7-16-2008

CoSaMP: Iterative Signal Recovery From Incomplete and Inaccurate Samples

Deanna Needell
Claremont McKenna College

J. A. Tropp
California Institute of Technology

Recommended Citation

COSAMP: ITERATIVE SIGNAL RECOVERY
FROM INCOMPLETE AND INACCURATE SAMPLES

D. NEEDELL AND J. A. TROPP

ABSTRACT. Compressive sampling offers a new paradigm for acquiring signals that are compressible with respect to an orthonormal basis. The major algorithmic challenge in compressive sampling is to approximate a compressible signal from noisy samples. This paper describes a new iterative recovery algorithm called CoSaMP that delivers the same guarantees as the best optimization-based approaches. Moreover, this algorithm offers rigorous bounds on computational cost and storage. It is likely to be extremely efficient for practical problems because it requires only matrix–vector multiplies with the sampling matrix. For compressible signals, the running time is just $O(N \log^2 N)$, where N is the length of the signal.

1. INTRODUCTION

Most signals of interest contain scant information relative to their ambient dimension, but the classical approach to signal acquisition ignores this fact. We usually collect a complete representation of the target signal and process this representation to sieve out the actionable information. Then we discard the rest. Contemplating this ugly inefficiency, one might ask if it is possible instead to acquire *compressive samples*. In other words, is there some type of measurement that automatically winnows out the information from a signal? Incredibly, the answer is sometimes *yes*.

Compressive sampling refers to the idea that, for certain types of signals, a small number of nonadaptive samples carries sufficient information to approximate the signal well. Research in this area has two major components:

Sampling: How many samples are necessary to reconstruct signals to a specified precision? What type of samples? How can these sampling schemes be implemented in practice?

Reconstruction: Given the compressive samples, what algorithms can efficiently construct a signal approximation?

The literature already contains a well-developed theory of sampling, which we summarize below. Although algorithmic work has been progressing, the state of knowledge is less than complete. We assert that a practical signal reconstruction algorithm should have all of the following properties.

- It should accept samples from a variety of sampling schemes.
- It should succeed using a minimal number of samples.
- It should be robust when samples are contaminated with noise.
- It should provide optimal error guarantees for every target signal.
- It should offer provably efficient resource usage.

To our knowledge, no approach in the literature simultaneously accomplishes all five goals.

This paper presents and analyzes a novel signal reconstruction algorithm that achieves these desiderata. The algorithm is called CoSaMP, from the acrostic *Compressive Sampling Matching Pursuit*. As the name suggests, the new method is ultimately based on orthogonal matching pursuit (OMP) [36], but it incorporates several other ideas from the literature to accelerate the algorithm and to provide strong guarantees that OMP cannot. Before we describe the algorithm, let us deliver an introduction to the theory of compressive sampling.

1.1. **Rudiments of Compressive Sampling.**

To enhance intuition, we focus on sparse and compressible signals. For vectors in $\mathbb{C}^N$, define the ℓ_0 quasi-norm

$$\|\boldsymbol{x}\|_0 = |\mathrm{supp}(\boldsymbol{x})| = |\{j : x_j \neq 0\}|.$$

We say that a signal $\boldsymbol{x}$ is *s-sparse* when $\|\boldsymbol{x}\|_0 \leq s$. Sparse signals are an idealization that we do not encounter in applications, but real signals are quite often *compressible*, which means that their entries decay rapidly when sorted by magnitude. As a result, compressible signals are well approximated by sparse signals. We can also talk about signals that are compressible with respect to other orthonormal bases, such as a Fourier or wavelet basis. In this case, the sequence of coefficients in the orthogonal expansion decays quickly. It represents no loss of generality to focus on signals that are compressible with respect to the standard basis, and we do so without regret. For a more precise definition of compressibility, turn to Section 2.6.

In the theory of compressive sampling, a *sample* is a linear functional applied to a signal. The process of collecting multiple samples is best viewed as the action of a *sampling matrix* $\boldsymbol{\Phi}$ on the target signal. If we take m samples, or *measurements*, of a signal in $\mathbb{C}^N$, then the sampling matrix $\boldsymbol{\Phi}$ has dimensions $m \times N$. A natural question now arises: *How many measurements are necessary to acquire s-sparse signals?*

The minimum number of measurements $m \geq 2s$ on account of the following simple argument. The sampling matrix must not map two different s-sparse signals to the same set of samples. Therefore, each collection of $2s$ columns from the sampling matrix must be nonsingular. It is easy to see that certain Vandermonde matrices satisfy this property, but these matrices are not really suitable for signal acquisition because they contain square minors that are very badly conditioned. As a result, some sparse signals are mapped to very similar sets of samples, and it is unstable to invert the sampling process numerically.

Instead, Candès and Tao proposed the stronger condition that the geometry of sparse signals should be preserved under the action of the sampling matrix [6]. To quantify this idea, they defined the rth *restricted isometry constant* of a matrix $\boldsymbol{\Phi}$ as the least number δ_r for which

$$(1 - \delta_r)\|\boldsymbol{x}\|_2^2 \leq \|\boldsymbol{\Phi}\boldsymbol{x}\|_2^2 \leq (1 + \delta_r)\|\boldsymbol{x}\|_2^2 \qquad \text{whenever } \|\boldsymbol{x}\|_0 \leq r. \qquad (1.1)$$

We have written $\|\cdot\|_2$ for the ℓ_2 vector norm. When $\delta_r < 1$, these inequalities imply that each collection of r columns from $\boldsymbol{\Phi}$ is nonsingular, which is the minimum requirement for acquiring $(r/2)$-sparse signals. When $\delta_r \ll 1$, the sampling operator very nearly maintains the ℓ_2 distance between each pair of $(r/2)$-sparse signals. In consequence, it is possible to invert the sampling process stably.

To acquire s-sparse signals, one therefore hopes to achieve a small restricted isometry constant δ_{2s} with as few samples as possible. A striking fact is that many types of random matrices have excellent restricted isometry behavior. For example, we can often obtain $\delta_{2s} \leq 0.1$ with

$$m = \mathrm{O}(s \log^\alpha N)$$

measurements, where α is a small integer. Unfortunately, no deterministic sampling matrix is known to satisfy a comparable bound. Even worse, it is computationally difficult to check the inequalities (1.1), so it may never be possible to exhibit an explicit example of a good sampling matrix.

As a result, it is important to understand how random sampling matrices behave. The two quintessential examples are Gaussian matrices and partial Fourier matrices.

Gaussian matrices: If the entries of $\sqrt{m}\boldsymbol{\Phi}$ are independent and identically distributed standard normal variables then

$$m \geq \frac{\mathrm{C}r\log(N/r)}{\varepsilon^2} \qquad \Longrightarrow \qquad \delta_r \leq \varepsilon \tag{1.2}$$

except with probability $\mathrm{e}^{-\mathrm{c}m}$. See [6] for details.

Partial Fourier matrices: If $\sqrt{m}\boldsymbol{\Phi}$ is a uniformly random set of m rows drawn from the $N \times N$ unitary discrete Fourier transform (DFT), then

$$m \geq \frac{\mathrm{C}r\log^5 N \cdot \log(\varepsilon^{-1})}{\varepsilon^2} \qquad \Longrightarrow \qquad \delta_r \leq \varepsilon \tag{1.3}$$

except with probability N^{-1}. See [32] for the proof. Experts believe that the power on the first logarithm should be no greater than two [30].

Here and elsewhere, we follow the analyst's convention that upright letters $(\mathrm{c}, \mathrm{C}, \dots)$ refer to positive, universal constants that may change from appearance to appearance.

The Gaussian matrix is important because it has optimal restricted isometry behavior. Indeed, for any $m \times N$ matrix,

$$\delta_r \leq 0.1 \qquad \Longrightarrow \qquad m \geq \mathrm{C}r\log(N/r),$$

on account of profound geometric results of Kashin [23] and Garnaev–Gluskin [15]. Even though partial Fourier matrices may require additional samples to achieve a small restricted isometry constant, they are more interesting for the following reasons [2].

- There are technologies that acquire random Fourier measurements at unit cost per sample.
- The sampling matrix can be applied to a vector in time $\mathrm{O}(N\log N)$.
- The sampling matrix requires only $\mathrm{O}(m\log N)$ storage.

Other types of sampling matrices, such as the *random demodulator* [35], enjoy similar qualities. These traits are essential for the translation of compressive sampling from theory into practice.

1.2. **Signal Recovery Algorithms.** The major algorithmic challenge in compressive sampling is to approximate a signal given a vector of noisy samples. The literature describes a huge number of approaches to solving this problem. They fall into three rough categories:

Greedy pursuits: These methods build up an approximation one step at a time by making locally optimal choices at each step. Examples include OMP [36], stagewise OMP (StOMP) [13], and regularized OMP (ROMP) [28, 27].

Convex relaxation: These techniques solve a convex program whose minimizer is known to approximate the target signal. Many algorithms have been proposed to complete the optimization, including interior-point methods [2, 24], projected gradient methods [14], and iterative thresholding [11].

Combinatorial algorithms: These methods acquire highly structured samples of the signal that support rapid reconstruction via group testing. This class includes Fourier sampling [19, 21], chaining pursuit [16], and HHS pursuit [17], as well as some algorithms of Cormode–Muthukrishnan [9] and Iwen [22].

At present, each type of algorithm has its native shortcomings. Many of the combinatorial algorithms are extremely fast—sublinear in the length of the target signal—but they require a large number of somewhat unusual samples that may not be easy to acquire. At the other extreme, convex relaxation algorithms succeed with a very small number of measurements, but they tend to be computationally burdensome. Greedy pursuits—in particular, the ROMP algorithm—are intermediate in their running time and sampling efficiency.

CoSaMP, the algorithm described in this paper, is at heart a greedy pursuit. It also incorporates ideas from the combinatorial algorithms to guarantee speed and to provide rigorous error bounds [17]. The analysis is inspired by the work on ROMP [28, 27] and the work of Candès–Romberg–Tao [3] on convex relaxation methods. In particular, we establish the following result.

Theorem A (CoSaMP). *Suppose that $\boldsymbol{\Phi}$ is an $m \times N$ sampling matrix with restricted isometry constant $\delta_{2s} \leq$ c. Let $\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{x} + \boldsymbol{e}$ be a vector of samples of an arbitrary signal, contaminated with arbitrary noise. For a given precision parameter η, the algorithm* CoSaMP *produces a $2s$-sparse approximation $\boldsymbol{a}$ that satisfies*

$$\|\boldsymbol{x} - \boldsymbol{a}\|_2 \leq \mathrm{C} \cdot \max\left\{\eta, \frac{1}{\sqrt{s}}\|\boldsymbol{x} - \boldsymbol{x}_s\|_1 + \|\boldsymbol{e}\|_2\right\}$$

where $\boldsymbol{x}_s$ is a best s-sparse approximation to $\boldsymbol{x}$. The running time is $\mathrm{O}(\mathscr{L} \cdot \log(\|\boldsymbol{x}\|_2 / \eta))$, where $\mathscr{L}$ bounds the cost of a matrix–vector multiply with $\boldsymbol{\Phi}$ or $\boldsymbol{\Phi}^*$. The working storage use is $\mathrm{O}(N)$.

Let us expand on the statement of this result. First, recall that many types of random sampling matrices satisfy the restricted isometry hypothesis when the number of samples $m = \mathrm{O}(s \log^\alpha N)$. Therefore, the theorem applies to a wide class of sampling schemes when the number of samples is proportional to the target sparsity and logarithmic in the ambient dimension of the signal space.

The algorithm produces a $2s$-sparse approximation whose ℓ_2 error is comparable with the scaled ℓ_1 error of the best s-sparse approximation to the signal. Of course, the algorithm cannot resolve the uncertainty due to the additive noise, so we also pay for the energy in the noise. This type of error bound is structurally optimal, as we discuss in Section 2.6. Some disparity in the sparsity levels (here, $2s$ versus s) seems to be necessary when the recovery algorithm is computationally efficient [31].

We can interpret the error guarantee as follows. In the absence of noise, the algorithm can recover an s-sparse signal to arbitrarily high precision. Performance degrades gracefully as the energy in the noise increases. Performance also degrades gracefully for compressible signals. The theorem is ultimately vacuous for signals that cannot be approximated by sparse signals, but compressive sampling is not an appropriate technique for this class.

The running time bound indicates that each matrix–vector multiplication reduces the error by a constant factor (if we amortize over the entire execution). That is, the algorithm has linear convergence[1]. We find that the total runtime is roughly proportional to (the negation of) the *reconstruction signal-to-noise ratio*

$$\mathsf{R\text{-}SNR} = 10 \log_{10}\left(\frac{\|\boldsymbol{x} - \boldsymbol{a}\|_2}{\|\boldsymbol{x}\|_2}\right) \quad \mathrm{dB}.$$

For compressible signals, one can show that $|\mathsf{R\text{-}SNR}| = \mathrm{O}(\log s)$. The runtime is also proportional to the cost of a matrix–vector multiply. For sampling matrices with a fast multiply, the algorithm is accelerated substantially. In particular, for the partial Fourier matrix, a matrix–vector multiply requires time $\mathrm{O}(N \log N)$. It follows that the total runtime is $\mathrm{O}(N \log N \cdot |\mathsf{R\text{-}SNR}|)$. For most signals of interest, this cost is nearly linear in the signal length!

