phi}_{\mathcal{S}}$. To do so we multiply $\mathbf{y}_i$ by the projection matrix $\mathbf{P}^{\perp}_{\mathcal{S}} = \mathbf{I} - \mathbf{P}_{\mathcal{S}} = \mathbf{I} - \mathbf{\Phi}_{\mathcal{S}}\left(\mathbf{\Phi}^{\mathsf{H}}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{S}}\right)^{-1}\mathbf{\Phi}^{\mathsf{H}}_{\mathcal{S}}$. This leaves us with $\mathbf{P}^{\perp}_{\mathcal{S}}\mathbf{y}_i = \mathbf{P}^{\perp}_{\mathcal{S}}\mathbf{w}_i$, which is zero mean Gaussian with covariance

$$\mathbf{K} = \mathbb{E}[(\mathbf{P}^{\perp}_{\mathcal{S}}\mathbf{w}_i)(\mathbf{P}^{\perp}_{\mathcal{S}}\mathbf{w}_i)^{\mathsf{H}}] = \mathbf{P}^{\perp}_{\mathcal{S}}\sigma^2_{\mathbf{w}}\mathbf{P}^{\perp\,\mathsf{H}}_{\mathcal{S}} = \sigma^2_{\mathbf{w}}\mathbf{P}^{\perp}_{\mathcal{S}}. \tag{6}$$

where $\sigma^2_{\mathbf{w}}$ is the noise variance. Thus we can write,

$$p(\mathbf{y}_i|\mathcal{S}) \simeq \frac{1}{\sqrt{(2\pi\sigma^2_{\mathbf{w}})^M}}\exp\left(-\frac{1}{2}\left(\mathbf{P}^{\perp}_{\mathcal{S}}\mathbf{y}_i\right)^{\mathsf{H}}\mathbf{K}^{-1}\left(\mathbf{P}^{\perp}_{\mathcal{S}}\mathbf{y}_i\right)\right). \tag{7}$$

Simplifying and dropping the pre-exponential factor yields,

$$p(\mathbf{Y}|\mathcal{S}) \simeq \exp\left(-\frac{1}{2\sigma^2_{\mathbf{w}}}\sum_{i=1}^{L}\left\|\mathbf{P}^{\perp}_{\mathcal{S}}\mathbf{y}_i\right\|^2\right). \tag{8}$$

While we now have all the ingredients to evaluate the sum in (2) this remains a challenging task when N is large as we have to evaluate the sum over 2^N terms. To go around this, we approximate the sum by evaluating over a few support sets corresponding to significant posteriors, yielding,

$$\hat{\mathbf{X}}_{\text{ammse}} = \sum_{\mathcal{S} \in \mathcal{S}^d} p(\mathcal{S}|\mathbf{Y})\mathbb{E}[\mathbf{X}|\mathbf{Y}, \mathcal{S}]. \tag{9}$$

where $\mathcal{S}^d$ is the set of supports corresponding to significant posteriors. In the next section, we propose a greedy algorithm to find $\mathcal{S}^d$. For convenience, we represent the posteriors in the log domain and define a dominant support selection metric $\nu(\mathcal{S})$, to be used by the greedy algorithm, as

$$\begin{aligned}\nu(\mathcal{S}) &\triangleq \ln p(\mathbf{Y}|\mathcal{S})p(\mathcal{S})\\ &= \ln\exp\Big(\frac{-1}{2\sigma^2_{\mathbf{w}}}\sum_{i=1}^{L}\left\|\mathbf{P}^{\perp}_{\mathcal{S}}\mathbf{y}_i\right\|^2\Big) + \ln(\lambda^{|\mathcal{S}|}(1-\lambda)^{N-|\mathcal{S}|})\\ &= \frac{1}{2\sigma^2_{\mathbf{w}}}\sum_{i=1}^{L}\left[\left\|\mathbf{\Phi}_{\mathcal{S}}(\mathbf{\Phi}^{\mathsf{H}}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{S}})^{-1}\mathbf{\Phi}^{\mathsf{H}}_{\mathcal{S}}\mathbf{y}_i\right\|^2 - \left\|\mathbf{y}_i\right\|^2\right]\\ &\quad + |\mathcal{S}|\ln\lambda + (N-|\mathcal{S}|)\ln(1-\lambda)\end{aligned} \tag{10}$$

