HYBRID GREEDY PURSUIT

Saikat Chatterjee, Dennis Sundman, Mikko Vehkaperä, Mikael Skoglund

Communication Theory Laboratory, School of Electrical Engineering
KTH - Royal Institute of Technology, Stockholm - 10044, Sweden
sach@kth.se, denniss@kth.se, mikkov@kth.se, skoglund@ee.kth.se

ABSTRACT

For constructing the support set of a sparse vector in the standard compressive sensing framework, we develop a hybrid greedy pursuit algorithm that combines the advantages of serial and parallel atom selection strategies. In an iterative framework, the hybrid algorithm uses a joint sparsity information extracted from the independent use of serial and parallel greedy pursuit algorithms. Through experimental evaluations, the hybrid algorithm is shown to provide a significant improvement for the support set recovery performance.

1. INTRODUCTION

The standard compressive sensing (CS) problem [1] is based on a sparse signal model and uses an under-determined system of linear equations. In the literature, a variety of CS reconstruction algorithms have been developed based on convex relaxation [2]-[3], non-convex [4]-[5] and iterative greedy search [6]-[12] strategies. Among these methods, convex relaxation based methods have attracted much attention due to their theoretical elegance and provable recovery performance. In practice, convex relaxation based methods are computationally intensive. On the other hand, the iterative greedy search methods are of lower complexity and hence their use may be practically viable in solving large-dimensional CS problems.

From a measurement vector, the main principle of the iterative greedy search methods is the estimation of the underlying support set of a sparse vector. The support set is the set of indices corresponding to non-zero elements of a sparse vector. To estimate the support set, iterative greedy search methods use linear algebraic tools (such as the inner-product based matched filter and the pseudo-inverse based least squares solution) iteratively. A better estimate of the support set leads to reduction in the norm of least squares error (or a fitting residual). The ability to estimate the support set from noisy measurements depends on an interplay between several factors, including the sparse signal vector dimension, measurement vector dimension, number of non-zero elements (level of sparsity), and measurement noise level.

Based on atom selection strategies for constructing a support set, iterative greedy search algorithms can be broadly classified into two categories: serial and parallel. Prominent examples of serial atom selection based greedy pursuits are matching pursuit (MP) [6] and orthogonal matching pursuit (OMP) [7]. The OMP is similar to MP with an additional aspect of least squares solution in each iteration, leading to significant performance improvement over the MP. On the other hand, the examples of parallel atom selection based greedy pursuits are subspace pursuit (SP) [11] and CoSaMP [10] (SP and CoSaMP are almost similar algorithms).

Let us consider a set of greedy pursuit algorithms. Assuming that all physical aspects remained same, an experimental evaluation reveals that the performances of the algorithms are input data dependent. For a particular instance/realization, the algorithms may provide different performance; it may happen that OMP provides better performance than SP or vice-versa. In this paper, considering the data dependent behaviors, we develop a hybrid greedy pursuit (HGP) algorithm by combining the serial atom selection based OMP and the parallel atom selection based SP. In an iterative framework, we execute both the algorithms simultaneously to extract the joint sparsity information. Here, the joint sparsity information is referred to as evaluating a joint support set; the joint support set is defined as the common support set provided by both the OMP and SP algorithms. In the iterative framework, the use of joint sparsity information helps to provide a better support set estimation when the developed HGP algorithm converges. Through experimental evaluations, we show a significant improvement than the OMP and SP algorithms.

Notations: Let $\mathbf{A} \in \mathbb{R}^{M \times N}$, $\mathbf{x} \in \mathbb{R}^N$, and $I \subset \{1,2,\ldots,N\}$. The matrix $\mathbf{A}_I \in \mathbb{R}^{M \times |I|}$ consists of the columns of $\mathbf{A}$ with indices $i \in I$, and $\mathbf{x}_I \in \mathbb{R}^{|I|}$ is composed of the components of $\mathbf{x}$ indexed by $i \in I$. We denote $\bar{I}$ as the compliment of set I. Also we denote $(.)^t$ and $(.)^\dagger$ as transpose and pseudo-inverse respectively.

