Non-Gaussian prior Fast Bayesian Matching Pursuit

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

***Abstract*—A fast matching pursuit method (nGpFBMP) is introduced which performs Bayesian estimates of sparse signals even when the signal prior is non-Gaussian/unknown. It is agnostic on signal statistics and utilizes a greedy approach and order-recursive updates to determine the approximate MMSE estimate of the sparse signal. Simulation results demonstrate the power and robustness of the method.**

***Index Terms*—sparse reconstruction, compressed sensing, Bayesian, matching pursuit, basis selection, greedy algorithm.**

I. Introduction

SPARSITY exists in many natural and man-made signals. Some examples of sparse signals include those from speech, images, videos, sensor arrays, seismic activity, and frequency hopping. Sparsity is an attractive property because its exploitation may be useful in the development of simple signal processing systems. Sparsity-aware estimators form the core of such systems. Some examples include estimators such as Lasso [1], basis pursuit [2], structure-based estimator [3], fast Bayesian matching pursuit [4], and those related to the area of compressed sensing (CS) [5]–[7].

CS algorithms have been shown to recover sparse signals from underdetermined systems of equations that take the form

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

where $\boldsymbol{x} \in \mathbb{C}^N$, and $\boldsymbol{y} \in \mathbb{C}^M$ are the unknown sparse and observed signal, respectively. Furthermore, $\boldsymbol{\Phi} \in \mathbb{C}^{M\times N}$ is the measurement matrix with $N \gg M$ and $\boldsymbol{n} \in \mathbb{C}^M$ is the additive Gaussian noise. CS uses structure-preserving linear projections of sparse signals for reconstruction using l_1-optimization.

$$\hat{\boldsymbol{x}} = \mathbf{argmin}\, \|\boldsymbol{x}\|_1 \;\text{ such that }\; \boldsymbol{\Phi}\boldsymbol{x} = \boldsymbol{y} \tag{2}$$

l_1-optimization is a convex optimization problem that conveniently reduces to basis pursuit having computational complexity of $\mathcal{O}(N^3)$. Since, usually, N is large, such an approach rapidly becomes unrealistic. Some efficient alternatives such as orthogonal matching pursuit (OMP) [8] and the algorithm proposed by Haupt et al. [9] have been proposed. These algorithms fall into the category of greedy algorithms that are relatively faster than basis pursuit. However, an inherent problem in these systems is that the only *a priori* information utilized is the sparsity information.

Another category of methods based on the Bayesian approach considers complete *a priori* statistical information of sparse signals. Two popular methods, fast Bayesian matching pursuit (FBMP) [4] and the other proposed by Larsson and Selén [10], adopt such an approach and assume Gaussian prior on the unknown sparse vector. They find MMSE estimate of the sparse signal via model selection and averaging, and the signal is described as a mixture of several components. FBMP was shown to outperform several sparse recovery algorithms, including OMP [8], StOMP [11], GPSR-Basic [12], Sparse Bayes [13], BCS [14] and a variational-Bayes implementation of BCS [15]. However, there are several drawbacks associated with both FBMP and [10]. They work successfully only for Gaussian priors. Gaussian assumption for additive noise is reasonable; however, for signal it is inadequate. Moreover, in situations where the assumption of a Gaussian prior is valid, its parameters 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 with regard to signal statistics.

In this paper, we pursue a sparse signal reconstruction approach that on one hand is Bayesian, acknowledging the noise statistics and the signal sparsity rate, while on the other hand is agnostic on the signal amplitude statistics. The approach can bootstrap itself and estimate the sparsity rate and noise variance when unknown. The algorithm is implemented in a greedy manner and pursues an order-recursive approach, helping it to enjoy low complexity.

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 and IV, we describe our greedy algorithm and a recursive method to make its computations efficient. This is followed by Section V, which describes our hyperparameter estimation process. In Section VI, we present our simulation results and Section VII concludes the paper.