1.3. **Notation.** Let us instate several pieces of notation that are carried throughout the paper. For $p \in [1, \infty]$, we write $\|\cdot\|_p$ for the usual ℓ_p vector norm. We reserve the symbol $\|\cdot\|$ for the spectral norm, i.e., the natural norm on linear maps from ℓ_2 to ℓ_2.

Suppose that $\boldsymbol{x}$ is a signal in $\mathbb{C}^N$ and r is a positive integer. We write $\boldsymbol{x}_r$ for the signal in $\mathbb{C}^N$ that is formed by restricting $\boldsymbol{x}$ to its r largest-magnitude components. Ties are broken lexicographically.

---

[1]Mathematicians sometimes refer to linear convergence as "exponential convergence."

This signal is a best r-sparse approximation to $\boldsymbol{x}$ with respect to any ℓ_p norm. Suppose now that T is a subset of $\{1, 2, \ldots, N\}$. We define the restriction of the signal to the set T as

$$
\boldsymbol{x}|_T = \begin{cases} x_i, & i \in T \\ 0, & \text{otherwise.} \end{cases}
$$

We occasionally abuse the notation and treat $\boldsymbol{x}|_T$ as an element of the vector space $\mathbb{C}^T$. We also define the restriction $\boldsymbol{\Phi}_T$ of the sampling matrix $\boldsymbol{\Phi}$ as the column submatrix whose columns are listed in the set T.

Finally, we define the pseudoinverse of a tall, full-rank matrix $\boldsymbol{A}$ by the formula $\boldsymbol{A}^\dagger = (\boldsymbol{A}^* \boldsymbol{A})^{-1} \boldsymbol{A}^*$.

1.4. **Organization.** The rest of the paper has the following structure. In Section 2 we introduce the CoSaMP algorithm, we state the major theorems in more detail, and we discuss implementation and resource requirements. Section 3 describes some consequences of the restricted isometry property that pervade our analysis. The central theorem is established for sparse signals in Sections 4 and 5. We extend this result to general signals in Section 6. Finally, Section 7 places the algorithm in the context of previous work. The first appendix presents variations on the algorithm. The second appendix contains a bound on the number of iterations required when the algorithm is implemented using exact arithmetic.

2. THE CoSaMP ALGORITHM

This section gives an overview of the algorithm, along with explicit pseudocode. It presents the major theorems on the performance of the algorithm. Then it covers details of implementation and bounds on resource requirements.

2.1. **Intuition.** The most difficult part of signal reconstruction is to identify the locations of the largest components in the target signal. CoSaMP uses an approach inspired by the restricted isometry property. Suppose that the sampling matrix $\boldsymbol{\Phi}$ has restricted isometry constant $\delta_s \ll 1$. For an s-sparse signal $\boldsymbol{x}$, the vector $\boldsymbol{y} = \boldsymbol{\Phi}^* \boldsymbol{\Phi} \boldsymbol{x}$ can serve as a proxy for the signal because the energy in each set of s components of $\boldsymbol{y}$ approximates the energy in the corresponding s components of $\boldsymbol{x}$. In particular, the largest s entries of the proxy $\boldsymbol{y}$ point toward the largest s entries of the signal $\boldsymbol{x}$. Since the samples have the form $\boldsymbol{u} = \boldsymbol{\Phi} \boldsymbol{x}$, we can obtain the proxy just by applying the matrix $\boldsymbol{\Phi}^*$ to the samples.

The algorithm invokes this idea iteratively to approximate the target signal. At each iteration, the current approximation induces a residual, the part of the target signal that has not been approximated. As the algorithm progresses, the samples are updated so that they reflect the current residual. These samples are used to construct a proxy for the residual, which permits us to identify the large components in the residual. This step yields a tentative support for the next approximation. We use the samples to estimate the approximation on this support set using least squares. This process is repeated until we have found the recoverable energy in the signal.

2.2. **Overview.** As input, the CoSaMP algorithm requires four pieces of information:

- Access to the sampling operator via matrix–vector multiplication.
- A vector of (noisy) samples of the unknown signal.
- The sparsity of the approximation to be produced.
- A halting criterion.

The algorithm is initialized with a trivial signal approximation, which means that the initial residual equals the unknown target signal. During each iteration, CoSaMP performs five major steps:

(1) **Identification.** The algorithm forms a proxy of the residual from the current samples and locates the largest components of the proxy.

(2) **Support Merger.** The set of newly identified components is united with the set of components that appear in the current approximation.

(3) **Estimation.** The algorithm solves a least-squares problem to approximate the target signal on the merged set of components.

(4) **Pruning.** The algorithm produces a new approximation by retaining only the largest entries in this least-squares signal approximation.

(5) **Sample Update.** Finally, the samples are updated so that they reflect the residual, the part of the signal that has not been approximated.

These steps are repeated until the halting criterion is triggered. In the body of this work, we concentrate on methods that use a fixed number of iterations. Appendix A discusses some other simple stopping rules that may also be useful in practice.

Pseudocode for CoSaMP appears as Algorithm 2.1. This code describes the version of the algorithm that we analyze in this paper. Nevertheless, there are several adjustable parameters that may improve performance: the number of components selected in the identification step and the number of components retained in the pruning step. For a brief discussion of other variations on the algorithm, turn to Appendix A.

ALGORITHM 2.1: CoSaMP Recovery Algorithm

CoSaMP($\mathbf{\Phi}$, $\boldsymbol{u}$, s)
Input: Sampling matrix $\mathbf{\Phi}$, noisy sample vector $\boldsymbol{u}$, sparsity level s
Output: An s-sparse approximation $\boldsymbol{a}$ of the target signal

---

$\boldsymbol{a}^0 \leftarrow \boldsymbol{0}$ { Trivial initial approximation }
$\boldsymbol{v} \leftarrow \boldsymbol{u}$ { Current samples = input samples }
$k \leftarrow 0$

repeat
    $k \leftarrow k + 1$

    $\boldsymbol{y} \leftarrow \mathbf{\Phi}^* \boldsymbol{v}$ { Form signal proxy }
    $\Omega \leftarrow \operatorname{supp}(\boldsymbol{y}_{2s})$ { Identify large components }
    $T \leftarrow \Omega \cup \operatorname{supp}(\boldsymbol{a}^{k-1})$ { Merge supports }

    $\boldsymbol{b}|_T \leftarrow \mathbf{\Phi}_T^\dagger \boldsymbol{u}$ { Signal estimation by least-squares }
    $\boldsymbol{b}|_{T^c} \leftarrow \boldsymbol{0}$

    $\boldsymbol{a}^k \leftarrow \boldsymbol{b}_s$ { Prune to obtain next approximation }
    $\boldsymbol{v} \leftarrow \boldsymbol{u} - \mathbf{\Phi}\boldsymbol{a}^k$ { Update current samples }

until halting criterion *true*

2.3. **Performance Guarantees.** This section describes our theoretical analysis of the behavior of CoSaMP. The next section covers the resource requirements of the algorithm. Afterward, Section 2.5 combines these materials to establish Theorem A.

Our results depend on a set of hypotheses that has become common in the compressive sampling literature. Let us frame the standing assumptions:

CoSaMP HYPOTHESES

- The sparsity level s is fixed.
- The $m \times N$ sampling operator $\mathbf{\Phi}$ has restricted isometry constant $\delta_{4s} \leq 0.1$.
- The signal $\boldsymbol{x} \in \mathbb{C}^N$ is arbitrary, except where noted.
- The noise vector $\boldsymbol{e} \in \mathbb{C}^m$ is arbitrary.
- The vector of samples $\boldsymbol{u} = \mathbf{\Phi}\boldsymbol{x} + \boldsymbol{e}$.

We also define the *unrecoverable energy* ν in the signal. This quantity measures the baseline error in our approximation that occurs because of noise in the samples or because the signal is not sparse.

$$\nu = \|\boldsymbol{x} - \boldsymbol{x}_s\|_2 + \frac{1}{\sqrt{s}}\|\boldsymbol{x} - \boldsymbol{x}_s\|_1 + \|\boldsymbol{e}\|_2. \tag{2.1}$$

We postpone a more detailed discussion of the unrecoverable energy until Section 2.6.

Our key result is that CoSaMP makes significant progress during each iteration where the approximation error is large relative to the unrecoverable energy.

Theorem 2.1 (Iteration Invariant). *For each iteration $k \geq 0$, the signal approximation $\boldsymbol{a}^k$ is s-sparse and*

$$\left\|\boldsymbol{x} - \boldsymbol{a}^{k+1}\right\|_2 \leq 0.5\left\|\boldsymbol{x} - \boldsymbol{a}^k\right\|_2 + 10\nu.$$

In particular,

$$\left\|\boldsymbol{x} - \boldsymbol{a}^k\right\|_2 \leq 2^{-k}\left\|\boldsymbol{x}\right\|_2 + 20\nu.$$

The proof of Theorem 2.1 will occupy us for most of this paper. In Section 4, we establish an analog for sparse signals. The version for general signals appears as a corollary in Section 6.

Theorem 2.1 has some immediate consequences for the quality of reconstruction with respect to standard signal metrics. In this setting, a sensible definition of the *signal-to-noise ratio* (SNR) is

$$\mathsf{SNR} = 10\log_{10}\left(\frac{\|\boldsymbol{x}\|_2}{\nu}\right).$$

The *reconstruction SNR* is defined as

$$\mathsf{R\text{-}SNR} = 10\log_{10}\left(\frac{\|\boldsymbol{x} - \boldsymbol{a}\|_2}{\|\boldsymbol{x}\|_2}\right).$$

Both quantities are measured in decibels. Theorem 2.1 implies that, after k iterations, the reconstruction SNR satisfies

$$\mathsf{R\text{-}SNR} \lesssim 3 - \min\{3k, \mathsf{SNR} - 13\}.$$

In words, each iteration reduces the reconstruction SNR by about 3 decibels until the error nears the noise floor. To reduce the error to its minimal value, the number of iterations is proportional to the SNR.

Let us consider a slightly different scenario. Suppose that the signal $\boldsymbol{x}$ is s-sparse, so the unrecoverable energy $\nu = \|\boldsymbol{e}\|_2$. Define the *dynamic range*

$$\Delta = 10\log_{10}\left(\frac{\max|x_i|}{\min|x_i|}\right) \qquad \text{where } i \text{ ranges over } \mathrm{supp}(\boldsymbol{x}).$$

Assume moreover that the minimum nonzero component of the signal is at least 40ν. Using the fact that $\|\boldsymbol{x}\|_2 \leq \sqrt{s}\,\|\boldsymbol{x}\|_\infty$, it is easy to check that $\|\boldsymbol{x} - \boldsymbol{a}\|_2 \leq \min |x_i|$ as soon as the number k of iterations satisfies

$$k \gtrsim 3.3\Delta + \log_2 \sqrt{s} + 1.$$

It follows that the support of the approximation $\boldsymbol{a}$ must contain every entry in the support of the signal $\boldsymbol{x}$.

This discussion suggests that the number of iterations might be substantial if we require a very low reconstruction SNR or if the signal has a very wide dynamic range. This initial impression is not entirely accurate. We have established that, when the algorithm performs arithmetic to high enough precision, then a fixed number of iterations suffices to reduce the approximation error to the same order as the unrecoverable energy. Here is a result for exact computations.

Theorem 2.2 (Iteration Count). *Suppose that* CoSaMP *is implemented with exact arithmetic. After at most $6(s+1)$ iterations,* CoSaMP *produces an s-sparse approximation $\boldsymbol{a}$ that satisfies*

$$\|\boldsymbol{x} - \boldsymbol{a}\|_2 \leq 20\nu.$$

In fact, even more is true. The number of iterations depends significantly on the structure of the signal. The only situation where the algorithm needs $\Omega(s)$ iterations occurs when the entries of the signal decay exponentially. For signals whose largest entries are comparable, the number of iterations may be as small as $\mathrm{O}(\log s)$. This claim is quantified in Appendix B, where we prove Theorem 2.2.

If we solve the least-squares problems to high precision, an analogous result holds, but the approximation guarantee contains an extra term that comes from solving the least-squares problems imperfectly. In practice, it may be more efficient overall to solve the least-squares problems to low precision. The correct amount of care seems to depend on the relative costs of forming the signal proxy and solving the least-squares problem, which are the two most expensive steps in the algorithm. We discuss this point in the next section. Ultimately, the question is best settled with empirical studies.

Remark 2.3. *In the hypotheses, a bound on the restricted isometry constant δ_{2s} also suffices. Indeed, Corollary 3.4 of the sequel implies that $\delta_{4s} \leq 0.1$ holds whenever $\delta_{2s} \leq 0.025$.*

Remark 2.4. *The expression* (2.1) *for the unrecoverable energy can be simplified using Lemma 7 from [17], which states that, for every signal $\boldsymbol{y} \in \mathbb{C}^N$ and every positive integer t, we have*

$$\|\boldsymbol{y} - \boldsymbol{y}_t\|_2 \leq \frac{1}{2\sqrt{t}}\,\|\boldsymbol{y}\|_1.$$

Choosing $\boldsymbol{y} = \boldsymbol{x} - \boldsymbol{x}_{s/2}$ and $t = s/2$, we reach

$$\nu \leq \frac{1.71}{\sqrt{s}}\,\|\boldsymbol{x} - \boldsymbol{x}_{s/2}\|_1 + \|\boldsymbol{e}\|_2. \tag{2.2}$$

In words, the unrecoverable energy is controlled by the scaled ℓ_1 norm of the signal tail.

