Stepwise Optimal Subspace Pursuit for Improving Sparse Recovery

Balakrishnan Varadarajan, Sanjeev Khudanpur, and Trac D. Tran

***Abstract*—We propose a new iterative algorithm to reconstruct an unknown sparse signal $\mathbf{x}$ from a set of projected measurements $\mathbf{y} = \mathbf{\Phi}\mathbf{x}$. Unlike existing methods, which rely crucially on the near orthogonality of the sampling matrix $\mathbf{\Phi}$, our approach makes stepwise optimal updates even when the columns of $\mathbf{\Phi}$ are *not* orthogonal. We invoke a block-wise matrix inversion formula to obtain a closed-form expression for the increase (reduction) in the L_2-norm of the residue obtained by removing (adding) a single element from (to) the presumed support of $\mathbf{x}$. We then use this expression to design a computationally tractable algorithm to search for the nonzero components of $\mathbf{x}$. We show that compared to currently popular sparsity seeking matching pursuit algorithms, each step of the proposed algorithm is locally optimal with respect to the actual objective function. We demonstrate experimentally that the algorithm significantly outperforms conventional techniques in recovering sparse signals whose nonzero values have exponentially decaying magnitudes or are distributed $\mathcal{N}(0,1)$.**

***Index Terms*—Compressed sensing, greedy algorithms, least squares, signal reconstruction.**

I. Sparse Signal Recovery

Let $\mathbf{x} \in \mathbb{R}^N$ be a sparse signal with $\|\mathbf{x}\|_0 = K \ll N$, and let $\mathbf{y} \in \mathbb{R}^M$ be an observation of $\mathbf{x}$ via M linear measurements represented by a matrix $\mathbf{\Phi}$. In other words, let $\mathbf{y} = \mathbf{\Phi}\mathbf{x}$. It is known that if M is sufficiently larger than K (we assume $M \geq 2K$), and $\mathbf{\Phi}$ satisfies some rank conditions, then

$$\mathbf{x} = \arg\min_{\mathbf{y}=\mathbf{\Phi}\mathbf{w}} \|\mathbf{w}\|_0. \tag{1}$$

It was established by Tropp *et al.* [1], for instance, that $\mathcal{O}(K \log N)$ probabilistic measurements are sufficient in practice to recover a K-sparse signal in $\mathbb{R}^N$. Solutions to (1) therefore lead to algorithms for recovering K-sparse signals from M linear measurements. In general, the minimization of (1) is NP-hard, since it requires searching through $\binom{N}{K}$ possible column-subsets of $\mathbf{\Phi}$. Is has been shown, however, that if the sampling matrix $\mathbf{\Phi}$ further satisfies certain properties relating to near orthogonality, then the following tractable minimization also recovers $\mathbf{x}$ exactly ([2]–[5]):

$$\mathbf{x} = \arg\min_{\mathbf{y}=\mathbf{\Phi}\mathbf{w}} \|\mathbf{w}\|_1. \tag{2}$$

Unfortunately, the complexity of the linear programming algorithms for solving (2), also known as basis pursuit (BP), is $\mathcal{O}(N^3)$, making them infeasible for practical, large-scale applications. Some fast convex relaxation algorithms have been proposed to solve or approximate BP, a popular example being the gradient projection method of [6].

An alternative approach to sparse signal recovery is based on the idea of iterative greedy pursuit, and tries to approximate the solution to (1) directly. The earliest examples include matching pursuit, orthogonal matching pursuit (OMP) [7], and their variants such as stagewise OMP (StOMP) [8] and regularized OMP (ROMP) [9]. The reconstruction complexity of these approximate algorithms is around $\mathcal{O}(KMN)$, which is significantly lower than the complexity of BP. However, they require more measurements M for accurate reconstruction, and they lack provable reconstruction quality. More recently, greedy algorithms with a backtracking mechanism, such as subpace pursuit (SP) [10] and compressive sampling matching pursuit (CoSaMP) [11], have offered comparable theoretical reconstruction quality to the linear programming methods along with low reconstruction complexity. Our proposed algorithm belongs to this latter class of recovery algorithms: it iteratively refines an estimate of the support set of $\mathbf{x}$, denoted by $(\mathbf{x})$, in a manner similar to SP [10].