II. Problem Formulation and MMSE Setup

A. The Signal Model

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

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

where $\boldsymbol{\Phi}$ is a known $M \times N$ regression matrix and $\boldsymbol{n} \sim \mathcal{CN}(\mathbf{0}, \boldsymbol{K_n})$ is the additive Gaussian noise vector. Here, $\boldsymbol{x}$ has a sparse structure and is modeled as $\boldsymbol{x} = \boldsymbol{x}_A \circ \boldsymbol{x}_B$ with $\circ$ indicating element-by-element multiplication. The vector $\boldsymbol{x}_A$ consists of elements that are drawn from some unknown distribution[1] and the entries of $\boldsymbol{x}_B$ are drawn i.i.d. from a Bernoulli distribution with success probability p.

Authors are with the Department of Electrical Engg., KAUST, Thuwal, Saudi Arabia. Tareq Al-Naffouri is also with the Department of Electrical Engg., KFUPM, Dhahran, Saudi Arabia. e-mail: {mudassir.masood, tareq.alnaffouri}@kaust.edu.sa.

[1]Typically, in Bayesian estimation, the signal entries are assumed to be drawn from a Gaussian distribution but here 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 $\boldsymbol{x}$

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

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

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

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

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

where $\boldsymbol{\Phi}_{\mathcal{S}}$ is a matrix formed by selecting columns of $\boldsymbol{\Phi}$ indexed by support $\mathcal{S}$. Similar explanation follows for $\boldsymbol{x}_{\mathcal{S}}$. Since the distribution of $\boldsymbol{x}$ is unknown, the best we can do is to use the BLUE as an estimate, i.e.,

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

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

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

The factor, $p(\boldsymbol{y})$, is common to all posterior probabilities and could be ignored. Since the elements of $\boldsymbol{x}$ are activated according to the Bernoulli distribution $\mathcal{B}(1,p)$, we have

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

Determining $p(\boldsymbol{y}|\mathcal{S})$ is in general very difficult when $\boldsymbol{x}_{\mathcal{S}}$ is non-Gaussian/unknown. To go around this, we eliminate the non-Gaussian component of $\boldsymbol{y}$ by projecting it onto the orthogonal complement space of $\boldsymbol{\Phi}_{\mathcal{S}}$. To do so we multiply $\boldsymbol{y}$ by the projection matrix $\boldsymbol{P}_{\mathcal{S}}^{\perp} = \boldsymbol{I} - \boldsymbol{\Phi}_{\mathcal{S}}(\boldsymbol{\Phi}_{\mathcal{S}}^{\mathsf{H}}\boldsymbol{\Phi}_{\mathcal{S}})^{-1}\boldsymbol{\Phi}_{\mathcal{S}}^{\mathsf{H}}$. This leaves us with $\boldsymbol{P}_{\mathcal{S}}^{\perp}\boldsymbol{y} = \boldsymbol{P}_{\mathcal{S}}^{\perp}\boldsymbol{n}$, which follows $\mathcal{N}(\boldsymbol{0}, \boldsymbol{P}_{\mathcal{S}}^{\perp}\boldsymbol{K}_{\boldsymbol{n}}\boldsymbol{P}_{\mathcal{S}}^{\perp\,\mathsf{H}})$, where $\boldsymbol{K}_{\boldsymbol{n}}$ is the noise covariance matrix. Thus, we have,

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

Substituting (8) and (9) into (7) yields an expression for the posterior probability providing us all the ingredients for (4). Computing (4) becomes intractable for large N; therefore, we compute it over a few significant support sets $\mathcal{S}^d$

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

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 follows:

$$\nu(\mathcal{S}) \triangleq \ln p(\boldsymbol{y}|\mathcal{S})p(\mathcal{S}) = \frac{1}{\sigma_{\boldsymbol{n}}^2}\left\|\boldsymbol{\Phi}_{\mathcal{S}}(\boldsymbol{\Phi}_{\mathcal{S}}^{\mathsf{H}}\boldsymbol{\Phi}_{\mathcal{S}})^{-1}\boldsymbol{\Phi}_{\mathcal{S}}^{\mathsf{H}}\boldsymbol{y}\right\|^2 - \frac{1}{\sigma_{\boldsymbol{n}}^2}\|\boldsymbol{y}\|^2 + |\mathcal{S}|\ln p + (N-|\mathcal{S}|)\ln(1-p) \tag{11}$$

