Sparse Reconstruction Using Distribution Agnostic Bayesian Matching Pursuit

Mudassir Masood, *Student Member, IEEE*, and Tareq Y. Al-Naffouri, *Member, IEEE*

***Abstract*—A fast matching pursuit method using a Bayesian approach is introduced for sparse signal recovery. This method performs Bayesian estimates of sparse signals even when the signal prior is non-Gaussian or unknown. It is agnostic on signal statistics and utilizes *a priori* statistics of additive noise and the sparsity rate of the signal, which are shown to be easily estimated from data if not available. The method 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 sparse signal. Simulation results demonstrate the power and robustness of our proposed estimator.**

***Index Terms*—Basis selection, Bayesian, compressed sensing, greedy algorithm, linear regression, matching pursuit, minimum mean-square error (MMSE) estimate, sparse reconstruction.**

I. Introduction

SPARSITY is a feature that is abundant in both natural and man-made signals. Some examples of sparse signals include those from speech, images, videos, sensor arrays (e.g., temperature and light sensors), seismic activity, galactic activities, biometric activity, radiology, and frequency hopping. Given the vast existence of signals, their sparsity is an attractive property because the exploitation of this sparsity may be useful in the development of simple signal processing systems. Some examples of systems in which *a priori* knowledge of signal sparsity is utilized include motion estimation [1], magnetic resonance imaging (MRI) [2], impulse noise estimation and cancellation in DSL [3], network tomography [4], and peak-to-average-power ratio reduction in OFDM [5], [25]. All of these systems are based on sparsity-aware estimators such as Lasso [6], basis pursuit [7], structure-based estimator [8], [26], fast Bayesian matching pursuit [9], and estimators related to the relatively new area of compressed sensing [10]–[12].

Compressed sensing (CS), otherwise known as compressive sampling, has found many applications in the fields of communications, image processing, medical imaging, and networking. CS algorithms have been shown to recover sparse signals from underdetermined systems of equations that take the form

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

where $\mathbf{x} \in \mathbb{C}^N$, and $\mathbf{y} \in \mathbb{C}^M$ are the unknown sparse signal and the observed signal, respectively. Furthermore, $\mathbf{\Phi} \in \mathbb{C}^{M\times N}$ is the measurement matrix and $\mathbf{n} \in \mathbb{C}^M$ is the additive Gaussian noise vector. Here, the number of unknown elements, N, is much larger than the number of observations, M. CS uses linear projections of sparse signals that preserve structure of signals; furthermore, these projections are used to reconstruct the sparse signal using ℓ_1-optimization with high probability.

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ such that } \|\mathbf{y} - \mathbf{\Phi}\mathbf{x}\|_2 \leq \delta, \tag{2}$$

where $\delta = \sqrt{\sigma_{\mathbf{n}}^2(M + \sqrt{2M})}$. ℓ_1-optimization is a convex optimization problem that conveniently reduces to a linear program known as basis pursuit, which has the high computational complexity of $\mathcal{O}(N^3)$. Other more efficient algorithms such as orthogonal matching pursuit (OMP) [13] and the algorithm proposed by Haupt *et al.* [14] have been proposed. These algorithms fall into the category of greedy algorithms that are relatively faster than basis pursuit. However, it should be noted that the only *a priori* information utilized by these systems is the sparsity information.[1]

Another category of methods based on the Bayesian approach considers complete *a priori* statistical information of sparse signals. The fast Bayesian matching pursuit (FBMP) [9], [15] adopts such an approach and assumes a Bernoulli-Gaussian prior on the unknown vector $\mathbf{x}$. This method performs sparse signal estimation via model selection and model averaging. The sparse vector is described as a mixture of several components, the selection of which is based on successive model expansion. FBMP obtains an approximate MMSE estimate of the unknown signal with high accuracy and low complexity. It was shown to outperform several sparse recovery algorithms, including OMP [13], StOMP [16], GPSR-Basic [17], Sparse Bayes [18], BCS [19] and a variational-Bayes implementation of BCS

[1]The advantage of these approaches is that they are distribution agnostic and hence demonstrate robust performance.

Manuscript received June 11, 2012; revised November 23, 2012, June 23, 2013, and July 30, 2013; accepted July 31, 2013. Date of publication August 15, 2013; date of current version September 23, 2013. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Piotr Indyk. This work was funded in part by a CRG2 grant CRG_R2_13_ALOU_KAUST_2 from the Office of Competitive Research (OCRF) at King Abdullah University of Science and Technology (KAUST). The work of T.Y. Al-Naffouri was also supported by King Abdulaziz City for Science and Technology (KACST) through the Science & Technology Unit at King Fahd University of Petroleum & Minerals (KFUPM) through Project No. 09-ELE763-04 as part of the National Science, Technology and Innovation Plan.

M. Masood is with the Department of Electrical Engineering, King Abdullah University of Science & Technology, Thuwal 23955-6900, Saudi Arabia (e-mail: mudassir.masood@kaust.edu.sa).

T. Y. Al-Naffouri is with the Department of Electrical Engineering, King Abdullah University of Science & Technology, Thuwal 23955-6900, Saudi Arabia, and also with the Department of Electrical Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia (e-mail: tareq.alnaffouri@kaust.edu.sa).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2013.2278814

1053-587X © 2013 IEEE

[20]. However, there are several drawbacks associated with this method. The assumption that the support distribution[2] is Gaussian is not realistic, because, in most real-world scenarios, it is not Gaussian, or it is unknown. In addition, its performance is dependent on the knowledge of parameters of the Gaussian and Bernoulli priors, which are usually difficult to compute. Although a parameter estimation process is proposed, it is dependent on knowledge of the initial estimates of these signal parameters. The estimation process, in turn, has a negative impact on the complexity of the method.

Another popular Bayesian method proposed by Larsson and Selén [21] computes the MMSE estimate of the unknown vector, $\mathbf{x}$. Its approach is similar to that of FBMP in the sense that the sparse vector is described as a mixture of several components that are selected based on successive model reduction. It also requires knowledge of the noise and signal statistics. However, it was found that the MMSE estimate is insensitive to the *a priori* parameters and therefore an empirical-Bayesian variant that does not require any *a priori* knowledge of the data was devised.

Babacan *et al.* [22] have also proposed a greedy algorithm based on a Bayesian framework. They utilize a hierarchical form of the Laplace prior to model the sparsity of unknown signal. Their fast approach is fully automatic and does not require user intervention. They have also shown that their technique outperforms several sparse recovery algorithms. The list includes all of the algorithms which FBMP used to compare their performance.

The Bayesian approaches mentioned above in [9], [15] and [21] assume Gaussian prior on the non-zero elements of the unknown sparse vector[3] $\mathbf{x}$, while the Bayesian approach of [22] assumes a Laplace prior. It is reasonable to assume that any additive noise, generated at the sensing end, is Gaussian. However, assuming the signal statistics to be Bernoulli-Gaussian or Laplacian does not always reflect reality. Moreover, regardless of whether the actual prior is Gaussian or not, the parameters (mean and variance) of the Gaussian prior to be used need to be estimated, which is challenging especially when the signal statistics are not i.i.d. In that respect, one can appreciate convex relaxation approaches that are agnostic to signal statistics.

In this paper, we pursue a Bayesian approach for sparse signal reconstruction 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 support distribution. 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. Therefore, it is agnostic to variations in distributions.
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.

The fact that our approach is agnostic to support distribution motivates us to call it Support Agnostic Bayesian Matching Pursuit (SABMP). The remainder of this paper is organized as follows. In Section II, we formulate the problem and present the MMSE setup in the non-Gaussian/unknown statistics case. In Section III, we describe our greedy algorithm that is able to obtain the approximate MMSE estimate of the sparse vector followed by a description of our hyperparameter estimation process. Section IV demonstrates how the greedy algorithm can be made faster by calculating various metrics in a recursive manner. This is followed by Section V in which we present our simulation results and in Section VI, we conclude the paper.

A. Notation

We denote scalars with small-case letters (e.g., x), vectors with small-case, bold-face letters (e.g., $\mathbf{x}$), matrices with upper-case, bold-face letters (e.g., $\mathbf{X}$), and we reserve calligraphic notation (e.g., $\mathcal{S}$) for sets. We use $\mathbf{x}_i$ to denote the ith column of matrix $\mathbf{X}$, $x(j)$ to denote the jth entry of vector $\mathbf{x}$, and $\mathcal{S}_i$ to denote a subset of a set $\mathcal{S}$. We also use $\mathbf{X}_{\mathcal{S}}$ to denote the sub-matrix formed by the columns $\{\mathbf{x}_i : i \in \mathcal{S}\}$, indexed by set $\mathcal{S}$. Finally, we use $\hat{\mathbf{x}}$, $\mathbf{x}^*$, $\mathbf{x}^{\mathsf{T}}$, and $\mathbf{x}^{\mathsf{H}}$ to respectively denote the estimate, conjugate, transpose, and conjugate transpose of the vector $\mathbf{x}$.

II. Problem Formulation and MMSE Setup

A. The Signal Model

The analysis in this paper considers estimating an $N \times 1$ sparse vector, $\mathbf{x}$, from an $M \times 1$ observations vector, $\mathbf{y}$. These observations obey the linear regression model

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x} + \mathbf{n}, \tag{3}$$

where $\mathbf{\Phi}$ is a known $M \times N$ regression matrix and $\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, \sigma_{\mathbf{n}}^2 \mathbf{I}_M)$ is the additive white Gaussian noise vector.