2. COMPRESSIVE SENSING AND GREEDY PURSUITS

In this section, we first discuss about the CS problem and then discuss two existing greedy pursuit algorithms (OMP and SP) briefly. Let us state the standard CS problem where we acquire a K-sparse signal $\mathbf{x} \in \mathbb{R}^N$ via the linear measurements

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w}, \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{M \times N}$ is a matrix representing the sensing system, $\mathbf{y} \in \mathbb{R}^M$ represents a vector of measurements and $\mathbf{w} \in \mathbb{R}^M$ is additive noise representing measurement errors. A K-sparse signal vector consists of at most K non-zero scalar components. With the setup of $K < M < N$ (under-determined system of linear equations), the task is to reconstruct $\mathbf{x}$ from $\mathbf{y}$ as $\hat{\mathbf{x}}$. Naturally the objective is to strive for a reduced number of measurements as well as achieving good reconstruction quality. Note that, in practice, we may wish to acquire a signal $\mathbf{x}$ that is sparse in a known orthonormal basis and the concerned problem can be easily recast as (1). A column of $\mathbf{A}$ is also called an 'atom' in the literature.

This work was funded in part by VINNOVA, the Strategic Research Area project RICSNET, and EU Fp7 FeedNetBack.

For the signal vector $\mathbf{x} = [x_1, x_2, \ldots, x_N]^t$, the support set $I_\mathbf{x} \subset \{1, 2, \ldots, N\}$ is defined as $I_\mathbf{x} = \{i : x_i \neq 0\}$. For a K-sparse vector $\mathbf{x} \in \mathbb{R}^N$, $|I_\mathbf{x}| = \|\mathbf{x}\|_0 \leq K$. In this paper, we assume that $|I_\mathbf{x}| = K$. Denoting the i'th column (atom) of the measurement matrix $\mathbf{A}$ as $\mathbf{a}_i$, note that

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w} = \sum_{i \in I_\mathbf{x}} x_i \mathbf{a}_i + \mathbf{w} = \mathbf{A}_{I_\mathbf{x}} \mathbf{x}_{I_\mathbf{x}} + \mathbf{w}. \quad (2)$$

From $\mathbf{y}$, if the underlying support set (containing the indices of atoms that are linearly combined) of cardinality K can be identified, then we can estimate the non-zero values of $\mathbf{x}$ using the standard least square (LS) solution (as $K < M$, we can use pseudo-inverse). Therefore, a better estimate of support set leads to a better reconstruction performance.

Next we consider the question of how to construct the sensing matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$. While it is possible to obtain deterministic constructions of $\mathbf{A} = \{a_{i,j}\}$ holding a specific structure, at present the most efficient designs (i.e., those requiring minimum number of rows) rely on random matrix constructions where the $a_{i,j}$'s are assumed realizations of independent and identically distributed (i.i.d.) random variables. A standard method is to draw $a_{i,j}$'s independently from a Gaussian source (i.e., $a_{i,j} \sim \mathcal{N}\left(0, \frac{1}{M}\right)$) and then to scale the columns of $\mathbf{A}$ as unit-norm [8].

Having introduced the basics of CS, we now discuss two existing greedy pursuit algorithms, OMP and SP, with a slight modification in the next subsections. We introduce the modification such that the algorithms become better suited for developing the hybrid greedy pursuit (HGP) algorithm later. The new modification is that the OMP and SP algorithms can use an initial support set information.

2.1 Serial atom selection in OMP

Let us discuss the serial atom selection based OMP algorithm with the supply of an initial support set. We summarize the main steps of the OMP algorithm below.

(OMP for CS Recovery)

Inputs: $\mathbf{A} = [\mathbf{a}_1\ \mathbf{a}_2 \ldots \mathbf{a}_N]$, measurement $\mathbf{y}$, sparsity level K, initial support set I_{ini} where $|I_{ini}| \leq K$.

Initialization:

$k = |I_{ini}|$; (• Set the iteration counter variable)
$I_k = I_{ini}$;
$\hat{\mathbf{x}}_{I_k} = \mathbf{A}_{I_k}^\dagger \mathbf{y}$; $\hat{\mathbf{x}}_{\bar{I}_k} = \mathbf{0}$; (• LS solution: Pseudo-inverse)
$\mathbf{r}_k = \mathbf{y} - \mathbf{A}_{I_k} \hat{\mathbf{x}}_{I_k}$; (• Initial residual)

