SUPPORT AGNOSTIC BAYESIAN MATCHING PURSUIT FOR BLOCK SPARSE SIGNALS

Mudassir Masood[1] *and Tareq Y. Al-Naffouri*[1,2]

[1,2]EE Department, King Abdullah University of Science & Technology, Thuwal, Makkah, Saudi Arabia
[2]EE Department, King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia
{mudassir.masood, tareq.alnaffouri}@kaust.edu.sa

ABSTRACT

A fast matching pursuit method using a Bayesian approach is introduced for block-sparse signal recovery. This method performs Bayesian estimates of block-sparse signals even when the distribution of active blocks is non-Gaussian or unknown. It is agnostic to the distribution of active blocks in the signal and utilizes a priori statistics of additive noise and the sparsity rate of the signal, which are shown to be easily estimated from data and no user intervention is required. The method requires a priori knowledge of block partition and utilizes a greedy approach and order-recursive updates of its metrics to find the most dominant sparse supports to determine the approximate minimum mean square error (MMSE) estimate of the block-sparse signal. Simulation results demonstrate the power and robustness of our proposed estimator.

Index Terms— Block sparse signals, sparse signal recovery, compressed sensing, Bayesian matching pursuit, SABMP

1. INTRODUCTION

The problem addressed by compressed sensing and sparse recovery algorithms is to recover an unknown sparse vector from an underdetermined system of linear equations. Sometimes the sparse signals under consideration are structured in nature. Specifically, a natural block structure might exist in the sparse signal where the few nonzero elements appear in groups. For example, an ideal sparse channel consisting of a few multipath components could be represented in a block sparse structure [1]. Some other interesting situations where block sparsity arises include, gene expression analysis [2], time series data analysis involving lagged variables forming a block, multiple measurement vector (MMV) [3], PAPR reduction in OFDM [4] and neural activity [5]. It has been known that the block structure can be exploited for enhanced recovery. Moreover, it was shown that if knowledge of the block structure (expected locations where blocks might occur and the block sizes) is available then under certain conditions it allows us to reduce the number of observations required for recovery [6].

Several algorithms have been proposed taking into account the knowledge of the block structure. The foundational work in this respect was [7] which proposed the group-LASSO algorithm. However, it has limited applicability as compared to other algorithms as it makes some assumptions on the dictionary being used. Block-OMP [8] is an extension of the classical orthogonal matching pursuit algorithm (OMP [9]). It was proposed by Eldar et. al. where they used the concept of block coherence to extend the OMP algorithm. Another algorithm by Eldar called mixed ℓ_2/ℓ_1-norm recovery algorithm proposed in [3] extended the BP method to tackle block sparsity. Similarly, extensions of the CoSAMP algorithm [10] and IHT [11] were used to propose an algorithm called Block-CoSAMP [6] which has provable recovery guarantees and robustness properties. The LaMP algorithm proposed in [12] used Markov Random Fields model to capture the structure of sparse signal. They demonstrated that their algorithm performed well using fewer number of measurements.

All of these methods mentioned above belong to the category of greedy algorithms. Their advantage is that they are agnostic to support distribution[1] and hence demonstrate robust performance. However, most of them work only in noiseless environment. Although many Bayesian approaches (e.g., [13]) could be extended to utilize block structure, these are not as common. A notable exception is the cluster-sparse Bayesian learning algorithm (cluster-SBL) proposed in [14] which assumes that the block supports follow a multivariate Gaussian distribution. Furthermore, they take into account the intra-block correlation to show that it improved the recovery performance.

The focus of the present paper is on developing a Bayesian approach for block-structured sparse signal recovery. Specifically, we pursue a Bayesian approach similar to that proposed in [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). While the approach depends on the sparsity rate and the noise variance, it does not require estimates of the parameters but is able to estimate these parameters in a robust manner. Specifically, the advantages of our approach are as follows

1. The approach provides a Bayesian estimate of the sparse signal even when the signal support prior is non-Gaussian or unknown.
2. The approach 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.
3. The approach utilizes the prior Gaussian statistics of the additive noise and the sparsity rate of the signal. The approach is able to estimate the noise variance and sparsity rate in a robust manner from the data.
4. The approach 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, Dhahran, Saudi Arabia (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 BLOCK SPARSE SIGNAL RECONSTRUCTION

In this paper we will consider the estimation of a block-structured sparse vector, $\mathbf{x} \in \mathbb{C}^{N\times 1}$, from an observations vector $\mathbf{y} \in \mathbb{C}^{M\times 1}$, obeying the linear regression model,