[2] Our algorithm is capable of modeling the scenario where multiple sensing matrices could be used to sense the unknown sparse vectors. However, in this paper we focus on the case where sensing matrices are same.

[3] The distribution may be unknown or known with unknown parameters or even Gaussian. Our developments are agnostic to the statistics of $\mathbf{x}_{A_i}$.

[4] This is essentially minimum-variance unbiased estimator (MVUE). The linear MMSE would have been a more faithful approach of the MMSE but that would depend on the second-order statistics of the support, defying our support agnostic approach.

3. THE GREEDY ALGORITHM M-SABMP

We now present a greedy algorithm to determine the set of dominant supports, $\mathcal{S}^d$, required to evaluate $\hat{\mathbf{X}}_{\text{ammse}}$ in (9). We search for the optimal support in a greedy manner. We first start by finding the best support of size 1, which involves evaluating $\nu(\mathcal{S})$ for $\mathcal{S} = \{1\}, \ldots, \{N\}$, i.e., a total of $\binom{N}{1}$ search points. Let $\mathcal{S}_1 = \{i_1^\star\}$ be the optimal support. Now, we look for the optimal support of size 2. Ideally, this involves a search over a space of size $\binom{N}{2}$. To reduce the search space, however, we pursue a greedy approach and look for the point $i_2^\star \neq i_1^\star$ such that $\mathcal{S}_2 = \{i_1^\star, i_2^\star\}$ maximizes $\nu(\mathcal{S}_2)$. This involves $\binom{N-1}{1}$ search points (as opposed to the optimal search over $\binom{N}{2}$ points). We continue in this manner by forming $\mathcal{S}_3 = \{i_1^\star, i_2^\star, i_3^\star\}$ and searching for $i_3^\star$ in the remaining $N-2$ points and so on until we reach $\mathcal{S}_P = \{i_1^\star, \ldots, i_P^\star\}$. The value of P is selected to be slightly larger than the expected number of nonzero elements in each of the constructed signal such that $\Pr(|\mathcal{S}| > P)$ is sufficiently small[5].

Note that in our greedy move from $\mathcal{S}_j$ to $\mathcal{S}_{j+1}$, we need to evaluate $\nu(\mathcal{S}_j \cup \{i_{j+1}\})$ around N times, which can be done in an order-recursive manner starting from that of $\nu(\mathcal{S}_j)$. Specifically, we note that every expansion, $\mathcal{S}_j \cup \{i_{j+1}\}$, from $\mathcal{S}_j$ requires a calculation of $\nu(\mathcal{S}_j \cup \{i_{j+1}\})$ using (10). This translates to appending a column $\boldsymbol{\phi}_{j+1}$ to $\boldsymbol{\Phi}_{\mathcal{S}_j}$ in the calculations of (10), which can be done in an order-recursive manner. We summarize these calculations in Section 4.

The nature of our greedy algorithm allows us to output not just the set of dominant supports but also the ingredients needed to compute $\mathbf{X}_{\text{ammse}}$ in (9) without any additional cost. Specifically, since $\nu(\mathcal{S})$ is simply $\ln p(\mathcal{S}|\mathbf{Y})$, we do not need to compute the posteriors separately. Similarly, the form of $\mathbb{E}[\mathbf{X}|\mathbf{Y}, \mathcal{S}]$ in (3) lends itself as an intermediate computation performed to calculate $\nu(\mathcal{S})$. A formal algorithmic description of our greedy algorithm is presented in Table 1.