The SP algorithm relies on the fact that high correlation of a column of $\mathbf{\Phi}$ with the observed $\mathbf{y}$ corresponds to a desirable index in the support set of $\mathbf{x}$. SP iterates over two key steps.

S1. *Expansion*: At iteration l, if the *current estimate* of $\mathrm{supp}(\mathbf{x})$ is a set of K indices denoted by T^{l-1}, then Δ more indices corresponding to the largest magnitude entries of the *residue* (a measure of error between the observed $\mathbf{y}$ and the inferred $\hat{\mathbf{y}}$ based on T^{l-1}) are added to T^{l-1} to create a new index set $\tilde{T}^l$ of size $K + \Delta$.

S2. *Contraction*: Projecting the observation $\mathbf{y}$ onto the set $\tilde{T}^l$ gives a new vector $\mathbf{x}_p$. Indices of the K largest elements of $\mathbf{x}_p$ yield a *revised estimate* T^l of $\mathrm{supp}(\mathbf{x})$.

An obvious drawback of SP is that there is no quantification of the overall reconstruction quality of the support sets $\tilde{T}^l$ or T^l. In particular, there is no guarantee that the residual error due to T^l is lower than that due to T^{l-1}. We propose to improve the SP algorithm [10] by modifying these two steps, as described

Manuscript received August 30, 2010; revised October 17, 2010; accepted October 25, 2010. Date of publication November 18, 2010; date of current version November 29, 2010. This work was supported in part by the National Science Foundation under Grants 0941362 (OIA/CDI) and 0931805 (CNS/CPS). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ce Zhu.

The authors are with the Department of Electrical & Computer Engineering, The Johns Hopkins University, Baltimore, MD 21210 USA (e-mail: blakris_v@yahoo.com; khudanpur@jhu.edu).

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/LSP.2010.2090143

below, while still ensuring that the overall computational (big $\mathcal{O}$) complexity remains competitive with SP.

Given[1] an interim estimate $\mathcal{I} \subseteq \{1, \ldots, N\}$ of the support set of $\mathbf{x}$, let $S_{\mathcal{I}} = \min_{\tilde{\mathbf{x}} \in \mathbb{R}^I} \|\mathbf{y} - \mathbf{\Phi}_{\mathcal{I}} \tilde{\mathbf{x}}\|_2^2$ denote the residual error in $\mathbf{y}$, where $I = |\mathcal{I}|$ and $\mathbf{\Phi}_{\mathcal{I}}$ is the submatrix of the columns of $\mathbf{\Phi}$ indexed by $\mathcal{I}$.

We note[2] that if $\mathcal{J} = \mathcal{I} \cup \{i\}$ for some $i \notin \mathcal{I}$, then the exact reduction in residual error is

$$S_{\mathcal{I}} - S_{\mathcal{J}} = \frac{\left(\xi_{\{i\}} - \xi_{\mathcal{I}}^T \mathbf{\Phi}_{\mathcal{I}}^{\ddagger} \mathbf{\Psi}[\mathcal{I}, i]\right)^2}{\mathbf{\Psi}[i, i] - \mathbf{\Psi}[\mathcal{I}, i]^T \mathbf{\Phi}_{\mathcal{I}}^{\ddagger} \mathbf{\Psi}[\mathcal{I}, i]} \tag{3}$$

where $\xi_{\mathcal{I}} = \mathbf{\Phi}_{\mathcal{I}}^T \mathbf{y}$, $\mathbf{\Phi}_{\mathcal{I}}^{\ddagger} = \left(\mathbf{\Phi}_{\mathcal{I}}^T \mathbf{\Phi}_{\mathcal{I}}\right)^{-1}$ and $\mathbf{\Psi}[\mathcal{I}, i]$ is, in MATLAB-like notation, the submatrix of $\mathbf{\Psi} = \mathbf{\Phi}^T \mathbf{\Phi}$ comprised of the rows indexed by $\mathcal{I}$ and the i^{th} column.

Similarly, note that if $\mathcal{I} = \mathcal{J} \setminus \{j\}$ for some $j \in \mathcal{J}$, then

$$S_{\mathcal{I}} - S_{\mathcal{J}} = \frac{\left(\xi_{\mathcal{J}}^T \mathbf{\Phi}_{\mathcal{J}}^{\ddagger}[:, j]\right)^2}{\mathbf{\Phi}_{\mathcal{J}}^{\ddagger}[j, j]} \tag{4}$$

where $\mathbf{\Phi}_{\mathcal{J}}^{\ddagger}[:, j]$ is the j^{th} column of $\mathbf{\Phi}_{\mathcal{J}}^{\ddagger}$.