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x} + \mathbf{n}. \tag{1}$$

Here $\mathbf{\Phi} \in \mathbb{C}^{M\times N}$ is a known regression matrix and $\mathbf{n}$ is the additive white Gaussian noise vector $\mathcal{CN}(\mathbf{0}, \sigma_{\mathbf{n}}^2 \mathbf{I}_M)$. We shall assume that $\mathbf{x}$ has a block-sparse structure and is modeled as $\mathbf{x} = \mathbf{x}_A \circ \mathbf{x}_B$ where $\circ$ indicates element-by-element multiplication. The vector $\mathbf{x}_A$ models the support distribution and consists of elements that are drawn from some unknown distribution[2] and $\mathbf{x}_B$ is a block-structered binary vector with K blocks of size C each as shown below:

$$\mathbf{x}_B = [\mathbf{0}^{\mathsf{T}}\, \mathbf{x}_{\mathcal{B}_1}^{\mathsf{T}}\, \mathbf{0}^{\mathsf{T}}\, \mathbf{x}_{\mathcal{B}_2}^{\mathsf{T}}\, \mathbf{0}^{\mathsf{T}} \cdots \mathbf{x}_{\mathcal{B}_K}^{\mathsf{T}}\, \mathbf{0}^{\mathsf{T}}]^{\mathsf{T}}. \tag{2}$$

where $\mathcal{B}_i$, $i = 1, \ldots K$ refer to the size C supports of each block. Equivalently, we can write $\mathbf{x}_B$ as[3]

$$\mathbf{x}_B = \mathbf{x}_b \otimes \mathbf{1}_C \tag{3}$$

where $\mathbf{1}_C$ is a $C \times 1$ vector of 1's and $\mathbf{x}_b$ is a $K \times 1$ binary vector. Block sparsity requires that only a few among the K blocks in $\mathbf{x}_B$ are non-zero. The blocks of $\mathbf{x}_B$ (or equivalently, the elements of $\mathbf{x}_b$) are activated according to a Bernoulli distribution with success probability λ. Thus, for example, probability of k blocks being active out of the total K blocks will be $\lambda^k(1-\lambda)^{K-k}$. We observe that the sparsity of vector $\mathbf{x}$ is controlled by λ and, therefore, we call it the sparsity parameter/rate.

We purse an MMSE estimate of $\mathbf{x}$ given observation $\mathbf{y}$ and the block partition (2) as follows

$$\hat{\mathbf{x}}_{mmse} \triangleq \mathbb{E}[\mathbf{x}|\mathbf{y}] = \sum_{\mathcal{S}} p(\mathcal{S}|\mathbf{y})\mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}], \tag{4}$$

where the sum is executed over all possible 2^K support sets formed by K blocks. Given the support $\mathcal{S}$ composed of $|\mathcal{S}|/C$ blocks, (1) becomes, $\mathbf{y} = \mathbf{\Phi}_{\mathcal{S}}\mathbf{x}_{\mathcal{S}} + \mathbf{n}$, where $\mathbf{\Phi}_{\mathcal{S}}$ is a matrix formed by selecting columns of $\mathbf{\Phi}$ indexed by support $\mathcal{S}$. Similarly, $\mathbf{x}_{\mathcal{S}}$ is formed by selecting entries of $\mathbf{x}$ indexed by $\mathcal{S}$. Since the distribution of $\mathbf{x}$ is unknown or possibly non-Gaussian, computation of $\mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}]$ in (4) is difficult or even impossible. Thus the best we could do is to replace it with the best linear unbiased estimator (BLUE)[4]

$$\mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}] \leftarrow (\mathbf{\Phi}_{\mathcal{S}}^{\mathsf{H}}\mathbf{\Phi}_{\mathcal{S}})^{-1}\mathbf{\Phi}_{\mathcal{S}}^{\mathsf{H}}\mathbf{y}. \tag{5}$$

Now, the posterior in (4) can be written using Bayes rule as

$$p(\mathcal{S}|\mathbf{y}) = p(\mathbf{y}|\mathcal{S})p(\mathcal{S})/p(\mathbf{y}) \tag{6}$$

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

$$p(\mathcal{S}) = \lambda^{|\mathcal{S}|/C}(1-\lambda)^{K-|\mathcal{S}|/C}. \tag{7}$$

