CoSaMP with Redundant Dictionaries

Mark A. Davenport
School of Electrical
and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332

Deanna Needell
Department of Mathematics
and Computer Science
Claremont McKenna College
Claremont, CA 91711

Michael B. Wakin
Department of Electrical Engineering
and Computer Science
Colorado School of Mines
Golden, CO 80401

***Abstract*—In this paper we describe a variant of the iterative reconstruction algorithm CoSaMP for the setting where the signal is not sparse in an orthonormal basis but in a truly redundant or overcomplete dictionary. We utilize the $\boldsymbol{D}$-RIP, a condition on the sensing matrix analogous to the well-known restricted isometry property. In contrast to prior work, the method and analysis are "signal-focused"; that is, they are oriented around recovering the *signal* rather than its dictionary *coefficients*. Under the assumption that we have a near-optimal scheme for projecting vectors in signal space onto the model family of candidate sparse signals, we provide provable recovery guarantees. We also provide a discussion of practical examples and empirical results.**

I. Introduction

Compressive sensing (CS) is a powerful new framework for signal acquisition [1], offering the promise that we can acquire a vector $\boldsymbol{x} \in \mathbb{C}^n$ via only $m \ll n$ linear measurements provided that $\boldsymbol{x}$ is *sparse* or *compressible*.[1] Specifically, CS considers the problem where we obtain measurements of the form $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e}$, where $\boldsymbol{A}$ is an $m \times n$ sensing matrix and $\boldsymbol{e}$ is a noise vector. If $\boldsymbol{x}$ is sparse or compressible and $\boldsymbol{A}$ satisfies certain conditions, then CS provides a mechanism to efficiently recover the signal $\boldsymbol{x}$ from the measurement vector $\boldsymbol{y}$.

Typically, however, signals of practical interest are not themselves sparse, but rather have a sparse expansion in some dictionary $\boldsymbol{D}$. By this we mean that there exists a sparse coefficient vector $\boldsymbol{\alpha}$ such that the signal $\boldsymbol{x}$ can be expressed as $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\alpha}$. One could then ask the simple question: How can we account for this signal model in CS? In some cases, there is a natural way to extend the standard CS formulation—since we can write the measurements as $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{D}\boldsymbol{\alpha} + \boldsymbol{e}$ we can use standard CS techniques to first obtain an estimate $\widehat{\boldsymbol{\alpha}}$ of the sparse coefficient vector. We can then synthesize an estimate $\widehat{\boldsymbol{x}} = \boldsymbol{D}\widehat{\boldsymbol{\alpha}}$ of the original signal. Unfortunately, this is a rather restrictive approach for two main reasons: (i) applying standard CS results to this problem will require that the product $\boldsymbol{A}\boldsymbol{D}$ satisfies certain properties, such as the *restricted isometry property* (RIP), that will not be satisfied for many interesting choices of $\boldsymbol{D}$, and (ii) we are not ultimately interested in recovering $\boldsymbol{\alpha}$ *per se*, but rather in obtaining an accurate estimate of $\boldsymbol{x}$. The distinction is that redundancy in $\boldsymbol{D}$ implies that, in general, the representation of a vector $\boldsymbol{x}$ in the dictionary is not unique—there may exist many possible coefficient vectors $\boldsymbol{\alpha}$ that can be used to synthesize $\boldsymbol{x}$. Moreover, the dictionary $\boldsymbol{D}$ may be poorly conditioned, and hence the signal space recovery error $\|\boldsymbol{x} - \widehat{\boldsymbol{x}}\|_2$ could be significantly smaller or larger than the coefficient space recovery error $\|\boldsymbol{\alpha} - \widehat{\boldsymbol{\alpha}}\|_2$. Thus, it may be possible to recover $\boldsymbol{x}$ in situations where recovering $\boldsymbol{\alpha}$ is impossible, and even if we could apply standard CS results to ensure that our estimate of $\boldsymbol{\alpha}$ is accurate, this would not necessarily translate into a recovery guarantee for $\boldsymbol{x}$.

All of these challenges essentially stem from the fact that extending standard CS algorithms in an attempt to recover $\boldsymbol{\alpha}$ is a *coefficient-focused* recovery strategy. By trying to go from the measurements $\boldsymbol{y}$ all the way back to the coefficient vector $\boldsymbol{\alpha}$, we encounter all the problems above due to the lack of orthogonality of the dictionary. In contrast, in this paper we propose a *signal-focused* recovery strategy for CS for which we can provide guarantees on the recovery of $\boldsymbol{x}$ while making no direct assumptions concerning our choice of $\boldsymbol{D}$. Our algorithm employs the model of sparsity in $\boldsymbol{D}$ but directly obtains an estimate of the signal $\boldsymbol{x}$, and we provide guarantees on the quality of this estimate in signal space. Our bounds require only that $\boldsymbol{A}$ satisfy the $\boldsymbol{D}$-RIP [2]—a less-restrictive condition than requiring $\boldsymbol{A}\boldsymbol{D}$ to satisfy the RIP. Our algorithm is a modification of CoSaMP [3], and in cases where $\boldsymbol{D}$ is unitary, our "Signal-Space CoSaMP" algorithm reduces to standard CoSaMP.