3.1. Refined Greedy Search

One of the advantages of the proposed greedy algorithm is that it is agnostic to the support distribution; the only parameters required are the noise variance, $\sigma_{\mathbf{w}}^2$, and the sparsity rate, λ. However, M-SABMP does not require the user to provide any initial estimate of λ and $\sigma_{\mathbf{w}}^2$. Instead the method starts by finding initial estimates of these parameters which are used to compute $\nu(\mathcal{S})$ in (10). Refining these parameters will improve our chances of selecting the right support. The refinement demands that the greedy algorithm be repeated with new estimates. In this way both the hyperparameters and support are refined simultaneously. The repetition continues until a predetermined criterion has been satisfied. A description of the M-SABMP algorithm is provided in Table 2.

```
procedure GREEDY($\boldsymbol{\Phi}, \mathbf{Y}, \lambda, \sigma_{\mathbf{w}}^2, P$)
    initialize $T \leftarrow \{1, 2, \ldots, N\}$, $i \leftarrow 1$
    initialize empty $\mathcal{S}_{\max}$, $\mathcal{S}^d$, $p(\mathcal{S}^d|\mathbf{Y})$, $\mathbb{E}[\mathbf{X}|\mathbf{Y}, \mathcal{S}^d]$
    $T_i \leftarrow T$
    while $i \leq P$ do
        $\Omega \leftarrow \{\mathcal{S}_{\max} \cup \{\alpha_1\}, \mathcal{S}_{\max} \cup \{\alpha_2\}, \cdots, \mathcal{S}_{\max} \cup \{\alpha_{|L_i|}\} \mid \alpha_k \in T_i\}$
        compute $\{\nu(\mathcal{S}_k) \mid \mathcal{S}_k \in \Omega\}$
        find $\mathcal{S}^\star \in \Omega$ such that $\nu(\mathcal{S}^\star) \geq \max_j \nu(\mathcal{S}_j)$
        $\mathcal{S}^d \leftarrow \{\mathcal{S}^d, \mathcal{S}^\star\}$
        compute $p(\mathcal{S}^\star|\mathbf{Y}), \mathbb{E}[\mathbf{X}|\mathbf{Y}, \mathcal{S}^\star]$
        $p(\mathcal{S}^d|\mathbf{Y}) \leftarrow \{p(\mathcal{S}^d|\mathbf{Y}), p(\mathcal{S}^\star|\mathbf{Y})\}$
        $\mathbb{E}[\mathbf{X}|\mathbf{Y}, \mathcal{S}^d] \leftarrow \{\mathbb{E}[\mathbf{X}|\mathbf{Y}, \mathcal{S}^d], \mathbb{E}[\mathbf{X}|\mathbf{Y}, \mathcal{S}^\star\}$
        $\mathcal{S}_{\max} \leftarrow \mathcal{S}^\star, T_{i+1} \leftarrow T \setminus \mathcal{S}^\star, i \leftarrow i + 1$
    end while
    return $\mathcal{S}^d, p(\mathcal{S}^d|\mathbf{Y}), \mathbb{E}[\mathbf{X}|\mathbf{Y}, \mathcal{S}^d]$
end procedure
```

Table 1: The Greedy Algorithm

3.2. Estimation & Refinement of λ and $\sigma_{\mathbf{w}}^2$

When the hyperparameters λ and $\sigma_{\mathbf{w}}^2$ are unknown, we need to refine them iteratively. This starts from some initial estimate usually supplied by the user. Here, we show how we can initialize the process from the observed data. Projecting a measurement vector $\mathbf{y}_j$ onto the basis vectors $\boldsymbol{\phi}_i$, $i = 1, \cdots, N$ (columns of $\boldsymbol{\Phi}$) provides one way to initialize λ as follows:

$$i^\star = \underset{i\in[1,\cdots,N]}{\arg\max} \left|\boldsymbol{\phi}_i^{\mathsf{H}}\mathbf{y}_j\right|,$$