Iterations:

repeat
$k = k + 1$;
$i_k =$ index of the highest amplitude component of $\mathbf{A}^t \mathbf{r}_{k-1}$ such that $i_k \notin I_{k-1}$; (• Matched filter)
$I_k = I_{k-1} \cup i_k$; (• Note: $|I_k| = k$.)
$\hat{\mathbf{x}}_{I_k} = \mathbf{A}_{I_k}^\dagger \mathbf{y}$; $\hat{\mathbf{x}}_{\bar{I}_k} = \mathbf{0}$; (• LS solution: Pseudo-inverse)
$\mathbf{r}_k = \mathbf{y} - \mathbf{A}_{I_k} \hat{\mathbf{x}}_{I_k}$; (• Residual)
until $((\|\mathbf{r}_k\|_2 > \|\mathbf{r}_{k-1}\|_2)$ or $(k > K))$
$k = k - 1$; (• Previous iteration count)

Outputs: $\hat{\mathbf{x}}^{(omp)}$ (satisfying $\hat{\mathbf{x}}_{I_k}^{(omp)} = \mathbf{A}_{I_k}^\dagger \mathbf{y}$ and $\hat{\mathbf{x}}_{\bar{I}_k}^{(omp)} = \mathbf{0}$); $\hat{I}^{(omp)} = I_k$; $\mathbf{r}^{(omp)} = \mathbf{r}_k$.

The OMP algorithm starts with an initial support set I_{ini}. We set the iteration counter variable $k = |I_{ini}|$ and then set the initial residual according to the LS fit. At the k'th iteration stage, the algorithm uses the 'matched filter' $\mathbf{A}^t \mathbf{r}_{k-1}$ output, identifies the new coordinate (corresponding to an atom) with highest amplitude, solves a LS problem with the selected coordinates, subtracts the LS fit and produces a new residual. Such an approach generally leads to the observation that the residual norm reduces over the iterations obeying the rule $\|\mathbf{r}_k\|_2 \leq \|\mathbf{r}_{k-1}\|_2$. In the case when the rule is violated, the iterative loop is terminated. Therefore $|\hat{I}^{(omp)}| \leq K$. Given the sparsity level K, the algorithm *usually* executes K iterations and forms a support set of cardinality K at the end. Note that the support set cardinality is increased one-by-one through identifying a new atom in each iteration; this is a *serial* (or sequential) approach for atom selection. An important point to note that once an element is deemed reliable in an iteration, the element is added to the support set and remains in the support set forever. We mention that the existing standard OMP algorithm [7] starts with a null support set, i.e., $I_{ini} = \emptyset$.

2.2 Parallel atom selection in SP

Next we summarize the main steps of the parallel atom selection based SP algorithm below where an initial support set is also supplied.

(SP for CS Recovery)

Inputs: $\mathbf{A}$, $\mathbf{y}$, K, I_{ini} where $|I_{ini}| \leq K$.

Initialization:

$\hat{\mathbf{x}}_{I_{ini}} = \mathbf{A}_{I_{ini}}^\dagger \mathbf{y}$; $\hat{\mathbf{x}}_{\bar{I}_{ini}} = \mathbf{0}$; (• LS solution: Pseudo-inverse)
$\mathbf{r}_{ini} = \mathbf{y} - \mathbf{A}_{I_{ini}} \hat{\mathbf{x}}_{I_{ini}}$; (• Residual)
$J =$ indices of the $(K - |I_{ini}|)$ highest amplitude components of $\mathbf{A}^t \mathbf{r}_{ini}$ such that $I_{ini} \cap J = \emptyset$;
$I_0 = I_{ini} \cup J$; (• Note: $|I_0| = K$.)
$\hat{\mathbf{x}}_{I_0} = \mathbf{A}_{I_0}^\dagger \mathbf{y}$; $\hat{\mathbf{x}}_{\bar{I}_0} = \mathbf{0}$; (• LS solution: Pseudo-inverse)
$\mathbf{r}_0 = \mathbf{y} - \mathbf{A}_{I_0} \hat{\mathbf{x}}_{I_0}$;
$k = 0$; (• Set the iteration counter variable)