Our work most closely relates to Blumensath's Projected Landweber Algorithm (PLA) [4], an extension of Iterative Hard Thresholding (IHT) [5] that operates in signal space and accounts for a union-of-subspaces signal model. In several ways, our work is a parallel of this one, except that we extend CoSaMP rather than IHT to operate in signal space. Both works assume that $\boldsymbol{A}$

This work was partially supported by NSF grants DMS-1004718 and CCF-0830320, NSF CAREER grant CCF-1149225, and AFOSR grant FA9550-09-1-0465.

[1] When we say that a vector $\boldsymbol{z}$ is k-sparse, we mean that $\|\boldsymbol{z}\|_0 \stackrel{\mathrm{def}}{=} |\mathrm{supp}(\boldsymbol{z})| \leq k \ll n$. A compressible vector is one that is well-approximated as being sparse.

satisfies the $\boldsymbol{D}$-RIP, and implementing both algorithms requires the ability to compute projections of vectors in the signal space onto the model family. One difference, however, is that our analysis allows for near-optimal projections whereas the PLA analysis does not.

II. Signal Space CoSaMP

To describe our algorithm, originally proposed in [6], we begin by establishing some notation. Suppose that $\boldsymbol{A} \in \mathbb{C}^{m\times n}$ and $\boldsymbol{D} \in \mathbb{C}^{n\times d}$ are given and that we observe measurements of the form $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e} = \boldsymbol{A}\boldsymbol{D}\boldsymbol{\alpha} + \boldsymbol{e}$. For an index set $\Lambda \in \{1, 2, \ldots, d\}$ (sometimes referred to as a *support set*), we let $\boldsymbol{D}_\Lambda$ denote the $n \times |\Lambda|$ submatrix of $\boldsymbol{D}$ corresponding to the columns indexed by Λ, and we let $\mathcal{R}(\boldsymbol{D}_\Lambda)$ denote the column span of $\boldsymbol{D}_\Lambda$. We also use $\mathcal{P}_\Lambda : \mathbb{C}^n \to \mathbb{C}^n$ to denote the orthogonal projection operator onto $\mathcal{R}(\boldsymbol{D}_\Lambda)$ and $\mathcal{P}_{\Lambda^\perp} : \mathbb{C}^n \to \mathbb{C}^n$ to denote the orthogonal projection operator onto the orthogonal complement of $\mathcal{R}(\boldsymbol{D}_\Lambda)$.[2]

Next, recall that one of the key steps in the traditional CoSaMP algorithm is to project a vector in signal space onto the set of candidate sparse signals. In the traditional setting (when $\boldsymbol{D}$ is an orthonormal basis), this step can be performed by simple thresholding of the entries of the coefficient vector. Our Signal Space version of CoSaMP, described in Algorithm 1, involves replacing thresholding with a more general projection of vectors in the signal space onto the signal model. Specifically, for a given vector $\boldsymbol{z} \in \mathbb{C}^n$ and a given sparsity level k, define

$$\Lambda_k(\boldsymbol{z}) := \underset{\Lambda:|\Lambda|=k}{\arg\min} \|\boldsymbol{z} - \mathcal{P}_\Lambda \boldsymbol{z}\|_2 .$$

The support $\Lambda_k(\boldsymbol{z})$—if we could compute it—could be used to generate the best k-sparse approximation to $\boldsymbol{z}$; in particular, the nearest neighbor to $\boldsymbol{z}$ among all signals that can be synthesized using k columns from $\boldsymbol{D}$ is given by $\mathcal{P}_{\Lambda_k(\boldsymbol{z})}\boldsymbol{z}$. Unfortunately, computing $\Lambda_k(\boldsymbol{z})$ may be difficult in general. Therefore, we allow near-optimal projections to be used in our algorithm. For a given vector $\boldsymbol{z} \in \mathbb{C}^n$ and a given sparsity level k, we assume a method is available for producing an estimate of $\Lambda_k(\boldsymbol{z})$, denoted $\mathcal{S}_k(\boldsymbol{z})$, such that $|\mathcal{S}_k(\boldsymbol{z})| = k$ and that there exist constants $\epsilon_1, \epsilon_2 \ge 0$ such that

$$\left\|\mathcal{P}_{\Lambda_k(\boldsymbol{z})}\boldsymbol{z} - \mathcal{P}_{\mathcal{S}_k(\boldsymbol{z})}\boldsymbol{z}\right\|_2 \le \epsilon_1 \left\|\mathcal{P}_{\Lambda_k(\boldsymbol{z})}\boldsymbol{z}\right\|_2 \tag{1}$$

and

$$\left\|\mathcal{P}_{\Lambda_k(\boldsymbol{z})}\boldsymbol{z} - \mathcal{P}_{\mathcal{S}_k(\boldsymbol{z})}\boldsymbol{z}\right\|_2 \le \epsilon_2 \left\|\boldsymbol{z} - \mathcal{P}_{\Lambda_k(\boldsymbol{z})}\boldsymbol{z}\right\|_2 . \tag{2}$$

Setting either ϵ_1 or ϵ_2 equal to 0 would lead to requiring that $\mathcal{P}_{\Lambda_k(\boldsymbol{z})}\boldsymbol{z} = \mathcal{P}_{\mathcal{S}_k(\boldsymbol{z})}\boldsymbol{z}$ exactly. Note that our metric for judging the quality of an approximation to $\Lambda_k(\boldsymbol{z})$ is entirely in terms of its impact in signal space. It might well be the case that $\mathcal{S}_k(\boldsymbol{z})$ could satisfy (1) and (2) while being substantially different (or even disjoint) from $\Lambda_k(\boldsymbol{z})$. It is important to note, however, that computing an optimal or even near-optimal support estimate satisfying (1) and (2) remains challenging in general.