$$\rho_i^\star = \begin{cases} 0, & \text{if } \left|\boldsymbol{\phi}_i^{\mathsf{H}}\mathbf{y}_j\right| < \left|\boldsymbol{\phi}_{i^\star}^{\mathsf{H}}\mathbf{y}_j\right|/2 \\ 1, & \text{otherwise} \end{cases},$$

$$\lambda_{\textbf{init}} = \sum_{i=1}^{N} \rho_i^\star / N. \tag{11}$$

Our algorithm is robust enough to find the right support even if λ is initialized badly. Note that the projections performed above are required by the first step of the greedy algorithm and, therefore, do not require any additional computation. As for the noise variance, our experimental results show that using an initial estimate as rough as $\sigma_{\mathbf{w}\,\textbf{init}}^2 = \frac{1}{4}\sigma_{\mathbf{Y}}^2$ is good enough ($\sigma_{\mathbf{Y}}^2$ is the variance of the elements of $\mathbf{Y}$ collectively) and the algorithm performs quite well in estimating the actual noise variance. We now explain how a fairly accurate estimation could be performed in a very simple manner as follows.

Recall that, our greedy algorithm returns a set of dominant supports $\mathcal{S}^d$ along with the corresponding $p(\mathcal{S}|\mathbf{Y})$ and $\mathbb{E}[\mathbf{X}|\mathbf{Y}, \mathcal{S}]$. These are used to compute $\hat{\mathbf{X}}_{\text{ammse}}$ from (9). Similarly, by determining $\hat{\mathcal{S}}_{\text{map}} = \arg\max_{\mathcal{S}\in\mathcal{S}^d} p(\mathcal{S}|\mathbf{Y})$ we are able to determine $\hat{\mathbf{x}}_{\text{map}} = \mathbb{E}[\mathbf{X}|\mathbf{Y}, \hat{\mathcal{S}}_{\text{map}}]$. Based on these

[5] $|\mathcal{S}|$, i.e., support of the constructed signal, follows the binomial distribution $\mathcal{B}(N, p)$, which can be approximated by the Gaussian distribution $\mathcal{N}(Np, Np(1-p))$ if $Np > 5$. For this case, $\Pr(|\mathcal{S}| > P) = \frac{1}{2}\text{erfc}\frac{P-Np}{\sqrt{2Np(1-p)}}$.

```
procedure M-SABMP($\mathbf{\Phi}, \mathbf{Y}, r_{stop}$)
    estimate $\lambda$, and $\sigma^2_{\mathbf{w}}$.
    repeat
        $P \leftarrow N\lambda + \epsilon$
        $\lambda_{\text{old}} \leftarrow \lambda$
        $\{\mathcal{S}^d, p(\mathcal{S}^d|\mathbf{Y}), \mathbb{E}[\mathbf{X}|\mathbf{Y}, \mathcal{S}^d]\} \leftarrow \text{G}(\mathbf{\Phi}, \mathbf{Y}, \lambda, \sigma^2_{\mathbf{w}}, P)$
        $\hat{\mathcal{S}}_{\text{map}} \leftarrow \arg\max_{\mathcal{S}} p(\mathcal{S}|\mathbf{Y})$
        $\hat{\mathbf{x}}_{\text{map}} \leftarrow \mathbb{E}[\mathbf{X}|\mathbf{Y}, \hat{\mathcal{S}}_{\text{map}}]$
        $\hat{\mathbf{x}}_{\text{ammse}} \leftarrow \sum_{\mathcal{S} \in \mathcal{S}^d} p(\mathcal{S}|\mathbf{Y}) \mathbb{E}[\mathbf{X}|\mathbf{Y}, \mathcal{S}]$
        $\lambda \leftarrow \|\hat{\mathbf{x}}_{\text{map}}\|_0 / LN$
        $\sigma^2_{\mathbf{w}} \leftarrow \text{var}(\mathbf{Y} - \mathbf{\Phi}\hat{\mathbf{X}}_{\text{ammse}})$
    until $|\lambda - \lambda_{\text{old}}|/\lambda_{\text{old}} < r_{\text{stop}}$
    return $\hat{\mathbf{X}}_{\text{ammse}}$
end procedure
```