Using these two identities, we propose the following replacements for the *Expansion* and *Contraction* steps of SP.

S ′ 1. *GREEDY-ADD*: Given the current estimate T^{l-1} of the support set, set $\mathcal{J} = T^{l-1}$ and add to it the index i that maximizes (3) to obtain a new set $\mathcal{I}_1$; set $\mathcal{J} = \mathcal{I}_1$ and repeat, adding one index at a time *according to* (3), to obtain $\mathcal{I}_2, \mathcal{I}_3, \ldots, \mathcal{I}_\Delta$; set $\tilde{T}^l = \mathcal{I}_\Delta$.

S ′ 2. *GREEDY-REMOVE*: Given $\tilde{T}^l$, set $\mathcal{I} = \tilde{T}^l$ and remove from it the index i that minimizes (4) to obtain $\mathcal{J}_1$; set $\mathcal{I} = \mathcal{J}_1$ and repeat, removing one index at a time *according to* (4), to obtain $\mathcal{J}_2, \mathcal{J}_3, \ldots, \mathcal{J}_\Delta$; set $T^l = \mathcal{J}_\Delta$.

The identities (3) and (4) guide SP by providing the exact residual error at each expansion/contraction step. More importantly, *they take into account any lack of orthogonality* among the columns of $\mathbf{\Phi}$ when adding/removing an index.

II. Stepwise Optimal Subspace Pursuit

Algorithm 1 describes in detail our GREEDY PURSUIT procedure for sparse signal recovery. It requires as input the sampling matrix $\mathbf{\Phi}$, the measurements $\mathbf{y}$ and the sparsity K of $\mathbf{x}$; the expansion/contraction step size Δ is an optional input that is set to K by default. The algorithm returns its estimate of $\operatorname{supp}(\mathbf{x})$, from which $\mathbf{x}$ may be easily computed.

Algorithm 2 (GREEDY ADD) describes stepwise optimal expansion using (3). It requires as input $\mathbf{y}$, $\mathbf{\Phi}$, a support set $\mathcal{J}$ and expansion size Δ and returns a support set $\mathcal{I} \supset \mathcal{J}$ of size $|\mathcal{J}| + \Delta$. Algorithm 3 (GREEDY REMOVE) describes stepwise optimal contraction using (4). It requires as input $\mathbf{y}$, $\mathbf{\Phi}$, a support set $\mathcal{I}$ and contraction size Δ and returns a support set $\mathcal{J} \subset \mathcal{I}$ of size $|\mathcal{I}| - \Delta$.

```
**Algorithm:** GREEDY-PURSUIT ($\mathbf{y}$, $\mathbf{\Phi}$, $K$, $\Delta := K$)
$T^0 := \text{GREEDY-ADD}(\mathbf{y}, \mathbf{\Phi}, \emptyset, K)$;
$S_{T^0} := \mathbf{y}^\mathbf{T}\mathbf{y} - \mathbf{y}^\mathbf{T}\mathbf{\Phi}_{\mathbf{T^0}}\left(\mathbf{\Phi}_{\mathbf{T^0}}^\mathbf{T}\mathbf{\Phi}_{\mathbf{T^0}}\right)^{-\mathbf{1}}\mathbf{\Phi}_{\mathbf{T^0}}^\mathbf{T}\mathbf{y}$;
$l := 0$;
while $\Delta > 0$ do
    $l := l + 1$;
    $\tilde{T}^l := \text{GREEDY-ADD}(\mathbf{y}, \mathbf{\Phi}, T^{l-1}, \Delta)$;
    $T^l := \text{GREEDY-REMOVE}(\mathbf{y}, \mathbf{\Phi}, \tilde{T}^l, \Delta)$;
    $S_{T^l} := \mathbf{y}^\mathbf{T}\mathbf{y} - \mathbf{y}^\mathbf{T}\mathbf{\Phi}_{\mathbf{T^l}}\left(\mathbf{\Phi}_{\mathbf{T^l}}^\mathbf{T}\mathbf{\Phi}_{\mathbf{T^l}}\right)^{-\mathbf{1}}\mathbf{\Phi}_{\mathbf{T^l}}^\mathbf{T}\mathbf{y}$;
    if $S_{T^l} \geq S_{T^{l-1}}$ then
        $T^l := T^{l-1}$;
        $S_{T^l} := S_{T^{l-1}}$;
        $\Delta := \Delta - 1$;
    end
end
return $T_l$
```