We shall assume that $\mathbf{x}$ has a sparse structure and is modeled as $\mathbf{x} = \mathbf{x}_A \circ \mathbf{x}_B$ where $\circ$ indicates Hadamard (element-by-element) multiplication. The vector $\mathbf{x}_A$ consists of elements that are drawn from some unknown distribution and the entries of $\mathbf{x}_B$ are drawn i.i.d. from a Bernoulli distribution with success probability p. We observe that the sparsity of vector $\mathbf{x}$ is controlled by p and, therefore, we call it the sparsity parameter/rate. Typically, in Bayesian estimation, the signal entries are assumed to be drawn from a Gaussian distribution but here we would like to emphasize that the distribution of the entries of $\mathbf{x}_A$ does not matter.[4]

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

[3]While the Bayesian approaches of [9], [15], and [21] assume a Gaussian prior, these approaches continued to work when this assumption is violated.

[4]The distribution may be unknown or known with unknown parameters or even Gaussian. Our developments are agnostic with regard to signal statistics.

B. MMSE Estimation of $\mathbf{x}$

To determine $\mathbf{x}$, we compute the MMSE estimate of $\mathbf{x}$ given observation $\mathbf{y}$. This estimate is formally defined by

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

where the sum is executed over all possible 2^N support sets of $\mathbf{x}$. In the following, we explain how the expectation $\mathbb{E}[\mathbf{x}|\mathbf{y},\mathcal{S}]$, the posterior $p(\mathcal{S}|\mathbf{y})$ and the sum in (4) can be evaluated.

Given the support $\mathcal{S}$, (3) becomes

$$\mathbf{y} = \mathbf{\Phi}_{\mathcal{S}}\mathbf{x}_{\mathcal{S}} + \mathbf{n}, \tag{5}$$

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, it is difficult or even impossible to compute the expectation $\mathbb{E}[\mathbf{x}|\mathbf{y},\mathcal{S}]$. Thus, the best we can do is to use the best linear unbiased estimate (BLUE)[5] as an estimate of $\mathbf{x}$. Therefore, we replace $\mathbb{E}[\mathbf{x}|\mathbf{y},\mathcal{S}]$ by the BLUE as follows,

$$\mathbb{E}[\mathbf{x}|\mathbf{y},\mathcal{S}] \leftarrow \left(\mathbf{\Phi}_{\mathcal{S}}^{\mathsf{H}}\mathbf{\Phi}_{\mathcal{S}}\right)^{-1}\mathbf{\Phi}_{\mathcal{S}}^{\mathsf{H}}\mathbf{y}. \tag{6}$$

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

$$p(\mathcal{S}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathcal{S})p(\mathcal{S})}{p(\mathbf{y})}. \tag{7}$$

The probability, $p(\mathbf{y})$, is a factor common to all posterior probabilities that appear in (7) and hence can be ignored. Since the elements of $\mathbf{x}$ are activated according to the Bernoulli distribution with success probability p, we have

$$p(\mathcal{S}) = p^{|\mathcal{S}|}(1-p)^{N-|\mathcal{S}|}. \tag{8}$$

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 $\mathbf{P}_{\mathcal{S}}^{\perp} = \mathbf{I} - \mathbf{P}_{\mathcal{S}} = \mathbf{I} - \mathbf{\Phi}_{\mathcal{S}}(\mathbf{\Phi}_{\mathcal{S}}^{\mathsf{H}}\mathbf{\Phi}_{\mathcal{S}})^{-1}\mathbf{\Phi}_{\mathcal{S}}^{\mathsf{H}}$. 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

$$\begin{aligned}\mathbf{K} &= \mathbb{E}\left[\left(\mathbf{P}_{\mathcal{S}}^{\perp}\mathbf{n}\right)\left(\mathbf{P}_{\mathcal{S}}^{\perp}\mathbf{n}\right)^{\mathsf{H}}\right] \\ &= \mathbf{P}_{\mathcal{S}}^{\perp}\mathbb{E}[\mathbf{n}\mathbf{n}^{\mathsf{H}}]\mathbf{P}_{\mathcal{S}}^{\perp\,\mathsf{H}} = \mathbf{P}_{\mathcal{S}}^{\perp}\sigma_{\mathbf{n}}^2\mathbf{P}_{\mathcal{S}}^{\perp\,\mathsf{H}} \\ &= \sigma_{\mathbf{n}}^2\mathbf{P}_{\mathcal{S}}^{\perp}.\end{aligned} \tag{9}$$

[5]This is essentially minimum-variance unbiased estimator (MVUE) which renders the estimate (6) 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.

Thus we have,[6]

$$p(\mathbf{y}|\mathcal{S}) \simeq \frac{1}{\sqrt{(2\pi\sigma_{\mathbf{n}}^2)^M}}\exp\left(-\frac{1}{2}\left(\mathbf{P}_{\mathcal{S}}^{\perp}\mathbf{y}\right)^{\mathsf{H}}\mathbf{K}^{-1}\left(\mathbf{P}_{\mathcal{S}}^{\perp}\mathbf{y}\right)\right). \tag{10}$$

Simplifying and dropping the pre-exponential factor yields,

$$p(\mathbf{y}|\mathcal{S}) \simeq \exp\left(-\frac{1}{2\sigma_{\mathbf{n}}^2}\left\|\mathbf{P}_{\mathcal{S}}^{\perp}\mathbf{y}\right\|^2\right). \tag{11}$$

Substituting (8) and (11) into (7) finally yields an expression for the posterior probability. In this way, we have all the ingredients to compute the sum in (4). Computing this sum is a challenging task when N is large because the number of support sets can be extremely large and the computational complexity can become unrealistic. To have a computationally feasible solution, this sum can be computed over a few support sets corresponding to significant posteriors. Let $\mathcal{S}^d$ be the set of supports for which the posteriors are significant. Hence, we arrive at an approximation to the MMSE estimate given by,

$$\hat{\mathbf{x}}_{ammse} = \sum_{\mathcal{S}\in\mathcal{S}^d} p(\mathcal{S}|\mathbf{y})\mathbb{E}[\mathbf{x}|\mathbf{y},\mathcal{S}]. \tag{12}$$

In the next section, we propose a greedy algorithm to find $\mathcal{S}^d$. Before proceeding, for ease of representation and convenience, we represent the posteriors in the log domain. In this regard, we define a dominant support selection metric, $\nu(\mathcal{S})$, to be used by the greedy algorithm as follows:

$$\begin{aligned}\nu(\mathcal{S}) &\triangleq \ln p(\mathbf{y}|\mathcal{S})p(\mathcal{S}) \\ &= \ln\exp\left(-\frac{1}{2\sigma_{\mathbf{n}}^2}\left\|\mathbf{P}_{\mathcal{S}}^{\perp}\mathbf{y}\right\|^2\right) + \ln\left(p^{|\mathcal{S}|}(1-p)^{N-|\mathcal{S}|}\right) \\ &= \frac{1}{2\sigma_{\mathbf{n}}^2}\left\|\mathbf{\Phi}_{\mathcal{S}}\left(\mathbf{\Phi}_{\mathcal{S}}^{\mathsf{H}}\mathbf{\Phi}_{\mathcal{S}}\right)^{-1}\mathbf{\Phi}_{\mathcal{S}}^{\mathsf{H}}\mathbf{y}\right\|^2 - \frac{1}{2\sigma_{\mathbf{n}}^2}\|\mathbf{y}\|^2 \ldots \\ &\quad + |\mathcal{S}|\ln p + (N-|\mathcal{S}|)\ln(1-p)\end{aligned} \tag{13}$$

III. Support Agnostic Bayesian Matching Pursuit (SABMP)

A. A Greedy Algorithm

We now present a greedy algorithm to determine the set of dominant supports, $\mathcal{S}^d$, required to evaluate $\hat{\mathbf{x}}_{ammse}$ in (12). 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

[6]Results in Section V show that indeed this approximation is justified. Moreover, to provide the reader a feel of the quality of this approximation, we are motivated to provide a relevant discussion in Appendix.

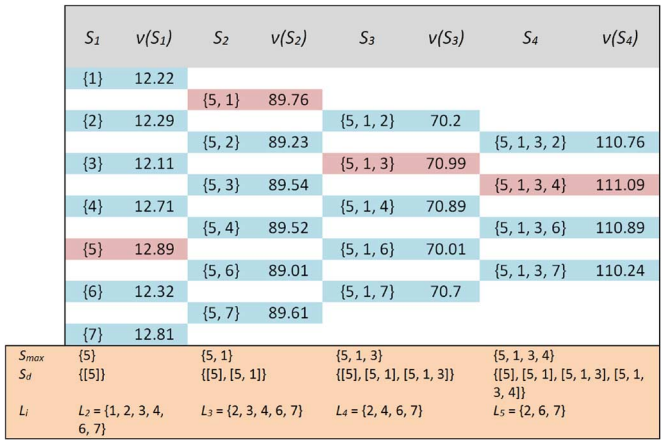


Fig. 1. An example run of the greedy algorithm for $N = 7$ and $P = 4$.

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 the constructed signal such that $\Pr(|\mathcal{S}| > P)$ is sufficiently small.[7] An example run of this algorithm for $N = 7$ and $P = 4$ is presented in Fig.1.