[2]Note that $\mathcal{P}_{\Lambda^\perp}$ does *not* represent the orthogonal projection operator onto $\mathcal{R}(\boldsymbol{D}_{\{1,2,\ldots,d\}\setminus\Lambda})$.

Algorithm 1 Signal Space CoSaMP [6]

input: $\boldsymbol{A}$, $\boldsymbol{D}$, $\boldsymbol{y}$, k, stopping criterion
initialize: $\boldsymbol{x}^0 = \boldsymbol{0}$, $\ell = 0$, $\Gamma = \emptyset$
while not converged **do**
  proxy: $\boldsymbol{h} = \boldsymbol{A}^*(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}^\ell)$
  identify: $\Omega = \mathcal{S}_{2k}(\boldsymbol{h})$
  merge: $T = \Omega \cup \Gamma$
  update: $\widetilde{\boldsymbol{x}} = \arg\min_{\boldsymbol{z}\in\mathcal{R}(\boldsymbol{D}_T)} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{z}\|_2$
    $\Gamma = \mathcal{S}_k(\widetilde{\boldsymbol{x}})$
    $\boldsymbol{x}^{\ell+1} = \mathcal{P}_\Gamma \widetilde{\boldsymbol{x}}$
    $\ell = \ell + 1$
end while
output: $\widehat{\boldsymbol{x}} = \boldsymbol{x}^\ell$

III. Recovery Guarantees

We now provide a simple guarantee on the performance of our algorithm in the special case where $\boldsymbol{x}$ is k-sparse and $\boldsymbol{e} = \boldsymbol{0}$. For a more thorough analysis of the general noisy/compressible setting, we refer the reader to [6]. We will approach our analysis under the assumption that the matrix $\boldsymbol{A}$ satisfies the $\boldsymbol{D}$-RIP [2]. Specifically, we say that $\boldsymbol{A}$ satisfies the $\boldsymbol{D}$-RIP of order k if there exists a constant $\delta_k \in (0, 1)$ such that

$$\sqrt{1-\delta_k} \le \frac{\|\boldsymbol{A}\boldsymbol{D}\boldsymbol{\alpha}\|_2}{\|\boldsymbol{D}\boldsymbol{\alpha}\|_2} \le \sqrt{1+\delta_k}$$

holds for all $\boldsymbol{\alpha}$ satisfying $\|\boldsymbol{\alpha}\|_0 \le k$. We note that for any choice of $\boldsymbol{D}$, if $\boldsymbol{A}$ is populated with independent and identically distributed (i.i.d.) random entries from a Gaussian or subgaussian distribution, then with high probability, $\boldsymbol{A}$ will satisfy the $\boldsymbol{D}$-RIP of order k as long as $m = O(k \log(d/k))$ [7, Corollary 3.1].

Supposing that $\boldsymbol{A}$ satisfies the $\boldsymbol{D}$-RIP, then for signals having a sparse representation in the dictionary $\boldsymbol{D}$, we have the following guarantee.

Theorem III.1. *Suppose there exists a k-sparse coefficient vector $\boldsymbol{\alpha}$ such that $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\alpha}$, and suppose that $\boldsymbol{A}$ satisfies the $\boldsymbol{D}$-RIP of order $4k$. If we observe $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$, then the signal estimate $\boldsymbol{x}^{\ell+1}$ obtained after $\ell + 1$ iterations of Signal Space CoSaMP satisfies*

$$\left\|\boldsymbol{x} - \boldsymbol{x}^{\ell+1}\right\|_2 \le C \left\|\boldsymbol{x} - \boldsymbol{x}^\ell\right\|_2 , \tag{3}$$

where

$$C = ((2+\epsilon_1)\delta_{4k} + \epsilon_1)(2+\epsilon_2)\sqrt{\frac{1+\delta_{4k}}{1-\delta_{4k}}}.$$

Our proof of Theorem III.1 appears in the Appendix and is a modification of the original CoSaMP proof [3]. Through various combinations of ϵ_1, ϵ_2, and δ_{4k}, it is possible to ensure that $C < 1$ and thus that the accuracy of Signal Space CoSaMP improves at each iteration. Taking $\epsilon_1 = \frac{1}{10}$, $\epsilon_2 = 1$, and $\delta_{4k} = 0.029$ as an example, we obtain $C \leq 0.5$. Applying the relation (3) recursively, we can then conclude that

$$\left\| \boldsymbol{x} - \boldsymbol{x}^{\ell} \right\|_2 \leq 2^{-\ell} \left\| \boldsymbol{x} \right\|_2 . \tag{4}$$

Thus, by taking a sufficient number of iterations ℓ, the right hand side of (4) can be made arbitrarily small. For sparse signal recovery, this result is fully in line with state-of-the-art bounds for traditional CS algorithms such as CoSaMP [3], except that it can be applied in settings where the dictionary $\boldsymbol{D}$ is not unitary.

