Greedy Sparse Signal Reconstruction Using Matching Pursuit Based on Hope-tree

Zhetao Li, Hongqing Zeng, Chengqing Li, and Jun Fang

Abstract—The reconstruction of sparse signals requires the solution of an $\ell_0$-norm minimization problem in Compressed Sensing. Previous research has focused on the investigation of a single candidate to identify the support (index of nonzero elements) of a sparse signal. To ensure that the optimal candidate can be obtained in each iteration, we propose here an iterative greedy reconstruction algorithm (GSRA). First, the intersection of the support sets estimated by the Orthogonal Matching Pursuit (OMP) and Subspace Pursuit (SP) is set as the initial support set. Then, a hope-tree is built to expand the set. Finally, a developed decreasing subspace pursuit method is used to rectify the candidate set. Detailed simulation results demonstrate that GSRA is more accurate than other typical methods in recovering Gaussian signals, 0–1 sparse signals, and synthetic signals.

Index Terms—Compressed Sensing, greedy reconstruction algorithm, hope-tree, matching pursuit, sparse signal recovery.

I. INTRODUCTION

In the past decade, Compressed Sensing (CS) has received considerable attention as a means to reconstruct sparse signals in underdetermined systems. The basic premise of CS is that a $K$-sparse signal $x \in \mathbb{R}^n$ can be recovered perfectly with far fewer compressed measurements $y = \Phi x \in \mathbb{R}^m$ than the traditional approaches [1], [2], [3], [4]. The problem of reconstructing the original signal can be formulated as an $\ell_0$-minimization problem

$$\min_{x} \|x\|_0 \quad \text{subject to} \quad y = \Phi x,$$

where $\|\cdot\|_p$ denotes the $p$th norm, $\Phi \in \mathbb{R}^{m \times n}$ is the sensing matrix, and $m < n$. Since the $\ell_0$-minimization problem is NP-hard [1], it is often transformed into an $\ell_1$-minimization problem to make it tractable. Concretely, some $\ell_1$-relaxation methods, such as Basis Pursuit (BP) and BP denoising (BPDN), can be used to solve it [5].

To greatly reduce the computational complexity of the $\ell_1$-relaxation methods, algorithms based on greedy pursuit have been adopted, which use the least squares method to estimate the sparse signal $x$ by finding the positions of its nonzero elements. The sparse signal can be represented as $x = \sum_{j \in S} e_j c_j$, where the $e_j$ are the standard unit vectors and $S$ is the support set of $x$. In Orthogonal Matching Pursuit (OMP), the index of the column possessing the strongest correlation with the modified measurements is chosen as the new element of the support in each iteration [6]. Note that if any index in the support set is not right, the result of OMP would be wrong. To mitigate this weakness of OMP, improved selection methods for indices have been developed, such as Stagewise Orthogonal Matching Pursuit (StOMP) [7], which adopts a threshold mechanism. Subspace pursuit (SP) and Compressive Sampling Matching Pursuit (CoSaMP) select indices exceeding the sparsity level by a pruning [8], [9]. Multipath Matching Pursuit (MMP) builds a tree to find the optimal support set [10]. Generalized Orthogonal Matching Pursuit (gOMP) selects multiple indices [11]. However, these algorithms cannot obtain an acceptable trade-off behavior between computational complexity and reconstruction performance.

In this letter, we propose an iterative greedy signal reconstruction algorithm (GSRA) using matching pursuit based on hope-tree. First, the intersection of the support sets is estimated by OMP and SP and taken as the initial support set. Then, a hope-tree is built to obtain a candidate support $S$ by setting the search depth, which is rectified by a decreasing subspace pursuit method. Finally, we calculate that the complexity of GSRA is $O(N_{max} \cdot mn \cdot iter)$, where $iter$ is the number of iterations and $N_{max}$ is the number of candidates in each iteration.

The rest of this paper is organized as follows. In Sec. II we introduce the matching pursuit used in the GSRA. In Sec. III we provide the efficiency analysis of the GSRA. Section IV presents detailed empirical experiments on its reconstruction performance. The last section concludes the paper.

II. MATCHING PURSUIT BASED ON HOPE-TREE

The proposed GSRA is composed of three stages: pre-selection, hope-tree search, and rectification. These are described in the following three subsections.

A. Pre-selection

The purpose of pre-selection is to obtain the indices belonging to the target support set $T$ with a high probability. The initial support set $\Theta$ can be taken to be the intersection of the support sets estimated by different types of reconstruction algorithms, such as OMP, SP and gOMP. If three or more such sets are used, the reconstruction performance cannot be improved further, but the whole complexity is greatly increased. As the index selection strategy of OMP is different from the others,
we only adopt OMP and SP, so as to satisfy the computational efficiency requirement.