Iterations:

repeat
$k = k + 1$;
$J = I_{k-1} \cup \{$indices of the K highest amplitude components of $\mathbf{A}^t \mathbf{r}_{k-1}\}$;
$\hat{\mathbf{x}}_J = \mathbf{A}_J^\dagger \mathbf{y}$; $\hat{\mathbf{x}}_{\bar{J}} = \mathbf{0}$; (• LS solution: Pseudo-inverse)
$I_k =$ indices of the K highest amplitude components of $\hat{\mathbf{x}}$;
$\hat{\mathbf{x}}_{I_k} = \mathbf{A}_{I_k}^\dagger \mathbf{y}$; $\hat{\mathbf{x}}_{\bar{I}_k} = \mathbf{0}$; (• LS solution: Pseudo-inverse)
$\mathbf{r}_k = \mathbf{y} - \mathbf{A}_{I_k} \hat{\mathbf{x}}_{I_k}$;
until $(\|\mathbf{r}_k\|_2 > \|\mathbf{r}_{k-1}\|_2)$
$k = k - 1$; (• Previous iteration count)

Outputs: $\hat{\mathbf{x}}^{(sp)}$ (satisfying $\hat{\mathbf{x}}_{I_k}^{(sp)} = \mathbf{A}_{I_k}^\dagger \mathbf{y}$ and $\hat{\mathbf{x}}_{\bar{I}_k}^{(sp)} = \mathbf{0}$); $\hat{I}^{(sp)} = I_k$; $\mathbf{r}^{(sp)} = \mathbf{r}_k$.

The SP algorithm starts with an initial support set I_{ini} where $|I_{ini}| \leq K$. Then it evaluates the initial residual and finds the $(K - |I_{ini}|)$ elements through using a matched filter to form a K-element support set I_0 and corresponding residual $\mathbf{r}_0 = \mathbf{y} - \mathbf{A}_{I_0} \hat{\mathbf{x}}_{I_0}$. At the k'th iteration stage, it uses the 'matched filter' $\mathbf{A}^t \mathbf{r}_{k-1}$ output, identifies the K highest amplitude coordinates, forms a dummy support set J satisfying $K \leq |J| \leq 2K$, refines out K-element support set I_k from J, solves a LS problem with the selected coordinates in I_k, subtracts the LS fit and produces a new residual. Given the sparsity level K, the algorithm usually iterates until the residual minimization condition (over the iterations) is violated

and estimates a support set of cardinality K in each iteration. Note that, as in the case of OMP, the support set cardinality is not increased one-by-one through iterations. Rather, a K-element support set is refined through iterations by addition of promising new atoms and deletion of unnecessary atoms. This procedure of atom selection is different from the OMP. Note that such a *parallel* atom selection strategy allows addition/deletion of atom-indices to/from the support set. We mention that the existing standard SP algorithm [11] starts with a null support set, i.e., $I_{ini} = \emptyset$.

3. HYBRID GREEDY PURSUIT

In this section, we develop the hybrid greedy pursuit (HGP) algorithm. For developing the HGP algorithm, we first need to define two functions. The functions execute the OMP and SP algorithms supplied with an initial support set.

For the OMP algorithm, the corresponding function is defined as

Function 1 *(OMP) Let $\mathbf{y} \in \mathbb{R}^M$, $\mathbf{A} \in \mathbb{R}^{M \times N}$ and K is the sparsity level. Suppose that the initial support set is I_{ini} (where $|I_{ini}| \leq K$). Then, using $\mathbf{A}$, $\mathbf{y}$, K and I_{ini}, let us define the following algorithmic function*

$$[\hat{\mathbf{x}}^{(omp)}, \hat{I}^{(omp)}, \mathbf{r}^{(omp)}] = \mathrm{OMP}(\mathbf{A}, \mathbf{y}, K, I_{ini}), \tag{3}$$

where the outputs are the estimated signal $\hat{\mathbf{x}}^{(omp)} \in \mathbb{R}^N$, its support set $\hat{I}^{(omp)}$ (where $|\hat{I}^{(omp)}| \leq K$), and corresponding residual $\mathbf{r}^{(omp)} \in \mathbb{R}^M$. Here the above function exactly executes the algorithm "OMP" shown in section 2.1.

In the same manner, for the SP algorithm, the corresponding function is defined as

Function 2 *(SP) Let $\mathbf{y} \in \mathbb{R}^M$, $\mathbf{A} \in \mathbb{R}^{M \times N}$ and K is the sparsity level. Suppose that the initial support set is I_{ini} (where $|I_{ini}| \leq K$). Then, using $\mathbf{A}$, $\mathbf{y}$, K and I_{ini}, let us define the following algorithmic function*

$$[\hat{\mathbf{x}}^{(sp)}, \hat{I}^{(sp)}, \mathbf{r}^{(sp)}] = \mathrm{SP}(\mathbf{A}, \mathbf{y}, K, I_{ini}), \tag{4}$$

where the outputs are the estimated signal $\hat{\mathbf{x}}^{(sp)} \in \mathbb{R}^N$, its support set $\hat{I}^{(sp)}$ (where $|\hat{I}^{(sp)}| = K$), and corresponding residual $\mathbf{r}^{(sp)} \in \mathbb{R}^M$. Here the above function exactly executes the algorithm "SP" shown in section 2.2.