III. A Greedy Algorithm

We now present a greedy algorithm to determine the set of dominant supports, $\mathcal{S}^d$, required to evaluate (10). 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, which involves a search of size $\binom{N}{2}$. To reduce the search space, we pursue a greedy approach and look for the point $i_2^\star \neq i_1^\star$ such that $\mathcal{S}_2 = \{i_1^\star, i_2^\star\}$ maximizes $\nu(\mathcal{S}_2)$. This involves $\binom{N-1}{1}$ search points (as opposed to the optimal search over $\binom{N}{2}$ points). We continue in this manner by forming $\mathcal{S}_3 = \{i_1^\star, i_2^\star, i_3^\star\}$ and searching for $i_3^\star$ in the remaining $N-2$ points and so on until we reach $\mathcal{S}_P = \{i_1^\star,\ldots,i_P^\star\}$. The value of P is selected to be slightly larger than the expected number of nonzero elements in the constructed signal such that $\Pr(|\mathcal{S}| > P)$ is sufficiently small[2]. Table I presents a formal algorithmic description.

TABLE I: The Greedy Algorithm

1) Initialize $L = \{1,2,\ldots,N\}$, $\mathcal{S}_{max} = \{\}$, $\mathcal{S}^d = \{\}$, $i = 1$, $L_i = L$.
2) If $i > P$, then stop.
3) Generate $\Omega = \{\mathcal{S}_{max} \cup \{\alpha_1\}, \mathcal{S}_{max} \cup \{\alpha_2\}, \cdots, \mathcal{S}_{max} \cup \{\alpha_{|L_i|}\} \mid \alpha_k \in L_i\}$
4) Compute $\{\nu(\mathcal{S}_k) \mid \mathcal{S}_k \in \Omega\}$.
5) Find $\mathcal{S}^\star \in \Omega$, such that $\nu(\mathcal{S}^\star) \geq \max_j \nu(\mathcal{S}_j)$.
6) Update, $\mathcal{S}^d = \{\mathcal{S}^d, \mathcal{S}^\star\}$, $\mathcal{S}_{max} = \mathcal{S}^\star$, $L_{i+1} = L \setminus \mathcal{S}^\star$.
7) Set $i \leftarrow i+1$ and repeat steps 2 - 7.

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 $\nu(\mathcal{S}_j \cup \{i_{j+1}\})$ from (11). This translates to appending a column, $\boldsymbol{\phi}_{j+1}$, to $\boldsymbol{\Phi}_{\mathcal{S}_j}$ in the calculations of (11), 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 approach. Therefore, the complexity we incur is of the order $\mathcal{O}(PMN)$ in our greedy search for the best P support.

A. A Repeated Greedy Search

The accuracy of the reconstructed signal is dependent on the number of support vectors in $\mathcal{S}^d$ and may be increased by repeating the greedy algorithm a number of times (e.g., D). This would result in $\mathcal{S}^d$ with a total of PD supports. The selection of supports in subsequent repetitions of the algorithm is performed by making sure not to select an element at a particular sparsity level that has been selected at the same sparsity level in any of the previous repetitions. We note that a repeated greedy search in this manner would incur a complexity of order $\mathcal{O}(DPMN)$. For a detailed description of the steps followed by the method, code is provided on the author's website[3].