B. Hope-Tree search

Once the pre-selection is finished, a hope-tree is built to search for the candidate support. As shown in Fig. 1, the search depth is set as \( N \) in the \( k \)-th iteration, and each path generates \( L \) (\( \leq K \)) child paths with indices \( \bar{\pi}, \tilde{\pi}, \cdots, \bar{\pi}_L \) of those columns possessing the strongest correlations with the residual, according to the condition

\[
\{\bar{\pi}_1, \bar{\pi}_2, \cdots, \bar{\pi}_L\} = \arg \max_{|\pi|=L} \| (\Phi^T r_{k-1})_\pi \|^2_2.
\]

When the search depth approaches \( N \), a sub-tree of the hope-tree is built. The modulo strategy in [10] is used to compute the layer order, and execute a priority search for promising candidates. Then, the indices of the optimal paths with the minimum residuals are selected as the best elements of \( T \) and added to the set \( \Theta \). Next, the last optimal index of the \( k \)-th iteration is chosen as the root index. The above procedures are repeated in the \((k+1)\)-th iteration. The tree search procedure is stopped when the length of the support \( \Theta \) is not smaller than \( K \). In each sub-process, the number of candidates increases by the factor \( L \), resulting in \( L^N \) candidates after \( N \) iterations. These operations are repeated until either the candidate support satisfies \( S = \Theta \) or the \( \ell_2 \)-norm of the residual falls below a preset threshold, i.e., \( \| r^k \|_2 \leq \varsigma \).

\begin{algorithm}
\caption{Matching Pursuit Based on Hope-Tree.}
\begin{algorithmic}
\State \textbf{Data:} measurement vector \( y \), Sensing Matrix \( \Phi \), sparsity level \( K \), number of path \( L \), search depth \( N \)
\State \( J_{\text{omp}} = \text{OMP}(\Phi, y, K) \), \hspace{1em} // OMP support set
\State \( J_{\text{sp}} = \text{SP}(\Phi, y, K) \), \hspace{1em} // SP support set
\State \text{initial candidate support set } \lambda = J_{\text{omp}} \cap J_{\text{sp}}.
\State \textbf{Result:} \( \tilde{x}, S \)
\State \textbf{Initialization:}
\State Residual vector \( r^0 = y - \Phi x_\lambda \), candidate support \( \Lambda^0 = \lambda \), backtracking subspace parameter \( \alpha \)
\State \( 0 < \alpha < 1 \), iteration index \( k = 0 \)
\While{\( \| \Lambda \|_0 < K \)}
\State \( k = k + 1 \)
\Comment{\( [\text{supp}, \sim] = \text{CreateHopeTree}(y, \Phi, r^k, K, N, L) \)}
\State \textbf{Update candidate support:} \( \Lambda^k = \Lambda^{k-1} \cup \text{supp} \)
\State \textbf{Estimate} \( \tilde{x}_{\Lambda^k} : \tilde{x}_{\Lambda^k} = \Phi_{\Lambda^k}^\dagger y \)
\State \textbf{Update residual:} \( r^k = y - \Phi_{\Lambda^k} \tilde{x}_{\Lambda^k} \)
\EndWhile
\State \( [\tilde{x}, S] = \text{RectifySupport}(\Lambda, y, \Phi, r_{\text{new}}, \alpha) \)
\end{algorithmic}
\end{algorithm}

Fig. 1. Illustration of the hope-tree search, where the black dots denote the optimal candidate.

C. Rectification

Because searching the hope-tree may ignore any correct indices falling outside of the hope-tree, a developed decreasing subspace pursuit method is used to obtain the indices from the latter and improve the accuracy of the candidate support set. In the first iteration, the size of the search space is \( T = K \), and the decreasing subspace pursuit method expands the selected support with the indices of the \( T \) largest magnitude entries of \( \Phi^T r \). Once the size of the extended support \( S^k \) reaches \( T + K \), the orthogonal projection coefficients of \( y \) onto \( \Phi_{S^k} \) are calculated, and the estimated support \( S^k \) is obtained by pruning it to contain only the \( K \) indices corresponding to the largest elements of \( \Phi_{S^k} y \), where \( k \) is the iteration index. In the next iteration, the length of the search space decreases according to \( T_{k+1} = \alpha T_k \), while the execution remains the same as in the first iteration, where \( 0 < \alpha < 1 \). When \( T \) becomes smaller than a certain threshold, the iteration is terminated. The final support set \( S \) and the estimated version of the signal \( x \) are determined.