2.4. **Implementation and Resource Requirements.** CoSaMP was designed to be a practical method for signal recovery. An efficient implementation of the algorithm requires some ideas from numerical linear algebra, as well as some basic techniques from the theory of algorithms. This section discusses the key issues and develops an analysis of the running time for the two most common scenarios.

We focus on the least-squares problem in the estimation step because it is the major obstacle to a fast implementation of the algorithm. The algorithm guarantees that the matrix $\boldsymbol{\Phi}_T$ never has more than $3s$ columns, so our assumption $\delta_{4s} \leq 0.1$ implies that the matrix $\boldsymbol{\Phi}_T$ is extremely well conditioned. As a result, we can apply the pseudoinverse $\boldsymbol{\Phi}_T^\dagger = (\boldsymbol{\Phi}_T^* \boldsymbol{\Phi}_T)^{-1}\boldsymbol{\Phi}_T^*$ very quickly using an iterative method, such as Richardson's iteration [1, Sec. 7.2.3] or conjugate gradient [1,

Sec. 7.4]. These techniques have the additional advantage that they only interact with the matrix $\mathbf{\Phi}_T$ through its action on vectors. It follows that the algorithm performs better when the sampling matrix has a fast matrix–vector multiply.

Section 5 contains an analysis of the performance of iterative least-squares algorithms in the context of CoSaMP. In summary, if we initialize the least-squares method with the current approximation $\boldsymbol{a}^{k-1}$, then the cost of solving the least-squares problem is $\mathrm{O}(\mathscr{L})$, where $\mathscr{L}$ bounds the cost of a matrix–vector multiply with $\mathbf{\Phi}_T$ or $\mathbf{\Phi}_T^*$. This implementation ensures that Theorem 2.1 holds at each iteration.

We emphasize that direct methods for least squares are likely to be extremely inefficient in this setting. The first reason is that each least-squares problem may contain substantially different sets of columns from $\mathbf{\Phi}$. As a result, it becomes necessary to perform a completely new QR or SVD factorization during each iteration at a cost of $\mathrm{O}(s^2 m)$. The second problem is that computing these factorizations typically requires direct access to the columns of the matrix, which is problematic when the matrix is accessed through its action on vectors. Third, direct methods have storage costs $\mathrm{O}(sm)$, which may be deadly for large-scale problems.

The remaining steps of the algorithm are standard. Let us estimate the operation counts.

Proxy: Forming the proxy is dominated by the cost of the matrix–vector multiply $\mathbf{\Phi}^* \boldsymbol{v}$.

Identification: We can locate the largest $2s$ entries of a vector in time $\mathrm{O}(N)$ using the approach in [8, Ch. 9]. In practice, it may be faster to sort the entries of the signal in decreasing order of magnitude at cost $\mathrm{O}(N \log N)$ and then select the first $2s$ of them. The latter procedure can be accomplished with quicksort, mergesort, or heapsort [8, Sec. II]. To implement the algorithm to the letter, the sorting method needs to be stable because we stipulate that ties are broken lexicographically. This point is not important in practice.

Support Merger: We can merge two sets of size $\mathrm{O}(s)$ in expected time $\mathrm{O}(s)$ using randomized hashing methods [8, Ch. 11]. One can also sort both sets first and use the elementary merge procedure [8, p. 29] for a total cost $\mathrm{O}(s \log s)$.

LS estimation: We use Richardson's iteration or conjugate gradient to compute $\mathbf{\Phi}_T^\dagger \boldsymbol{u}$. Initializing the least-squares algorithm requires a matrix–vector multiply with $\mathbf{\Phi}_T^*$. Each iteration of the least-squares method requires one matrix–vector multiply each with $\mathbf{\Phi}_T$ and $\mathbf{\Phi}_T^*$. Since $\mathbf{\Phi}_T$ is a submatrix of $\mathbf{\Phi}$, the matrix–vector multiplies can also be obtained from multiplication with the full matrix. We prove in Section 5 that a constant number of least-squares iterations suffices for Theorem 2.1 to hold.

Pruning: This step is similar to identification. Pruning can be implemented in time $\mathrm{O}(s)$, but it may be preferable to sort the components of the vector by magnitude and then select the first s at a cost of $\mathrm{O}(s \log s)$.

Sample Update: This step is dominated by the cost of the multiplication of $\mathbf{\Phi}$ with the s-sparse vector $\boldsymbol{a}^k$.

Table 1 summarizes this discussion in two particular cases. The first column shows what happens when the sampling matrix $\mathbf{\Phi}$ is applied to vectors in the standard way, but we have random access to submatrices. The second, column shows what happens when the sampling matrix $\mathbf{\Phi}$ and its adjoint $\mathbf{\Phi}^*$ both have a fast multiply with cost $\mathscr{L}$, where we assume that $\mathscr{L} \geq N$. A typical value is $\mathscr{L} = \mathrm{O}(N \log N)$. In particular, a partial Fourier matrix satisfies this bound.

Finally, we note that the storage requirements of the algorithm are also favorable. Aside from the storage required by the sampling matrix, the algorithm constructs only one vector of length N, the signal proxy. The sample vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ have length m, so they require $\mathrm{O}(m)$ storage. The signal approximations can be stored using sparse data structures, so they require at most $\mathrm{O}(s \log N)$ storage. Similarly, the index sets that appear require only $\mathrm{O}(s \log N)$ storage. The total storage is $\mathrm{O}(N)$.

TABLE 1. Operation count for CoSaMP. Big-O notation is omitted for legibility. The dimensions of the sampling matrix $\boldsymbol{\Phi}$ are $m \times N$; the sparsity level is s. The number $\mathscr{L}$ bounds the cost of a matrix–vector multiply with $\boldsymbol{\Phi}$ or $\boldsymbol{\Phi}^*$.

Step	Standard multiply	Fast multiply
Form proxy	mN	$\mathscr{L}$
Identification	N	N
Support merger	s	s
LS estimation	sm	$\mathscr{L}$
Pruning	s	s
Sample update	sm	$\mathscr{L}$
Total per iteration	$\mathrm{O}(mN)$	$\mathrm{O}(\mathscr{L})$

The following result summarizes this discussion.

Theorem 2.5 (Resource Requirements). *Each iteration of* CoSaMP *requires* $\mathrm{O}(\mathscr{L})$ *time, where* $\mathscr{L}$ *bounds the cost of a multiplication with the matrix* $\boldsymbol{\Phi}$ *or* $\boldsymbol{\Phi}^*$*. The algorithm uses storage* $\mathrm{O}(N)$*.*

Remark 2.6. *We have been able to show that Theorem 2.2 holds when the least-squares problems are solved iteratively with a delicately chosen stopping threshold. In this case, the total number of least-squares iterations performed over the entire execution of the algorithm is at most* $\mathrm{O}(\log(\|\boldsymbol{x}\|_2 / \eta))$ *if we wish to achieve error* $\mathrm{O}(\eta + \nu)$*. When the cost of forming the signal proxy is much higher than the cost of solving the least-squares problem, this analysis may yield a sharper result. For example, using standard matrix–vector multiplication, we have a runtime bound*

$$\mathrm{O}(s \cdot mN + \log(\|\boldsymbol{x}\|_2 / \eta) \cdot sm).$$

The first term reflects the number of CoSaMP *iterations times the cost of forming the signal proxy. The second term reflects the total cost of the least-squares iterations. Unless the relative precision* $\|\boldsymbol{x}\|_2 / \eta$ *is superexponential in the signal length, we obtain running time* $\mathrm{O}(smN)$*. This bound is comparable with the worst-case cost of OMP or ROMP. As we discuss in Appendix B, the number of* CoSaMP *iterations may be much smaller than* s*, which also improves the estimate.*

2.5. **Proof of Theorem A.** We have now collected all the material we need to establish the main result. Fix a precision parameter η. After at most $\mathrm{O}(\log(\|\boldsymbol{x}\|_2 / \eta))$ iterations, CoSaMP produces an s-sparse approximation $\boldsymbol{a}$ that satisfies

$$\|\boldsymbol{x} - \boldsymbol{a}\|_2 \leq \mathrm{C} \cdot (\eta + \nu)$$

in consequence of Theorem 2.1. Apply inequality (2.2) to bound the unrecoverable energy ν in terms of the ℓ_1 norm. We see that the approximation error satisfies

$$\|\boldsymbol{x} - \boldsymbol{a}\|_2 \leq \mathrm{C} \cdot \max \left\{ \eta, \frac{1}{\sqrt{s}} \|\boldsymbol{x} - \boldsymbol{x}_{s/2}\|_1 + \|\boldsymbol{e}\|_2 \right\}.$$

According to Theorem 2.5, each iteration of CoSaMP is completed in time $\mathrm{O}(\mathscr{L})$, where $\mathscr{L}$ bounds the cost of a matrix–vector multiplication with $\boldsymbol{\Phi}$ or $\boldsymbol{\Phi}^*$. The total runtime, therefore, is $\mathrm{O}(\mathscr{L} \log(\|\boldsymbol{x}\|_2 / \eta))$. The total storage is $\mathrm{O}(N)$.

In the statement of the theorem, perform the substitution $s/2 \mapsto s$. Finally, we replace δ_{8s} with δ_{2s} by means of Corollary 3.4, which states that $\delta_{cr} \leq c \cdot \delta_{2r}$ for any positive integers c and r.

2.6. **The Unrecoverable Energy.** Since the unrecoverable energy ν plays a central role in our analysis of CoSaMP, it merits some additional discussion. In particular, it is informative to examine the unrecoverable energy in a compressible signal. Let p be a number in the interval $(0, 1)$. We say that $\boldsymbol{x}$ is *p-compressible* with magnitude R if the sorted components of the signal decay at the rate

$$|x|_{(i)} \le R \cdot i^{-1/p} \qquad \text{for } i = 1, 2, 3, \ldots.$$

When $p = 1$, this definition implies that $\|\boldsymbol{x}\|_1 \le R \cdot (1 + \log N)$. Therefore, the unit ball of 1-compressible signals is similar to the ℓ_1 unit ball. When $p \approx 0$, this definition implies that p-compressible signals are very nearly sparse. In general, compressible signals are well approximated by sparse signals:

$$\|\boldsymbol{x} - \boldsymbol{x}_s\|_1 \le \mathrm{C}_p \cdot R \cdot s^{1-1/p}$$
$$\|\boldsymbol{x} - \boldsymbol{x}_s\|_2 \le \mathrm{D}_p \cdot R \cdot s^{1/2-1/p}$$

where $\mathrm{C}_p = (1/p - 1)^{-1}$ and $\mathrm{D}_p = (2/p - 1)^{-1/2}$. These results follow by writing each norm as a sum and approximating the sum with an integral. We see that the unrecoverable energy (2.1) in a p-compressible signal is bounded as

$$\nu \le 2\mathrm{C}_p \cdot R \cdot s^{1/2-1/p} + \|\boldsymbol{e}\|_2 . \tag{2.3}$$

When p is small, the first term in the unrecoverable energy decays rapidly as the sparsity level s increases. For the class of p-compressible signals, the bound (2.3) on the unrecoverable energy is sharp, modulo the exact values of the constants.

With these inequalities, we can see that CoSaMP recovers compressible signals efficiently. Let us calculate the number of iterations required to reduce the approximation error from $\|\boldsymbol{x}\|_2$ to the optimal level (2.3). For compressible signals, the energy $\|\boldsymbol{x}\|_2 \le 2R$, so

$$\log \left(\frac{2R}{2\mathrm{C}_p \cdot R \cdot s^{1/2-1/p}} \right) = \log(1/p - 1) + (1/p - 1/2) \log s.$$

Therefore, the number of iterations required to recover a generic p-compressible signal is $\mathrm{O}(\log s)$, where the constant in the big-O notation depends on p.

The term "unrecoverable energy" is justified by several facts. First, we must pay for the ℓ_2 error contaminating the samples. To check this point, define $S = \operatorname{supp}(\boldsymbol{x}_s)$. The matrix $\boldsymbol{\Phi}_S$ is nearly an isometry from ℓ_2^S to ℓ_2^m, so an error in the large components of the signal induces an error of equivalent size in the samples. Clearly, we can never resolve this uncertainty.