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

[3] http://faculty.kfupm.edu.sa/ee/naffouri/publications.html

IV. Efficient Computation of the Dominant Support Selection Metric

The computational complexity of the proposed algorithm is dependent upon the way $\nu(\mathcal{S})$ is computed. The efficient computation of $\nu(\mathcal{S})$ depends mainly on the term $\xi_{\mathcal{S}} = \|(\mathbf{\Phi}_{\mathcal{S}}(\mathbf{\Phi}_{\mathcal{S}}^{\mathsf{H}}\mathbf{\Phi}_{\mathcal{S}})^{-1}\mathbf{\Phi}_{\mathcal{S}}^{\mathsf{H}}\boldsymbol{y})\|^2 = \|\mathbf{\Phi}_{\mathcal{S}}\mathbb{E}[\boldsymbol{x}|\boldsymbol{y},\mathcal{S}]\|^2$. Our focus is therefore on computing $\mathbb{E}[\boldsymbol{x}|\boldsymbol{y},\mathcal{S}]$ efficiently.

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 $\mathbf{e}_{\boldsymbol{y},k-1}(\underline{\mathcal{S}}) \triangleq \mathbb{E}[\boldsymbol{x}_{\underline{\mathcal{S}}}|\,\boldsymbol{y}]$ to obtain[4] $\mathbf{e}_{\boldsymbol{y},k}(\mathcal{S}) = \mathbb{E}[\boldsymbol{x}_{\mathcal{S}}|\,\boldsymbol{y}]$. We note that

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

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

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

where $\Gamma = \frac{1}{f_{\mathcal{S}}}(\boldsymbol{q}_{\boldsymbol{\phi},k}^{\mathsf{H}}(\mathcal{S})\mathbf{e}_{\boldsymbol{y},k-1}(\underline{\mathcal{S}}) - \mathbf{e}_{\boldsymbol{y},1}(s_k))$. This recursion is initialized by $\mathbf{e}_{\boldsymbol{y},1}(i) = (\boldsymbol{\phi}_s^{\mathsf{H}}\boldsymbol{\phi}_s)^{-1}\boldsymbol{\phi}_s^{\mathsf{H}}\boldsymbol{y}$. The recursion also depends on $\boldsymbol{q}_{\boldsymbol{\phi},k}(\mathcal{S}) \triangleq \mathbf{\Phi}_{\underline{\mathcal{S}}}^{\mathsf{H}}\boldsymbol{\phi}_{s_k}$, $\mathbf{e}_{\boldsymbol{\phi},k}(\mathcal{S}) \triangleq (\mathbf{\Phi}_{\underline{\mathcal{S}}}^{\mathsf{H}}\mathbf{\Phi}_{\underline{\mathcal{S}}})^{-1}\mathbf{\Phi}_{\underline{\mathcal{S}}}^{\mathsf{H}}\boldsymbol{\phi}_{s_k}$ and $f_{\mathcal{S}} \triangleq 1 - \boldsymbol{q}_{\boldsymbol{\phi},k}^{\mathsf{H}}(\mathcal{S})\mathbf{e}_{\boldsymbol{\phi},k}(\mathcal{S})$. The recursions for $\boldsymbol{q}_{\boldsymbol{\phi},k}(\mathcal{S})$, and $\mathbf{e}_{\boldsymbol{\phi},k}(\mathcal{S})$ may be determined as follows[5]

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

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

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

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 $\boldsymbol{q}_{\boldsymbol{\phi},2}(s_1; s_2)$ for $s_1, s_2 = 1, 2, \ldots, N$. This completes the recursion of $\mathbf{e}_{\boldsymbol{y},k}(\mathcal{S})$ which we utilize for recursive evaluation of $\nu(\mathcal{S})$.

V. Estimation of the hyperparameters p and $\sigma_{\boldsymbol{n}}^2$

One of the advantages of the proposed nGpFBMP is that it is agnostic with regard to signal statistics; the only parameters required are the noise variance, $\sigma_{\boldsymbol{n}}^2$, and the sparsity rate, p. Note from (11) that the selection of dominant supports at each sparsity level is independent of these quantities. This allows accurate and rapid estimation of these parameters. To determine these estimates, a computationally efficient method is described next.