Using the above definitions, we now develop the HGP algorithm. The main steps of the HGP algorithm are summarized below.

(HGP for CS Recovery)

Inputs: $\mathbf{A}$, $\mathbf{y}$, K.

Initialization:

$\hat{\mathbf{x}}_0 = \mathbf{0}$, $\mathbf{r}_0 = \mathbf{y}$, $I_0 = \emptyset$;

$J_0 = \emptyset$; (• To keep joint sparsity information)

$k = 0$; (• Set the iteration counter variable)

Iterations:

repeat

$k = k + 1$;

$\left[\hat{\mathbf{x}}^{(omp)}, \hat{I}^{(omp)}, \mathbf{r}^{(omp)}\right] = \mathrm{OMP}(\mathbf{A}, \mathbf{y}, K, J_{k-1})$;

$\left[\hat{\mathbf{x}}^{(sp)}, \hat{I}^{(sp)}, \mathbf{r}^{(sp)}\right] = \mathrm{SP}(\mathbf{A}, \mathbf{y}, K, J_{k-1})$;

$J_k = \hat{I}^{(omp)} \cap \hat{I}^{(sp)}$; (• Joint support estimation)

(• Next: Choice of best algorithm using if-else)

if $(\|\mathbf{r}^{(omp)}\|_2 \leq \|\mathbf{r}^{(sp)}\|_2)$ **then**

$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}^{(omp)}$, $\hat{I}_k = I^{(omp)}$, $\mathbf{r}_k = \mathbf{r}^{(omp)}$;

else

$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}^{(sp)}$, $\hat{I}_k = I^{(sp)}$, $\mathbf{r}_k = \mathbf{r}^{(sp)}$;

end if

until $(\|\mathbf{r}_k\|_2 \geq \|\mathbf{r}_{k-1}\|_2)$

$k = k - 1$; (• Previous iteration count)

Outputs: $\hat{\mathbf{x}}^{(hgp)} = \hat{\mathbf{x}}_k$, $\hat{I}^{(hgp)} = I_k$, $\mathbf{r}^{(hgp)} = \mathbf{r}_k$.

In the HGP algorithm, both the OMP and SP algorithms are executed. The OMP and SP algorithms estimate the underlying support set independently. We find the joint support set information from the estimated two individual support sets and use the joint support set information in an iterative loop. For each iteration, we choose the best performing algorithm between the two competing (OMP and SP) algorithms on the basis of providing minimum residual norm and set the HGP algorithm parameters accordingly. The HGP loop iteration starts with an empty joint support set $J_0 = \emptyset$. Then, through the iterations, the joint support set is improved in such a way that the overall fitting residual decreases. Here, the joint support set is referred to as the common support set provided by both the OMP and SP algorithms. In practice, the joint support set is evaluated as the intersection between two support sets provided by OMP and SP algorithms. For the HGP algorithm, the iteration loop stops when the residual norm shows non-decreasing behavior. The non-decreasing behavior ensures convergence in case the joint support set is not improved further.

A natural observation is that the proposed HGP algorithm is computationally more demanding than the individual OMP and SP algorithms. The HGP will at-least have a computational complexity which is more than the added complexity of OMP and SP algorithms. In the backdrop of HGP, let us considering another simple non-iterative scheme where the joint support set is not used, but the decision is only made on the basis of residual norms corresponding to OMP and SP methods. That means, we execute both the OMP and SP algorithms once, but choose the solution of the OMP algorithm if the OMP residual norm is lower than the SP residual norm or vise versa. The algorithm can be seen as the HGP algorithm where the iteration is performed only once. Let us refer to such an algorithm as HGP-1 ('1' corresponds to only one iteration allowed.). It is interesting to observe the performance of HGP-1 algorithm and compare with the HGP algorithm.