IV. Simulations

The main challenge in implementing our algorithm is in computing $\mathcal{S}_k(\boldsymbol{z})$. Although our theoretical analysis can accommodate near-optimal support estimates $\mathcal{S}_k(\boldsymbol{z})$ that satisfy (1) and (2), computing even near-optimal supports can be a challenging task for many dictionaries of practical interest. In this section, we present simulation results using practical (but heuristic) methods for attempting to find near-optimal supports $\mathcal{S}_k(\boldsymbol{z})$. We see that the resulting algorithms—even though they are not quite covered by our theory—can nevertheless outperform classical CS reconstruction techniques.

In our simulations, we let $\boldsymbol{D}$ be a 256×1024 overcomplete DFT dictionary. In this dictionary, neighboring columns are highly coherent, while distant columns are not. We then construct a length-d coefficient vector $\boldsymbol{\alpha}$ with $k = 8$ nonzero entries chosen as i.i.d. Gaussian random variables. We set $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\alpha}$, construct $\boldsymbol{A}$ with i.i.d. Gaussian entries, and collect noiseless measurements $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$. After reconstructing an estimate of $\boldsymbol{x}$, we declare this recovery to be perfect if the SNR of the recovered signal estimate is above 100 dB. We consider two scenarios: one in which the nonzeros of $\boldsymbol{\alpha}$ are randomly positioned but well-separated (with a minimum spacing of 8 zeros in between any pair of nonzeros), and one in which the nonzeros cluster together in a single, randomly-positioned block. Because of the nature of the columns in $\boldsymbol{D}$, we see that many recovery algorithms perform differently in these two scenarios. See [6] for additional simulations and details.[3]

1) Well-separated coefficients: Figure 1(a) plots the performance of six different recovery algorithms for the scenario where the nonzero entries of $\boldsymbol{\alpha}$ are well-separated. Two of these algorithms are the traditional OMP and CoSaMP algorithms from CS, each using the combined dictionary $\boldsymbol{A}\boldsymbol{D}$ to first recover $\boldsymbol{\alpha}$. We actually see that OMP performs substantially better than CoSaMP in this scenario, apparently because it can select one coefficient at a time and is less affected by the coherence of $\boldsymbol{D}$. It is somewhat remarkable that OMP succeeds at all, given that $\boldsymbol{A}\boldsymbol{D}$ will not satisfy the RIP and we are not aware of any existing theory that would guarantee the performance of OMP in this scenario.

We also show in Figure 1(a) two variants of Signal Space CoSaMP: one in which OMP is used for computing $\mathcal{S}_k(\boldsymbol{z})$ (labeled "SSCoSaMP (OMP)"), and one in which CoSaMP is used for computing $\mathcal{S}_k(\boldsymbol{z})$ (labeled "SSCoSaMP (CoSaMP)"). That is, these algorithms actually use OMP or CoSaMP as an inner loop inside of Signal Space CoSaMP to find a sparse solution to the equation $\boldsymbol{z} = \boldsymbol{D}\boldsymbol{\alpha}$. In this scenario, we see that the performance of SSCoSaMP (OMP) is substantially better than OMP, while the performance of SSCoSaMP (CoSaMP) is poor. We believe that this happens for the same reason that traditional OMP outperforms traditional CoSaMP. In general, we have found that when OMP performs well, SSCoSaMP (OMP) may perform even better, and when CoSaMP performs poorly, SSCoSaMP (CoSaMP) may still perform poorly.

Figure 1(a) also shows the performance of two algorithms that involve convex optimization for sparse regularization. One, labeled "ℓ_1," uses ℓ_1-minimization [8] to find a sparse vector $\boldsymbol{\alpha}'$ subject to the constraint that $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{D}\boldsymbol{\alpha}'$. This algorithm outperforms traditional OMP in this scenario. The other, labeled "SSCoSaMP (ℓ_1)," is a variant of Signal Space CoSaMP in which ℓ_1-minimization is used for computing $\mathcal{S}_k(\boldsymbol{z})$.[4] Specifically, to compute $\mathcal{S}_k(\boldsymbol{z})$, we find the vector $\boldsymbol{\alpha}'$ with the smallest ℓ_1 norm subject to $\boldsymbol{z} = \boldsymbol{D}\boldsymbol{\alpha}'$, and we then choose the support that contains the k largest entries of $\boldsymbol{\alpha}'$. Remarkably, this algorithm performs best of all. We believe that this is due to the fact that, for the overcomplete DFT dictionary, ℓ_1-minimization is capable of finding $\Lambda_k(\boldsymbol{z})$ exactly when $\boldsymbol{z} = \mathcal{P}_{\Lambda_k(\boldsymbol{z})}\boldsymbol{z}$ and the entries of $\Lambda_k(\boldsymbol{z})$ are sufficiently well-separated [9]. While we do not guarantee that this condition will be met within every iteration of Signal Space CoSaMP, the fact that the original coefficient vector $\boldsymbol{\alpha}$ has well-separated coefficients seems to be intimately related to the success of ℓ_1 and SSCoSaMP (ℓ_1) here.

2) Clustered coefficients: Finally, Figure 1(b) plots the performance of the same six recovery algorithms for the scenario where the nonzero entries of $\boldsymbol{\alpha}$ are clustered into a single block. Although one could of course employ a block-sparse recovery algorithm in this scenario, our

[3]All of our simulations were performed via a MATLAB software package that we have made available for download at http://users.ece.gatech.edu/~mdavenport/software.