It remains to evaluate the likelihood $p(\mathbf{y}|\mathcal{S})$. 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 $\mathbf{y}$ is formed by a vector in the subspace spanned by the columns of $\mathbf{\Phi}_{\mathcal{S}}$ plus a Gaussian noise vector, $\mathbf{n}$. This motivates us to eliminate the non-Gaussian component by projecting $\mathbf{y}$ onto the orthogonal complement space of $\mathbf{\Phi}_{\mathcal{S}}$. This is done by multiplying $\mathbf{y}$ by the projection matrix $\mathbf{P}_{\mathcal{S}}^{\perp} = \mathbf{I} - \mathbf{P}_{\mathcal{S}} = \mathbf{I} - \mathbf{\Phi}_{\mathcal{S}}\left(\mathbf{\Phi}_{\mathcal{S}}^{\mathsf{H}}\mathbf{\Phi}_{\mathcal{S}}\right)^{-1}\mathbf{\Phi}_{\mathcal{S}}^{\mathsf{H}}$. This leaves us with $\mathbf{P}_{\mathcal{S}}^{\perp}\mathbf{y} = \mathbf{P}_{\mathcal{S}}^{\perp}\mathbf{n}$, which is Gaussian with a zero mean and covariance

$$\begin{aligned}\mathbf{K} &= \mathbb{E}[(\mathbf{P}_{\mathcal{S}}^{\perp}\mathbf{n})(\mathbf{P}_{\mathcal{S}}^{\perp}\mathbf{n})^{\mathsf{H}}] \\ &= \mathbf{P}_{\mathcal{S}}^{\perp}\sigma_{\mathbf{n}}^2\mathbf{P}_{\mathcal{S}}^{\perp\,\mathsf{H}} = \sigma_{\mathbf{n}}^2\mathbf{P}_{\mathcal{S}}^{\perp}.\end{aligned}$$

Thus we can approximate the likelihood $p(\mathbf{y}|\mathcal{S})$ by,

$$p(\mathbf{y}|\mathcal{S}) \simeq \frac{1}{\sqrt{(2\pi\sigma_{\mathbf{n}}^2)^M}} \exp\left(-\frac{1}{2}\left(\mathbf{P}_{\mathcal{S}}^{\perp}\mathbf{y}\right)^{\mathsf{H}}\mathbf{K}^{-1}\left(\mathbf{P}_{\mathcal{S}}^{\perp}\mathbf{y}\right)\right). \tag{8}$$

Simplifying and dropping the pre-exponential factor yields,

$$p(\mathbf{y}|\mathcal{S}) \simeq \exp\left(-\frac{1}{2\sigma_{\mathbf{n}}^2}\left\|\mathbf{P}_{\mathcal{S}}^{\perp}\mathbf{y}\right\|^2\right). \tag{9}$$

While we now have all the ingredients to evaluate the sum in (4) this remains a challenging task when K is large as we have to evaluate the sum over 2^K terms. To go around this, we approximate the sum by evaluating over a few support sets corresponding to significant posteriors, yielding,

$$\hat{\mathbf{x}}_{ammse} = \sum_{\mathcal{S}\in\mathcal{S}^d} p(\mathcal{S}|\mathbf{y})\mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}]. \tag{10}$$

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

$$\begin{aligned}\nu(\mathcal{S}) &\triangleq \ln p(\mathbf{y}|\mathcal{S})p(\mathcal{S}) \\ &= \ln\exp\left(\frac{-1}{2\sigma_{\mathbf{n}}^2}\left\|\mathbf{P}_{\mathcal{S}}^{\perp}\mathbf{y}\right\|^2\right) + \ln(\lambda^{|\mathcal{S}|/C}(1-\lambda)^{K-|\mathcal{S}|/C}) \\ &= \frac{1}{2\sigma_{\mathbf{n}}^2}\left\|\mathbf{\Phi}_{\mathcal{S}}(\mathbf{\Phi}_{\mathcal{S}}^{\mathsf{H}}\mathbf{\Phi}_{\mathcal{S}})^{-1}\mathbf{\Phi}_{\mathcal{S}}^{\mathsf{H}}\mathbf{y}\right\|^2 - \frac{1}{2\sigma_{\mathbf{n}}^2}\|\mathbf{y}\|^2 \\ &\quad + \frac{|\mathcal{S}|}{C}\ln\lambda + (K - \frac{|\mathcal{S}|}{C})\ln(1-\lambda)\end{aligned} \tag{11}$$