Table 2: M-SABMP

quantities we update λ and $\sigma^2_{\mathbf{w}}$ as follows:

$$\hat{\lambda} = \left\|\hat{\mathbf{X}}_{\text{map}}\right\|_0 / LN, \quad \hat{\sigma}^2_{\mathbf{w}} = \text{var}(\mathbf{Y} - \mathbf{\Phi}\hat{\mathbf{X}}_{\text{ammse}}) \qquad (12)$$

The greedy algorithm is called again with this new set of parameters. The output of which is then used to update λ and $\sigma^2_{\mathbf{w}}$ again using (12). This process continues until a prespecified criteria has been satisfied. Simulation results show that both $\hat{\lambda}$ and $\hat{\sigma}^2_{\mathbf{w}}$ converge to fairly accurate values.

We would also highlight that the nature of our algorithm allows us to penalize sparsity. Unlike normal matching pursuit algorithms, our algorithm could select supports of varying sizes based on $\nu(\mathcal{S})$ and thus is able to penalize sparsity. For a detailed description of the steps followed by the method the algorithms are provided in Tables 1 and 2.

4. EFFICIENT COMPUTATION OF $\nu(\mathcal{S})$

By inspection of $\nu(\mathcal{S})$ in (10), we see that the main challenge is in calculating the term $\|(\mathbf{\Phi}_{\mathcal{S}}(\mathbf{\Phi}^{\mathsf{H}}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{S}})^{-1}\mathbf{\Phi}^{\mathsf{H}}_{\mathcal{S}}\mathbf{y}_i)\|^2$ which can be written in terms of the expectation as $\|\mathbf{\Phi}_{\mathcal{S}}\mathbb{E}[\mathbf{x}_i|\mathbf{y}_i, \mathcal{S}]\|^2$. So, we mainly need to update $\mathbb{E}[\mathbf{x}_i|\mathbf{y}_i, \mathcal{S}]$. To this end, consider the general support $\mathcal{S} = \{s_1, s_2, s_3, \ldots, s_k\}$ with $s_1 < s_2 < \cdots < s_k$ and let $\underline{\mathcal{S}}$ and $\overline{\mathcal{S}}$ denote the subset $\underline{\mathcal{S}} = \{s_1, s_2, s_3, \ldots, s_{k-1}\}$ and superset $\overline{\mathcal{S}} = \{s_1, s_2, s_3, \ldots, s_{k+1}\}$, respectively, where $s_k < s_{k+1}$. In the following, we demonstrate how to update $\mathbf{e}_{\mathbf{y},k-1}(\underline{\mathcal{S}}) \triangleq \mathbb{E}[\mathbf{x}_i|\mathbf{y}_i, \underline{\mathcal{S}}]$ to obtain[6] $\mathbf{e}_{\mathbf{y},k}(\mathcal{S}) = \mathbb{E}[\mathbf{x}_i|\mathbf{y}_i, \mathcal{S}]$. Note that since $\mathcal{S} = \underline{\mathcal{S}} \cup \{s_k\}$, we can write

$$\mathbf{e}_{\mathbf{y},k}(\mathcal{S}) = \left(\begin{bmatrix} \mathbf{\Phi}^{\mathsf{H}}_{\underline{\mathcal{S}}} \\ \\ \boldsymbol{\phi}^{\mathsf{H}}_{s_k} \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_{\underline{\mathcal{S}}} \boldsymbol{\phi}_{s_k} \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{\Phi}^{\mathsf{H}}_{\underline{\mathcal{S}}}\mathbf{y} \\ \\ \boldsymbol{\phi}^{\mathsf{H}}_{s_k}\mathbf{y} \end{bmatrix}. \qquad (13)$$