[4]We are not unaware of the irony of using ℓ_1-minimization inside of a greedy algorithm.

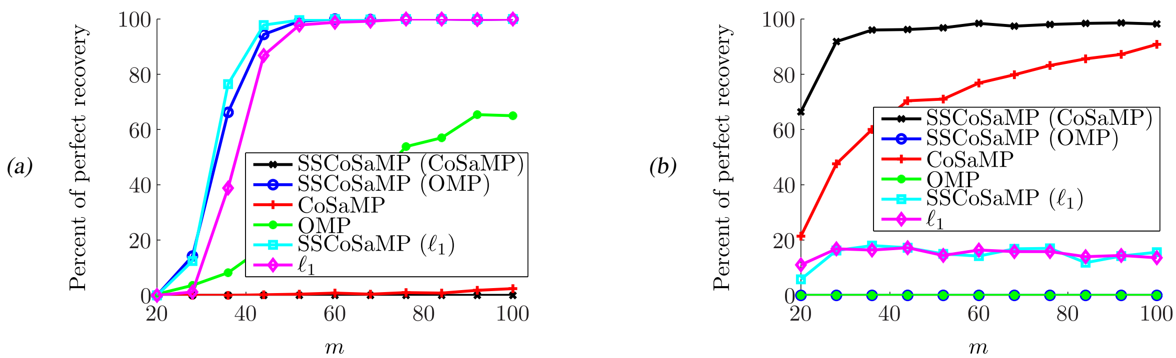

Fig. 1. *Performance in recovering signals having a $k = 8$ sparse representation in a $4\times$ overcomplete DFT dictionary. Two scenarios are shown: (a) one in which the $k = 8$ nonzero entries of $\boldsymbol{\alpha}$ are randomly positioned but well-separated, and (b) one in which the $k = 8$ nonzero entries all cluster together in a single, randomly-positioned block.*

intent is more to study the impact that neighboring active atoms have on the algorithms above.

In this scenario, between the traditional greedy algorithms, CoSaMP now outperforms OMP, apparently because it is designed to select multiple indices at each step and will not be as affected by the coherence of neighboring active columns in $\boldsymbol{D}$. We also see that the performance of SSCoSaMP (CoSaMP) is somewhat better than CoSaMP, while the performance of SSCoSaMP (OMP) is poor. We believe that this happens for the same reason that traditional CoSaMP outperforms traditional OMP. In general, we have found that when CoSaMP performs well, SSCoSaMP (CoSaMP) may perform even better, and when OMP performs poorly, SSCoSaMP (OMP) may still perform poorly.

In terms of our condition for perfect recovery (estimating $\boldsymbol{x}$ to within an SNR of 100 dB or more), neither of the algorithms that involve ℓ_1-minimization perform well in this scenario. However, we do note that both ℓ_1 and SSCoSaMP (ℓ_1) do frequently recover an estimate of $\boldsymbol{x}$ with an SNR of 50 dB or more, though still not quite as frequently as SSCoSaMP (CoSaMP) does.

Appendix

The proof of Theorem III.1 requires four main lemmas, which are listed below and proved in Sections A–D. In the lemmas below, $\boldsymbol{v} = \boldsymbol{x} - \boldsymbol{x}^\ell$ denotes the recovery error in signal space after ℓ iterations.

Lemma A.1. $\left\|\boldsymbol{x} - \boldsymbol{x}^{\ell+1}\right\|_2 \leq (2+\epsilon_2)\left\|\boldsymbol{x} - \widetilde{\boldsymbol{x}}\right\|_2 .$

Lemma A.2. $\left\|\boldsymbol{x} - \widetilde{\boldsymbol{x}}\right\|_2 \leq \sqrt{\frac{1+\delta_{4k}}{1-\delta_{4k}}}\left\|\mathcal{P}_{T^\perp}\boldsymbol{x}\right\|_2 .$

Lemma A.3. $\left\|\mathcal{P}_{T^\perp}\boldsymbol{x}\right\|_2 \leq \left\|\mathcal{P}_{\Omega^\perp}\boldsymbol{v}\right\|_2 .$

Lemma A.4. $\left\|\mathcal{P}_{\Omega^\perp}\boldsymbol{v}\right\|_2 \leq ((2+\epsilon_1)\delta_{4k} + \epsilon_1)\left\|\boldsymbol{v}\right\|_2 .$

Combining all four statements above, we have

$$
\begin{aligned}
\left\|\boldsymbol{x} - \boldsymbol{x}^{\ell+1}\right\|_2 &\leq (2+\epsilon_2)\left\|\boldsymbol{x} - \widetilde{\boldsymbol{x}}\right\|_2 \\
&\leq (2+\epsilon_2)\sqrt{\frac{1+\delta_{4k}}{1-\delta_{4k}}}\left\|\mathcal{P}_{T^\perp}\boldsymbol{x}\right\|_2 \\
&\leq (2+\epsilon_2)\sqrt{\frac{1+\delta_{4k}}{1-\delta_{4k}}}\left\|\mathcal{P}_{\Omega^\perp}\boldsymbol{v}\right\|_2 \\
&\leq C\left\|\boldsymbol{v}\right\|_2 = C\left\|\boldsymbol{x} - \boldsymbol{x}^\ell\right\|_2 .
\end{aligned}
$$

This completes the proof of Theorem III.1.