The pseudocode of GSRA is summarized as Algorithm 1 where the process of building the hope-tree and the subprocess of rectifying the candidate support set are summarized as Algorithm 2 and Algorithm 3 respectively. The main merit of the reconstruction algorithm of GSRA is that it creates a process of searching the hope-tree and rectifying the candidate support set, which can balance the contradiction between computational complexity and reconstruction performance.

III. EFFICIENCY ANALYSIS

A. Parameter setting

A sensing matrix \( \Phi \) is said to satisfy the restricted isometry property (RIP) of order \( K \) if there exists a constant \( \delta \in [0, 1] \) such that

\[
(1 - \delta) \| x \|_2^2 \leq \| x \|_2^2 \leq (1 + \delta) \| x \|_2^2
\]

for any \( K \)-sparse vector \( x \). In particular, the minimum of all constants \( \delta \) ensuring that the correct indices can be obtained is referred to as the isometry constant \( \delta_K \). In the clean case, it is assumed that \( \Phi \) satisfies the RIP of order \( NK \), i.e., \( \delta_{NK} \in (0, 1) \). So, one has \( 0 < 1 - \delta_{NK} \leq \lambda_{\min}(\Phi_D^T \Phi_D) \) for any index set \( N \) satisfying \( |D| \leq NK \), which indicates that all eigenvalues of \( \Phi_D \) are positive. Thus, \( \Phi_D \) has full column rank (i.e., \( K \leq m/N \)) and \( N \leq \sqrt{m} \). Therefore, one has \( N < \min(K, \sqrt{m}) \). However, in the noisy case, a noisy disturbance makes the GSRA improve its performance.
Algorithm 2: \([supp, \sim]=CreateHopeTree(y, \Phi, r, K, N, L)\)

**Data:** Initialization: maximum number of search candidate \(N_{\text{max}}\), stop threshold \(\varsigma\), candidate order \(\ell = 0\), minimal magnitude of residual \(\rho = \infty\)

**Result:** \(supp\)

while \(\ell < N_{\text{max}}\) and \(\varsigma < \rho\) do

\(\ell = \ell + 1\), \(r^0 = r\), \(\text{temp} = \ell - 1\)

for \(k = 1\) to \(N\) do

\(c_k = \text{temp} \mod L + 1\) \(//\) compute layer order

\(\text{temp} = \lceil\text{temp}/L\rceil\)

\(c_k = c_k \cup c_{k-1}\)

end for \(k = 1\) to \(N\) do

\(\bar{\pi} = \arg \max_{|\pi| = \ell} \| (\Phi^T r^{k-1})_{\pi} \|_2^2 \) \(//\) choose \(L\) best indices

\(s^k = s^{k-1} \cup \bar{\pi}_{c_k}\)

\(\hat{x} = \Phi_{s^k} y\)

\(r^k = y - \Phi_{\hat{x}} \hat{x}^k\)

end

if \(\|r^N\| \leq \rho\) then \(supp = s\), \(\rho = \|r^N\|\); end

Algorithm 3: \([\hat{x}, S]=RectifySupport(\Lambda, y, \Phi, r_{\text{new}}, \alpha)\)

**Data:** the support set \(\Lambda\), backtrack subspace parameter \(T\) (\(T \leq K\)) and residual \(r_{\text{new}}\)

**Result:** \(x\) (estimated signal), \(S\) (estimated support)

while \(T != 0\) do

\(S = T\) indices of highest amplitude components of \(\Phi^T r_{\text{new}}, \Lambda = \Lambda \cup S; \hat{x} = A_{\Lambda}^T y\)

\(S = K\) indices corresponding to the largest magnitude components of \(x, \; \tilde{x} = \Phi_{s^k} y\); \(r_{\text{new}} = y - \Phi \tilde{x}; \; T = \alpha T\)

end

at the cost of increasing the value of the search depth. To make \(\Phi\) satisfy RIP and \(S\) include the correct indices in each optimal path as much as possible, the value of \(N\) can be set appropriately with the upper bound \(K\).