3. SUPPORT AGNOSTIC BAYESIAN MATCHING PURSUIT FOR BLOCK-STRUCTURED SPARSE SIGNALS

We now present a greedy algorithm to determine the set of dominant supports $\mathcal{S}^d$ required to evaluate $\hat{\mathbf{x}}_{ammse}$ in (10). We search for the optimal support in a greedy manner. We first start by finding the best block which involves evaluating $\nu(\mathcal{S})$ for $\mathcal{S} = \mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_K$, i.e., a total of $\binom{K}{1}$ searches. Let $\mathcal{S}_1 = \mathcal{B}_1^{\star}$ be the optimal support. Now, we look for the optimal support composed of two blocks. Ide-

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

[3]Our algorithm applies to the general case when the C sized blocks could be placed arbitrarily within $\mathbf{x}_B$. However, due to space limitation we focus on the special case (3).

[4]This is essentially minimum-variance unbiased estimator (MVUE) which renders the estimate (5) itself an MVU estimate. 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.

1: **procedure** BLOCK-SABMP($\mathbf{\Phi}, \mathbf{y}, \mathbf{x}_B, r_{stop}$)
2: **estimate** λ, and $\sigma_{\mathbf{n}}^2$ as follows
3: $i^\star \leftarrow \underset{i \in [1,\cdots,N]}{\arg\max} \left|\boldsymbol{\phi}_i^{\mathsf{H}}\mathbf{y}\right|$
4: $\rho_i^\star \leftarrow \begin{cases} 0, & \text{if } \left|\boldsymbol{\phi}_i^{\mathsf{H}}\mathbf{y}\right| < \left|\boldsymbol{\phi}_{i^\star}^{\mathsf{H}}\mathbf{y}\right|/2 \\ 1, & \text{otherwise} \end{cases}$
5: $\lambda \leftarrow \left(\left\lceil \sum_{i=1}^{N} \rho_i^\star / C \right\rceil / K \right)$
6: **repeat**
7: $P \leftarrow K\lambda + \epsilon$
8: $\lambda_{old} \leftarrow \lambda$
9: $\{\mathcal{S}^d, p(\mathcal{S}^d|\mathbf{y}), \mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}^d]\} \leftarrow \mathrm{G}(\mathbf{\Phi}, \mathbf{y}, \lambda, \sigma_{\mathbf{n}}^2, P, \mathbf{x}_B)$
10: $\hat{\mathcal{S}}_{map} \leftarrow \arg\max_{\mathcal{S}} p(\mathcal{S}|\mathbf{y})$
11: $\hat{\mathbf{x}}_{map} \leftarrow \mathbb{E}[\mathbf{x}|\mathbf{y}, \hat{\mathcal{S}}_{map}]$
12: $\hat{\mathbf{x}}_{ammse} \leftarrow \sum_{\mathcal{S} \in \mathcal{S}^d} p(\mathcal{S}|\mathbf{y})\mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}]$
13: $\lambda \leftarrow \left\lceil \|\hat{\mathbf{x}}_{map}\|_0 / C \right\rceil / K$
14: $\sigma_{\mathbf{n}}^2 \leftarrow \mathrm{var}(\mathbf{y} - \mathbf{\Phi}\hat{\mathbf{x}}_{ammse})$
15: **until** $|\lambda - \lambda_{old}|/\lambda_{old} < r_{stop}$
16: **return** $\hat{\mathbf{x}}_{ammse}$
17: **end procedure**

Table 1: Block-SABMP

ally, this involves a search over a space of size $\binom{K}{2}$. To reduce the search space, however, we pursue a greedy approach and look for the block $\mathcal{B}_2^\star \neq \mathcal{B}_1^\star$ such that $\mathcal{S}_2 = \{\mathcal{B}_1^\star, \mathcal{B}_2^\star\}$ maximizes $\nu(\mathcal{S}_2)$. This involves $\binom{K-1}{1}$ searches (as opposed to the optimal search over a space of size $\binom{K}{2}$). We continue in this manner by forming $\mathcal{S}_3 = \{\mathcal{B}_1^\star, \mathcal{B}_2^\star, \mathcal{B}_3^\star\}$ and searching for $\mathcal{B}_3^\star$ in the remaining $K-2$ blocks and so on until we reach $\mathcal{S}_P = \{\mathcal{B}_1^\star, \ldots, \mathcal{B}_P^\star\}$. The value of P is selected to be slightly larger than the expected number of active blocks in the constructed signal such that $\Pr(\frac{|\mathcal{S}|}{C} > P)$ is sufficiently small[5].