A. Proof of Lemma A.1

Using the triangle inequality, we have

$$
\left\|\boldsymbol{x} - \boldsymbol{x}^{\ell+1}\right\|_2 \leq \left\|\boldsymbol{x} - \widetilde{\boldsymbol{x}}\right\|_2 + \left\|\widetilde{\boldsymbol{x}} - \boldsymbol{x}^{\ell+1}\right\|_2 .
$$

Recall that $\Gamma = \mathcal{S}_k(\widetilde{\boldsymbol{x}})$ and $\boldsymbol{x}^{\ell+1} = \mathcal{P}_\Gamma\widetilde{\boldsymbol{x}}$. If we let $\Gamma^* = \Lambda_k(\widetilde{\boldsymbol{x}})$, then we can write

$$
\begin{aligned}
\left\|\widetilde{\boldsymbol{x}} - \boldsymbol{x}^{\ell+1}\right\|_2 &\leq \left\|\widetilde{\boldsymbol{x}} - \mathcal{P}_{\Gamma^*}\widetilde{\boldsymbol{x}}\right\|_2 + \left\|\mathcal{P}_{\Gamma^*}\widetilde{\boldsymbol{x}} - \mathcal{P}_\Gamma\widetilde{\boldsymbol{x}}\right\|_2 \\
&\leq \left\|\widetilde{\boldsymbol{x}} - \mathcal{P}_{\Gamma^*}\widetilde{\boldsymbol{x}}\right\|_2 + \epsilon_2\left\|\widetilde{\boldsymbol{x}} - \mathcal{P}_{\Gamma^*}\widetilde{\boldsymbol{x}}\right\|_2 ,
\end{aligned}
$$

where the first line follows from the triangle inequality, and the second line uses (2). Combining this, we have

$$
\begin{aligned}
\left\|\boldsymbol{x} - \boldsymbol{x}^{\ell+1}\right\|_2 &\leq \left\|\boldsymbol{x} - \widetilde{\boldsymbol{x}}\right\|_2 + (1+\epsilon_2)\left\|\widetilde{\boldsymbol{x}} - \mathcal{P}_{\Gamma^*}\widetilde{\boldsymbol{x}}\right\|_2 \\
&\leq \left\|\boldsymbol{x} - \widetilde{\boldsymbol{x}}\right\|_2 + (1+\epsilon_2)\left\|\widetilde{\boldsymbol{x}} - \boldsymbol{x}\right\|_2 \\
&= (2+\epsilon_2)\left\|\boldsymbol{x} - \widetilde{\boldsymbol{x}}\right\|_2 ,
\end{aligned}
$$

where the second line follows from the fact that $\mathcal{P}_{\Gamma^*}\widetilde{\boldsymbol{x}}$ is the nearest neighbor to $\widetilde{\boldsymbol{x}}$ among all vectors having a k-sparse representation in $\boldsymbol{D}$.

B. Proof of Lemma A.2

To begin, we note that $\boldsymbol{x} - \widetilde{\boldsymbol{x}}$ has a $4k$-sparse representation in $\boldsymbol{D}$, thus, applying the $\boldsymbol{D}$-RIP we have

$$
\left\|\boldsymbol{x} - \widetilde{\boldsymbol{x}}\right\|_2 \leq \frac{\left\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\widetilde{\boldsymbol{x}}\right\|_2}{\sqrt{1-\delta_{4k}}} .
$$

By construction,

$$
\left\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\widetilde{\boldsymbol{x}}\right\|_2 \leq \left\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\boldsymbol{z}\right\|_2
$$

for any $\boldsymbol{z} \in \mathcal{R}(\boldsymbol{D}_T)$, in particular for $\boldsymbol{z} = \mathcal{P}_T \boldsymbol{x}$. Thus,

$$\|\boldsymbol{x} - \widetilde{\boldsymbol{x}}\|_2 \leq \frac{\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\widetilde{\boldsymbol{x}}\|_2}{\sqrt{1-\delta_{4k}}} \leq \frac{\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\mathcal{P}_T\boldsymbol{x}\|_2}{\sqrt{1-\delta_{4k}}}. \quad (5)$$

By applying the $\boldsymbol{D}$-RIP we obtain

$$\begin{aligned} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\mathcal{P}_T\boldsymbol{x}\|_2 &\leq \sqrt{1+\delta_{4k}}\, \|\boldsymbol{x} - \mathcal{P}_T\boldsymbol{x}\|_2 \\ &= \sqrt{1+\delta_{4k}}\, \|\mathcal{P}_{T^\perp}\boldsymbol{x}\|_2 . \end{aligned} \quad (6)$$

Combining (5) and (6) establishes the lemma.