The term $s^{-1/2} \|\boldsymbol{x} - \boldsymbol{x}_s\|_1$ is also required on account of classical results about the Gel'fand widths of the ℓ_1^N ball in ℓ_2^N, due to Kashin [23] and Garnaev–Gluskin [15]. In the language of compressive sampling, their work has the following interpretation. Let $\boldsymbol{\Phi}$ be a fixed $m \times N$ sampling matrix. Suppose that, for every signal $\boldsymbol{x} \in \mathbb{C}^N$, there is an algorithm that uses the samples $\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{x}$ to construct an approximation $\boldsymbol{a}$ that achieves the error bound

$$\|\boldsymbol{x} - \boldsymbol{a}\|_2 \le \frac{\mathrm{C}}{\sqrt{s}} \|\boldsymbol{x}\|_1 .$$

Then the number m of measurements must satisfy $m \ge \mathrm{c}s \log(N/s)$.

3. Restricted Isometry Consequences

When the sampling matrix satisfies the restricted isometry inequalities (1.1), it has several other properties that we require repeatedly in the proof that the CoSaMP algorithm is correct. Our first observation is a simple translation of (1.1) into other terms.

Proposition 3.1. *Suppose $\mathbf{\Phi}$ has restricted isometry constant δ_r. Let T be a set of r indices or fewer. Then*

$$\left\| \mathbf{\Phi}_T^* \boldsymbol{u} \right\|_2 \leq \sqrt{1 + \delta_r} \left\| \boldsymbol{u} \right\|_2$$

$$\left\| \mathbf{\Phi}_T^\dagger \boldsymbol{u} \right\|_2 \leq \frac{1}{\sqrt{1 - \delta_r}} \left\| \boldsymbol{u} \right\|_2$$

$$\left\| \mathbf{\Phi}_T^* \mathbf{\Phi}_T \boldsymbol{x} \right\|_2 \gtrless^{\leq} (1 \pm \delta_r) \left\| \boldsymbol{x} \right\|_2$$

$$\left\| (\mathbf{\Phi}_T^* \mathbf{\Phi}_T)^{-1} \boldsymbol{x} \right\|_2 \gtrless^{\leq} \frac{1}{1 \pm \delta_r} \left\| \boldsymbol{x} \right\|_2 .$$

where the last two statements contain an upper and lower bound, depending on the sign chosen.

Proof. The restricted isometry inequalities (1.1) imply that the singular values of $\mathbf{\Phi}_T$ lie between $\sqrt{1 - \delta_r}$ and $\sqrt{1 + \delta_r}$. The bounds follow from standard relationships between the singular values of $\mathbf{\Phi}_T$ and the singular values of basic functions of $\mathbf{\Phi}_T$. $\square$

A second consequence is that disjoint sets of columns from the sampling matrix span nearly orthogonal subspaces. The following result quantifies this observation.

Proposition 3.2 (Approximate Orthogonality). *Suppose $\mathbf{\Phi}$ has restricted isometry constant δ_r. Let S and T be disjoint sets of indices whose combined cardinality does not exceed r. Then*

$$\left\| \mathbf{\Phi}_S^* \mathbf{\Phi}_T \right\| \leq \delta_r.$$

Proof. Abbreviate $R = S \cup T$, and observe that $\mathbf{\Phi}_S^* \mathbf{\Phi}_T$ is a submatrix of $\mathbf{\Phi}_R^* \mathbf{\Phi}_R - \mathbf{I}$. The spectral norm of a submatrix never exceeds the norm of the entire matrix. We discern that

$$\left\| \mathbf{\Phi}_S^* \mathbf{\Phi}_T \right\| \leq \left\| \mathbf{\Phi}_R^* \mathbf{\Phi}_R - \mathbf{I} \right\| \leq \max\{(1 + \delta_r) - 1, 1 - (1 - \delta_r)\} = \delta_r$$

because the eigenvalues of $\mathbf{\Phi}_R^* \mathbf{\Phi}_R$ lie between $1 - \delta_r$ and $1 + \delta_r$. $\square$

This result will be applied through the following corollary.

Corollary 3.3. *Suppose $\mathbf{\Phi}$ has restricted isometry constant δ_r. Let T be a set of indices, and let $\boldsymbol{x}$ be a vector. Provided that $r \geq |T \cup \mathrm{supp}(\boldsymbol{x})|$,*

$$\left\| \mathbf{\Phi}_T^* \mathbf{\Phi} \cdot \boldsymbol{x}|_{T^c} \right\|_2 \leq \delta_r \left\| \boldsymbol{x}|_{T^c} \right\|_2 .$$

Proof. Define $S = \mathrm{supp}(\boldsymbol{x}) \setminus T$, so we have $\boldsymbol{x}|_S = \boldsymbol{x}|_{T^c}$. Thus,

$$\left\| \mathbf{\Phi}_T^* \mathbf{\Phi} \cdot \boldsymbol{x}|_{T^c} \right\|_2 = \left\| \mathbf{\Phi}_T^* \mathbf{\Phi} \cdot \boldsymbol{x}|_S \right\|_2 \leq \left\| \mathbf{\Phi}_T^* \mathbf{\Phi}_S \right\| \left\| \boldsymbol{x}|_S \right\|_2 \leq \delta_r \left\| \boldsymbol{x}|_{T^c} \right\|_2 ,$$

owing to Proposition 3.2. $\square$

As a second corollary, we show that δ_{2r} gives weak control over the higher restricted isometry constants.

Corollary 3.4. *Let c and r be positive integers. Then $\delta_{cr} \leq c \cdot \delta_{2r}$.*

Proof. The result is clearly true for $c = 1, 2$, so we assume $c \geq 3$. Let S be an arbitrary index set of size cr, and let $\boldsymbol{M} = \mathbf{\Phi}_S^* \mathbf{\Phi}_S - \mathbf{I}$. It suffices to check that $\|\boldsymbol{M}\| \leq c \cdot \delta_{2r}$. To that end, we break the matrix $\boldsymbol{M}$ into $r \times r$ blocks, which we denote $\boldsymbol{M}_{ij}$. A block version of Gershgorin's theorem states that $\|\boldsymbol{M}\|$ satisfies at least one of the inequalities

$$\left| \|\boldsymbol{M}\| - \|\boldsymbol{M}_{ii}\| \right| \leq \sum_{j \neq i} \|\boldsymbol{M}_{ij}\| \qquad \text{where } i = 1, 2, \ldots, c.$$

The derivation is entirely analogous with the usual proof of Gershgorin's theorem, so we omit the details. For each diagonal block, we have $\|\boldsymbol{M}_{ii}\| \leq \delta_r$ because of the restricted isometry inequalities (1.1). For each off-diagonal block, we have $\|\boldsymbol{M}_{ij}\| \leq \delta_{2r}$ because of Proposition 3.2. Substitute these bounds into the block Gershgorin theorem and rearrange to complete the proof. $\square$

Finally, we present a result that measures how much the sampling matrix inflates nonsparse vectors. This bound permits us to establish the major results for sparse signals and then transfer the conclusions to the general case.

Proposition 3.5 (Energy Bound). *Suppose that* $\boldsymbol{\Phi}$ *verifies the upper inequality of* (1.1)*, viz.*

$$\|\boldsymbol{\Phi}\boldsymbol{x}\|_2 \leq \sqrt{1+\delta_r}\,\|\boldsymbol{x}\|_2 \qquad when \qquad \|\boldsymbol{x}\|_0 \leq r.$$

Then, for every signal $\boldsymbol{x}$,

$$\|\boldsymbol{\Phi}\boldsymbol{x}\|_2 \leq \sqrt{1+\delta_r}\left[\|\boldsymbol{x}\|_2 + \frac{1}{\sqrt{r}}\,\|\boldsymbol{x}\|_1\right].$$

Proof. We repeat the geometric argument of Rudelson that is presented in [17].

First, observe that the hypothesis of the proposition can be regarded as a statement about the operator norm of $\boldsymbol{\Phi}$ as a map between two Banach spaces. For a set $I \subset \{1, 2, \ldots, N\}$, write B_2^I for the unit ball in $\ell_2(I)$. Define the convex body

$$S = \operatorname{conv}\left\{\bigcup\nolimits_{|I|\leq r} B_2^I\right\} \subset \mathbb{C}^N,$$

and notice that, by hypothesis, the operator norm

$$\|\boldsymbol{\Phi}\|_{S\to 2} = \max_{\boldsymbol{x}\in S}\|\boldsymbol{\Phi}\boldsymbol{x}\|_2 \leq \sqrt{1+\delta_r}.$$

Define a second convex body

$$K = \left\{\boldsymbol{x} : \|\boldsymbol{x}\|_2 + \frac{1}{\sqrt{r}}\,\|\boldsymbol{x}\|_1 \leq 1\right\} \subset \mathbb{C}^N,$$

and consider the operator norm

$$\|\boldsymbol{\Phi}\|_{K\to 2} = \max_{\boldsymbol{x}\in K}\|\boldsymbol{\Phi}\boldsymbol{x}\|_2.$$

The content of the proposition is the claim that

$$\|\boldsymbol{\Phi}\|_{K\to 2} \leq \|\boldsymbol{\Phi}\|_{S\to 2}.$$

To establish this point, it suffices to check that $K \subset S$.

Instead, we prove the reverse inclusion for the polars: $S^\circ \subset K^\circ$. The norm with unit ball S° is calculated as

$$\|\boldsymbol{u}\|_{S^\circ} = \max_{|I|\leq r}\|\boldsymbol{u}|_I\|_2.$$

Consider a vector $\boldsymbol{u}$ in the unit ball S°, and let I be a set of r coordinates where $\boldsymbol{u}$ is largest in magnitude. We must have

$$\|\boldsymbol{u}|_{I^c}\|_\infty \leq \frac{1}{\sqrt{r}},$$

or else $|u_i| > \frac{1}{\sqrt{r}}$ for each $i \in I$. But then $\|\boldsymbol{u}\|_{S^\circ} \geq \|\boldsymbol{u}|_I\|_2 > 1$, a contradiction. Therefore, we may write

$$\boldsymbol{u} = \boldsymbol{u}|_I + \boldsymbol{u}|_{I^c} \in B_2 + \frac{1}{\sqrt{r}}B_\infty,$$

where B_p is the unit ball in ℓ_p^N. But the set on the right-hand side is precisely the unit ball of K° since

$$\sup\nolimits_{\boldsymbol{v}\in K^\circ}\langle \boldsymbol{x},\ \boldsymbol{v}\rangle = \|\boldsymbol{x}\|_K = \|\boldsymbol{x}\|_2 + \frac{1}{\sqrt{r}}\,\|\boldsymbol{x}\|_1$$

$$= \sup\nolimits_{\boldsymbol{v}\in B_2}\langle \boldsymbol{x},\ \boldsymbol{v}\rangle + \sup\nolimits_{\boldsymbol{w}\in \frac{1}{\sqrt{r}}B_\infty}\langle \boldsymbol{x},\ \boldsymbol{w}\rangle = \sup\nolimits_{\boldsymbol{v}\in B_2 + \frac{1}{\sqrt{r}}B_\infty}\langle \boldsymbol{x},\ \boldsymbol{v}\rangle.$$

In summary, $S^\circ \subset K^\circ$. $\qquad\qquad\square$

4. The Iteration Invariant: Sparse Case

We now commence the proof of Theorem 2.1. For the moment, let us assume that the signal is actually sparse. Section 6 removes this assumption.

The result states that each iteration of the algorithm reduces the approximation error by a constant factor, while adding a small multiple of the noise. As a consequence, when the approximation error is large in comparison with the noise, the algorithm makes substantial progress in identifying the unknown signal.

Theorem 4.1 (Iteration Invariant: Sparse Case). *Assume that $\boldsymbol{x}$ is s-sparse. For each $k \geq 0$, the signal approximation $\boldsymbol{a}^k$ is s-sparse, and*

$$\left\| \boldsymbol{x} - \boldsymbol{a}^{k+1} \right\|_2 \leq 0.5 \left\| \boldsymbol{x} - \boldsymbol{a}^k \right\|_2 + 7.5 \left\| \boldsymbol{e} \right\|_2.$$

In particular,

$$\left\| \boldsymbol{x} - \boldsymbol{a}^k \right\|_2 \leq 2^{-k} \left\| \boldsymbol{x} \right\|_2 + 15 \left\| \boldsymbol{e} \right\|_2.$$

The argument proceeds in a sequence of short lemmas, each corresponding to one step in the algorithm. Throughout this section, we retain the assumption that $\boldsymbol{x}$ is s-sparse.

4.1. Approximations, Residuals, etc.

Fix an iteration $k \geq 1$. We write $\boldsymbol{a} = \boldsymbol{a}^{k-1}$ for the signal approximation at the beginning of the iteration. Define the residual $\boldsymbol{r} = \boldsymbol{x} - \boldsymbol{a}$, which we interpret as the part of the signal we have not yet recovered. Since the approximation $\boldsymbol{a}$ is always s-sparse, the residual $\boldsymbol{r}$ must be $2s$-sparse. Notice that the vector $\boldsymbol{v}$ of updated samples can be viewed as noisy samples of the residual:

$$\boldsymbol{v} \stackrel{\text{def}}{=} \boldsymbol{u} - \boldsymbol{\Phi a} = \boldsymbol{\Phi}(\boldsymbol{x} - \boldsymbol{a}) + \boldsymbol{e} = \boldsymbol{\Phi r} + \boldsymbol{e}.$$

4.2. Identification.

The identification phase produces a set of components where the residual signal still has a lot of energy.