One point to note here is that in our greedy move from $\mathcal{S}_j$ to $\mathcal{S}_{j+1}$, we need to evaluate $\nu(\mathcal{S}_j \cup \mathcal{B}_{j+1})$ around K times, which can be done in an order-recursive manner starting from that of $\nu(\mathcal{S}_j)$. Specifically, we note that every expansion, $\mathcal{S}_j \cup \mathcal{B}_{j+1}$, from $\mathcal{S}_j$ requires a calculation of $\nu(\mathcal{S}_j \cup \mathcal{B}_{j+1})$ using (11) which can be done in an order-recursive manner by considering the elements of $\mathcal{B}_{j+1}$ one at a time. 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 $\mathbf{x}_{ammse}$ in (10) 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 (5) lends itself as an intermediate computation performed to calculate $\nu(\mathcal{S})$.

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, $\sigma_{\mathbf{n}}^2$, and the sparsity rate, λ. However, the proposed method can bootstrap itself and does not require the user to provide any initial estimate of λ and $\sigma_{\mathbf{n}}^2$. Instead the method starts by finding initial estimates of these parameters and repeat the greedy algorithm until the parameters converge. The estimate of $\mathbf{x}_{mmse}$ available at this stage is then returned. The formal algorithmic description of our algorithm is presented in Table 1 where the procedure 'G' on line 9 refers to the greedy steps mentioned above. Moreover, the MATLAB code for our algorithm, called block-support agnostic Bayesian matching pursuit (block-SABMP), is provided on the author's website.[6]

4. EFFICIENT COMPUTATION OF THE DOMINANT SUPPORT SELECTION METRIC

Assume that the likelihood has been calculated for $\mathcal{S}^\star$ and we know $\nu(\mathcal{S}^\star)$. Let us see how we can update the value of $\nu(\mathcal{S}^\star)$ to $\nu(\mathcal{S}^\star \cup \mathcal{B}_i)$ (i.e., how to calculate the likelihood to include an additional block $\mathcal{B}_i = \{b_{i_1}, b_{i_2}, \ldots, b_{i_C}\}$). This can be done by adding the C elements one by one, i.e., by performing the sequence of recursive updates $\nu(\mathcal{S}^\star) \rightarrow \nu(\mathcal{S}^\star \cup \{b_{i_1}\}) \rightarrow \nu(\mathcal{S}^\star \cup \{b_{i_1}, b_{i_2}\}) \cdots \rightarrow \nu(\mathcal{S}^\star \cup \{b_{i_1}, \ldots, b_{i_C}\})$.

By inspection of $\nu(\mathcal{S})$ in (11), we see that the main challenge is in calculating the term $\left\|\left(\mathbf{\Phi}_{\mathcal{S}}(\mathbf{\Phi}_{\mathcal{S}}^{\mathsf{H}}\mathbf{\Phi}_{\mathcal{S}})^{-1}\mathbf{\Phi}_{\mathcal{S}}^{\mathsf{H}}\mathbf{y}\right)\right\|^2$ which can be written in terms of the expectation as $\|\mathbf{\Phi}_{\mathcal{S}}\mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}]\|^2$. So, we mainly need to update $\mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}]$. 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 $\mathbf{e}_{\mathbf{y},k-1}(\underline{\mathcal{S}}) \triangleq \mathbb{E}[\mathbf{x}|\mathbf{y}, \underline{\mathcal{S}}]$ to obtain[7] $\mathbf{e}_{\mathbf{y},k}(\mathcal{S}) = \mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}]$. Note that since $\mathcal{S} = \underline{\mathcal{S}} \cup \{s_k\}$, we can write

$$\mathbf{e}_{\mathbf{y},k}(\mathcal{S}) = \left(\begin{bmatrix} \mathbf{\Phi}_{\underline{\mathcal{S}}}^{\mathsf{H}} \\ \\ \boldsymbol{\phi}_{s_k}^{\mathsf{H}} \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_{\underline{\mathcal{S}}} \boldsymbol{\phi}_{s_k} \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{\Phi}_{\underline{\mathcal{S}}}^{\mathsf{H}}\mathbf{y} \\ \\ \boldsymbol{\phi}_{s_k}^{\mathsf{H}}\mathbf{y} \end{bmatrix}. \tag{12}$$