Algorithm 1: GREEDY-PURSUIT for sparse recovery.

```
**Algorithm:** GREEDY-ADD ($\mathbf{y}$, $\mathbf{\Phi}$, $\mathcal{J}$, $\Delta$)
$\xi := \mathbf{\Phi}^T\mathbf{y}$;
$\mathbf{\Psi} := \mathbf{\Phi}^\mathbf{T}\mathbf{\Phi}$;
$\chi[\mathcal{J}, :] := \left(\mathbf{\Phi}_\mathcal{J}^\mathbf{T}\mathbf{\Phi}_\mathcal{J}\right)^{-1}\mathbf{\Phi}_\mathcal{J}^\mathbf{T}\mathbf{\Phi}$;
$\mathcal{I} := \mathcal{J}$;
while $|\mathcal{I}| < |\mathcal{J}| + \Delta$ do
    $\hat{i} := \underset{i \in \{1,\ldots,N\} \setminus \mathcal{I}}{\operatorname{argmax}} \frac{\left(\xi_{\{i\}} - \xi_\mathcal{I}^T \chi[\mathcal{I}, i]\right)^2}{\mathbf{\Psi}[i,i] - \mathbf{\Psi}[\mathcal{I}, i]^T \chi[\mathcal{I}, i]}$;
    $\mathbf{w} := \chi[\mathcal{I}, \hat{i}]^T {\mathbf{\Phi}_\mathcal{I}}^T$;
    $\mathbf{v} := \frac{\mathbf{w}\mathbf{\Phi} - \mathbf{\Psi}[\hat{i}, :]}{\Psi[\hat{i}, \hat{i}] - \mathbf{\Psi}[\mathcal{J}, i]^T \chi[\mathcal{I}, \hat{i}]}$;
    $\chi[\mathcal{I}, :] := \chi[\mathcal{I}, :] + \chi[\mathcal{I}, \hat{i}]\mathbf{v}$;
    $\chi[\hat{i}, :] := -\mathbf{v}$;
    $\mathcal{I} := \mathcal{I} \cup \{\hat{i}\}$;
end
return $\mathcal{I}$
```

Algorithm 2: GREEDY-ADD adds Δ indices to $\mathcal{J}$.

```
**Algorithm:** GREEDY-REMOVE ($\mathbf{y}$, $\mathbf{\Phi}$, $\mathcal{I}$, $\Delta$)
$\xi := \mathbf{\Phi}^T\mathbf{y}$;
$\mathbf{\Phi}_\mathcal{I}^\ddagger := \left(\mathbf{\Phi}_\mathcal{I}^T\mathbf{\Phi}_\mathcal{I}\right)^{-1}$;
$\mathcal{J} := \mathcal{I}$;
while $|\mathcal{J}| > |\mathcal{I}| - \Delta$ do
    $\hat{j} := \underset{j \in \mathcal{J}}{\operatorname{argmin}} \frac{\left(\xi_\mathcal{J}^T \mathbf{\Phi}_\mathcal{J}^\ddagger[:, j]\right)^2}{\mathbf{\Phi}_\mathcal{J}^\ddagger[j, j]}$;
    $\mathcal{J} := \mathcal{J} \setminus \{\hat{j}\}$;
    $\mathbf{\Phi}_\mathcal{J}^\ddagger := \mathbf{\Phi}_\mathcal{J}^\ddagger - \frac{1}{\mathbf{\Phi}_\mathcal{J}^\ddagger[\hat{j}, \hat{j}]}\mathbf{\Phi}_\mathcal{J}^\ddagger[\mathcal{J}, \hat{j}]\mathbf{\Phi}_\mathcal{J}^\ddagger[\hat{j}, \mathcal{J}]$;
end
return $\mathcal{J}$
```