An important point is that here we develop the HGP algorithm using only two greedy pursuit algorithms: OMP and SP. However, in the same algorithmic structure of HGP, it may be possible to use several other existing greedy pursuit algorithms (along-with SP and OMP) if those algorithms can be modified to use a joint sparsity information. That means, those algorithms require to use a joint support set as the initial support set . For example, the standard CoSaMP [10] may be modified to use a joint support set.

4. EXPERIMENTS AND RESULTS

We performed computer simulations in order to compare the performance of four CS reconstruction algorithms: OMP, SP, HGP and HGP-1. We first discuss the reconstruction performance measure and experimental setups, and then show the

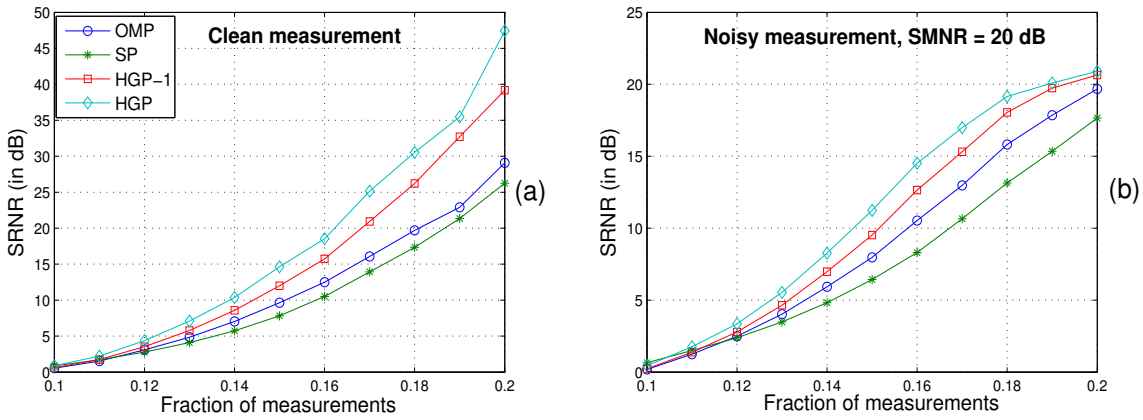

Figure 1: Signal-to-reconstruction-noise error (SRNR) versus fraction of measurements (α). The performances are shown for clean measurement and noisy measurement (with SMNR=20 dB).

experimental results for both clean and noisy setups.

4.1 Performance measure and experimental setups

We use two performance measures. For the first performance measure, we use signal-to-reconstruction-noise ratio (SRNR) defined as

$$\text{SRNR} = \frac{\mathscr{E}\{\|\mathbf{x}\|_2^2\}}{\mathscr{E}\{\|\mathbf{x}-\hat{\mathbf{x}}\|_2^2\}}, \tag{5}$$

where $\hat{\mathbf{x}}$ is the reconstruction of the signal $\mathbf{x}$ using a CS method. Next we define another performance measure which provides a direct measure of estimating the underlying support set. For a K-sparse signal vector $\mathbf{x}$, the support set was denoted as $I_{\mathbf{x}}$ with cardinality K. Let us denote the support set of reconstructed vector $\hat{\mathbf{x}}$ as $I_{\hat{\mathbf{x}}}$. We assume that $\hat{\mathbf{x}}$ is also a K-sparse signal vector, i.e. $|I_{\hat{\mathbf{x}}}| = K$. To compare the methods, we consider to use the distortion $d(I_{\mathbf{x}}, I_{\hat{\mathbf{x}}}) = 1-(|I_{\mathbf{x}} \cap I_{\hat{\mathbf{x}}}|/K)$ to measure errors [13]. Considering a large number of realizations (data vectors), we can compute the average of $d(I_{\mathbf{x}}, I_{\hat{\mathbf{x}}})$. We define the average support-cardinality error (ASCE) as follows

$$ASCE = \mathscr{E}\{d(I_{\mathbf{x}}, I_{\hat{\mathbf{x}}})\} = 1 - \frac{1}{K}\mathscr{E}\{|I_{\mathbf{x}} \cap I_{\hat{\mathbf{x}}}|\}. \tag{6}$$

Along-with SRNR, the ASCE is used as the second performance evaluation measure because our main objective is to estimate the underlying support set.