By using the block inversion formula to express the inverse of the above and simplifying, we get

$$\mathbf{e}_{\mathbf{y},k}(\mathcal{S}) = \begin{bmatrix} \Gamma\mathbf{e}_{\boldsymbol{\phi},k}(\mathcal{S}) + \mathbf{e}_{\mathbf{y},k-1}(\underline{\mathcal{S}}) \\ \\ -\Gamma \end{bmatrix} \tag{13}$$

where $\Gamma = \frac{1}{f_{\mathcal{S}}}(\mathbf{q}_{\boldsymbol{\phi},k}^{\mathsf{H}}(\mathcal{S})\mathbf{e}_{\mathbf{y},k-1}(\underline{\mathcal{S}}) - \mathbf{e}_{\mathbf{y},1}(s_k))$. This recursion is initialized by $\mathbf{e}_{\mathbf{y},1}(i) = (\boldsymbol{\phi}_s^{\mathsf{H}}\boldsymbol{\phi}_s)^{-1}\boldsymbol{\phi}_s^{\mathsf{H}}\mathbf{y}$. The recursion also depends on $\mathbf{q}_{\boldsymbol{\phi},k}(\mathcal{S}) \triangleq \mathbf{\Phi}_{\underline{\mathcal{S}}}^{\mathsf{H}}\boldsymbol{\phi}_{s_k}$, $\mathbf{e}_{\boldsymbol{\phi},k}(\mathcal{S}) \triangleq (\mathbf{\Phi}_{\underline{\mathcal{S}}}^{\mathsf{H}}\mathbf{\Phi}_{\underline{\mathcal{S}}})^{-1}\mathbf{\Phi}_{\underline{\mathcal{S}}}^{\mathsf{H}}\boldsymbol{\phi}_{s_k}$ and $f_{\mathcal{S}} \triangleq 1 - \mathbf{q}_{\boldsymbol{\phi},k}^{\mathsf{H}}(\mathcal{S})\mathbf{e}_{\boldsymbol{\phi},k}(\mathcal{S})$. The recursions for $\mathbf{e}_{\boldsymbol{\phi},k}(\mathcal{S})$, and $\mathbf{q}_{\boldsymbol{\phi},k}(\mathcal{S})$ may be determined as follows[8]

$$\mathbf{e}_{\boldsymbol{\phi},k+1}(\overline{\mathcal{S}}) = \begin{bmatrix} \Lambda\mathbf{e}_{\boldsymbol{\phi},k}(\mathcal{S}) + \mathbf{e}_{\boldsymbol{\phi},k}(\underline{\mathcal{S}}; s_{k+1}) \\ \\ -\Lambda \end{bmatrix} \tag{14}$$

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

$$\mathbf{q}_{\boldsymbol{\phi},k+1}(\overline{\mathcal{S}}) = \begin{bmatrix} \mathbf{\Phi}_{\underline{\mathcal{S}}}^{\mathsf{H}} \\ \boldsymbol{\phi}_{s_k}^{\mathsf{H}} \end{bmatrix} \boldsymbol{\phi}_{s_{k+1}} = \begin{bmatrix} \mathbf{q}_{\boldsymbol{\phi},k}(\underline{\mathcal{S}}; s_{k+1}) \\ \mathbf{q}_{\boldsymbol{\phi},2}(s_k; s_{k+1}) \end{bmatrix} \tag{15}$$

The two recursions (14) and (15) 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 $s_1, s_2 = 1, 2, \ldots, N$.

[5] Support of the constructed signal, follows the binomial distribution $\mathcal{B}(K, \lambda)$, which can be approximated by the Gaussian distribution $\mathcal{N}(K\lambda, K\lambda(1-\lambda))$ if $K\lambda > 5$. For this case, $\Pr(\frac{|\mathcal{S}|}{C} > P) = \frac{1}{2}\mathrm{erfc}\frac{P - K\lambda}{\sqrt{2K\lambda(1-\lambda)}}$.

[6] The MATLAB code of the block-SABMP algorithm presented in this paper is provided at http://faculty.kfupm.edu.sa/ee/naffouri/publications.html

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

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

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 block-SABMP algorithm, we compare it with the known block partition version of cluster-SBL algorithm [14] and Block-CoSaMP [6]. Since there are various versions of cluster SBL algorithms, we selected the one (BSBL-EM) which performed best among its other versions. This algorithm takes into account the intra-block coherence and performs EM updates of required parameters. The reason cluster-SBL was selected is that it was shown to outperform a number of algorithms, including Block-OMP [8], CluSS-MCMC [16], DGS [17], and Mixed ℓ_2/ℓ_1 norm [3] algorithms. However, Block-CoSaMP was selected due to its known robustness. Comparisons show that block-SABMP performs where both algorithms fail.