Lemma 4.2 (Identification). *The set $\Omega = \operatorname{supp}(\boldsymbol{y}_{2s})$ contains at most $2s$ indices, and*

$$\left\| \boldsymbol{r}|_{\Omega^c} \right\|_2 \leq 0.2223 \left\| \boldsymbol{r} \right\|_2 + 2.34 \left\| \boldsymbol{e} \right\|_2.$$

Proof. The identification phase forms a proxy $\boldsymbol{y} = \boldsymbol{\Phi}^* \boldsymbol{v}$ for the residual signal. The algorithm then selects a set Ω of $2s$ components from $\boldsymbol{y}$ that have the largest magnitudes. The goal of the proof is to show that the energy in the residual on the set Ω^c is small in comparison with the total energy in the residual.

Define the set $R = \operatorname{supp}(\boldsymbol{r})$. Since R contains at most $2s$ elements, our choice of Ω ensures that $\left\| \boldsymbol{y}|_R \right\|_2 \leq \left\| \boldsymbol{y}|_\Omega \right\|_2$. By squaring this inequality and canceling the terms in $R \cap \Omega$, we discover that

$$\left\| \boldsymbol{y}|_{R \setminus \Omega} \right\|_2 \leq \left\| \boldsymbol{y}|_{\Omega \setminus R} \right\|_2.$$

Since the coordinate subsets here contain few elements, we can use the restricted isometry constants to provide bounds on both sides.

First, observe that the set $\Omega \setminus R$ contains at most $2s$ elements. Therefore, we may apply Proposition 3.1 and Corollary 3.3 to obtain

$$
\begin{aligned}
\left\| \boldsymbol{y}|_{\Omega \setminus R} \right\|_2 &= \left\| \boldsymbol{\Phi}_{\Omega \setminus R}^* (\boldsymbol{\Phi r} + \boldsymbol{e}) \right\|_2 \\
&\leq \left\| \boldsymbol{\Phi}_{\Omega \setminus R}^* \boldsymbol{\Phi r} \right\|_2 + \left\| \boldsymbol{\Phi}_{\Omega \setminus R}^* \boldsymbol{e} \right\|_2 \\
&\leq \delta_{4s} \left\| \boldsymbol{r} \right\|_2 + \sqrt{1 + \delta_{2s}} \left\| \boldsymbol{e} \right\|_2.
\end{aligned}
$$

Likewise, the set $R \setminus \Omega$ contains $2s$ elements or fewer, so Proposition 3.1 and Corollary 3.3 yield

$$\|\boldsymbol{y}|_{R\setminus\Omega}\|_2 = \|\boldsymbol{\Phi}^*_{R\setminus\Omega}(\boldsymbol{\Phi r} + \boldsymbol{e})\|_2$$
$$\geq \|\boldsymbol{\Phi}^*_{R\setminus\Omega}\boldsymbol{\Phi} \cdot \boldsymbol{r}|_{R\setminus\Omega}\|_2 - \|\boldsymbol{\Phi}^*_{R\setminus\Omega}\boldsymbol{\Phi} \cdot \boldsymbol{r}|_\Omega\|_2 - \|\boldsymbol{\Phi}^*_{R\setminus\Omega}\boldsymbol{e}\|_2$$
$$\geq (1 - \delta_{2s})\|\boldsymbol{r}|_{R\setminus\Omega}\|_2 - \delta_{2s}\|\boldsymbol{r}\|_2 - \sqrt{1 + \delta_{2s}}\|\boldsymbol{e}\|_2.$$

Since the residual is supported on R, we can rewrite $\boldsymbol{r}|_{R\setminus\Omega} = \boldsymbol{r}|_{\Omega^c}$. Finally, combine the last three inequalities and rearrange to obtain

$$\|\boldsymbol{r}|_{\Omega^c}\|_2 \leq \frac{(\delta_{2s} + \delta_{4s})\|\boldsymbol{r}\|_2 + 2\sqrt{1 + \delta_{2s}}\|\boldsymbol{e}\|_2}{1 - \delta_{2s}}.$$

Invoke the numerical hypothesis that $\delta_{2s} \leq \delta_{4s} \leq 0.1$ to complete the argument. $\qquad\square$

4.3. Support Merger. The next step of the algorithm merges the support of the current signal approximation $\boldsymbol{a}$ with the newly identified set of components. The following result shows that components of the signal $\boldsymbol{x}$ outside this set have very little energy.

Lemma 4.3 (Support Merger). *Let Ω be a set of at most $2s$ indices. The set $T = \Omega \cup \mathrm{supp}(\boldsymbol{a})$ contains at most $3s$ indices, and*

$$\|\boldsymbol{x}|_{T^c}\|_2 \leq \|\boldsymbol{r}|_{\Omega^c}\|_2.$$

Proof. Since $\mathrm{supp}(\boldsymbol{a}) \subset T$, we find that

$$\|\boldsymbol{x}|_{T^c}\|_2 = \|(\boldsymbol{x} - \boldsymbol{a})|_{T^c}\|_2 = \|\boldsymbol{r}|_{T^c}\|_2 \leq \|\boldsymbol{r}|_{\Omega^c}\|_2,$$

where the inequality follows from the containment $T^c \subset \Omega^c$. $\qquad\square$

4.4. Estimation. The estimation step of the algorithm solves a least-squares problem to obtain values for the coefficients in the set T. We need a bound on the error of this approximation.

Lemma 4.4 (Estimation). *Let T be a set of at most $3s$ indices, and define the least-squares signal estimate $\boldsymbol{b}$ by the formulae*

$$\boldsymbol{b}|_T = \boldsymbol{\Phi}^\dagger_T \boldsymbol{u} \qquad and \qquad \boldsymbol{b}|_{T^c} = \boldsymbol{0}.$$

Then

$$\|\boldsymbol{x} - \boldsymbol{b}\|_2 \leq 1.112\,\|\boldsymbol{x}|_{T^c}\|_2 + 1.06\,\|\boldsymbol{e}\|_2.$$

This result assumes that we solve the least-squares problem in infinite precision. In practice, the right-hand side of the bound contains an extra term owing to the error from the iterative least-squares solver. In Section 5, we study how many iterations of the least-squares solver are required to make the least-squares error negligible in the present argument.

Proof. Note first that

$$\|\boldsymbol{x} - \boldsymbol{b}\|_2 \leq \|\boldsymbol{x}|_{T^c}\|_2 + \|\boldsymbol{x}|_T - \boldsymbol{b}|_T\|_2.$$

Using the expression $\boldsymbol{u} = \boldsymbol{\Phi x} + \boldsymbol{e}$ and the fact $\boldsymbol{\Phi}^\dagger_T \boldsymbol{\Phi}_T = \mathbf{I}_T$, we calculate that

$$\|\boldsymbol{x}|_T - \boldsymbol{b}|_T\|_2 = \|\boldsymbol{x}|_T - \boldsymbol{\Phi}^\dagger_T(\boldsymbol{\Phi} \cdot \boldsymbol{x}|_T + \boldsymbol{\Phi} \cdot \boldsymbol{x}|_{T^c} + \boldsymbol{e})\|_2$$
$$= \|\boldsymbol{\Phi}^\dagger_T(\boldsymbol{\Phi} \cdot \boldsymbol{x}|_{T^c} + \boldsymbol{e})\|_2$$
$$\leq \|(\boldsymbol{\Phi}^*_T\boldsymbol{\Phi}_T)^{-1}\boldsymbol{\Phi}^*_T\boldsymbol{\Phi} \cdot \boldsymbol{x}|_{T^c}\|_2 + \|\boldsymbol{\Phi}^\dagger_T\boldsymbol{e}\|_2.$$

The cardinality of T is at most $3s$, and $\boldsymbol{x}$ is s-sparse, so Proposition 3.1 and Corollary 3.3 imply that

$$\|\boldsymbol{x}|_T - \boldsymbol{b}|_T\|_2 \leq \frac{1}{1 - \delta_{3s}}\|\boldsymbol{\Phi}^*_T\boldsymbol{\Phi} \cdot \boldsymbol{x}|_{T^c}\|_2 + \frac{1}{\sqrt{1 - \delta_{3s}}}\|\boldsymbol{e}\|_2$$
$$\leq \frac{\delta_{4s}}{1 - \delta_{3s}}\|\boldsymbol{x}|_{T^c}\|_2 + \frac{\|\boldsymbol{e}\|_2}{\sqrt{1 - \delta_{3s}}}.$$

Combine the bounds to reach

$$\|\boldsymbol{x} - \boldsymbol{b}\|_2 \leq \left[1 + \frac{\delta_{4s}}{1 - \delta_{3s}}\right] \|\boldsymbol{x}|_{T^c}\|_2 + \frac{\|\boldsymbol{e}\|_2}{\sqrt{1 - \delta_{3s}}}.$$

Finally, invoke the hypothesis that $\delta_{3s} \leq \delta_{4s} \leq 0.1$. $\qquad\square$

4.5. **Pruning.** The final step of each iteration is to prune the intermediate approximation to its largest s terms. The following lemma provides a bound on the error in the pruned approximation.

Lemma 4.5 (Pruning)**.** *The pruned approximation* $\boldsymbol{b}_s$ *satisfies*

$$\|\boldsymbol{x} - \boldsymbol{b}_s\|_2 \leq 2 \|\boldsymbol{x} - \boldsymbol{b}\|_2.$$

Proof. The intuition is that $\boldsymbol{b}_s$ is close to $\boldsymbol{b}$, which is close to $\boldsymbol{x}$. Rigorously,

$$\|\boldsymbol{x} - \boldsymbol{b}_s\|_2 \leq \|\boldsymbol{x} - \boldsymbol{b}\|_2 + \|\boldsymbol{b} - \boldsymbol{b}_s\|_2 \leq 2 \|\boldsymbol{x} - \boldsymbol{b}\|_2.$$

The second inequality holds because $\boldsymbol{b}_s$ is the best s-sparse approximation to $\boldsymbol{b}$. In particular, the s-sparse vector $\boldsymbol{x}$ is a worse approximation. $\qquad\square$

4.6. **Proof of Theorem 4.1.** We now complete the proof of the iteration invariant for sparse signals, Theorem 4.1. At the end of an iteration, the algorithm forms a new approximation $\boldsymbol{a}^k = \boldsymbol{b}_s$, which is evidently s-sparse. Applying the lemmas we have established, we easily bound the error:

$$
\begin{aligned}
\left\|\boldsymbol{x} - \boldsymbol{a}^k\right\|_2 &= \|\boldsymbol{x} - \boldsymbol{b}_s\|_2 \\
&\leq 2 \|\boldsymbol{x} - \boldsymbol{b}\|_2 && \text{Pruning (Lemma 4.5)} \\
&\leq 2 \cdot (1.112 \|\boldsymbol{x}|_{T^c}\|_2 + 1.06 \|\boldsymbol{e}\|_2) && \text{Estimation (Lemma 4.4)} \\
&\leq 2.224 \|\boldsymbol{r}|_{\Omega^c}\|_2 + 2.12 \|\boldsymbol{e}\|_2 && \text{Support merger (Lemma 4.3)} \\
&\leq 2.224 \cdot (0.2223 \|\boldsymbol{r}\|_2 + 2.34 \|\boldsymbol{e}\|_2) + 2.12 \|\boldsymbol{e}\|_2 && \text{Identification (Lemma 4.2)} \\
&< 0.5 \|\boldsymbol{r}\|_2 + 7.5 \|\boldsymbol{e}\|_2 \\
&= 0.5 \left\|\boldsymbol{x} - \boldsymbol{a}^{k-1}\right\|_2 + 7.5 \|\boldsymbol{e}\|_2.
\end{aligned}
$$

To obtain the second bound in Theorem 4.1, simply solve the error recursion and note that

$$(1 + 0.5 + 0.25 + \dots) \cdot 7.5 \|\boldsymbol{e}\|_2 \leq 15 \|\boldsymbol{e}\|_2.$$

This point completes the argument.

5. Analysis of Iterative Least-squares

To develop an efficient implementation of CoSaMP, it is critical to use an iterative method when we solve the least-squares problem in the estimation step. The two natural choices are Richardson's iteration and conjugate gradient. The efficacy of these methods rests on the assumption that the sampling operator has small restricted isometry constants. Indeed, since the set T constructed in the support merger step contains at most $3s$ components, the hypothesis $\delta_{4s} \leq 0.1$ ensures that the condition number

$$\kappa(\boldsymbol{\Phi}_T^* \boldsymbol{\Phi}_T) = \frac{\lambda_{\max}(\boldsymbol{\Phi}_T^* \boldsymbol{\Phi}_T)}{\lambda_{\min}(\boldsymbol{\Phi}_T^* \boldsymbol{\Phi}_T)} \leq \frac{1 + \delta_{3s}}{1 - \delta_{3s}} < 1.223.$$

This condition number is closely connected with the performance of Richardson's iteration and conjugate gradient. In this section, we show that Theorem 4.1 holds if we perform a constant number of iterations of either least-squares algorithm.