We use the MAP estimate of support $\mathcal{S}$ i.e. $\hat{\mathcal{S}}_{map}$ to get the MAP estimate of $\boldsymbol{x}$, i.e., $\hat{\boldsymbol{x}}_{map} = \mathbb{E}[\boldsymbol{x}|\boldsymbol{y}, \hat{\mathcal{S}}_{map}]$. This $\hat{\boldsymbol{x}}_{map}$ is in turn used to estimate p, iteratively, as $\hat{p}^{(i+1)} = \left\|\hat{\boldsymbol{x}}_{map}^{(i)}\right\|_0 / N$ at $(i+1)$th iteration. The estimate is computed iteratively where in the first iteration of nGpFBMP, $\hat{p}^{(1)}$ is initialized by $p_{\textbf{init}}$, to compute $\hat{\boldsymbol{x}}_{map}^{(1)}$. This is used to find the new estimate, $\hat{p}^{(2)}$, which is then used by nGpFBMP to compute $\hat{\boldsymbol{x}}_{map}^{(2)}$. This process is repeated until the estimate of p changes by less than 2% or until a predetermined maximum number of iterations has been performed. At this stage, the estimate of the noise variance can be computed as $\hat{\sigma}_n^2 = \text{var}(\boldsymbol{y} - \mathbf{\Phi}\hat{\boldsymbol{x}}_{map})$.

VI. Results

To demonstrate the performance of nGpFBMP, we compare it with FBMP [4] and the convex relaxation-based (l_1) approach. FBMP was selected as it was shown to outperform a number of contemporary algorithms. The following signal configurations were used for the experiments:

1) Gaussian i.i.d., $\mu_{\boldsymbol{x}} = 10, \quad \sigma_{\boldsymbol{x}}^2 = 2$
2) Uniform non-i.i.d., $5 \leq \mu_{\boldsymbol{x}} \leq 10, \quad 1 \leq \sigma_{\boldsymbol{x}}^2 \leq 2$

where $\mu_{\boldsymbol{x}}$ and $\sigma_{\boldsymbol{x}}^2$ refer to the mean and variance respectively.

Entries of $M \times N$ sensing 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 $M = 256, N = 1024$. The noise had a zero mean and was white and Gaussian, $\mathcal{CN}(\mathbf{0}, \sigma_{\boldsymbol{n}}^2\mathbf{I}_M)$, with $\sigma_{\boldsymbol{n}}^2$ determined according to the desired signal-to-noise ratio (SNR). Initial estimates of the hyperparameters used for the simulations were $\mu_{\boldsymbol{x}\ \textbf{est}} = 0$, $\sigma_{\boldsymbol{x}\ \textbf{est}}^2 = \frac{1}{10} \times \sigma_{\boldsymbol{x}}^2$, $\sigma_{\boldsymbol{n}\ \textbf{est}}^2 = 10 \times \sigma_{\boldsymbol{n}}^2$, and $p_{\textbf{est}} = 0.003$, where estimates of the signal mean and variance were needed for FBMP.

In all of the experiments, parameter refinement was performed for both algorithms for a maximum of 10 iterations. For fairness, support and amplitude refinement [16] procedures were performed on the results of the CS algorithm[6]. Finally, the normalized mean-squared error (NMSE) between the original signal, $\boldsymbol{x}$, and its MMSE estimate, $\hat{\boldsymbol{x}}_{ammse}$, was used as the performance measure.

Experiment 1 (Performance comparison for varying SNR)

NMSEs were measured for values of SNR between 0 dB and 30 dB and plotted to compare the performance of nGpFBMP with FBMP and the CS algorithm (Fig. 1). Sparsity rate selected was $p = 0.005$. Experiments showed that the proposed method has better NMSE performance than both FBMP and CS for all considered signals.

Experiment 2 (Performance comparison for varying p)

In a similar set of experiments, NMSE and mean runtime were measured for different values of sparsity parameter p with SNR set to 20 dB. Figs. 2 and 3 demonstrate the superiority of nGpFBMP over FBMP and CS. We also observe that performance of nGpFBMP is relatively insensitive to changes in p as the corresponding changes in NMSE are very small, thus demonstrating the strength of the proposed algorithm.