C. Proof of Lemma A.3

First note that by the definition of T, $\boldsymbol{x}^\ell \in \mathcal{R}(\boldsymbol{D}_T)$, and hence $\mathcal{P}_{T^\perp}\boldsymbol{x}^\ell = \boldsymbol{0}$. Thus we can write,

$$\|\mathcal{P}_{T^\perp}\boldsymbol{x}\|_2 = \left\|\mathcal{P}_{T^\perp}(\boldsymbol{x} - \boldsymbol{x}^\ell)\right\|_2 = \|\mathcal{P}_{T^\perp}\boldsymbol{v}\|_2 .$$

Finally, since $\Omega \subseteq T$, we have that

$$\|\mathcal{P}_{T^\perp}\boldsymbol{v}\|_2 \leq \|\mathcal{P}_{\Omega^\perp}\boldsymbol{v}\|_2 .$$

D. Proof of Lemma A.4

In order to prove the final lemma, we require two supplemental lemmas:

Lemma A.5 (Consequence of $\boldsymbol{D}$-RIP [6])**.** *For any index set B and any vector $\boldsymbol{z} \in \mathbb{C}^n$,*

$$\|\mathcal{P}_B \boldsymbol{A}^*\boldsymbol{A}\mathcal{P}_B\boldsymbol{z} - \mathcal{P}_B\boldsymbol{z}\|_2 \leq \delta_{|B|}\, \|\boldsymbol{z}\|_2 .$$

Lemma A.6 (Nested projections)**.** *For any pair of index sets A, B with $A \subset B$, $\mathcal{P}_A = \mathcal{P}_A\mathcal{P}_B$.*

Now, to make the notation simpler, let $\widetilde{\boldsymbol{v}} = \boldsymbol{A}^*\boldsymbol{A}\boldsymbol{v}$ and note that $\Omega = \mathcal{S}_{2k}(\widetilde{\boldsymbol{v}})$. Let $\Omega^* = \Lambda_{2k}(\widetilde{\boldsymbol{v}})$ and set $R = \mathcal{S}_{2k}(\boldsymbol{v})$. Using this notation we have

$$\begin{aligned} \|\mathcal{P}_{\Omega^\perp}\boldsymbol{v}\|_2 &= \|\boldsymbol{v} - \mathcal{P}_\Omega\boldsymbol{v}\|_2 \\ &\leq \|\boldsymbol{v} - \mathcal{P}_\Omega\widetilde{\boldsymbol{v}}\|_2 \\ &\leq \|\boldsymbol{v} - \mathcal{P}_{R\cup\Omega^*}\widetilde{\boldsymbol{v}}\|_2 + \|\mathcal{P}_{R\cup\Omega^*}\widetilde{\boldsymbol{v}} - \mathcal{P}_{\Omega^*}\widetilde{\boldsymbol{v}}\|_2 \\ &\qquad + \|\mathcal{P}_{\Omega^*}\widetilde{\boldsymbol{v}} - \mathcal{P}_\Omega\widetilde{\boldsymbol{v}}\|_2 , \end{aligned} \quad (7)$$

where the second line follows from the fact that $\mathcal{P}_\Omega\boldsymbol{v}$ is the nearest neighbor to $\boldsymbol{v}$ among all vectors in $\mathcal{R}(\boldsymbol{D}_\Omega)$ and the third line uses the triangle inequality.

Below, we provide a bound on the first term in (7). To deal with the second term in (7), note that for any Π which is a subset of $R \cup \Omega^*$, we can write

$$\widetilde{\boldsymbol{v}} - \mathcal{P}_\Pi\widetilde{\boldsymbol{v}} = (\widetilde{\boldsymbol{v}} - \mathcal{P}_{R\cup\Omega^*}\widetilde{\boldsymbol{v}}) + (\mathcal{P}_{R\cup\Omega^*}\widetilde{\boldsymbol{v}} - \mathcal{P}_\Pi\widetilde{\boldsymbol{v}}),$$

where $\widetilde{\boldsymbol{v}} - \mathcal{P}_{R\cup\Omega^*}\widetilde{\boldsymbol{v}}$ is orthogonal to $\mathcal{R}(\boldsymbol{D}_{R\cup\Omega^*})$, and $\mathcal{P}_{R\cup\Omega^*}\widetilde{\boldsymbol{v}} - \mathcal{P}_\Pi\widetilde{\boldsymbol{v}} \in \mathcal{R}(\boldsymbol{D}_{R\cup\Omega^*})$. Thus we can write

$$\|\widetilde{\boldsymbol{v}} - \mathcal{P}_\Pi\widetilde{\boldsymbol{v}}\|_2^2 = \|\widetilde{\boldsymbol{v}} - \mathcal{P}_{R\cup\Omega^*}\widetilde{\boldsymbol{v}}\|_2^2 + \|\mathcal{P}_{R\cup\Omega^*}\widetilde{\boldsymbol{v}} - \mathcal{P}_\Pi\widetilde{\boldsymbol{v}}\|_2^2 .$$

Recall that over all index sets Π with $|\Pi| = 2k$, $\|\widetilde{\boldsymbol{v}} - \mathcal{P}_\Pi\widetilde{\boldsymbol{v}}\|_2$ is minimized by choosing $\Pi = \Omega^*$. Thus, over all Π which are subsets of $R \cup \Omega^*$ with $|\Pi| = 2k$, $\|\mathcal{P}_{R\cup\Omega^*}\widetilde{\boldsymbol{v}} - \mathcal{P}_\Pi\widetilde{\boldsymbol{v}}\|_2^2$ must be minimized by choosing $\Pi = \Omega^*$. In particular, we have the first inequality below:

$$\begin{aligned} \|\mathcal{P}_{R\cup\Omega^*}\widetilde{\boldsymbol{v}} - \mathcal{P}_{\Omega^*}\widetilde{\boldsymbol{v}}\|_2 &\leq \|\mathcal{P}_{R\cup\Omega^*}\widetilde{\boldsymbol{v}} - \mathcal{P}_R\widetilde{\boldsymbol{v}}\|_2 \\ &= \|\mathcal{P}_{R\cup\Omega^*}\widetilde{\boldsymbol{v}} - \mathcal{P}_R\mathcal{P}_{R\cup\Omega^*}\widetilde{\boldsymbol{v}}\|_2 \\ &\leq \|\mathcal{P}_{R\cup\Omega^*}\widetilde{\boldsymbol{v}} - \mathcal{P}_R\boldsymbol{v}\|_2 \\ &= \|(\mathcal{P}_{R\cup\Omega^*}\widetilde{\boldsymbol{v}} - \boldsymbol{v})\|_2 . \end{aligned} \quad (8)$$