Experiments were conducted for signals whose active elements were drawn from Gaussian as well as non-Gaussian distributions. Entries of $M \times N$ measurement matrix $\mathbf{\Phi}$ were i.i.d., with zero means and complex Gaussian distribution. Columns of Φ were normalized to the unit norm. The size of $\mathbf{\Phi}$ was different for both experiments and is mentioned therein. The noise had a zero mean and was white and Gaussian, $\mathcal{CN}(\mathbf{0}, \sigma_{\mathbf{n}}^2\mathbf{I}_M)$, with $\sigma_{\mathbf{n}}^2$ determined according to the desired signal-to-noise ratio (SNR). Finally, we used two different metrics for performance measure; the normalized mean-squared error (NMSE) between the original signal, $\mathbf{x}$, and its MMSE estimate, $\hat{\mathbf{x}}_{ammse}$ defined by $10\log_{10}\left(\|\hat{\mathbf{x}}_k - \mathbf{x}_k\|^2 / \|\mathbf{x}_k\|^2\right)$, 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 was considered successful when the condition NMSE ≤ -10 dB was satisfied. The number of trials performed for computing NMSE was 200 while that for success rate was 1000.

5.1. Experiment 1: Performance under different undersampling ratios

In this experiment we fixed $N = 256$ and varied M to study the behavior of the two algorithms for different undersampling ratios. The block sparse signal consisted of 16 blocks, out of which 6 were active. For the signal size of $N = 256$ this amounted to a sparsity of 0.375. SNR was kept at 20 dB. Fig. 1 shows the success rate plotted versus varying number of measurements. It is obvious that Block-SABMP required lesser number of measurements as compared to cluster-SBL, for both Gaussian and non-Gaussian inputs, to achieve a perfect success rate. Block-CoSaMP only started performing for $M > 130$ therefore its results are not shown in this figure. The figure also shows that block-SABMP is robust to the distribution of the non-zero elements of vector $\mathbf{x}$ while the performance of BSBL-EM degraded a little when the distribution was non-Gaussian. This ascertains our claim that our algorithm is agnostic to signal statistics.

5.2. Experiment 2: Performance under varying sparsity rate

Recall that the value of sparsity rate used for Experiment 1 was quite high (37.5% non-zero entries) which meant that the signal had more active blocks and thus was less sparse. Though we demonstrated the robust behavior of our algorithm for less sparse signals, we would also like to compare our algorithm's performance for low values of sparsity rate. In this scenario, since the errors are very small, comparing success rate will not give the complete picture. Thus we plot NMSE versus sparsity in Fig. 2. Specifically, we consider $M = 128, N = 512$ and SNR$= 10$ dB. Blocks of size $C = 4$ were considered and the number of active blocks were varied from 1 to 10 in order to study the performance for signals having up to 8% non-zero entries. It is obvious from the graphs that block-SABMP outperforms both the cluster-SBL and block-CoSaMP algorithms. It is also obvious from the figure that unlike other algorithms the performance of block-SABMP does not depend on the distribution of the non-zero elements of vector $\mathbf{x}$. Note that in this experiment the SNR was kept low as compared to the previous one to demonstrate the effectiveness of our algorithm.

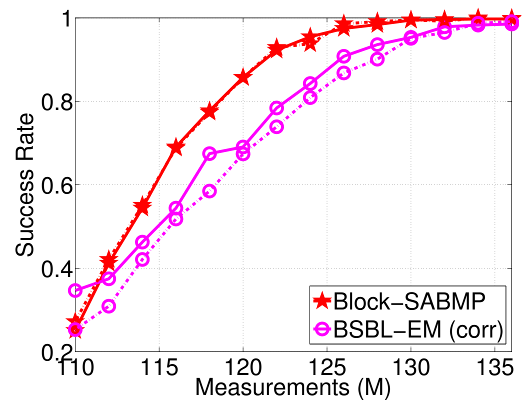

Fig. 1: Experiment 1: Success rate vs measurements. Distribution of non-zero elements: Gaussian (solid), non-Gaussian (dotted)