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

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

[6]Actual parameter values were provided to the CS algorithm instead of estimates; furthermore, support and amplitude refinement was also performed to demonstrate that, despite these measures, its performance was inferior to that of nGpFBMP.

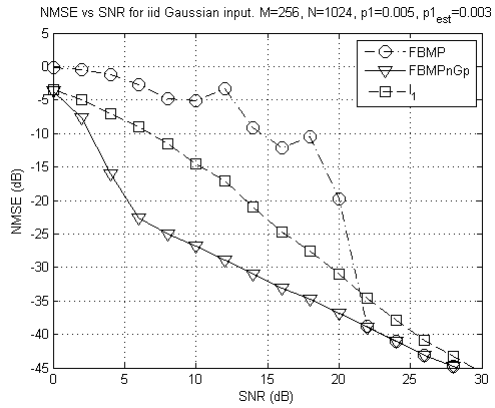

Fig. 1: NMSE vs SNR for Gaussian i.i.d. input

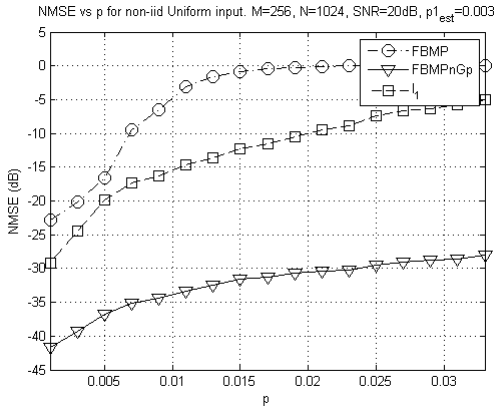

Fig. 2: NMSE vs p for uniform non-i.i.d. input

Experiment 3 (Performance comparison when the initial statistics of signal and noise are very close to the actual values)

Table II compares the average NMSEs of FBMP and nGpFBMP for different types of signals when the initial estimates ($\mu_{\boldsymbol{x}}$, $\sigma^2_{\boldsymbol{x}}$, and $\sigma^2_{\boldsymbol{n}}$) were chosen to be very near to their actual values. Since nGpFBMP is independent of these initial estimates its performance did not change. On the other hand, performance of FBMP improved, although it did not outperform nGpFBMP.

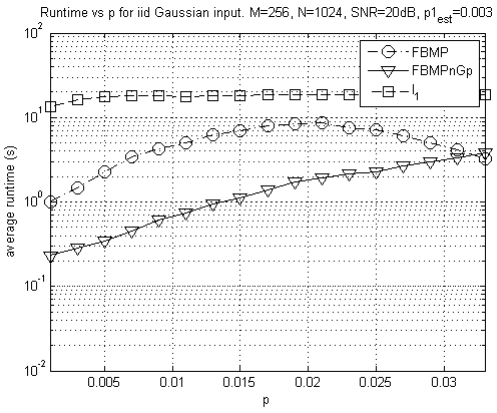

Fig. 3: Runtime vs p for Gaussian i.i.d. input

TABLE II: Average NMSE (dB) comparison between FBMP and nGpFBMP when the initial estimates of the hyperparameters are close to the actual values

Signal type	FBMP	nGpFBMP
Gaussian	-20.55	-31.103
Uniform (i.i.d.)	-24.2	-30.98
Uniform (non-i.i.d.)	-23.87	-30

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

References

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

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

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

[4] P. Schniter, L. C. Potter, and J. Ziniel, "Fast Bayesian Matching Pursuit: Model Uncertainty and Parameter Estimation for Sparse Linear Models," *IEEE Trans. Signal Process.*, to be published. [Online]. Available: http://www.ece.osu.edu/~zinielj/FBMP_TransSP.pdf

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

[6] 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.

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

[8] 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.

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

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

[11] 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.

[12] 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.

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

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

[15] C. M. Bishop and M. E. Tipping, "Variational relevance vector machines," in *Proc. of the 16th Conf. on Uncertainty in Artificial Intelligence*, 2000, pp. 46–53.

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