**B. Computational Complexity Analysis**

The complexities of the three stages of GSRA can be estimated separately. The complexity for obtaining the initial support set is

\[ O_{\text{omp}}(mn \cdot K) + O_{sp}(mn \cdot K) \approx O(mn \cdot K), \]

where \(O_{\text{omp}}(mn \cdot K)\) and \(O_{sp}(mn \cdot K)\) are derived in [8]. The number of candidates in the \(k\)th iteration is \(L^N\). To decrease the computational complexity, the maximum number of candidates \(N_{\text{max}}\) is less than \(L^N\). Obtaining one index requires \(O(mn)\) according to the expression \(\Phi^T r^k\). Choosing \(N\) indices from \(N_{\text{max}}\) candidates in each iteration means that the complexity of searching the hope-tree is \(O(N_{\text{max}} \cdot mn)\). As the number of iterations is set to be \(iter < K\), the complexity of searching the hope-tree is bounded by \(O(N_{\text{max}} \cdot mn \cdot iter)\). In addition, the number of selecting elements is reduced in each iteration; the complexity of the decreasing subspace method is bounded by \(O(mn)\). Therefore, the total complexity of GSRA is \(O(N_{\text{max}} \cdot n \cdot iter) + O(mn \cdot K) + O(mn) \approx O(N_{\text{max}} \cdot mn \cdot iter)\). When the value of \(K\) is low, the complexity of GSRA is close to that of OMP. In addition, the complexity of GSRA monotonously increases with respect to \(K\) and far less than that of the breadth-first search method.

**IV. SIMULATION RESULTS**

To check the real performance of GSRA for solving the sparse signal recovery problem, it was compared with OMP, gOMP, MMP-DFS, MMP-BFS using the same setup adopted in [8].

**A. Reconstruction of Sparse Signals Whose Nonzero Values Follow \(N(0, 1)\)**

In this experiment, some important parameters were set as follows:

1) Generate a random matrix \(\Phi\) of size \(m \times n\) with i.i.d. \(N(0, 1)\) entries. To facilitate comparison, we set \(m = 128\) and \(n = 256\).

2) To reduce the time complexity, set the maximum number of candidates of sub-tree \(N_{\text{max}} = 30\), the number of paths \(L = 2\), and the search depth \(N = 7\).

3) Choose a \(K\)-subset of \(\{1, 2, \cdots, n\}\), \(K = 1, 2, \cdots, 60\).

4) Set the value of \(x\) in the \(K\) chosen indices as random nonzero values obeying \(N(0, 1)\) and that in the remaining indices as zero.

5) Conduct 1,000 simulations to calculate the frequency of the exact reconstruction and the mean time needed for the different algorithms.

Figure 2a) demonstrates that GSRA can achieve an exact reconstruction with a greater frequency than OMP, gOMP, MMP-DFS and MMP-BFS for Gaussian signal of any sparsity level. Concretely, OMP, gOMP, MMP-DFS and MMP-BFS can accurately reconstruct a signal of various sparsity levels, such as 21, 40, 35, and 43. In contrast, GSRA can still do this when the level reaches 50. Figure 2b) shows that the time needed increases with an increase in the sparsity level, and the mean time needed by GSRA is shorter than that of MMP-DFS and MMP-BFS, but longer than that of OMP and gOMP.

**B. Reconstruction of 0–1 Sparse Signals**

In this experiment, we choose a \(K\)-subset of \(\{1, 2, \cdots, n\}\), \(K = 1, 2, \cdots, 50\). For each value of \(K\), we generate a signal \(x\) of sparsity level \(K\), where the nonzero coefficients are set to one and those in the remaining indices to zero. The other configurations are the same as in the above sub-section.

Figure 3a) shows that the GSRA performs better than OMP, gOMP, MMP-DFS and MMP-BFS in terms of the frequency of exact reconstruction for 0–1 sparse signals of any sparsity level. OMP, gOMP, MMP-DFS and MMP-BFS can accurately reconstruct signals of various sparsity levels: 10, 28, 20, and
Fig. 2. Performance comparison of reconstructing typical sparse signals in terms of two metrics: a) frequency of exact reconstruction; b) mean time needed.

23. Meanwhile, GSRA can obtain satisfying results when the sparsity level is increased to 30. As shown in Figure 3b), the comparison of the results of GSRA with the other reference algorithms in terms of the mean time needed of GSRA are the same as that in the above subsection.

C. Reconstruction of Synthetic Signals

To further verify the performance of GSRA in comparison with the four reference algorithms, some experiments were done for the recovery of an additive synthetic signal

\[ y = \Phi x + w. \tag{4} \]

The synthesis degree is measured by the Signal to Measurement-Noise Ratio

\[ SMNR = 10 \log_{10} \frac{E \|x\|_2^2}{E \|w\|_2^2} = 10 \log_{10} \frac{K \cdot \sigma_s^2}{m \cdot \sigma_l^2}, \tag{5} \]

where \( E(\bullet) \) denotes the mathematical expectation operator, \( \sigma_s^2 \) and \( \sigma_l^2 \) denote the power of each element of the signal and noise vector, respectively. To measure the recovery degree more accurately, the Signal-to-Reconstruction-Error Ratio

\[ SRER = 10 \log_{10} \frac{E \|x\|_2^2}{E \|x - \hat{x}\|_2^2} \]

is adopted here instead.