One point to note here is that in our greedy move from $\mathcal{S}_j$ to $\mathcal{S}_{j+1}$, we need to evaluate $\nu(\mathcal{S}_j \cup \{i_{j+1}\})$ 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 (13). This translates to appending a column $\boldsymbol{\phi}_{j+1}$ to $\boldsymbol{\Phi}_{\mathcal{S}_j}$ in the calculations of (13), which can be done in an order-recursive manner. We summarize these calculations in Section IV. This order-recursive approach reduces the calculation in each search step to an order of $\mathcal{O}(MN)$ operations down from $\mathcal{O}(MN^2)$ in the direct (non-recursive) approach. Since, we are searching for the best support of size up to P, we need to repeat this process P times and so the complexity we incur is of the order $\mathcal{O}(MNP)$.

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 (12) 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 (6) lends itself as an intermediate computation performed to calculate $\nu(\mathcal{S})$. We now present a formal algorithmic description of our greedy algorithm in Table I.

TABLE I
THE GREEDY ALGORITHM

procedure G($\boldsymbol{\Phi}, \mathbf{y}, p, \sigma_{\mathbf{n}}^2, P$)
  initialize $L \leftarrow \{1, 2, \ldots, N\}$, $i \leftarrow 1$
  initialize empty sets $\mathcal{S}_{max}$, $\mathcal{S}^d$, $p(\mathcal{S}^d|\mathbf{y})$, $\mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}^d]$
  $L_i \leftarrow L$
  while $i \leq P$ **do**
    $\Omega \leftarrow \{\mathcal{S}_{max} \cup \{\alpha_1\}, \mathcal{S}_{max} \cup \{\alpha_2\}, \cdots, \mathcal{S}_{max} \cup \{\alpha_{|L_i|}\} \mid \alpha_k \in L_i\}$
    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 $p(\mathcal{S}^\star|\mathbf{y}), \mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}^\star]$
    $p(\mathcal{S}^d|\mathbf{y}) \leftarrow \{p(\mathcal{S}^d|\mathbf{y}), p(\mathcal{S}^\star|\mathbf{y})\}$
    $\mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}^d] \leftarrow \{\mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}^d], \mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}^\star]\}$
    $\mathcal{S}_{max} \leftarrow \mathcal{S}^\star$
    $L_{i+1} \leftarrow L \setminus \mathcal{S}^\star$
    $i \leftarrow i + 1$
  end while
  return $\mathcal{S}^d, p(\mathcal{S}^d|\mathbf{y}), \mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}^d]$
end procedure

TABLE II
SUPPORT AGNOSTIC BAYESIAN MATCHING PURSUIT (SABMP)

procedure SABMP($\boldsymbol{\Phi}, \mathbf{y}, r_{stop}$)
  estimate p, $\sigma_{\mathbf{n}}^2$
  repeat
    $p_{old} \leftarrow p$
    $\{\mathcal{S}^d, p(\mathcal{S}^d|\mathbf{y}), \mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}^d]\} \leftarrow$ G($\boldsymbol{\Phi}, \mathbf{y}, p, \sigma_{\mathbf{n}}^2$)
    $\hat{\mathcal{S}}_{map} \leftarrow \arg\max_{\mathcal{S}} p(\mathcal{S}|\mathbf{y})$
    $\hat{\mathbf{x}}_{map} \leftarrow \mathbb{E}[\mathbf{x}|\mathbf{y}, \hat{\mathcal{S}}_{map}]$
    $\hat{\mathbf{x}}_{ammse} \leftarrow \sum_{\mathcal{S} \in \mathcal{S}^d} p(\mathcal{S}|\mathbf{y}) \mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}]$
    $p \leftarrow \|\hat{\mathbf{x}}_{map}\|_0 / N$
    $\sigma_{\mathbf{n}}^2 \leftarrow \text{var}(\mathbf{y} - \boldsymbol{\Phi}\hat{\mathbf{x}}_{ammse})$
  until $|p - p_{old}|/p_{old} < r_{stop}$
  return $\hat{\mathbf{x}}_{ammse}$
end procedure

B. 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, $\sigma_{\mathbf{n}}^2$, and the sparsity rate, p. However, the proposed SABMP method can bootstrap itself and does not require the user to provide any initial estimate of p and $\sigma_{\mathbf{n}}^2$. Instead the method starts by finding initial estimates of these parameters which are used to compute the dominant support selection metric $\nu(\mathcal{S})$ in (13). Since the decisions made by our greedy algorithm in support selection are influenced by the values of these parameters, we expect that refining these parameters will improve our chances of selecting the right support. The refinement demands that the greedy algorithm mentioned above be repeated with new estimates of sparsity rate and noise variance. In this way both the hyperparameters (p and $\sigma_{\mathbf{n}}^2$) and support are refined simultaneously. The repetition continues until a predetermined criterion has been satisfied. A description of the SABMP algorithm which repeatedly calls the greedy algorithm to estimate the hyperparameters and the unknown signal is provided in Table II.

C. Estimation and Refinement of the Hyperparameters p and $\sigma_{\mathbf{n}}^2$

When the hyperparameters p and $\sigma_{\mathbf{n}}^2$ are unknown, we need to calculate 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.

To calculate the initial estimate of the sparsity rate p, we project the observation vector $\mathbf{y}$ onto the basis vectors $\boldsymbol{\phi}_i$, $i =$

[7] $|\mathcal{S}|$, 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, $\Pr(|\mathcal{S}| > P) = \frac{1}{2}\text{erfc}\frac{P - Np}{\sqrt{2Np(1-p)}}$.

$1, \cdots, N$ (columns of $\mathbf{\Phi}$). We observe that these projections tend to be higher for those $\boldsymbol{\phi}_i$ which determine the actual support of the unknown signal. However, because of the underdetermined nature of the problem at hand, other projections could also be sufficiently high. Therefore, to start with, we could use this observation to find a rough estimate of p as follows:

$$p_{\text{init}} = \frac{\left|\left\{j \in \mathbb{N} : 1 \leq j \leq N \wedge \left|\phi_j^{\mathsf{H}}\mathbf{y}\right| \geq \frac{\|\phi^{\mathsf{H}}\mathbf{y}\|_\infty}{2}\right\}\right|}{N}$$

where $\|\phi^{\mathsf{H}}\mathbf{y}\|_\infty = \sup\{|\phi_i^{\mathsf{H}}\mathbf{y}| : 1 \leq i \leq N\}$ is the infinity-norm, $p_{\mathbf{init}}$ is the initial estimate of the sparsity rate p and the outer-most $|.|$ refers to the cardinality of set bounded by it. Note that our algorithm is robust enough to find the right support even if p is initialized badly. This feature of our algorithm has been demonstrated in Experiment 1 of the Results section. We would also like to point out 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 $\sigma^2_{\mathbf{n}_{\mathbf{init}}} = \frac{1}{4}\sigma^2_{\mathbf{y}}$ is good enough. We would like to point out that our method is robust to this initial estimate and performs quite well in estimating the actual noise variance.

In order to refine these parameters we might opt for finding the maximum-likelihood (ML) or maximum *a posteriori* (MAP) estimates using the expectation maximization (EM) algorithm. However, this will add to the computational complexity and is unnecessary as 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 posteriors $p(\mathcal{S}|\mathbf{y})$ and expectations $\mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}]$. These are used to compute the approximate MMSE $\hat{\mathbf{x}}_{ammse}$ from (12). Similarly, by determining $\hat{\mathcal{S}}_{map} = \arg\max_{\mathcal{S}\in\mathcal{S}^d} p(\mathcal{S}|\mathbf{y})$ we are able to determine $\hat{\mathbf{x}}_{map} = \mathbb{E}[\mathbf{x}|\mathbf{y}, \hat{\mathcal{S}}_{map}]$. Based on these quantities we update p and $\sigma^2_{\mathbf{n}}$ as follows:

$$\hat{p} = \frac{\|\hat{\mathbf{x}}_{map}\|_0}{N} \tag{14}$$

$$\hat{\sigma}^2_{\mathbf{n}} = \text{var}(\mathbf{y} - \mathbf{\Phi}\hat{\mathbf{x}}_{ammse}) \tag{15}$$

The greedy algorithm is called again with this new set of parameters. The output of which is then used to update p and $\sigma^2_{\mathbf{n}}$ again using (14) and (15). This process continues until the estimate of p changes by less than a prespecified factor (eg., we use 2% in simulations), or until a predetermined maximum number of iterations have been performed. The process is effective as the simulation results show that, in most case, $\hat{p}$ converges rapidly and the corresponding estimate of $\sigma^2_{\mathbf{n}}$ is also close to the actual noise variance (for example, see Fig. 6).

We would also like to highlight that the nature of our algorithm allows us to penalize the growth of support, thus encouraging sparser estimated signal. More specifically, since the description of $\nu(\mathcal{S})$ includes factors which grow as the support size $|\mathcal{S}|$ grows, the metric has an inherent capability of discouraging denser solutions. As explained in Section III-A, we compute the dominant support selection metric $\nu(\mathcal{S})$ at each stage of our greedy algorithm and based on the values of $\nu(\mathcal{S})$ we select the best supports. Therefore, unlike normal matching pursuit algorithms, our algorithm could select supports of varying sizes based on the values of $\nu(\mathcal{S})$.

In Section III-A, we mentioned that the greedy algorithm incurs a complexity of order $\mathcal{O}(MNP)$. However, in the refinement process the algorithm is repeated a number of times. Therefore, the computational complexity will increase to an order of $\mathcal{O}(E_{max}MNP)$, where E_{max} is the maximum number of times the greedy algorithm was repeated. For a detailed description of the steps followed by the method the algorithms are provided in Tables I and II. The MATLAB code for our algorithm, called support agnostic Bayesian matching pursuit (SABMP), is provided on the author's website.[8]

Remarks: While we use a greedy approach similar to those proposed by [9], [15], [21], [22], we would like to point out the distinctive features of our approach in the following.