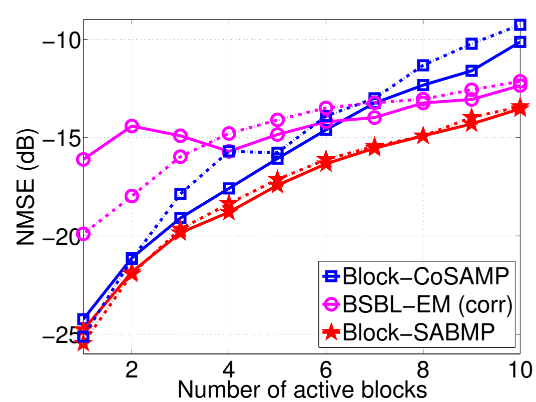

Fig. 2: Experiment 2: NMSE vs sparsity rate. Distribution of non-zero elements: Gaussian (solid), non-Gaussian (dotted)

6. CONCLUSION

In this paper, we introduced a robust Bayesian matching pursuit algorithm based on a fast recursive method for block-sparse signal recovery. Compared with other Bayesian approaches, it does not require the active blocks 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 under extreme conditions (e.g., very high sparsity/measurement ratio, low SNR) irrespective of the distribution of the active blocks.

7. REFERENCES

[1] Hwanjoon Kwon and Bhaskar D. Rao, "On the benefits of the block-sparsity structure in sparse signal recovery," in *ICASSP*, 2012, pp. 3685–3688.

[2] Farzad Parvaresh, Haris Vikalo, Sidhant Misra, and Babak Hassibi, "Recovering Sparse Signals Using Sparse Measurement Matrices in Compressed DNA Microarrays," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 3, pp. 275–285, June 2008.

[3] Yonina C. Eldar and Moshe Mishali, "Robust Recovery of Signals From a Structured Union of Subspaces," *IEEE Transactions on Information Theory*, vol. 55, no. 11, pp. 5302–5316, Nov. 2009.

[4] E. B. Al-Safadi and Tareq Y. Al-Naffouri, "Peak Reduction and Clipping Mitigation in OFDM by Augmented Compressive Sensing," *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 3834–3839, Jul. 2012.

[5] I.F. Gorodnitsky, J.S. George, and B.D. Rao, "Neuromagnetic Source Imaging with FOCUSS - A Recursive Weighted Minimum Norm Algorithm," *Electroencephalography and Clinical Neurophysiology*, vol. 95, no. 4, pp. 231–251, Oct. 1995.

[6] Richard G. Baraniuk, Volkan Cevher, Marco F. Duarte, and Chinmay Hegde, "Model-Based Compressive Sensing," *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 1982–2001, Apr. 2010.

[7] M Yuan and Y Lin, "Model selection and estimation in regression with grouped variables," *Journal of The Royal Statistical Society Series B-Statistical Methodology*, vol. 68, no. Part 1, pp. 49–67, 2006.

[8] Yonina C. Eldar, Patrick Kuppinger, and Helmut Boelcskei, "Block-Sparse Signals: Uncertainty Relations and Efficient Recovery," *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 3042–3054, Jun. 2010.

[9] Y. C. Pati, R. Rezaifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to Wavelet decomposition," in *Proc. 27th Asilomar Conf. Signals, Systems, Comput.*, Nov 1993, pp. 40–44.

[10] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, May 2009.

[11] Thomas Blumensath and Mike E. Davies, "Iterative hard thresholding for compressed sensing," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, Nov. 2009.

[12] V. Cevher, , M. F. Duarte, C. Hegde, and R. G. Baraniuk, "Sparse signal recovery using Markov Random Fields," in *NIPS*, Vancouver, B.C., Canada, 8–11 December 2008.

[13] Ahmed A. Quadeer and Tareq Y. Al-Naffouri, "Structure-Based Bayesian Sparse Reconstruction," *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6354–6367, DEC 2012.

[14] Zhilin Zhang and Bhaskar D. Rao, "Recovery of block sparse signals using the framework of block sparse bayesian learning," in *ICASSP*, 2012, pp. 3345–3348.

[15] Mudassir Masood and Tareq Y. Al-Naffouri, "A Fast Non-Gaussian Bayesian Matching Pursuit Method for Sparse Reconstruction," *IEEE Transactions on Signal Processing*, submitted.

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

[17] Junzhou Huang, Xiaolei Huang, and Dimitris N. Metaxas, "Learning with dynamic group sparsity," in *ICCV*, 2009, pp. 64–71.