5.1. **Richardson's Iteration.** For completeness, let us explain how Richardson's iteration can be applied to solve the least-squares problems that arise in CoSaMP. Suppose we wish to compute $\boldsymbol{A}^\dagger \boldsymbol{u}$ where $\boldsymbol{A}$ is a tall, full-rank matrix. Recalling the definition of the pseudoinverse, we realize that this amounts to solving a linear system of the form

$$(\boldsymbol{A}^* \boldsymbol{A})\boldsymbol{b} = \boldsymbol{A}^* \boldsymbol{u}.$$

This problem can be approached by *splitting* the Gram matrix:

$$\boldsymbol{A}^* \boldsymbol{A} = \mathbf{I} + \boldsymbol{M}$$

where $\boldsymbol{M} = \boldsymbol{A}^* \boldsymbol{A} - \mathbf{I}$. Given an initial iterate $\boldsymbol{z}^0$, Richardon's method produces the subsequent iterates via the formula

$$\boldsymbol{z}^{\ell+1} = \boldsymbol{A}^* \boldsymbol{u} - \boldsymbol{M} \boldsymbol{z}^\ell.$$

Evidently, this iteration requires only matrix–vector multiplies with $\boldsymbol{A}$ and $\boldsymbol{A}^*$. It is worth noting that Richardson's method can be accelerated [1, Sec. 7.2.5], but we omit the details.

It is quite easy to analyze Richardson's iteration [1, Sec. 7.2.1]. Observe that

$$\left\| \boldsymbol{z}^{\ell+1} - \boldsymbol{A}^\dagger \boldsymbol{u} \right\|_2 = \left\| \boldsymbol{M}(\boldsymbol{z}^\ell - \boldsymbol{A}^\dagger \boldsymbol{u}) \right\|_2 \leq \left\| \boldsymbol{M} \right\| \left\| \boldsymbol{z}^\ell - \boldsymbol{A}^\dagger \boldsymbol{u} \right\|_2.$$

This recursion delivers

$$\left\| \boldsymbol{z}^\ell - \boldsymbol{A}^\dagger \boldsymbol{u} \right\|_2 \leq \left\| \boldsymbol{M} \right\|^\ell \left\| \boldsymbol{z}^0 - \boldsymbol{A}^\dagger \boldsymbol{u} \right\|_2 \qquad \text{for } \ell = 0, 1, 2, \dots.$$

In words, the iteration converges linearly.

In our setting, $\boldsymbol{A} = \boldsymbol{\Phi}_T$ where T is a set of at most $3s$ indices. Therefore, the restricted isometry inequalities (1.1) imply that

$$\left\| \boldsymbol{M} \right\| = \left\| \boldsymbol{\Phi}_T^* \boldsymbol{\Phi}_T - \mathbf{I} \right\| \leq \delta_{3s}.$$

We have assumed that $\delta_{3s} \leq \delta_{4s} \leq 0.1$, which means that the iteration converges quite fast. Once again, the restricted isometry behavior of the sampling matrix plays an essential role in the performance of the CoSaMP algorithm.

Conjugate gradient provides even better guarantees for solving the least-squares problem, but it is somewhat more complicated to describe and rather more difficult to analyze. We refer the reader to [1, Sec. 7.4] for more information. The following lemma summarizes the behavior of both Richardson's iteration and conjugate gradient in our setting.

Lemma 5.1 (Error Bound for LS)**.** *Richardson's iteration produces a sequence $\{\boldsymbol{z}^\ell\}$ of iterates that satisfy*

$$\left\| \boldsymbol{z}^\ell - \boldsymbol{\Phi}_T^\dagger \boldsymbol{u} \right\|_2 \leq 0.1^\ell \cdot \left\| \boldsymbol{z}^0 - \boldsymbol{\Phi}_T^\dagger \boldsymbol{u} \right\|_2 \qquad \text{for } \ell = 0, 1, 2, \dots.$$

Conjugate gradient produces a sequence of iterates that satisfy

$$\left\| \boldsymbol{z}^\ell - \boldsymbol{\Phi}_T^\dagger \boldsymbol{u} \right\|_2 \leq 2 \cdot \rho^\ell \cdot \left\| \boldsymbol{z}^0 - \boldsymbol{\Phi}_T^\dagger \boldsymbol{u} \right\|_2 \qquad \text{for } \ell = 0, 1, 2, \dots.$$

where

$$\rho = \frac{\sqrt{\kappa(\boldsymbol{\Phi}_T^* \boldsymbol{\Phi}_T)} - 1}{\sqrt{\kappa(\boldsymbol{\Phi}_T^* \boldsymbol{\Phi}_T)} + 1} \leq 0.072.$$

5.2. **Initialization.** Iterative least-squares algorithms must be seeded with an initial iterate, and their performance depends heavily on a wise selection thereof. CoSaMP offers a natural choice for the initializer: the current signal approximation. As the algorithm progresses, the current signal approximation provides an increasingly good starting point for solving the least-squares problem.

Lemma 5.2 (Initial Iterate for LS)**.** *Let $\boldsymbol{x}$ be an s-sparse signal with noisy samples $\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{x} + \boldsymbol{e}$. Let $\boldsymbol{a}^{k-1}$ be the signal approximation at the end of the $(k-1)$th iteration, and let T be the set of components identified by the support merger. Then*

$$\left\| \boldsymbol{a}^{k-1} - \boldsymbol{\Phi}_T^\dagger \boldsymbol{u} \right\|_2 \leq 2.112 \left\| \boldsymbol{x} - \boldsymbol{a}^{k-1} \right\|_2 + 1.06 \left\| \boldsymbol{e} \right\|_2$$

Proof. By construction of T, the approximation $\boldsymbol{a}^{k-1}$ is supported inside T, so

$$\left\|\boldsymbol{x}|_{T^c}\right\|_2 = \left\|(\boldsymbol{x} - \boldsymbol{a}^{k-1})|_{T^c}\right\|_2 \leq \left\|\boldsymbol{x} - \boldsymbol{a}^{k-1}\right\|_2.$$

Using Lemma 4.4, we may calculate how far $\boldsymbol{a}^{k-1}$ lies from the solution to the least-squares problem.

$$\begin{aligned}
\left\|\boldsymbol{a}^{k-1} - \boldsymbol{\Phi}_T^\dagger \boldsymbol{u}\right\|_2 &\leq \left\|\boldsymbol{x} - \boldsymbol{a}^{k-1}\right\|_2 + \left\|\boldsymbol{x} - \boldsymbol{\Phi}_T^\dagger \boldsymbol{u}\right\|_2 \\
&\leq \left\|\boldsymbol{x} - \boldsymbol{a}^{k-1}\right\|_2 + 1.112\left\|\boldsymbol{x}|_{T^c}\right\|_2 + 1.06\left\|\boldsymbol{e}\right\|_2 \\
&\leq 2.112\left\|\boldsymbol{x} - \boldsymbol{a}^{k-1}\right\|_2 + 1.06\left\|\boldsymbol{e}\right\|_2.
\end{aligned}$$

Roughly, the error in the initial iterate is controlled by the current approximation error. $\qquad\square$

5.3. **Iteration Count.** We need to determine how many iterations of the least-squares algorithm are required to ensure that the approximation produced is sufficiently good to support the performance of CoSaMP.

Corollary 5.3 (Estimation by Iterative LS). *Suppose that we initialize the LS algorithm with* $\boldsymbol{z}^0 = \boldsymbol{a}^{k-1}$. *After at most three iterations, both Richardson's iteration and conjugate gradient produce a signal estimate* $\boldsymbol{b}$ *that satisfies*

$$\left\|\boldsymbol{x} - \boldsymbol{b}\right\|_2 \leq 1.112\left\|\boldsymbol{x}|_{T^c}\right\|_2 + 0.0022\left\|\boldsymbol{x} - \boldsymbol{a}^{k-1}\right\|_2 + 1.062\left\|\boldsymbol{e}\right\|_2.$$

Proof. Combine Lemma 5.1 and Lemma 5.2 to see that three iterations of Richardson's method yield

$$\left\|\boldsymbol{z}^3 - \boldsymbol{\Phi}_T^\dagger \boldsymbol{u}\right\|_2 \leq 0.002112\left\|\boldsymbol{x} - \boldsymbol{a}^{k-1}\right\|_2 + 0.00106\left\|\boldsymbol{e}\right\|_2.$$

The bound for conjugate gradient is slightly better. Let $\boldsymbol{b}|_T = \boldsymbol{z}^3$. According to the estimation result, Lemma 4.4, we have

$$\left\|\boldsymbol{x} - \boldsymbol{\Phi}_T^\dagger \boldsymbol{u}\right\|_2 \leq 1.112\left\|\boldsymbol{x}|_{T^c}\right\|_2 + 1.06\left\|\boldsymbol{e}\right\|_2.$$

An application of the triangle inequality completes the argument. $\qquad\square$

5.4. CoSaMP **with Iterative least-squares.** Finally, we need to check that the sparse iteration invariant, Theorem 4.1 still holds when we use an iterative least-squares algorithm.

Theorem 5.4 (Sparse Iteration Invariant with Iterative LS). *Suppose that we use Richardson's iteration or conjugate gradient for the estimation step, initializing the LS algorithm with the current approximation* $\boldsymbol{a}^{k-1}$ *and performing three LS iterations. Then Theorem 4.1 still holds.*

Proof. We repeat the calculation in Section 4.6 using Corollary 5.3 instead of the simple estimation lemma. To that end, recall that the residual $\boldsymbol{r} = \boldsymbol{x} - \boldsymbol{a}^{k-1}$. Then

$$\begin{aligned}
\left\|\boldsymbol{x} - \boldsymbol{a}^k\right\|_2 &\leq 2\left\|\boldsymbol{x} - \boldsymbol{b}\right\|_2 \\
&\leq 2 \cdot (1.112\left\|\boldsymbol{x}|_{T^c}\right\|_2 + 0.0022\left\|\boldsymbol{r}\right\|_2 + 1.062\left\|\boldsymbol{e}\right\|_2) \\
&\leq 2.224\left\|\boldsymbol{r}|_{\Omega^c}\right\|_2 + 0.0044\left\|\boldsymbol{r}\right\|_2 + 2.124\left\|\boldsymbol{e}\right\|_2 \\
&\leq 2.224 \cdot (0.2223\left\|\boldsymbol{r}\right\|_2 + 2.34\left\|\boldsymbol{e}\right\|_2) + 0.0044\left\|\boldsymbol{r}\right\|_2 + 2.124\left\|\boldsymbol{e}\right\|_2 \\
&< 0.5\left\|\boldsymbol{r}\right\|_2 + 7.5\left\|\boldsymbol{e}\right\|_2 \\
&= 0.5\left\|\boldsymbol{x} - \boldsymbol{a}^{k-1}\right\|_2 + 7.5\left\|\boldsymbol{e}\right\|_2.
\end{aligned}$$

This bound is precisely what is required for the theorem to hold. $\qquad\square$

6. EXTENSION TO GENERAL SIGNALS

In this section, we finally complete the proof of the main result for CoSaMP, Theorem 2.1. The remaining challenge is to remove the hypothesis that the target signal is sparse, which we framed in Theorems 4.1 and 5.4. Although this difficulty might seem large, the solution is simple and elegant. It turns out that we can view the noisy samples of a general signal as samples of a sparse signal contaminated with a different noise vector that implicitly reflects the tail of the original signal.

Lemma 6.1 (Reduction to Sparse Case). *Let $\boldsymbol{x}$ be an arbitrary vector in $\mathbb{C}^N$. The sample vector $\boldsymbol{u} = \boldsymbol{\Phi x} + \boldsymbol{e}$ can also be expressed as $\boldsymbol{u} = \boldsymbol{\Phi x}_s + \widetilde{\boldsymbol{e}}$ where*

$$\|\widetilde{\boldsymbol{e}}\|_2 \leq 1.05 \left[\|\boldsymbol{x} - \boldsymbol{x}_s\|_2 + \frac{1}{\sqrt{s}} \|\boldsymbol{x} - \boldsymbol{x}_s\|_1 \right] + \|\boldsymbol{e}\|_2 .$$

Proof. Decompose $\boldsymbol{x} = \boldsymbol{x}_s + (\boldsymbol{x} - \boldsymbol{x}_s)$ to obtain $\boldsymbol{u} = \boldsymbol{\Phi x}_s + \widetilde{\boldsymbol{e}}$ where $\widetilde{\boldsymbol{e}} = \boldsymbol{\Phi}(\boldsymbol{x} - \boldsymbol{x}_s) + \boldsymbol{e}$. To compute the size of the error term, we simply apply the triangle inequality and Proposition 3.5:

$$\|\widetilde{\boldsymbol{e}}\|_2 \leq \sqrt{1 + \delta_s} \left[\|\boldsymbol{x} - \boldsymbol{x}_s\|_2 + \frac{1}{\sqrt{s}} \|\boldsymbol{x} - \boldsymbol{x}_s\|_1 \right] + \|\boldsymbol{e}\|_2 .$$

Finally, invoke the fact that $\delta_s \leq \delta_{4s} \leq 0.1$ to obtain $\sqrt{1 + \delta_s} \leq 1.05$. $\square$

This lemma is just the tool we require to complete the remaining argument.