1) Our approach is agnostic to the distribution of active elements of the sparse signal.
2) The updates required in the Bayesian matching pursuit with Gaussian prior (FBMP) are essentially those of order recursive least squares [23]. However, our method depends on the projection matrix $\mathbf{P}^{\perp}_{\mathcal{S}}$ which is not commonly encountered in signal processing algorithms. So the recursions required for updates in our algorithm had to be developed from scratch.
3) Since the method is support distribution agnostic, we do not need to estimate the parameters of distribution.
4) Our greedy algorithm still requires the sparsity rate and noise variance. That said, the nature of the algorithm does not change whether we know these parameters perfectly or not. In addition, the performance of the algorithm is robust to the initial estimates of these parameters (which are supplied directly from the data with no need for user intervention). In other words, our algorithm is automatic and is able to bootstrap itself. Contrast this with Gaussian Bayesian matching pursuit (FBMP) in which the algorithm needs to run many more times when the parameters are unknown and might actually not lead to satisfactory results if not initialized properly.[9]

IV. Efficient Computation of the Dominant Support Selection Metric

As explained in Section III-A, $\nu(\mathcal{S})$ requires extensive computation to determine the dominant supports. The computational complexity of the proposed algorithm is therefore largely dependent upon the way $\nu(\mathcal{S})$ is computed. In this section, a computationally efficient procedure to calculate this quantity is presented.

We note that the efficient computation of $\nu(\mathcal{S})$ depends mainly on the efficient computation of the term $\boldsymbol{\xi}_{\mathcal{S}} = \|(\mathbf{\Phi}_{\mathcal{S}}(\mathbf{\Phi}^{\mathsf{H}}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{S}})^{-1}\mathbf{\Phi}^{\mathsf{H}}_{\mathcal{S}}\mathbf{y})\|^2 = \|\mathbf{\Phi}_{\mathcal{S}}\mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}]\|^2$. Our focus is therefore on computing $\mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}]$ efficiently.

[8]The MATLAB code of the SABMP algorithm and the results from various experiments discussed in this paper are provided at http://faculty.kfupm.edu.sa/ee/naffouri/publications.html.

[9]See in particular Figs. 2(a) and 3(a) in Experiment 1 for the effect of initial p and Fig. 6(b) in Experiment 4 for the effect of parameter estimation on the run time.

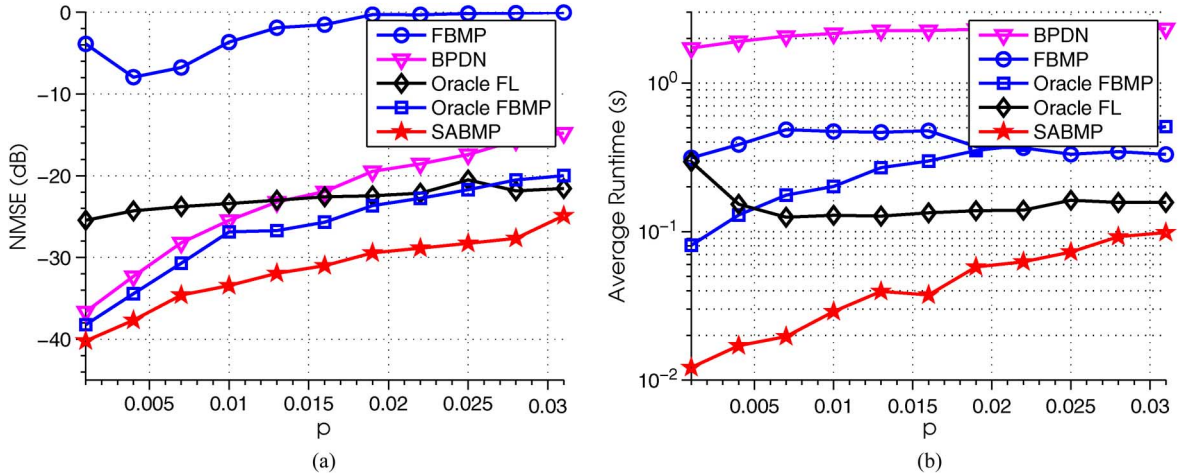


Fig. 2. NMSE and average runtime versus p graphs for Gaussian-i.i.d. input. $N = 512$, $M = 128$, $SNR = 20$ dB, $p_{\text{est}} = 0.003$. (a) NMSE versus p. (b) Runtime versus p.

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 following subset and superset, respectively:

$$\underline{\mathcal{S}} = \{s_1, s_2, s_3, \ldots, s_{k-1}\},$$
$$\overline{\mathcal{S}} = \{s_1, s_2, s_3, \ldots, s_{k+1}\},$$

where $s_k < s_{k+1}$. In the following, we demonstrate how to update $\mathbf{e}_{\mathbf{y},k-1}(\underline{\mathcal{S}}) \triangleq \mathbb{E}[\mathbf{x}_{\underline{\mathcal{S}}}|\mathbf{y}]$ to obtain[10] $\mathbf{e}_{\mathbf{y},k}(\mathcal{S}) = \mathbb{E}[\mathbf{x}_{\mathcal{S}}|\mathbf{y}]$. Here, we use $\mathcal{S}$ to designate the supports and thus $\mathbb{E}[\mathbf{x}_{\mathcal{S}}|\mathbf{y}]$ refers to $\mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}]$. We note that $\mathbf{e}_{\mathbf{y},k}(\mathcal{S})$ could be written as given in (16).

$$\mathbf{e}_{\mathbf{y},k}(\mathcal{S}) = \left(\mathbf{\Phi}_{\mathcal{S}}^{\mathsf{H}}\mathbf{\Phi}_{\mathcal{S}}\right)^{-1}\mathbf{\Phi}_{\mathcal{S}}^{\mathsf{H}}\mathbf{y} = \left(\begin{bmatrix}\mathbf{\Phi}_{\underline{\mathcal{S}}}^{\mathsf{H}}\\ \boldsymbol{\phi}_{s_k}^{\mathsf{H}}\end{bmatrix}[\mathbf{\Phi}_{\underline{\mathcal{S}}}\boldsymbol{\phi}_{s_k}]\right)^{-1}\begin{bmatrix}\mathbf{\Phi}_{\underline{\mathcal{S}}}^{\mathsf{H}}\mathbf{y}\\ \boldsymbol{\phi}_{s_k}^{\mathsf{H}}\mathbf{y}\end{bmatrix} \tag{16}$$

By using the block inversion formula to express the inverse of (16) and simplifying, we get the result as given in (17), shown at the bottom of the page.

This recursion is initialized by $\mathbf{e}_{\mathbf{y},1}(i) = (\boldsymbol{\phi}_s^{\mathsf{H}}\boldsymbol{\phi}_s)^{-1}\boldsymbol{\phi}_s^{\mathsf{H}}\mathbf{y}$ for $i = 1, 2, \ldots, N$. 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 $\mathbf{q}_{\boldsymbol{\phi},k}(\mathcal{S})$, and $\mathbf{e}_{\boldsymbol{\phi},k}(\mathcal{S})$ may be determined in a similar fashion as given in (18), shown at the bottom of the page, and[11]

$$\mathbf{q}_{\boldsymbol{\phi},k+1}(\overline{\mathcal{S}}) = \begin{bmatrix}\mathbf{\Phi}_{\underline{\mathcal{S}}}^{\mathsf{H}}\\ \boldsymbol{\phi}_{s_k}^{\mathsf{H}}\end{bmatrix}\boldsymbol{\phi}_{s_{k+1}} = \begin{bmatrix}\mathbf{q}_{\boldsymbol{\phi},k}(\underline{\mathcal{S}}; s_{k+1})\\ \mathbf{q}_{\boldsymbol{\phi},2}(s_k; s_{k+1})\end{bmatrix}. \tag{20}$$

The two recursions (18) and (20) 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 $\nu(\mathcal{S})$ as shown in (19), at the bottom of the page.

V. Results

To demonstrate the performance of the proposed SABMP, we compare it here with Fast Bayesian Matching Pursuit (FBMP) [9], [15], Fast Laplace (FL) algorithm [22] and the convex relaxation-based ℓ_1-optimization approach. The ℓ_1 problem given in (2) was solved using CVX, a package for specifying and solving convex programs [24]. This particular ℓ_1 problem is also referred to as Basis Pursuit Denoising (BPDN). For FBMP and FL their MATLAB implementations available on their respective websites were used. The reason FBMP and FL were selected is