The concrete steps of the experiment are described as follows.

1) Set the sparsity level \( K = 20 \) and fix \( n = 500 \). Then, adjust the value of the sampling rate \( \varphi = m/n \) so that the number of measurements \( m \) is an integer.
2) Generate elements of the \( \Phi \) independently from a source following \( \mathcal{N}(0, 1) \) and normalize each column norm to unity.
3) For the noisy region of the signal $x$, the additive noise $w$ is set as a Gaussian random vector whose elements are independently chosen with distribution $N(0, \sigma^2_w)$.

4) Select a measurement vector with $\text{SNMR} = 20$.

5) Apply the reconstruction methods in Algorithm 1.

6) Repeat steps 3)–5) $T = 10$ times.

7) Repeat steps 2)–6) $Q = 100$ times.

8) Calculate the average value of the $\text{SRER}$ scores for the $T \cdot Q$ signals.

As for GSRA, the other parameters are same as in Sec. IV-A except the search depth. With an increase of depth, the performance of GSRA is improved. Typically, the search depth is set to be 16. The above eight experimental steps were executed with the sampling rate $\varphi$ ranging from 0.15 to 0.20, and the obtained results are listed in Table I which demonstrates that GSRA performs better than any other reference scheme in each case.

| $\varphi$ | OMP | gOMP | MMP-DFS | MMP-BFS | GSRA |
|-----------|-----|------|---------|---------|------|
| 0.15      | 8.11| 7.75 | 15.74   | 15.38   | 16.05|
| 0.16      | 9.57| 9.08 | 18.09   | 17.63   | 18.30|
| 0.17      | 12.24| 11.37| 18.47   | 17.51   | 19.07|
| 0.18      | 14.71| 13.48| 19.67   | 18.51   | 19.20|
| 0.19      | 17.49| 15.08| 20.28   | 19.43   | 20.39|
| 0.20      | 19.31| 15.90| 20.89   | 20.04   | 20.98|

V. CONCLUSION

To improve the accuracy of sparse signal recovery, this paper proposed an iterative greedy reconstruction algorithm by examining multiple promising candidates with the help of greedy search. The key feature of the algorithm is to use a hope-tree and the decreasing subspace pursuit method to obtain the final support set. Detailed experimental results demonstrated that the proposed algorithm performs well for Gaussian signals, 0–1 sparse signals, and synthetic signals. Research on making the depth of the GSRA adaptive deserves further exploration.

REFERENCES

[1] E. J. Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” IEEE Trans. Inf. Theory, vol. 52, no. 2, pp. 489–509, 2006.

[2] E. J. Candès and T. Tao, “Decoding by linear programming,” IEEE Trans. Inf. Theory, vol. 51, no. 12, pp. 4203–4215, 2005.

[3] Y. Shen, J. Fang, and H. Li, “Exact reconstruction analysis of log-sum minimization for compressed sensing,” IEEE Signal Process. Lett., vol. 20, no. 12, pp. 1223–1226, 2013.

[4] J.-F. Determe, J. Louveaux, L. Jacques, and F. Horlin, “On the exact recovery condition of simultaneous orthogonal matching pursuit,” IEEE Signal Process. Lett., vol. 23, no. 1, pp. 164–168, 2016.

[5] S. S. Chen, D. L. Donoho, and M. A. Saunders, “Atomic decomposition by basis pursuit,” SIAM Rev., vol. 43, no. 1, pp. 129–159, 2001.

[6] J. Wang and B. Shim, “On the recovery limit of sparse signals using orthogonal matching pursuit,” IEEE Trans. Signal Process., vol. 60, no. 9, pp. 4973–4976, 2012.

[7] 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, no. 2, pp. 1094–1121, 2012.

[8] W. Dai and O. Milenkovic, “Subspace pursuit for compressive sensing signal reconstruction,” IEEE Trans. Inf. Theory, vol. 55, no. 5, pp. 2230–2249, 2009.

[9] D. Needell and J. A. Tropp, “Cosamp: iterative signal recovery from incomplete and inaccurate samples,” Commun. ACM, vol. 53, no. 12, pp. 93–100, 2010.

[10] S. Kwon, J. Wang, and B. Shim, “Multipath matching pursuit,” IEEE Trans. Inf. Theory, vol. 60, no. 5, pp. 2986–3001, 2014.

[11] J. Wang, S. Kwon, and B. Shim, “Generalized orthogonal matching pursuit,” IEEE Trans. Signal Process., vol. 60, no. 12, pp. 6202–6216, 2012.