Next we discuss experimental setups. In a CS setup, all sparse signal vectors are expected to be exactly reconstructed if the number of measurements is more than a certain threshold value. However, the computational complexity to test this uniform reconstruction ability is exponentially high. Instead, for empirical testing, we can devise a strategy that can compute SRNR and ASCE for random measurement matrix ensemble. Let us define the fraction of measurements (FoM)

$$\alpha = \frac{M}{N}. \tag{7}$$

Using α, steps of the testing strategy are listed as follows:

1. For given values of the parameters K and N, choose α such that number of measurements M is an integer.
2. Randomly generate an $M \times N$ sensing matrix where the components are drawn independently from a Gaussian source (i.e., $a_{i,j} \sim \mathscr{N}\left(0, \frac{1}{M}\right)$) and then to scale the columns of $\mathbf{A}$ as unit-norm.
3. Randomly generate a set of K-sparse data where the support set $I_{\mathbf{x}}$ is chosen uniformly over the set $\{1,2,\ldots,N\}$. Let we denote the size of data as S (i.e. the number of signal vectors is S). The non-zero components of $\mathbf{x}$ are independently drawn from a standard Gaussian source. This type of signal is referred to as Gaussian sparse signal. Note that the Gaussian sparse signal is compressible in nature. That means, in the descending order, the sorted amplitudes of a sparse signal vector's components decay fast with respect to the sorted indices. This decaying trend corroborates with several natural signals (for example, wavelet coefficients of an image).
4. For each data, compute the measurement $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w}$ and apply the CS reconstruction methods independently. Here we assume that the measurement noise $\mathbf{w} \in \mathbb{R}^M$ has a multidimensional pdf as $\mathbf{w} \sim \mathscr{N}\left(\mathbf{0}, \sigma_w^2 \mathbf{I}_M\right)$, where $\mathbf{I}_M$ is an $M \times M$ identity matrix.
5. Repeat steps 2-4 for a given times (let T times). Then evaluate the CS performance evaluation measures: SRNR and ASCE (averaging over ST data).
6. Repeat steps 1-5 for a new α.

This testing strategy can be performed for any chosen K and N.

Now we define the signal-to-measurement-noise-ratio (SMNR) as

$$\text{SMNR} = \frac{\mathscr{E}\{\|\mathbf{x}\|_2^2\}}{\mathscr{E}\{\|\mathbf{w}\|_2^2\}}. \tag{8}$$

4.2 Experimental results

Using $N = 500$, $K = 20$, $S = 100$ and $T = 100$, we performed experiments. That means, we used 500-dimensional sparse signal vectors with sparsity level $K = 20$. Such a 4% sparsity level is chosen in accordance with real life scenarios, such as most of the energy of an image signal in the wavelet domain is concentrated within $2-4\%$ coefficients. We used 100 realizations of $\mathbf{A}$ ($T = 100$). For each realization of $\mathbf{A}$, we used 100 signal vectors that are randomly generated ($S = 100$).

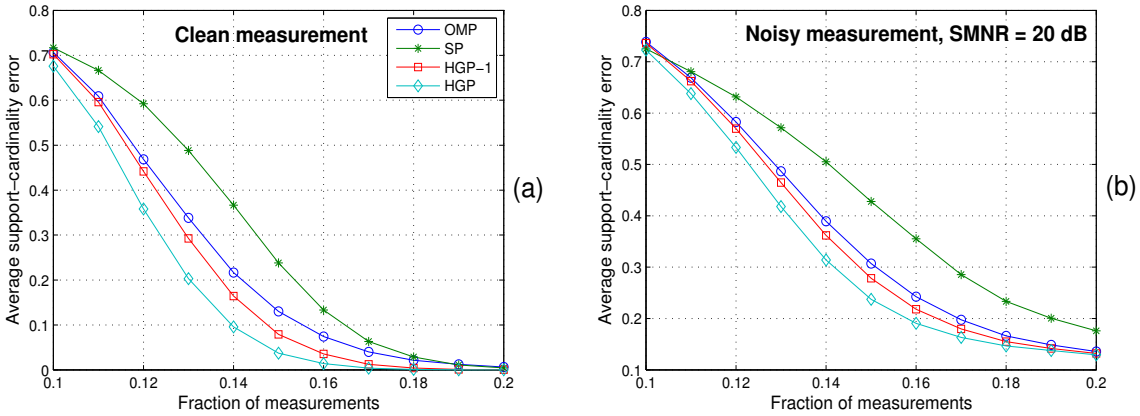

Figure 2: Average support-cardinality error (ASCE) versus fraction of measurements (α). The performances are shown for clean measurement and noisy measurement (with SMNR=20 dB).