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

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

$$\mathbf{e}_{\mathbf{y},k}(\mathcal{S}) = \begin{bmatrix}\frac{1}{f_{\mathcal{S}}}\left(\mathbf{q}_{\boldsymbol{\phi},k}^{\mathsf{H}}(\mathcal{S})\mathbf{e}_{\mathbf{y},k-1}(\underline{\mathcal{S}}) - \mathbf{e}_{\mathbf{y},1}(s_k)\right)\mathbf{e}_{\boldsymbol{\phi},k}(\mathcal{S}) + \mathbf{e}_{\mathbf{y},k-1}(\underline{\mathcal{S}})\\ \frac{-1}{f_{\mathcal{S}}}\left(\mathbf{q}_{\boldsymbol{\phi},k}^{\mathsf{H}}(\mathcal{S})\mathbf{e}_{\mathbf{y},k-1}(\underline{\mathcal{S}}) - \mathbf{e}_{\mathbf{y},1}(s_k)\right)\end{bmatrix} \tag{17}$$

$$\mathbf{e}_{\boldsymbol{\phi},k+1}(\overline{\mathcal{S}}) = \begin{bmatrix}\frac{1}{f_{\mathcal{S}}}\left(\mathbf{q}_{\boldsymbol{\phi},k}^{\mathsf{H}}(\mathcal{S})\mathbf{e}_{\boldsymbol{\phi},k}(\underline{\mathcal{S}}; s_{k+1}) - \mathbf{e}_{\boldsymbol{\phi},2}(s_k; s_{k+1})\right)\mathbf{e}_{\boldsymbol{\phi},k}(\mathcal{S}) + \mathbf{e}_{\boldsymbol{\phi},k}(\underline{\mathcal{S}}; s_{k+1})\\ \frac{-1}{f_{\mathcal{S}}}\left(\mathbf{q}_{\boldsymbol{\phi},k}^{\mathsf{H}}(\mathcal{S})\mathbf{e}_{\boldsymbol{\phi},k}(\underline{\mathcal{S}}; s_{k+1}) - \mathbf{e}_{\boldsymbol{\phi},2}(s_k; s_{k+1})\right)\end{bmatrix} \tag{18}$$

$$\nu_k(\mathcal{S}) = \frac{1}{\sigma_{\mathbf{n}}^2}\|\mathbf{\Phi}_{\mathcal{S}}\mathbf{e}_{\mathbf{y},k}(\mathcal{S})\|^2 - \frac{1}{\sigma_{\mathbf{n}}^2}\|\mathbf{y}\|^2 + |\mathcal{S}|\ln p + (N - |\mathcal{S}|)\ln(1-p). \tag{19}$$

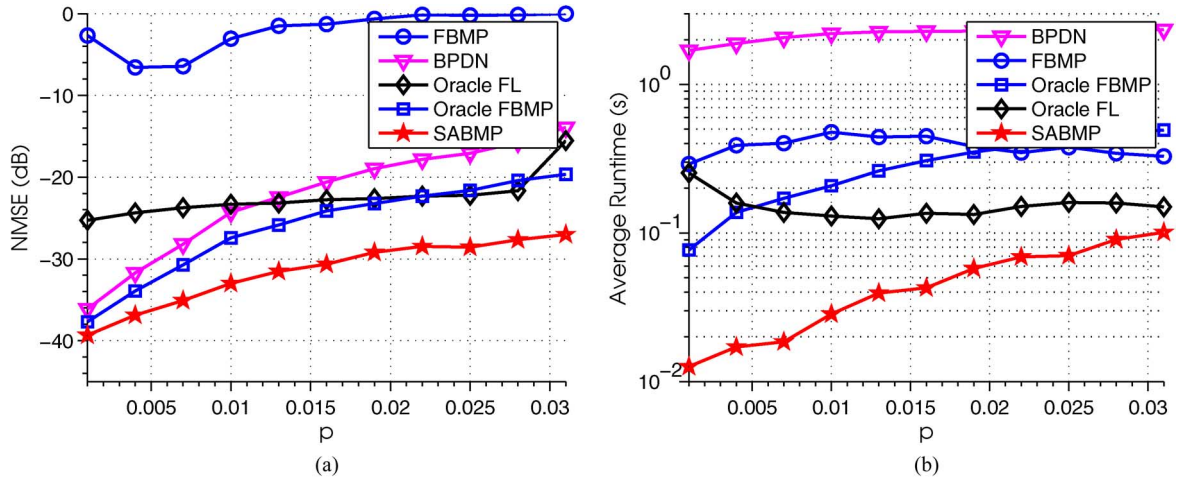


Fig. 3. NMSE and average runtime versus p graphs for Uniform non-i.i.d. input. $N = 512$, $M = 128$, SNR $= 20$ dB, $p_{\text{est}} = 0.003$. (a) NMSE versus p. (b) Runtime versus p.

that they have shown to outperform a number of the contemporary sparse signal recovery algorithms, including OMP [13], StOMP [16], GPSR-Basic [17], and BCS [19]. Comparisons with FBMP, FL and BPDN show that SABMP performs better than all of these algorithms and is robust to situations where these algorithms fail. We present comparisons for various signal settings which are discussed in detail in the following.

Signal Setup: Experiments were conducted for signals whose active elements are drawn from Gaussian as well as non-Gaussian distributions. The following configurations were used for the experiments:

1) Gaussian (i.i.d. ($\mu_{\mathbf{x}} = 10, \sigma_{\mathbf{x}}^2 = 2$)),
2) Non-Gaussian (Uniform, non-i.i.d. ($5 \leq \mu_{\mathbf{x}} \leq 10, 1 \leq \sigma_{\mathbf{x}}^2 \leq 2$)),
3) Unknown distribution (compressible signals)
4) Unknown distribution (for this example, different images with unknown statistics were used),

where $\mu_{\mathbf{x}}$ and $\sigma_{\mathbf{x}}^2$ refer to the mean and variance of the corresponding distributions, respectively.

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 size of $\mathbf{\Phi}$ selected for the experiments was $M = 128$, $N = 512$. 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). Initial estimates of the hyperparameters used for the simulations were $\mu_{\mathbf{x}\text{est}} = 0$, $\sigma_{\mathbf{x}\text{est}}^2 = \frac{1}{10} \times \sigma_{\mathbf{x}}^2$, $\sigma_{\mathbf{n}\text{est}}^2 = 10 \times \sigma_{\mathbf{n}}^2$, and $p_{\text{est}} = 0.003$, where these estimates were needed for FBMP.

In FBMP the greedy search is repeated a number of times D. For simulation purposes we used $D = 10$ unless otherwise stated. In all of the experiments, parameter refinement was performed for both SABMP and FBMP. For FBMP, the surrogate EM method proposed by its authors was used to refine the hyperparameters. The refinements were allowed to perform for a maximum of $E_{max} = 10$ iterations unless otherwise stated. For FL the default settings recommended by its authors were used. For fairness, support and amplitude refinement [3] procedures were performed on the results of the BPDN.[12] Moreover, the experiments also include results when true values of required parameters were provided to FBMP, FL and SABMP. Since there is no need for parameter estimation in this case, we used the versions of algorithms where parameter updation is not performed. We would refer to these particular instances of algorithms as oracle-FBMP, oracle-FL and oracle-SABMP while reserve FBMP, FL and SABMP, respectively, for their parameter refinement versions. Note that the code of FL does not allow us to control the refinement process. Therefore, oracle-FL will refer to the version of FL where we provide true values of parameters and refinement is also performed. Moreover, since FL works only on real data, in all of the experiments we generate real data for FL and complex for the other three algorithms. In doing so we make sure that all parameters of generated signal realizations are same. Please also note that our algorithm can deal with both real and complex data. Finally, the normalized mean-squared error (NMSE) between the original signal, $\mathbf{x}$, and its MMSE estimate, $\hat{\mathbf{x}}_{ammse}$, was used as the performance measure:

$$\text{NMSE} = 10 \log_{10} \left(\frac{1}{K} \sum_{k=1}^{K} \frac{\|\hat{\mathbf{x}}_k - \mathbf{x}_k\|^2}{\|\mathbf{x}_k\|^2} \right), \qquad (21)$$

where K is the number of trials NMSE was averaged over. The value of K used in experiments vary and is provided therein.

Experiment 1 (Signal Estimation Performance Comparison for Varying Sparsity Parameter p): In the first set of experiments, NMSE and mean runtime were measured for different values of sparsity parameter p and plotted to compare the performance of SABMP with FBMP, oracle-FBMP, FL and BPDN. The value of SNR selected for these experiments was 20 dB and the results were averaged over $K = 250$ trials.

Figs. 2 and 3 demonstrate the superiority of SABMP over all other algorithms. The figures suggest that, 1) even though true

[12]Actual parameter values were provided to BPDN instead of estimates; furthermore, support and amplitude refinement was also performed to demonstrate that, despite these measures, its performance was inferior to that of SABMP.

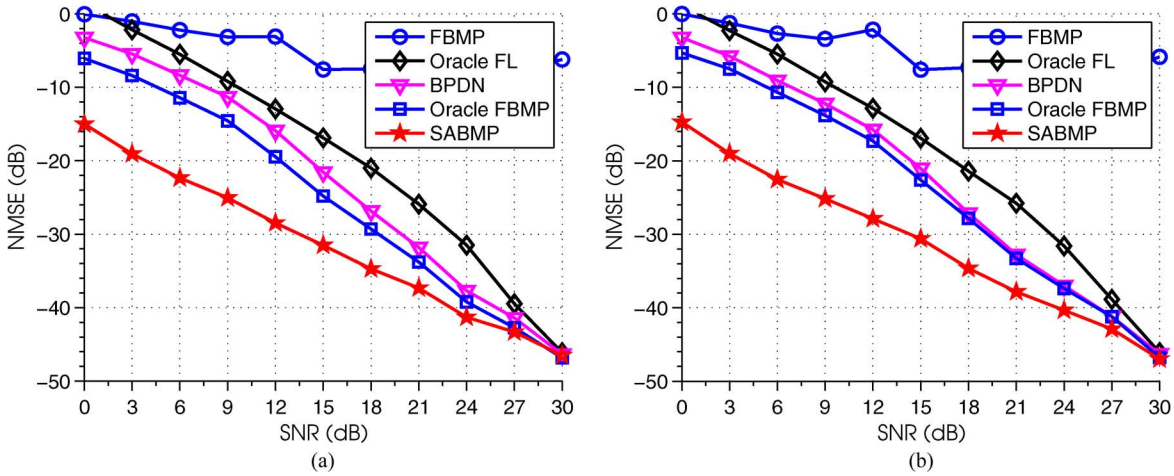


Fig. 4. NMSE versus SNR graphs for uniform non-i.i.d. and Gaussian i.i.d. inputs. $N = 512$, $M = 128$, $p = 0.005$, $p_{\text{est}} = 0.003$. (a) NMSE versus SNR for uniform non-i.i.d. input. (b) NMSE versus SNR for Gaussian i.i.d. input.

parameters were provided to all other algorithms, they were still outperformed by SABMP, and, 2) the huge gap in performance of FBMP and SABMP is due, in part, to the superior parameter estimation and refinement capability of SABMP. We would like to point out that the discrepancy between oracle-aided versions of algorithms and SABMP is due to incorrect support estimation. This shows that our method is better able to estimate the support; which explains why we perform better even when we do not know the exact prior. The figures also show that for both i.i.d. Gaussian and non-i.i.d. Uniform inputs, the performance of SABMP is exactly the same while that of oracle-FBMP deteriorates for the later case. This shows the robustness of our algorithm and ascertain our claim that it is agnostic to signal statistics.