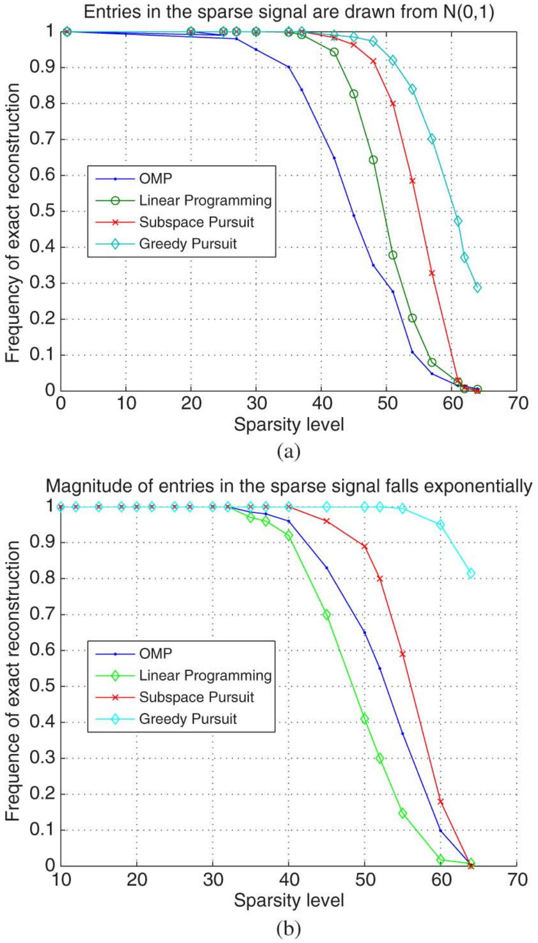
Algorithm 3: GREEDY-REMOVE removes Δ indices from $\mathcal{I}$.

GREEDY PURSUIT starts by calling GREEDY ADD with an empty support set to obtain an initial estimate[3] T^0 of size K . GREEDY ADD and GREEDY REMOVE then alternately expand this set by Δ , then shrink it back, to iteratively produce $T^1, T^2, \ldots$ so long as residual error is reducing. Note that when $\Delta = 1$, the expansion and contraction steps are both provably optimal by construction, and $S_{T^l} - S_{T^{l-1}} \leq 0$. When $\Delta > 1$, however, there is no such guarantee. Therefore, when no error reduction results at an iteration with $\Delta > 1$, we discard the update from T^{l-1} to T^l , and try again with a smaller value of Δ . This is another difference between GREEDY PURSUIT and SP. As an aside, smaller Δ s are also computationally more efficient, as will be shown later; this is useful when speed is of

[1] Assume that $\mathbf{\Phi}_{\mathcal{I}}$ has full column rank; this makes $\mathbf{\Phi}_{\mathcal{I}}^T \mathbf{\Phi}_{\mathcal{I}}$ invertible.

[2] (3) & (4) are obtained via block-wise matrix inversion formulae (cf. [12]).

[3] We note that in practice this estimate is already of very high quality.



(a)

(b)

Fig. 1. The fraction of times a 256-dimensional, K-sparse signal $(K \leq 64)$ is correctly recovered from 128 (Gaussian) random projections by different reconstruction algorithms. (a) Nonzero entries of $\mathbf{x}$ are distributed $\mathcal{N}(0,1)$. (b) Magnitudes of nonzero entries of $\mathbf{x}$ decay exponentially.

the essence. GREEDY PURSUIT terminates when there is no improvement even with $\Delta = 1$.

A. *Empirical Comparison of Reconstruction Accuracy*

We compare the empirical performance of the linear programming (LP), OMP, SP and GREEDY PURSUIT solutions to the sparse recovery problem using the setup described in [10].

1) We generate a Gaussian $\mathcal{N}(0,1)$ random matrix $\mathbf{\Phi}$ of size $M \times N$. We use $M = 128$ and $N = 256$ respectively to be able to compare directly with [10].
2) We choose a K-subset of $\{1,\ldots,N\}$, $K = 1,\ldots,(M/2)$.
3) We set the value of $\mathbf{x}$ at the K chosen indices to
   (a) random nonzero values drawn from $\mathcal{N}(0,1)$; or
   (b) a random permutation of K exponentially decaying values from 1.0 to 0.1;

   we set the value of $\mathbf{x}$ at the remaining indices to 0.
4) We estimate $\operatorname{supp}(\mathbf{x})$ using each method, and check if it matches the subset of K chosen indices exactly.

We repeat the simulation 1000 times for each value of K and note the frequency of exact reconstruction for each method.