Note that we have dropped the subscript i for simplicity. By using the block inversion formula to express the inverse of the above and simplifying, we get

$$\mathbf{e}_{\mathbf{y},k}(\mathcal{S}) = \begin{bmatrix} \Gamma \mathbf{e}_{\boldsymbol{\phi},k}(\mathcal{S}) + \mathbf{e}_{\mathbf{y},k-1}(\underline{\mathcal{S}}) \\ \\ -\Gamma \end{bmatrix} \qquad (14)$$

where $\Gamma = \frac{1}{f_{\mathcal{S}}}(\mathbf{q}^{\mathsf{H}}_{\boldsymbol{\phi},k}(\mathcal{S})\mathbf{e}_{\mathbf{y},k-1}(\underline{\mathcal{S}}) - \mathbf{e}_{\mathbf{y},1}(s_k))$. This recursion is initialized by $\mathbf{e}_{\mathbf{y},1}(i) = (\boldsymbol{\phi}^{\mathsf{H}}_s\boldsymbol{\phi}_s)^{-1}\boldsymbol{\phi}^{\mathsf{H}}_s\mathbf{y}$. The recursion also depends on $\mathbf{q}_{\boldsymbol{\phi},k}(\mathcal{S}) \triangleq \mathbf{\Phi}^{\mathsf{H}}_{\underline{\mathcal{S}}}\boldsymbol{\phi}_{s_k}$, $\mathbf{e}_{\boldsymbol{\phi},k}(\mathcal{S}) \triangleq (\mathbf{\Phi}^{\mathsf{H}}_{\underline{\mathcal{S}}}\mathbf{\Phi}_{\underline{\mathcal{S}}})^{-1}\mathbf{\Phi}^{\mathsf{H}}_{\underline{\mathcal{S}}}\boldsymbol{\phi}_{s_k}$ and $f_{\mathcal{S}} \triangleq 1 - \mathbf{q}^{\mathsf{H}}_{\boldsymbol{\phi},k}(\mathcal{S})\mathbf{e}_{\boldsymbol{\phi},k}(\mathcal{S})$. The recursions for $\mathbf{e}_{\boldsymbol{\phi},k}(\mathcal{S})$, and $\mathbf{q}_{\boldsymbol{\phi},k}(\mathcal{S})$ may be determined as follows[7]

$$\mathbf{e}_{\boldsymbol{\phi},k+1}(\overline{\mathcal{S}}) = \begin{bmatrix} \Lambda \mathbf{e}_{\boldsymbol{\phi},k}(\mathcal{S}) + \mathbf{e}_{\boldsymbol{\phi},k}(\underline{\mathcal{S}}; s_{k+1}) \\ \\ -\Lambda \end{bmatrix} \qquad (15)$$

where $\Lambda = \frac{1}{f_{\mathcal{S}}}(\mathbf{q}^{\mathsf{H}}_{\boldsymbol{\phi},k}(\mathcal{S})\mathbf{e}_{\boldsymbol{\phi},k}(\underline{\mathcal{S}}; s_{k+1}) - \mathbf{e}_{\boldsymbol{\phi},2}(s_k; s_{k+1}))$,

$$\mathbf{q}_{\boldsymbol{\phi},k+1}(\overline{\mathcal{S}}) = \begin{bmatrix} \mathbf{\Phi}^{\mathsf{H}}_{\underline{\mathcal{S}}} \\ \boldsymbol{\phi}^{\mathsf{H}}_{s_k} \end{bmatrix} \boldsymbol{\phi}_{s_{k+1}} = \begin{bmatrix} \mathbf{q}_{\boldsymbol{\phi},k}(\underline{\mathcal{S}}; s_{k+1}) \\ \mathbf{q}_{\boldsymbol{\phi},2}(s_k; s_{k+1}) \end{bmatrix} \qquad (16)$$