Runtime graphs of Figs. 2 and 3 depict that SABMP is faster than all of the algorithms. However, the runtime of SABMP increases for higher values of p. This occurs because the initial estimate of p was 0.003, and as the sparsity rate of $\mathbf{x}$ increased, more iterations were required to estimate the value of p. With higher values of p, the difference in runtime is insignificant given the excellent NMSE performance of our method. In the figures, we did not plot the results for oracle-SABMP as its performance closely follows that of SABMP.

Experiment 2 (Signal Estimation Performance Comparison for Varying SNR): In the second set of experiments, NMSEs were measured for values of SNR between 0 dB and 30 dB and plotted to compare the performance of SABMP with FBMP, oracle-FBMP, FL and BPDN. The signal sparsity rate selected for these experiments was $p = 0.005$. The results were averaged over $K = 250$ trials.

Figs. 4(a) and 4(b) show that the proposed method has better NMSE performance than all other algorithms for all considered signals. Only at very high values of SNR does the NMSE performance of these algorithms approach at each other. As in the previous experiment oracle-SABMP is not plotted as its performance closely followed that of SABMP.

Note that the performance of BPDN, as shown in experiments 1 and 2, is for the case when true parameter values were provided to it. Moreover, support and amplitude refinement procedures were also performed on the results. Since its performance is not comparable with our algorithm, we will not consider it for the rest of the experiments. We will continue to provide comparisons between our algorithm and the plain and oracle-aided versions of FBMP and FL.

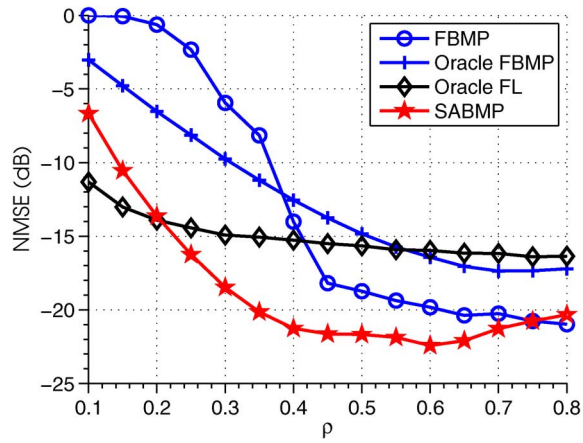


Fig. 5. Comparison for compressible signals. NMSE versus ρ. $N = 512$, $M = 128$, $E_{\max} = 20$, $D = 10$, SNR $= 15$ dB.

Experiment 3 (Recovery of Compressible Signals): In this experiment, we performed signal recovery for compressible signals using SABMP, FBMP, FL and oracle-FBMP. The signal given in [9] was selected for this purpose and its components were generated using $x_k = (-1)^k \exp(-\rho k)$ for $k = 0, \cdots, N-1$ with $\rho \in (0, 1)$. Here ρ controls the sparsity, with higher values giving sparser compressible signals. To make the experiment more challenging, SNR was deliberately kept low as there will be higher chances of confusing noise with components of compressible signal. All results were averaged over $K = 3000$ trials. We conducted two different experiments. In the first experiment $E_{max} = 20$ was selected for both SABMP and FBMP so that both algorithms have ample chance to refine their estimates. All other required parameters were intialized as mentioned in the Signal Setup section. Since we can not tell beforehand, with certainty, the true sparsity rate

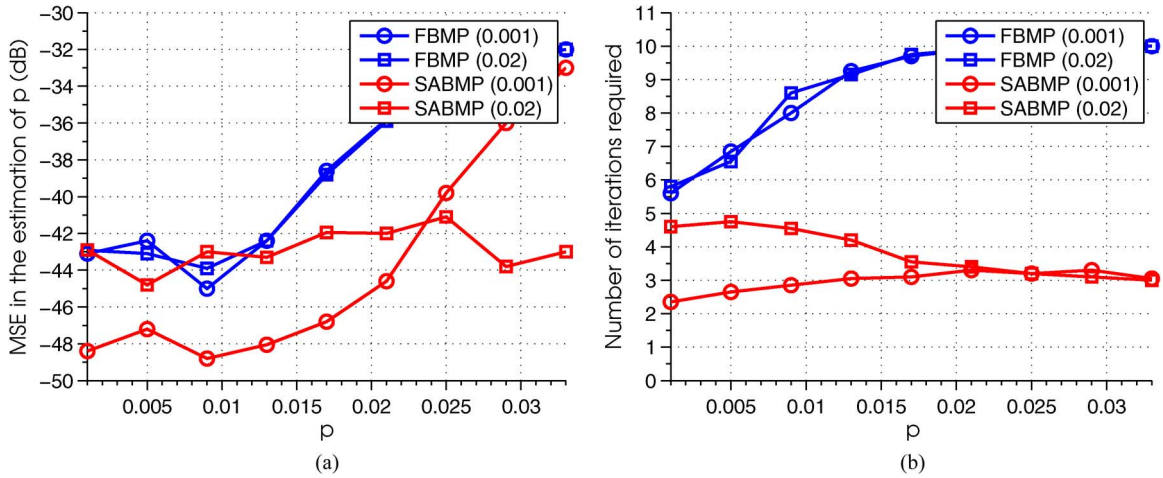


Fig. 6. Estimation of p performance for i.i.d. Gaussian inputs. $N = 512$, $M = 128$, $D = 10$, $E_{\max} = 10$, SNR $= 15$ dB. (a) Average MSE between actual and estimated p. (b) Average number of iterations required to estimate p.

for these particular type of signals, we provided $p = 0.01$ to both oracle-FBMP and oracle-SABMP. NMSE performance resulting from all considered algorithms is plotted in Fig. 5. We observe that, while FBMP failed to perform well at lower values of ρ, it is able to catch up to the performance of SABMP at $\rho = 0.75$.

Experiment 4 (Parameter Estimation): In the previous experiment we saw a comparison of how well the noise variance is estimated. In this experiment, we will study the performance of sparsity rate (p) estimation procedure. In this regard, signals were generated with sparsity rates ranging from $p = 0.001$ to $p = 0.033$. Both FBMP and SABMP algorithms were tested by providing two different initial estimates of the sparsity parameter, i.e., $p_{\mathbf{est}} = 0.001$ and $p_{\mathbf{est}} = 0.02$. True values for all other parameter were provided to FBMP for fairness. Both algorithms were allowed a maximum of $E_{max} = 10$ iterations for refinement of p. Moreover, the refinement procedure was stopped if the two consecutive estimates of p differed by less than 2%. The results were averaged over $K = 250$ trials. Interestingly, Fig. 6 shows that SABMP took much less iterations to refine the sparsity rate and even then its estimates were better than those of FBMP in almost all cases.

Experiment 5 (How Good are the Selected Models?): In this experiment, we plot rank ordered posterior probabilities $p(\mathcal{S}|\mathbf{y})$ of the models selected by our greedy algorithm in Fig. 7. Note that SABMP selects models based on the dominant support selection metric given in (13). Therefore, by looking at the posterior probabilities of the selected models we can have an idea of the effectiveness of our metric. As shown in Fig. 7, we found that indeed our greedy algorithm was able to return models with high posterior probabilities. The results were generated for $N = 1024$, $M = 256$, SNR $= 10$ dB, $p = 0.02$, and $E_{\max} = 20$. The results were averaged over $K = 200$ trials.

Experiment 6 (Performance Under Different Undersampling Ratios (N/M)): In this experiment, we study the performance under different undersampling ratios (N/M). For this purpose we used oracle FBMP, oracle FL and SABMP. The input signal size was fixed at $N = 512$ while the observation size M was varied from 10 to 200. From Fig. 8 it is evident that as the ratio decreases (i.e., M increases) the performance of all three algorithms improve. It is evident that SABMP performs much better than the two techniques. Furthermore, as expected, when the

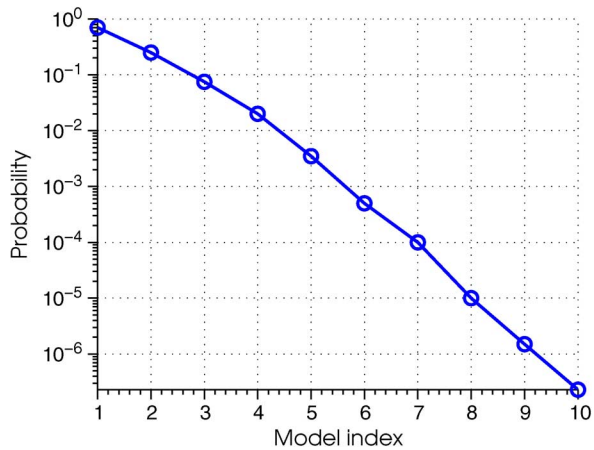


Fig. 7. Rank ordered posterior probabilities of the models selected by SABMP greedy algorithm.