Fig. 1(a) shows that GREEDY PURSUIT performs better than LP, OMP and SP when the nonzero entries of the sparse signal are drawn according to $\mathcal{N}(0,1)$. As depicted in Fig. 1(b), GREEDY PURSUIT significantly outperforms existing methods for the exponential case.

Additional simulations show that GREEDY PURSUIT also outperforms SP and LP when the $\mathbf{y}$ are noisy (10 dB SNR). In reconstructing a *compressible* signal with $\operatorname{supp}(\mathbf{x}) = N$ from $K \ll N$ measurements, it yields 1-to-3 dB higher SNR than SP and OMP. (Details are omitted due to space limitations.)

The results for recovering a 0–1 sparse signal,[4], however, are negative: LP $>$ SP $>$ GREEDY PURSUIT . This is consistent with the finding in [10] that LP outperforms SP for 0–1 sparse signals, and merits further study.

III. Complexity Analysis of Greedy Pursuit

The following complexity analysis of GREEDY PURSUIT shows that for each iteration it requires essentially the same computation as SP, namely $\mathcal{O}(KMN)$.

A. *Computational Complexity of GREEDY REMOVE*

Computing $\mathbf{\Phi}_{\mathcal{I}}^{\ddagger}$ in line 2 of Algorithm 3 requires inverting a $|\mathcal{I}| \times |\mathcal{I}|$ matrix which is $\mathcal{O}(|\mathcal{I}|^3)$. The minimum over j in line 5 requires computing $|\mathcal{J}|$ dot-products of $|\mathcal{J}|$-dimensions each, which is $\mathcal{O}(|\mathcal{J}^2|)$. Updating $\mathbf{\Phi}_{\mathcal{J}}^{\ddagger}$ is also $\mathcal{O}(|\mathcal{J}|^2)$. Since $|\mathcal{J}| \leq |\mathcal{I}|$, the entire while-loop takes $\mathcal{O}(|\mathcal{I}|^3)$ time. Therefore GREEDY REMOVE is $\mathcal{O}(|\mathcal{I}|^3) = O(|K+\Delta|^3)$.

1) A Generalized GREEDY REMOVE Operation: In (4) and on line 7 of Algorithm 3, we compute the increase in residual error upon removing a single index $\hat{j} \in \mathcal{J}$. We next provide a general expression for the increase in error upon removing a *set* of indices $\hat{\mathcal{J}} \subseteq \mathcal{J}$. Let $\mathcal{I} = \mathcal{J} \setminus \hat{\mathcal{J}}$ denote the retained indices, $\boldsymbol{\mathcal{A}} = \mathbf{\Phi}_{\mathcal{J}}^{\ddagger}[\mathcal{I},\mathcal{I}]$, $\mathcal{B} = \mathbf{\Phi}_{\mathcal{J}}^{\ddagger}[\mathcal{I},\hat{\mathcal{J}}]$ and $\mathcal{D} = \mathbf{\Phi}_{\mathcal{J}}^{\ddagger}[\hat{\mathcal{J}},\hat{\mathcal{J}}]$. Then it is easy to establish that

$$S_{\mathcal{I}} - S_{\mathcal{J}} = \left(\xi_{\mathcal{I}}^T \mathcal{B} + \xi_{\hat{\mathcal{J}}}^T \mathcal{D}\right) \mathcal{D}^{-1} \left(\xi_{\mathcal{I}}^T \mathcal{B} + \xi_{\hat{\mathcal{J}}}^T \mathcal{D}\right)^T. \qquad (5)$$

A generalized GREEDY REMOVE procedure may therefore be formulated with an additional parameter $\delta \geq 1$. Each time we enter the while-loop in Algorithm 3, we use (5) to compute the increase in squared error for every size-δ subset $\hat{\mathcal{J}}$ of $\mathcal{J}$, and remove the minimizer. There are $\mathcal{O}(|\mathcal{J}|^{\delta})$ size-δ subsets of $\mathcal{J}$. For each $\hat{\mathcal{J}}$, the quantity in (5) requires computing the vector $\xi_{\mathcal{I}}^T \mathcal{B} + \xi_{\hat{\mathcal{J}}}^T \mathcal{D}$ and the matrix $\mathcal{D}^{-1}$, which is $\mathcal{O}(\delta|\mathcal{I}| + \delta^3)$. Finally, the while-loop is executed $\lceil \Delta/\delta \rceil$ times. Thus the overall complexity of this generalized GREEDY REMOVE procedure is $\mathcal{O}(|K+\Delta|^{\delta}(K+\Delta+\delta^2)\Delta)$.