The two recursions (15) and (16) start at $k = 2$ and are thus initialized by $\mathbf{e}_{\boldsymbol{\phi},2}(s_1; s_2)$ and $\mathbf{q}_{\boldsymbol{\phi},2}(s_1; s_2)$ for $s_1, s_2 = 1, 2, \ldots, N$. This completes the recursion of $\mathbf{e}_{\mathbf{y},k}(\mathcal{S})$ which we utilize for recursive evaluation of $\nu(\mathcal{S})$.

5. SIMULATION RESULTS

To demonstrate the performance of the proposed M-SABMP algorithm, we performed two experiments. In all of the experiments we chose $N = 128$, and $M = 64$. Experiments were conducted for signals whose active elements are drawn from Gaussian as well as non-Gaussian distributions. Entries of $M \times N$ sensing/measurement matrix $\mathbf{\Phi}$ were i.i.d., with zero means and complex Gaussian distribution where the columns were normalized to the unit norm. The noise had a zero mean and was white and Gaussian with variance $\sigma^2_{\mathbf{w}}$ which was determined according to the desired SNR. Finally, we used two different metrics for performance measure; the normalized mean-squared error (NMSE) between the original $\mathbf{X}$, and its MMSE estimate, $\hat{\mathbf{X}}_{\text{ammse}}$ defined by $10\log_{10}(\|\hat{\mathbf{X}}_k - \mathbf{X}_k\|^2 / \|\mathbf{X}_k\|^2)$, and the success rate. Success rate is defined as the ratio of the number of successful trials to the number of total trials, where a trial is successful when the condition NMSE ≤ -10 dB was satisfied. The number of trials performed for computing both NMSE and success rate was 200. We compared the performance of our algorithm with M-FOCUSS [11] and M-BP [16] where default settings as proposed by their authors were used.

In the first experiment, success rate was computed for different sparsity levels. Simulations were performed for $L = 3, 5,$ and 7 while SNR was 25dB. It is obvious from Fig. 1

[6] We explicitly indicate the size k of $\mathcal{S}$ in this notation as it elucidates the recursive nature of the developed algorithms.

[7] Notation such as $\mathbf{e}_{\boldsymbol{\phi},k}(\underline{\mathcal{S}}; s_{k+1})$ is a short hand for $\mathbf{e}_{\boldsymbol{\phi},k}(\underline{\mathcal{S}} \cup \{s_{k+1}\})$.

that M-SABMP outperforms both algorithms; especially it performs well for higher values of the sparsity rate and is able to detect the unknown support with high accuracy.

In Fig. 2, error performance of the MMV algorithms is compared for various values of SNR and $\lambda = 0.2$. Observe that M-SABMP outperforms other algorithms for all values of SNR. Additionally, Fig. 3 shows that in most of the cases our algorithm requires the least amount of time which is due to its recursive nature.