Then, we incremented α from a lower limit to a higher limit using some small step-sizes (with the constraint that corresponding M is an integer for a value of α). Therefore, for each CS method at a chosen α, the performance is evaluated through averaging over $100 \times 100 = 10000$ realizations.

Fig. (1) shows the SRNR performance (in dB) of CS reconstruction methods for both clean and noisy measurement conditions. In case of the noisy measurement, we choose the SMNR=20 dB. We show the results for the range of α from 0.1 to 0.2. This range of α corresponds to the range of M from $M = 50$ to $M = 100$. We observe that the HGP-1 shows better performance than the OMP and SP methods and HGP is found to be the best. For the clean measurement case, the HGP shows considerable improvement over HGP-1 as the number of measurements increases. In the range $\alpha = 0.16$ to 0.2, the HGP provides more than 4 dB improvement compared to the HGP-1. Compared to the OMP and SP methods, the HGP provides significant improvement (more than 10 dB in the range $0.16 < \alpha \leq 0.2$). For the noisy measurement case, the HGP provides more than 2 dB improvement than the HGP-1 in the range $\alpha = 0.14$ to 0.18. Compared to the OMP and SP methods, the HGP provides a significant improvement (nearly 5 dB in the range $0.14 \leq \alpha \leq 0.18$).

Next we show the ASCE performance of CS reconstruction methods in Fig. (2). We note that the HGP provides the best performance. For clean measurement, the HGP algorithm provides nearly zero ASCE distortion for the range $\alpha \geq 0.17$. On the other hand, for the noisy case, the ASCE gradually reduces and maintains a nearly saturated behavior for the range $\alpha \geq 0.18$.

5. CONCLUSIONS

In an iterative framework, we have shown that simultaneous use of greedy pursuit algorithms leads to better compressive sensing recovery performance through exploiting joint sparsity information. At the expense of higher computational complexity, such an engineering approach has a similarity with a universal source coding method [14], [15], where several coders are jointly used to provide better quantization performance.

REFERENCES

[1] E.J. Candes and M.B. Wakin, "An introduction to compressive sampling," *IEEE Signal Proc. Magazine*, vol. 25, pp. 21-30, March 2008.

[2] S.S. Chen, D.L. Donoho, and M.A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Scientif. Comput.*, vol. 20, no. 1, pp. 33-61, 1998.

[3] E.J. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. Info. Theory*, vol. 51, no. 12, pp. 4203-4215, 2005.

[4] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Trans. Signal Proc.*, vol. 56, no. 6, pp. 2346-2356, 2008.

[5] D.P. Wipf and B.D. Rao, "Sparse Bayesian learning for basis selection," *IEEE Trans. Signal Proc.*, vol. 52, no. 8, pp. 2153-2164, 2004.

[6] S.G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Proc.*, vol. 41, no. 12, pp. 3397-3415, 1993.

[7] J.A. Tropp and A.C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Info. Theory*, vol. 53, no. 12, pp. 4655-4666, 2007.

[8] D.L. Donoho, et. al., "Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit," http://www-stat.stanford.edu/∼donoho/reports.html, March 2006.

[9] D. Needell and R. Vershynin, "Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit," *J. Found Comput. Math.*, vol. 9, pp. 317-334, 2009.

[10] D. Needell and J.A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," Tech. Report, no. 2008-01, California Inst. Tech., July 2008.

[11] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Trans. Info. Theory*, vol. 55, no. 5, pp. 2230-2249, 2009.

[12] S. Chatterjee, D. Sundman and M. Skoglund, "Look ahead orthogonal matching pursuit," *accepted for ICASSP 2011*.

[13] G. Reeves and M. Gastpar, "A note on optimal support recovery in compressed sensing," *Proc. Asilomar conf. Signals, System, and Computers*, pp. 1576-1580, 2009.

[14] P.A. Chou, M. Effros and R.M. Gray, "A vector quantization approach to universal noiseless coding and quantization," *IEEE Trans. Info. Theory*, vol. 42, no. 4, pp. 1109-1138, July 1996.

[15] B. Bessette, R. Lefebvre and R. Salami, "Universal speech/audio coding using hybrid ACELP/TCX techniques," *Proc. ICASSP*, Philadelphia, Vol. III, pp. 301-304, March 2005.