Proof of Theorem 2.1. Let $\boldsymbol{x}$ be a general signal, and use Lemma 6.1 to write the noisy vector of samples $\boldsymbol{u} = \boldsymbol{\Phi x}_s + \widetilde{\boldsymbol{e}}$. Apply the sparse iteration invariant, Theorem 4.1, or the analog for iterative least-squares, Theorem 5.4. We obtain

$$\left\|\boldsymbol{x}_s - \boldsymbol{a}^{k+1}\right\|_2 \leq 0.5\left\|\boldsymbol{x}_s - \boldsymbol{a}^k\right\|_2 + 7.5\|\widetilde{\boldsymbol{e}}\|_2 .$$

Invoke the lower and upper triangle inequalities to obtain

$$\left\|\boldsymbol{x} - \boldsymbol{a}^{k+1}\right\|_2 \leq 0.5\left\|\boldsymbol{x} - \boldsymbol{a}^k\right\|_2 + 7.5\|\widetilde{\boldsymbol{e}}\|_2 + 1.5\|\boldsymbol{x} - \boldsymbol{x}_s\|_2 .$$

Finally, recall the estimate for $\|\widetilde{\boldsymbol{e}}\|_2$ from Lemma 6.1, and simplify to reach

$$\left\|\boldsymbol{x} - \boldsymbol{a}^{k+1}\right\|_2 \leq 0.5\left\|\boldsymbol{x} - \boldsymbol{a}^k\right\|_2 + 9.375\|\boldsymbol{x} - \boldsymbol{x}_s\|_2 + \frac{7.875}{\sqrt{s}}\|\boldsymbol{x} - \boldsymbol{x}_s\|_1 + 7.5\|\boldsymbol{e}\|_2$$

$$< 0.5\left\|\boldsymbol{x} - \boldsymbol{a}^k\right\|_2 + 10\nu.$$

where ν is the unrecoverable energy (2.1). $\square$

7. DISCUSSION AND RELATED WORK

CoSaMP draws on both algorithmic ideas and analytic techniques that have appeared before. This section describes the other major signal recovery algorithms, and it compares them with CoSaMP. It also attempts to trace the key ideas in the algorithm back to their sources.

7.1. **Algorithms for Compressive Sampling.** We begin with a short discussion of the major algorithmic approaches to signal recovery from compressive samples. We focus on provably correct methods, although we acknowledge that some *ad hoc* techniques provide excellent empirical results.

The initial discovery works on compressive sampling proposed to perform signal recovery by solving a convex optimization problem [2, 12]. Given a sampling matrix $\boldsymbol{\Phi}$ and a noisy vector of samples $\boldsymbol{u} = \boldsymbol{\Phi x} + \boldsymbol{e}$, consider the mathematical program

$$\min \|\boldsymbol{y}\|_1 \qquad \text{subject to} \qquad \boldsymbol{\Phi y} = \boldsymbol{u}. \tag{7.1}$$

In words, we look for a signal reconstruction that is consistent with the samples but has minimal ℓ_1 norm. The intuition behind this approach is that minimizing the ℓ_1 norm promotes sparsity, so

allows the approximate recovery of compressible signals. Candès, Romberg, and Tao established in [3] that a minimizer $\boldsymbol{a}$ of (7.1) satisfies

$$\|\boldsymbol{x} - \boldsymbol{a}\|_2 \leq \mathrm{C}\left[\frac{1}{\sqrt{s}}\|\boldsymbol{x} - \boldsymbol{x}_s\|_1 + \|\boldsymbol{e}\|_2\right] \tag{7.2}$$

provided that the sampling matrix $\boldsymbol{\Phi}$ has restricted isometry constant $\delta_{4s} \leq 0.2$. In [4], the hypothesis on the restricted isometry constant is sharpened to $\delta_{2s} \leq \sqrt{2} - 1$. The error bound for CoSaMP is equivalent, modulo the exact value of the constants.

The literature describes a huge variety of algorithms for solving the optimization problem (7.1). The most common approaches involve interior-point methods [2, 24], projected gradient methods [14], or iterative thresholding [11] The interior-point methods are guaranteed to solve the problem to a fixed precision in time $\mathrm{O}(m^2 N^{1.5})$, where m is the number of measurements and N is the signal length [29]. Note that the constant in the big-O notation depends on some of the problem data. The other convex relaxation algorithms, while sometimes faster in practice, do not currently offer rigorous guarantees. CoSaMP provides rigorous bounds on the runtime that are much better than the available results for interior-point methods.

Tropp and Gilbert proposed the use of a greedy iterative algorithm called *orthogonal matching pursuit* (OMP) for signal recovery [36]. The algorithm initializes the current sample vector $\boldsymbol{v} = \boldsymbol{u}$. In each iteration, it forms the signal proxy $\boldsymbol{y} = \boldsymbol{\Phi}^*\boldsymbol{v}$ and identifies a component of the proxy with largest magnitude. It adds the new component to the set T of previously identified components. Then OMP forms a new signal approximation by solving a least-squares problem: $\boldsymbol{a} = \boldsymbol{\Phi}_T^\dagger \boldsymbol{u}$. Finally, it updates the samples $\boldsymbol{v} = \boldsymbol{u} - \boldsymbol{\Phi}\boldsymbol{a}$. These steps are repeated until a halting criterion is satisfied.

Tropp and Gilbert were able to prove a weak result for the performance of OMP [36]. Suppose that $\boldsymbol{x}$ is a fixed, s-sparse signal, and let $m = \mathrm{C}s \log N$. Draw an $m \times N$ sampling matrix $\boldsymbol{\Phi}$ whose entries are independent, zero-mean subgaussian[2] random variables with equal variances. Given noiseless measurements $\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{x}$, OMP reconstructs $\boldsymbol{x}$ after s iterations, except with probability N^{-1}. In this setting, OMP must fail for some sparse signals, so it does not provide the same uniform guarantees as convex relaxation. It is unknown whether OMP succeeds for compressible signals or whether it succeeds when the samples are contaminated with noise.

Donoho et al. invented another greedy iterative method called *stagewise OMP*, or StOMP [13]. This algorithm uses the signal proxy to select multiple components at each step, using a rule inspired by ideas from wireless communications. The algorithm is faster than OMP because of the selection rule, and it sometimes provides good performance, although parameter tuning can be difficult. There are no rigorous results available for StOMP.

Very recently, Needell and Vershynin developed and analyzed another greedy approach, called *regularized OMP*, or ROMP [28, 27]. This algorithm is similar to OMP but uses a more sophisticated selection rule. Among the s largest entries of the signal proxy, it identifies the largest subset whose entries differ in magnitude by at most a factor of two. The work on ROMP represents an advance because the authors establish under restricted isometry hypotheses that their algorithm can approximately recover any compressible signal from noisy samples. More precisely, suppose that the sampling matrix $\boldsymbol{\Phi}$ has restricted isometry constant $\delta_{8s} \leq 0.01/\sqrt{\log s}$. Given noisy samples $\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{x} + \boldsymbol{e}$, ROMP produces a $2s$-sparse signal approximation $\boldsymbol{a}$ that satisfies

$$\|\boldsymbol{x} - \boldsymbol{a}\|_2 \leq \mathrm{C}\sqrt{\log s}\left[\frac{1}{\sqrt{s}}\|\boldsymbol{x} - \boldsymbol{x}_s\|_1 + \|\boldsymbol{e}\|_2\right].$$

This result is comparable with the result for convex relaxation, aside from the extra logarithmic factor in the restricted isometry hypothesis and the error bound. The results for CoSaMP show that it does not suffer these parasitic factors, so its performance is essentially optimal.

---

[2]A subgaussian random variable Z satisfies $\mathbb{P}\{|Z| > t\} \leq \mathrm{c}e^{-ct^2}$ for all $t > 0$.

After we initially presented this work, Dai and Milenkovic developed an algorithm called Subspace Pursuit that is very similar to CoSaMP. They established that their algorithm offers performance guarantees analogous with those for CoSaMP. See [10] for details.

Finally, we note that there is a class of sublinear algorithms for signal reconstruction from compressive samples. A sublinear algorithm uses time and space resources that are asymptotically smaller than the length of the signal. One of the earliest such techniques is the Fourier sampling algorithm of Gilbert et al. [19, 21]. This algorithm uses random (but structured) time samples to recover signals that are compressible with respect to the discrete Fourier basis. Given $s \operatorname{polylog}(N)$ samples[3], Fourier sampling produces a signal approximation $\boldsymbol{a}$ that satisfies

$$\|\boldsymbol{x} - \boldsymbol{a}\|_2 \leq \mathrm{C} \|\boldsymbol{x} - \boldsymbol{x}_s\|_2$$

except with probability N^{-1}. The result for Fourier sampling holds for each signal (rather than for all). Later, Gilbert et al. developed two other sublinear algorithms, chaining pursuit [16] and HHS pursuit [17], that offer uniform guarantees for all signals. Chaining pursuit has an error bound

$$\|\boldsymbol{x} - \boldsymbol{a}\|_1 \leq \mathrm{C} \log N \|\boldsymbol{x} - \boldsymbol{x}_s\|_1$$

which is somewhat worse than (7.2). HHS pursuit achieves the error bound (7.2). These methods all require more measurements than the linear and superlinear algorithms (by logarithmic factors), and these measurements must be highly structured. As a result, the sublinear algorithms may not be useful in practice.

The sublinear algorithms are all combinatorial in nature. They use ideas from group testing to identify the support of the signal quickly. There are several other combinatorial signal recovery methods due to Cormode–Muthukrishnan [9] and Iwen [22]. These algorithms have drawbacks similar to the sublinear approaches.

7.2. **Relative Performance.** Table 2 summarizes the relative behavior of these algorithms in terms of the following criteria.

General samples: Does the algorithm work for a variety of sampling schemes? Or does it require structured samples? The designation "RIP" means that a bound on a restricted isometry constant suffices. "Subgauss." means that the algorithm succeeds for the class of subgaussian sampling matrices.

Optimal number of samples: Can the algorithm recover s sparse signals from $\mathrm{O}(s \log N)$ measurements? Or are its sampling requirements higher (by logarithmic factors)?

Uniformity: Does the algorithm recover all signals given a fixed sampling matrix? Or do the results require a sampling matrix to be drawn at random for each signal?

Stability: Does the algorithm succeed when (a) the signal is compressible but not sparse and (b) when the samples are contaminated with noise? In most cases, stable algorithms have error bounds similar to (7.2). See the discussion above for details.

Running time: What is the worst-case cost of the algorithm to recover a real-valued s-sparse signal to a fixed relative precision, given a sampling matrix with no special structure? The designation $\mathrm{LP}(N, m)$ indicates the cost of solving a linear program with N variables and m constraints, which is $\mathrm{O}(m^2 N^{1.5})$ for an interior-point method. Note that most of the algorithms can also take advantage of fast matrix–vector multiplies to obtain better running times.

Of the linear and superlinear algorithms, CoSaMP achieves the best performance on all these metrics. Although CoSaMP is slower than the sublinear algorithms, it makes up for this shortcoming by allowing more general sampling matrices and requiring fewer samples.

---

[3]The term polylog indicates a function that is dominated by a polynomial in the logarithm of its argument.

	CoSaMP	OMP	ROMP	Convex opt.	Fourier Samp.	HHS Pursuit
General samples	RIP	Subgauss.	RIP	RIP	no	no
Opt. # samples	yes	yes	no	yes	no	no
Uniformity	yes	no	yes	yes	no	yes
Stability	yes	?	yes	yes	yes	yes
Running time	$O(mN)$	$O(smN)$	$O(smN)$	$LP(N, m)$	$s\,\text{polylog}(N)$	$\text{poly}(s \log N)$

TABLE 2. Comparison of several signal recovery algorithms. The notation s refers to the sparsity level; m refers the number of measurements; N refers to the signal length. See the text for comments on specific designations.

7.3. Key Ideas.

We conclude with a historical overview of the ideas that inform the CoSaMP algorithm and its analysis.

The overall greedy iterative structure of CoSaMP has a long history. The idea of approaching sparse approximation problems in this manner dates to the earliest algorithms. In particular, methods for variable selection in regression, such as forward selection and its relatives, all take this form [26]. Temlyakov's survey [33] describes the historical role of greedy algorithms in nonlinear approximation. Mallat and Zhang introduced greedy algorithms into the signal processing literature and proposed the name *matching pursuit* [25]. Gilbert, Strauss, and their collaborators showed how to incorporate greedy iterative strategies into fast algorithms for sparse approximation problems, and they established the first rigorous guarantees for greedy methods [18, 19]. Tropp provided a new theoretical analysis of OMP in his work [34]. Subsequently, Tropp and Gilbert proved that OMP was effective for compressive sampling [36].

Unlike the simplest greedy algorithms, CoSaMP identifies many components during each iteration, which allows the algorithm to run faster for many types of signals. It is not entirely clear where this idea first appeared. Several early algorithms of Gilbert et al. incorporate this approach [20, 37], and it is an essential feature of the Fourier sampling algorithm [19, 21]. More recent compressive sampling recovery algorithms also select multiple indices, including chaining pursuit [16], HHS pursuit [17], StOMP [13], and ROMP [28].