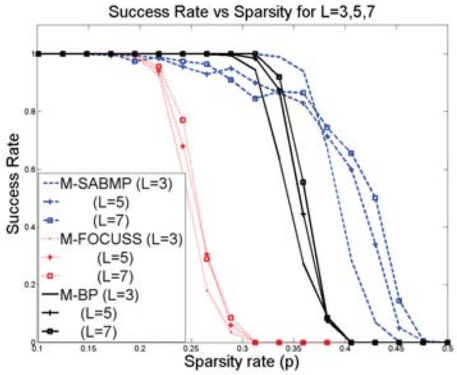

Fig. 1: Signal recovery success rate

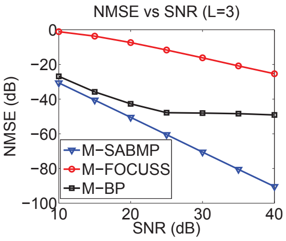

Fig. 2: NMSE vs SNR

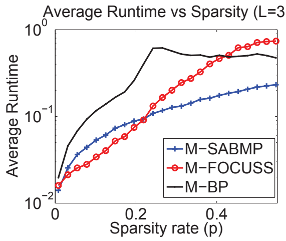

Fig. 3: Runtime vs Sparsity

6. CONCLUSION

A robust Bayesian matching pursuit algorithm based on a fast recursive method for joint-sparse signal recovery is presented. It does not require the active elements in signals to be derived from some known distribution. This is useful when we cannot estimate the parameters of the signal distributions. The algorithm does not require the initial estimates of signal sparsity and noise variance and is able to boot strap itself. We demonstrated that the algorithm is robust and performs well as compared to other algorithms.

REFERENCES

[1] S. F. Cotter and B. D. Rao, "Sparse channel estimation via matching pursuit with application to equalization," *IEEE Trans. on Commun.*, vol. 50, no. 3, Mar 2002.

[2] M. Masood, L. H. Afify, and T. Y. Al-Naffouri, "Efficient coordinated recovery of sparse channels in massive MIMO," *IEEE Trans. on Sig. Process. (submitted)*, 2014.

[3] M. Masood, L. H. Afify, and T. Y. Al-Naffouri, "Efficient coordinated recovery of sparse channels in massive MIMO," in *IEEE Globecom (submitted)*, 2014.

[4] A. Cichocki, "Blind source separation: New tools for extraction of source signals and denoising," *SPIE*, pp. 11–25, 2005.

[5] D. Wipf and S. Nagarajan, "A unified Bayesian framework for MEG/EEG source imaging," *NeuroImage*, pp. 947–966, 2008.

[6] F. Sana, T. Ballal, T. Y. Al-Naffouri, and I. Hoteit, "Low-complexity wireless monitoring of respiratory movements using ultra-wideband ir estimation," *Biomedical SP & Control*, vol. 10, pp. 192 – 200, 2014.

[7] G. Obozinski, M. J. Wainwright, and M. I. Jordan, "Support union recovery in high-dimensional MV regression," *The Annals of Statistics*, , no. 1, pp. 1–47.

[8] Yuzhe Jin and B. D. Rao, "Support recovery of sparse signals in the presence of multiple measurement vectors," *IEEE Trans. on Info. Theory*, vol. 59, no. 5, pp. 3139–3157, May 2013.

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

[10] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Simultaneous sparse approximation via greedy pursuit," in *IEEE Int. Conf. on Acoustics Speech and Sig. Process. (ICASSP)*, March 2005, vol. 5, pp. 721 – 724.

[11] S. F. Cotter, B. D. Rao, Kjersti Engan, and K. Kreutz-Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors," *IEEE Trans. on Sig. Process.*, vol. 53, no. 7, pp. 2477 – 2488, July 2005.

[12] M. Mishali and Y. C. Eldar, "ReMBo: Recovering arbitrary sets of jointly sparse vectors," *IEEE Trans. on Sig. Process.*, vol. 56, no. 10, pp. 4692 –4702, Oct. 2008.

[13] Kiryung Lee, Y. Bresler, and M. Junge, "Subspace methods for joint sparse recovery," *IEEE Trans. on Info. Theory*, vol. 58, no. 6, pp. 3613 –3641, June 2012.

[14] M. Masood and T. Y. Al-Naffouri, "Sparse reconstruction using distribution agnostic Bayesian matching pursuit," *IEEE Trans. on Sig. Process.*, vol. 61, no. 21, pp. 5298–5309, Nov 2013.

[15] T. Y. Al-Naffouri and M. Masood, "Distribution agnostic structured sparsity recovery algorithms," in *8th International WoSSPA*, 2013, pp. 283–290.

[16] E. van den Berg and M. P. Friedlander, "SPGL1: A solver for large-scale sparse reconstruction," June 2007.