The second line above uses Lemma A.6, the third line follows from the fact that $\mathcal{P}_R\mathcal{P}_{R\cup\Omega^*}\widetilde{\boldsymbol{v}}$ must be the nearest neighbor to $\mathcal{P}_{R\cup\Omega^*}\widetilde{\boldsymbol{v}}$ among all vectors in $\mathcal{R}(\boldsymbol{D}_R)$, and the fourth line uses the fact that $\mathcal{P}_R\boldsymbol{v} = \boldsymbol{v}$ because $R = \mathcal{S}_{2k}(\boldsymbol{v})$ and both $\boldsymbol{x}$ and $\boldsymbol{x}^\ell$ are k-sparse in $\boldsymbol{D}$.

To deal with the third term in (7), note that

$$\begin{aligned} \|\mathcal{P}_{\Omega^*}\widetilde{\boldsymbol{v}} - \mathcal{P}_\Omega\widetilde{\boldsymbol{v}}\|_2 &\leq \epsilon_1\, \|\mathcal{P}_{\Omega^*}\widetilde{\boldsymbol{v}}\|_2 \\ &\leq \epsilon_1\, \|\mathcal{P}_{\Omega^*}\mathcal{P}_{R\cup\Omega^*}\widetilde{\boldsymbol{v}}\|_2 \\ &\leq \epsilon_1\, (\|\boldsymbol{v}\|_2 + \|\mathcal{P}_{R\cup\Omega^*}\widetilde{\boldsymbol{v}} - \boldsymbol{v}\|_2) . \end{aligned} \quad (9)$$

The first line above follows from the definition of Ω^* and from (1), the second line uses Lemma A.6 and the third line uses the triangle inequality.

Combining (7), (8), and (9) we see that

$$\|\mathcal{P}_{\Omega^\perp}\boldsymbol{v}\|_2 \leq (2+\epsilon_1)\, \|\mathcal{P}_{R\cup\Omega^*}\widetilde{\boldsymbol{v}} - \boldsymbol{v}\|_2 + \epsilon_1\, \|\boldsymbol{v}\|_2 .$$

Since $\boldsymbol{v} \in \mathcal{R}(\boldsymbol{D}_R)$, we have that $\boldsymbol{v} \in \mathcal{R}(\boldsymbol{D}_{R\cup\Omega^*})$, and so

$$\begin{aligned} \|\mathcal{P}_{R\cup\Omega^*}\widetilde{\boldsymbol{v}} - \boldsymbol{v}\|_2 &= \|\mathcal{P}_{R\cup\Omega^*}\boldsymbol{A}^*\boldsymbol{A}\mathcal{P}_{R\cup\Omega^*}\boldsymbol{v} - \mathcal{P}_{R\cup\Omega^*}\boldsymbol{v}\|_2 \\ &\leq \delta_{4k}\, \|\boldsymbol{v}\|_2 , \end{aligned}$$

where we have used Lemma A.5 to get the inequality above. Putting all of this together establishes the lemma.

References

[1] M. Davenport, M. Duarte, Y. Eldar, and G. Kutyniok, "Introduction to compressed sensing," in *Compressed Sensing: Theory and Applications*, Y. Eldar and G. Kutyniok, Eds. Cambridge, UK: Cambridge University Press, 2012.

[2] E. Candès, Y. Eldar, D. Needell, and P. Randall, "Compressed sensing with coherent and redundant dictionaries," *Appl. Comput. Harmon. Anal.*, vol. 31, no. 1, pp. 59–73, 2011.

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

[4] T. Blumensath, "Sampling and reconstructing signals from a union of linear subspaces," *IEEE Trans. Inform. Theory*, vol. 57, no. 7, pp. 4660–4671, 2011.

[5] T. Blumensath and M. Davies, "Iterative hard thresholding for compressive sensing," *Appl. Comput. Harmon. Anal.*, vol. 27, no. 3, pp. 265–274, 2009.

[6] M. Davenport, D. Needell, and M. Wakin, "Signal Space CoSaMP for sparse recovery with redundant dictionaries," *Arxiv preprint arXiv:1208.0353*, 2012.

[7] M. Davenport and M. Wakin, "Compressive sensing of analog signals using discrete prolate spheroidal sequences," *Appl. Comput. Harmon. Anal.*, vol. 33, no. 3, pp. 438–472, 2012.

[8] E. van den Berg and M. Friedlander, "Probing the Pareto frontier for basis pursuit solutions," *SIAM J. Sci. Comput.*, vol. 31, no. 2, pp. 890–912, 2008.

[9] E. Candès and C. Fernandez-Granda, "Towards a mathematical theory of super-resolution," *Arxiv preprint arXiv:1203.5871*, 2012.