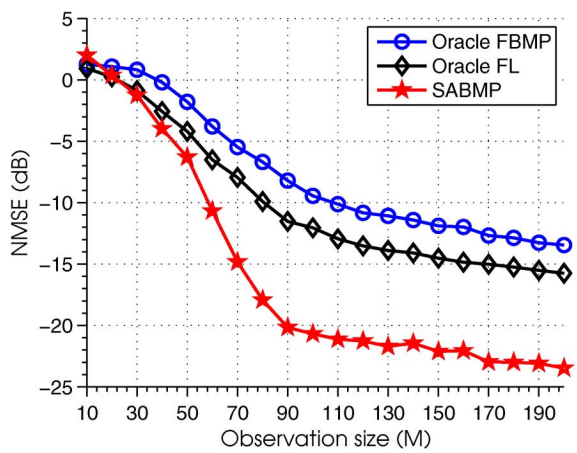


Fig. 8. NMSE performance for various N/M ratios for Gaussian iid input. $N = 512$, SNR $= 10$ dB, $p = 0.015$.

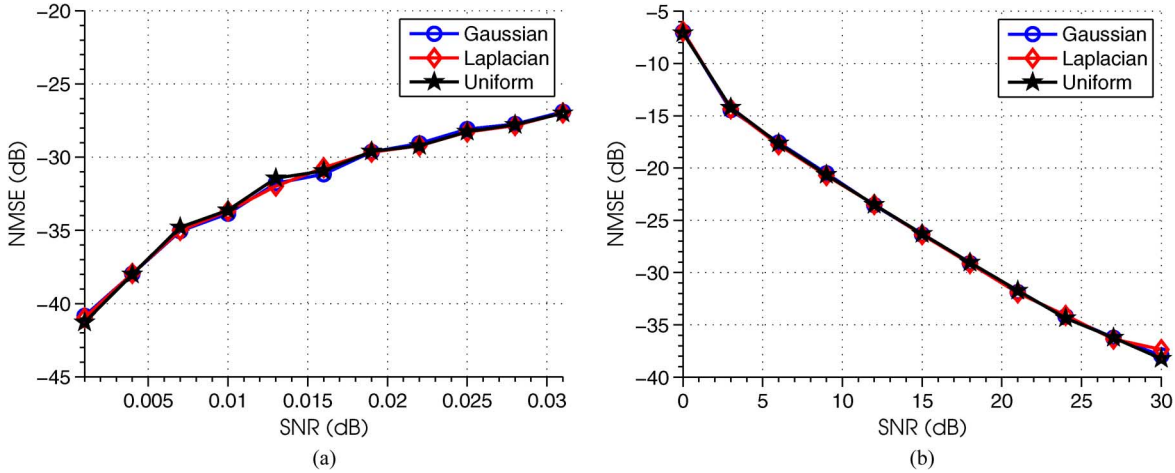


Fig. 9. SABMP performance for Gaussian, Laplacian and Uniform noise. $N = 512$, $M = 128$, $E_{max} = 10$. (a) NMSE versus sparsity rate p. SNR $= 20$ dB. (b) NMSE versus SNR. $p = 0.015$.

number of observations decreased the performance of all algorithms deteriorated. $K = 200$ trials were performed to generate the graphs.

Experiment 7 (Performance Under Non-Gaussian Noise Assumption): In this experiment, we study the performance of SABMP when the assumption of Gaussian noise is not valid. We performed this experiment for the Laplacian and Uniform noise and compared it with the Gaussian noise case. Noise were considered to be zero-mean and the noise power adjusted according to the desired SNR. Fig. 9(a) shows the performance for varying sparsity rate p. In this case the SNR was kept at 20 dB. Similarly, in Fig. 9(b), a comparison is performed for varying SNR while the sparsity rate was fixed at 0.015. Both figures show that the performance is same irrespective of the noise distribution and thus highlight the robustness of SABMP against noise distribution.

Experiment 8 (Comparison of Multiscale Image Recovery Performance): In another experiment, we carried out multiscale recovery of Mondrian image of 128 $\times$ 128 pixels in size. The image is shown in Fig. 10. One-level Haar wavelet decomposition of the image was performed, resulting in one 'approximation' (low-frequency) and three 'detail' (high-frequency) images. Unlike the approximation component, the detail components are compressible in nature. We, therefore, decided to conduct two experiments; one on the compressible components and the other on their sparse versions which were obtained by applying a suitable threshold. Noisy random measurements were acquired later from both versions for SNR $= 20$ dB. The number of measurements taken were 1/4 of the number of elements in detail components. For each of the case, the detail components were reconstructed from these measurements through SABMP. Finally, inverse wavelet transform was applied to reconstruct the image from the recovered details and the original approximations. Reconstruction errors were recorded and, for comparison, recoveries were obtained using FL and FBMP. The reconstructions were performed with $D = 10$, $E_{max} = 5$, and $p_{\mathbf{est}} = 0.005$ where, as mentioned earlier, D is a parameter required by FBMP. All other parameters were initialized as mentioned in the Signal Setup section. The results were averaged over $K = 10$ trials. The resulting

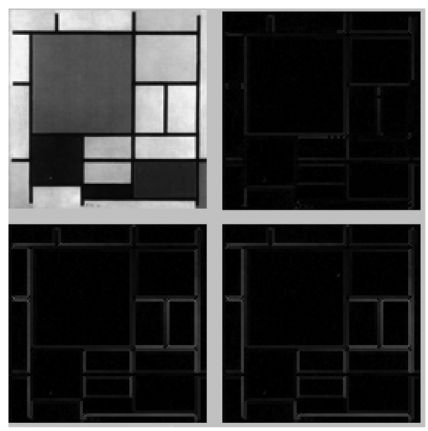


Fig. 10. Clockwise from top left: Original image, difference images for SABMP, FL, and FBMP.

difference images showing the reconstruction error for the threshold version (i.e., when the detail components are strictly sparse) are shown in Fig. 10. Numerical details of the results for these experiments are given in Table III. It is obvious that images reconstructed using SABMP have lower NMSEs when the detail components were strictly sparse. Similarly, for the case of compressible components, SABMP performed better. Here, we would like to emphasize that any gain in NMSE is crucial for image recovery. Edges in any image define its distinct features and the detail components contain the information of these edges. Therefore, if we are able to recover the detail components with less error, we will be preserving more edges; hence, the reconstructed image quality will improve to a great degree. For example, Fig. 10 shows the difference images highlighting the regions where reconstruction errors occurred. It is obvious from this image that SABMP was able to recover the detail components in a better way.

VI. Conclusion

In this paper, we presented a robust Bayesian matching pursuit algorithm based on a fast recursive method. Compared with other robust algorithms, our algorithm does not require signals to be derived from some known distribution. This is useful when

TABLE III
NMSE (dB) COMPARISONS BETWEEN FBMP, FL AND SABMP FOR THE TEST IMAGE (MONDRIAN) SHOWN IN FIG. 10 WHEN RECONSTRUCTION WAS PERFORMED USING BOTH COMPRESSIBLE AND SPARSE COMPONENTS

	FBMP	SABMP	FL
Sparse	-19.28	-36.51	-18.81
Compressible	-18.69	-27.42	-17.9

we can not estimate the parameters of the signal distributions. Application of the proposed method on several different signal types demonstrated its superiority and robustness.

APPENDIX
ON THE QUALITY OF THE APPROXIMATION OF LIKELIHOOD $p(\mathbf{y}|\mathcal{S})$

In order to have an idea of how good of an approximation our equation for likelihood (11) is to the actual $p(\mathbf{y}|\mathcal{S})$, let us compare it to the case when the distribution is known. So, let us assume for simplicity that $\mathbf{x}|\mathcal{S} \sim \mathcal{CN}(\mathbf{0}, \sigma_{\mathbf{x}}^2\mathbf{I})$. In this case $\mathbf{y}$ is also Gaussian $\mathbf{y} \sim \mathcal{CN}(\mathbf{0}, \Sigma_{\mathcal{S}})$, where $\Sigma_{\mathcal{S}} = \sigma_{\mathbf{n}}^2\mathbf{I} + \sigma_{\mathbf{x}}^2\mathbf{\Phi}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{S}}^{\mathsf{H}}$. Let us compare the log likelihood in the two cases

$$\mathcal{L}_G = -\frac{1}{2}\|\mathbf{y}\|^2_{\Sigma_{\mathcal{S}}^{-1}} \tag{22}$$

$$\mathcal{L}_U = -\frac{1}{2\sigma_{\mathbf{n}}^2}\left\|\mathbf{P}_{\mathcal{S}}^{\perp}\mathbf{y}\right\|^2 \tag{23}$$