B. *Computational Complexity of GREEDY ADD*

Since $\mathbf{\Psi}$ may be pre-computed outside GREEDY ADD, initializations up to line 4 of Algorithm 2 are $\mathcal{O}(|\mathcal{J}|^3)$. χ is initially is a $|\mathcal{J}| \times N$ matrix and ξ is a $M \times 1$ vector. Therefore $\xi_{\{i\}} - \xi_{\mathcal{I}}^T \chi[\mathcal{I},i]$ is a scalar that requires $\mathcal{O}(|\mathcal{I}|)$ time to compute. Similarly computing $\mathbf{\Psi}[i,i] - \mathbf{\Psi}[\mathcal{I},i]^T \chi[\mathcal{I},i]$ requires $O(|\mathcal{I}|)$ time. Hence maximizing over all $i \in \{1,\ldots,N\} \setminus \mathcal{I}$ requires $\mathcal{O}(N|\mathcal{I}|)$ time. Computing $\chi[\mathcal{I},\hat{i}]^T \mathbf{\Phi}_{\mathcal{I}}^T$ requires $\mathcal{O}(N|\mathcal{I}|)$ time. The computational bottleneck is the vector $\mathbf{v}$ which requires computing $\mathbf{w\Phi}$. This takes $\mathcal{O}(MN)$ time. Hence the complexity of each iteration of the while-loop is $\mathcal{O}(MN)$ and the net complexity of the algorithm is

[4] The value of $\mathbf{x}$ at the indices chosen in Step 2 is set to 1.0 in Step 3.

$\mathcal{O}(|\mathcal{J}|^3) + MN\Delta)$. Since $|\mathcal{J}| < M$, $\Delta = \mathcal{O}(K)$ and $N \gg M$, the complexity is essentially $\mathcal{O}(MNK)$.

It is possible to generalize (3) for adding more than one new index i , just as (5) generalizes (4). One could then use it to replace line 6 of Algorithm 2, maximizing the reduction in residual error over all size- δ subsets $\hat{\mathcal{I}} \subseteq \{1, \ldots, N\} \setminus \mathcal{I}$. We omit this generalization due to space limitations.

IV. Discussion and Conclusion

We have presented a new technique for recovering sparse signals from linear measurements. Although the original problem (1) is NP-hard, the technique performs an accurate, locally optimal update to a working solution, amounting to "gradient ascent" in a discrete search space. In particular, the one-step update accounts for any lack of orthogonality in the linear measurements. Hence the final solution is locally optimal. We also have outlined ways to generalize the technique to optimally add or remove $\delta \geq 1$ indices at a time. Finally, we have shown that our technique performs well for Gaussian and exponentially sparse signals (cf. Fig. 1).

References

[1] 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, 2007.

[2] Y. Tsaig and D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, pp. 1289–1306, 2006.

[3] R. Venkataramani and Y. Bresler, "Perfect reconstruction formulas and bounds on aliasing error in sub-nyquist nonuniform sampling of multiband signals," *IEEE Trans. Inf. Theory*, vol. 46, pp. 2173–2183, 2000.

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

[5] E. J. Candes 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.

[6] 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, no. 4, pp. 586–597, 2008.

[7] D. Needell and R. Vershynin, "Uniform uncertainty principle and signal recovery viaregularized orthogonal matching pursuit," *Found. Comput. Math.*, vol. 9, no. 3, pp. 317–334, 2009.

[8] D. L. Donoho, I. Drori, Y. Tsaig, and J. L. Starck, Sparse Solution of Underdetermined Linear Equations by Stagewise Orthogonal Matching Pursuit Citeseer, 2006.

[9] D. Needell and R. Vershynin, Signal Recovery From Incomplete and Inaccurate Measurements via Regularized Orthogonal Matching Pursuit *ArXiv e-prints*, Dec. 2007.

[10] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing: Closing the gap between performance and complexity," *CoRR*, vol. abs/0803.0811, 2008.

[11] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Appl. Comput. Harmon. Anal.*, vol. 26, no. 3, pp. 301–321, 2009.

[12] Wikipedia, Invertible Matrix 2004.