By the matrix inversion lemma, we have that

$$\Sigma_{\mathcal{S}} = \left(\sigma_{\mathbf{n}}^2\mathbf{I} + \sigma_{\mathbf{x}}^2\mathbf{\Phi}_{\mathcal{S}}\mathbf{\Phi}_{\mathcal{S}}^{\mathsf{H}}\right)^{-1} \tag{24}$$

$$\Sigma_{\mathcal{S}}^{-1} = \frac{1}{\sigma_{\mathbf{n}}^2}\mathbf{I} - \frac{\sigma_{\mathbf{x}}^2}{\sigma_{\mathbf{n}}^2\sigma_{\mathbf{n}}^2}\mathbf{\Phi}_{\mathcal{S}}\left(\mathbf{I} + \frac{\sigma_{\mathbf{x}}^2}{\sigma_{\mathbf{n}}^2}\mathbf{\Phi}_{\mathcal{S}}^{\mathsf{H}}\mathbf{\Phi}_{\mathcal{S}}\right)^{-1}\mathbf{\Phi}_{\mathcal{S}}^{\mathsf{H}} \tag{25}$$

$$\Sigma_{\mathcal{S}}^{-1} = \frac{1}{\sigma_{\mathbf{n}}^2}\left(\mathbf{I} - \mathbf{\Phi}_{\mathcal{S}}\left(\frac{\sigma_{\mathbf{n}}^2}{\sigma_{\mathbf{x}}^2}\mathbf{I} + \mathbf{\Phi}_{\mathcal{S}}^{\mathsf{H}}\mathbf{\Phi}_{\mathcal{S}}\right)^{-1}\mathbf{\Phi}_{\mathcal{S}}^{\mathsf{H}}\right). \tag{26}$$

which if the $\mathrm{SNR} = \frac{\sigma_{\mathbf{x}}^2}{\sigma_{\mathbf{n}}^2}$ is high enough reads,

$$\Sigma_{\mathcal{S}}^{-1} \simeq \frac{1}{\sigma_{\mathbf{n}}^2}\left(\mathbf{I} - \mathbf{\Phi}_{\mathcal{S}}\left(\mathbf{\Phi}_{\mathcal{S}}^{\mathsf{H}}\mathbf{\Phi}_{\mathcal{S}}\right)^{-1}\mathbf{\Phi}_{\mathcal{S}}^{\mathsf{H}}\right) \tag{27}$$

$$= -\frac{1}{\sigma_{\mathbf{n}}^2}\mathbf{P}_{\mathcal{S}}^{\perp} \tag{28}$$

It thus follows that

$$\mathcal{L}_G \simeq -\frac{1}{2\sigma_{\mathbf{n}}^2}\mathbf{y}\mathbf{P}_{\mathcal{S}}^{\perp}\mathbf{y} \tag{29}$$

$$= -\frac{1}{2\sigma_{\mathbf{n}}^2}\left\|\mathbf{P}_{\mathcal{S}}^{\perp}\mathbf{y}\right\|^2 \tag{30}$$

$$= \mathcal{L}_U \tag{31}$$

Unfortunately in non-Gaussian case it is difficult to evaluate the actual $p(\mathbf{y}|\mathcal{S})$ and so it is not easy to tell how the proposed likelihood $\exp\left(-\frac{1}{2\sigma_{\mathbf{n}}^2}\|\mathbf{P}_{\mathcal{S}}^{\perp}\mathbf{y}\|^2\right)$ is good of an approximation. However, our experimental results which were conducted under a multitude of diverse situations attest to the robustness of the proposed likelihood.

REFERENCES

[1] K. Jia, X. Wang, and X. Tang, "Optical flow estimation using learned sparse model," in *Proc. IEEE Int. Conf. Comput. Vision (ICCV)*, Nov. 2011, pp. 2391–2398.

[2] M. Lustig, D. Donoho, J. Santos, and J. Pauly, "Compressed sensing MRI," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 72–82, Mar. 2008.

[3] T. Al-Naffouri, A. Quadeer, F. Al-Shaalan, and H. Hmida, "Impulsive noise estimation and cancellation in DSL using compressive sampling," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2011, pp. 2133–2136.

[4] M. Firooz and S. Roy, "Network tomography via compressed sensing," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM 2010)*, Dec. 2010, pp. 1–5.

[5] T. Y. Al-Naffouri, E. B. Al-Safadi, and M. E. Eltayeb, "OFDM Peak-to-Average Power Ratio Reduction Method," U.S. Patent 2011/0 122 930, May 2011.

[6] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *J. Roy. Statist. Soc. Series B-Methodological*, vol. 58, no. 1, pp. 267–288, 1996.

[7] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1998.

[8] A. Quadeer, S. Ahmed, and T. Al-Naffouri, "Structure based bayesian sparse reconstruction using non-gaussian prior," in *Proc. 49th Annu. Allerton Conf. Commun., Control, Comput.*, Sep. 2011, pp. 277–283.

[9] P. Schniter, L. C. Potter, and J. Ziniel, Fast Bayesian Matching Pursuit: Model Uncertainty and Parameter Estimation for Sparse Linear Models 2009 [Online]. Available: http://www.ece.osu.edu/~zinielj/FBMP_TechRep.pdf, OSU ECE Technical Report

[10] E. Candes and J. Romberg, "Sparsity and incoherence in compressive sampling," *Inverse Problems*, vol. 23, no. 3, pp. 969–985, Jun. 2007.

[11] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, pp. 489–509, Feb. 2006.

[12] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, pp. 1289–1306, Apr. 2006.

[13] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, pp. 4655–4666, Dec. 2007.

[14] J. Haupt and R. Nowak, "Signal reconstruction from noisy random projections," *IEEE Trans. Inf. Theory*, vol. 52, pp. 4036–4048, Sep. 2006.

[15] P. Schniter, L. C. Potter, and J. Ziniel, "Fast bayesian matching pursuit," in *Proc. Workshop Inform. Theory Appl. (ITA)*, 2008.

[16] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 58, pp. 1094–1121, Feb. 2012.

[17] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE J. Sel. Topics Signal Process.*, vol. 1, pp. 586–597, Dec. 2007.

[18] M. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learning Res.*, vol. 1, no. 3, pp. 211–244, 2001.

[19] S. Ji and L. Carin, "Bayesian compressive sensing and projection optimization," in *Proc. 24th Int. Conf. Mach. Learning*, 2007, pp. 377–384.

[20] C. M. Bishop and M. E. Tipping, "Variational relevance vector machines," in *Proc. 16th Conf. Uncertainty Artif. Intell.*, 2000, pp. 46–53.

[21] E. G. Larsson and Y. Selén, "Linear regression with a sparse parameter vector," *IEEE Trans. Signal Process.*, vol. 55, pp. 451–460, Feb. 2007.

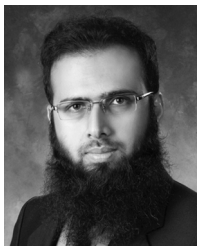
[22] S. Babacan, R. Molina, and A. Katsaggelos, "Bayesian compressive sensing using Laplace priors," *IEEE Trans. Image Process.*, vol. 19, no. 1, pp. 53–63, Jan. 2010.

[23] A. Sayed, *Adaptive Filters*. Hoboken, NJ, USA: Wiley, 2008.

[24] C. R. Inc., "CVX: Matlab software for disciplined convex programming, version 1.22," [Online]. Available: http://cvxr.com/cvx Feb. 2012

[25] A. Al-Rabah, M. Masood, A. Ali, and T.Y. Al-Naffouri, "Receiver-based Bayesian PAPR reduction in OFDM," in *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2013.

[26] A. Quadeer and T. Y. Al-Naffouri, "Structure-based Bayesian sparse reconstruction," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6354–6367, Dec. 2012.



Mudassir Masood (S'13) received the B.E. degree from NED University of Engineering and Technology, Karachi, Pakistan, in 2001, and the M.S. degree in electrical engineering from King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, in 2005. He is currently working toward the Ph.D. degree in electrical engineering from King Abdullah University of Science and Technology, Thuwal, Saudi Arabia. His research interests lie in the areas of signal processing, compressed sensing and its applications.

Mr. Masood is the recipient of a Best Student Poster Award at the ICCSPA 2013 and the KAUST Fellowship and Academic Excellence Award in 2010.

Tareq Y. Al-Naffouri (M'10) received the B.S. degree in mathematics and electrical engineering (with first honors) from King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, in 1994, the M.S. degree in electrical engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 1998, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, USA, in 2004.

He was a Visiting Scholar at the California Institute of Technology, Pasadena, CA, USA, from January to August 2005 and during the summer of 2006. He was a Fulbright Scholar at the University of Southern California from February to September 2008. He held internship positions at NEC Research Labs, Tokyo, Japan, in 1998, Adaptive Systems Lab., University of California, Los Angeles, CA, USA, in 1999, National Semiconductor, Santa Clara, CA, USA, in 2001 and 2002, and Beceem Communications Santa Clara, CA, USA, in 2004. He is currently an Associate Professor in the Electrical Engineering Department, King Fahd University of Petroleum and Minerals, Saudi Arabia, and jointly at the Electrical Engineering Department, King Abdullah University of Science and Technology. He has over 80 publications in journal and conference proceedings, 9 standard contributions, and four issued and four pending patents. His research interests lie in the areas of adaptive and statistical signal processing and their applications to wireless communications, seismic signal processing, and in multiuser information theory. He has recently been interested in compressive sensing and random matrix theory and their applications.

Dr.Al-Naffouri has been serving as an Associate Editor of IEEE TRANSACTIONS ON SIGNAL PROCESSING since August 2013. He is the recipient of a Best Student Paper Award at the 2001 IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing for his work on adaptive filtering analysis, the IEEE Education Society Chapter Achievement Award in 2008, and the Al-Marai Award for innovative research in communication in 2009.