CoSaMP uses the restricted isometry properties of the sampling matrix to ensure that the identification step is successful. Candès and Tao isolated the restricted isometry conditions in their work on convex relaxation methods for compressive sampling [5]. The observation that restricted isometries can also be used to ensure the success of greedy methods is relatively new. This idea plays a role in HHS pursuit [17], but it is expressed more completely in the analysis of ROMP [28].

The pruning step of CoSaMP is essential to maintain the sparsity of the approximation, which is what permits us to use restricted isometries in the analysis of the algorithm. It also has significant ramifications for the running time because it impacts the speed of the iterative least-squares algorithms. This technique originally appeared in HHS pursuit [17].

The iteration invariant, Theorem 2.1, states that if the error is large then CoSaMP makes substantial progress. This approach to the overall analysis echoes the analysis of other greedy iterative algorithms, including the Fourier sampling method [19, 21] and HHS Pursuit [17].

Finally, mixed-norm error bounds, such as that in Theorem A, have become an important feature of the compressive sampling literature. This idea appears in the work of Candès–Romberg–Tao on convex relaxation [3]; it is used in the analysis of HHS pursuit [17]; it also plays a role in the theoretical treatment of Cohen–Dahmen–DeVore [7].

Appendix A. Algorithmic Variations

This appendix describes other possible halting criteria and their consequences. It also proposes some other variations on the algorithm.

A.1. **Halting Rules.** There are three natural approaches to halting the algorithm. The first, which we have discussed in the body of the paper, is to stop after a fixed number of iterations. Another possibility is to use the norm $\|v\|_2$ of the current samples as evidence about the norm $\|r\|_2$ of the residual. A third possibility is to use the magnitude $\|y\|_\infty$ of the entries of the proxy to bound the magnitude $\|r\|_\infty$ of the entries of the residual.

It suffices to discuss halting criteria for sparse signals because Lemma 6.1 shows that the general case can be viewed in terms of sampling a sparse signal. Let x be an s-sparse signal, and let a be an s-sparse approximation. The residual $r = x - a$. We write $v = \Phi r + e$ for the induced noisy samples of the residual and $y = \Phi^* v$ for the signal proxy.

The discussion proceeds in two steps. First, we argue that an *a priori* halting criterion will result in a guarantee about the quality of the final signal approximation.

Theorem A.1 (Halting I). *The halting criterion* $\|v\|_2 \le \varepsilon$ *ensures that*

$$\|x - a\|_2 \le 1.06 \cdot (\varepsilon + \|e\|_2).$$

The halting criterion $\|y\|_\infty \le \eta/\sqrt{2s}$ *ensures that*

$$\|x - a\|_\infty \le 1.12\eta + 1.17 \|e\|_2.$$

Proof. Since r is $2s$-sparse, Proposition 3.1 ensures that

$$\sqrt{1 - \delta_{2s}} \|r\|_2 - \|e\|_2 \le \|v\|_2.$$

If $\|v\|_2 \le \varepsilon$, it is immediate that

$$\|r\|_2 \le \frac{\varepsilon + \|e\|_2}{\sqrt{1 - \delta_{2s}}}.$$

The definition $r = x - a$ and the numerical bound $\delta_{2s} \le \delta_{4s} \le 0.1$ dispatch the first claim.

Let $R = \operatorname{supp}(r)$, and note that $|R| \le 2s$. Proposition 3.1 results in

$$(1 - \delta_{2s}) \|r\|_2 - \sqrt{1 + \delta_{2s}} \|e\|_2 \le \|y|_R\|_2.$$

Since

$$\|y|_R\|_2 \le \sqrt{2s} \|y|_R\|_\infty \le \sqrt{2s} \|y\|_\infty,$$

we find that the requirement $\|y\|_\infty \le \eta/\sqrt{2s}$ ensures that

$$\|r\|_\infty \le \frac{\eta + \sqrt{1 + \delta_{2s}} \|e\|_2}{1 - \delta_{2s}}.$$

The numerical bound $\delta_{2s} \le 0.1$ completes the proof. $\qquad\square$

Second, we check that each halting criterion is triggered when the residual has the desired property.

Theorem A.2 (Halting II). *The halting criterion* $\|v\|_2 \le \varepsilon$ *is triggered as soon as*

$$\|x - a\|_2 \le 0.95 \cdot (\varepsilon - \|e\|_2).$$

The halting criterion $\|y\|_\infty \le \eta/\sqrt{2s}$ *is triggered as soon as*

$$\|x - a\|_\infty \le \frac{0.45\eta}{s} - \frac{0.68 \|e\|_2}{\sqrt{s}}.$$

Proof. Proposition 3.1 shows that

$$\|\boldsymbol{v}\|_2 \leq \sqrt{1 + \delta_{2s}}\,\|\boldsymbol{r}\|_2 + \|\boldsymbol{e}\|_2 \,.$$

Therefore, the condition

$$\|\boldsymbol{r}\|_2 \leq \frac{\varepsilon - \|\boldsymbol{e}\|_2}{\sqrt{1 + \delta_{2s}}}$$

ensures that $\|\boldsymbol{v}\|_2 \leq \varepsilon$. Note that $\delta_{2s} \leq 0.1$ to complete the first part of the argument.

Now let R be the singleton containing the index of a largest-magnitude coefficient of $\boldsymbol{y}$. Proposition 3.1 implies that

$$\|\boldsymbol{y}\|_\infty = \|\boldsymbol{y}|_R\|_2 \leq \sqrt{1 + \delta_1}\,\|\boldsymbol{v}\|_2 \,.$$

By the first part of this theorem, the halting criterion $\|\boldsymbol{y}\|_\infty \leq \eta/\sqrt{2s}$ is triggered as soon as

$$\|\boldsymbol{x} - \boldsymbol{a}\|_2 \leq 0.95 \cdot \left(\frac{\eta}{\sqrt{2s}\sqrt{1 + \delta_1}} - \|\boldsymbol{e}\|_2 \right).$$

Since $\boldsymbol{x} - \boldsymbol{a}$ is $2s$-sparse, we have the bound $\|\boldsymbol{x} - \boldsymbol{a}\|_2 \leq \sqrt{2s}\,\|\boldsymbol{x} - \boldsymbol{a}\|_\infty$. To wrap up, recall that $\delta_1 \leq \delta_{2s} \leq 0.1$. □

A.2. **Other Variations.** This section briefly describes several natural variations on CoSaMP that may improve its performance.

(1) Here is a version of the algorithm that is, perhaps, simpler than Algorithm 2.1. At each iteration, we approximate the current residual rather than the entire signal. This approach is similar to HHS Pursuit [17]. The inner loop changes in the following manner.

Identification: As before, select $\Omega = \mathrm{supp}(\boldsymbol{y}_{2s})$.

Estimation: Solve a least-squares problem with the *current samples* instead of the original samples to obtain an approximation of the *residual signal*. Formally, $\boldsymbol{b} = \boldsymbol{\Phi}_\Omega^\dagger \boldsymbol{v}$. In this case, one initializes the iterative least-squares algorithm with the zero vector to take advantage of the fact that the residual is becoming small.

Approximation Merger: Add this approximation of the residual to the previous approximation of the signal to obtain a new approximation of the signal: $\boldsymbol{c} = \boldsymbol{a}^{k-1} + \boldsymbol{b}$.

Pruning: Construct the s-sparse signal approximation: $\boldsymbol{a}^k = \boldsymbol{c}_s$.

Sample Update: Update the samples as before: $\boldsymbol{v} = \boldsymbol{u} - \boldsymbol{\Phi}\boldsymbol{a}^k$.

We can show that this algorithm satisfies a result similar to Theorem 2.1 by adapting the current argument. We were unable to verify an analog of Theorem 2.2. Nevertheless, we believe that this version of the algorithm is also promising.

(2) After the inner loop of the algorithm is complete, we can solve another least-squares problem in an effort to improve the final result. If $\boldsymbol{a}$ is the approximation at the end of the loop, we set $T = \mathrm{supp}(\boldsymbol{a})$. Then solve $\boldsymbol{b} = \boldsymbol{\Phi}_T^\dagger \boldsymbol{u}$ and output the s-sparse signal approximation $\boldsymbol{b}$. Note that the output approximation is not guaranteed to be better than $\boldsymbol{a}$ because of the noise vector $\boldsymbol{e}$, but it should never be much worse.

(3) Another variation is to prune the merged support T down to s entries *before* solving the least-squares problem. One may use the values of the proxy $\boldsymbol{y}$ as surrogates for the unknown values of the new approximation on the set Ω. Since the least-squares problem is solved at the end of the iteration, the columns of $\boldsymbol{\Phi}$ that are used in the least-squares approximation are orthogonal to the current samples $\boldsymbol{v}$. As a result, the identification step always selects new components in each iteration. We have not attempted an analysis of this algorithm.

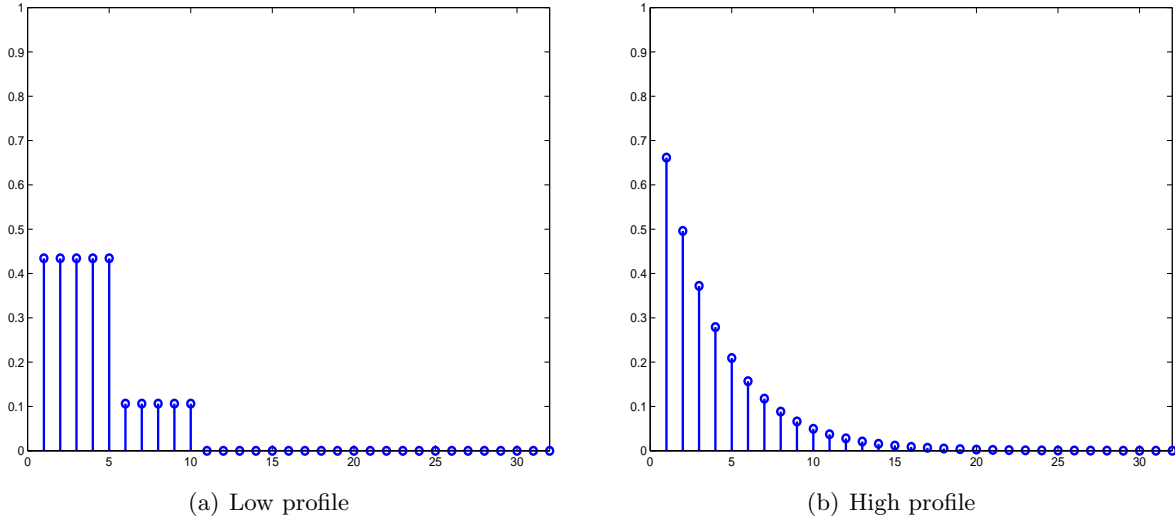(a) Low profile   (b) High profile

FIGURE 1. Illustration of two unit-norm signals with sharply different profiles.

APPENDIX B. ITERATION COUNT

In this appendix, we obtain an estimate on the number of iterations of the CoSaMP algorithm necessary to identify the recoverable energy in a sparse signal, assuming exact arithmetic. Except where stated explicitly, we assume that $\boldsymbol{x}$ is s-sparse. It turns out that the number of iterations depends heavily on the signal structure. Let us explain the intuition behind this fact.

When the entries in the signal decay rapidly, the algorithm must identify and remove the largest remaining entry from the residual before it can make further progress on the smaller entries. Indeed, the large component in the residual contaminates each component of the signal proxy. In this case, the algorithm may require an iteration or more to find each component in the signal.

On the other hand, when the s entries of the signal are comparable, the algorithm can simultaneously locate many entries just by reducing the norm of the residual below the magnitude of the smallest entry. Since the largest entry of the signal has magnitude at least $s^{-1/2}$ times the ℓ_2 norm of the signal, the algorithm can find all s components of the signal after about $\log s$ iterations.

To quantify these intuitions, we want to collect the components of the signal into groups that are comparable with each other. To that end, define the *component bands* of a signal $\boldsymbol{x}$ by the formulae

$$B_j \stackrel{\text{def}}{=} \left\{ i : 2^{-(j+1)} \|\boldsymbol{x}\|_2^2 < |x_i|^2 \leq 2^{-j} \|\boldsymbol{x}\|_2^2 \right\} \qquad \text{for } j = 0, 1, 2, \ldots. \tag{B.1}$$

The *profile* of the signal is the number of bands that are nonempty:

$$\text{profile}(\boldsymbol{x}) \stackrel{\text{def}}{=} \#\{j : B_j \neq \emptyset\}.$$

In words, the profile counts how many orders of magnitude at which the signal has coefficients. It is clear that the profile of an s-sparse signal is at most s. See Figure 1 for images of stylized signals with different profiles.

First, we prove a result on the number of iterations needed to acquire an s-sparse signal. At the end of the section, we extend this result to general signals, which yields Theorem 2.2.

Theorem B.1 (Iteration Count: Sparse Case)**.** *Let $\boldsymbol{x}$ be an s-sparse signal, and define $p = \text{profile}(\boldsymbol{x})$. After at most*

$$p \log_{4/3}(1 + 4.6\sqrt{s/p}) + 6$$

iterations, CoSaMP *produces an approximation* $\boldsymbol{a}$ *that satisfies*

$$\|\boldsymbol{x} - \boldsymbol{a}\|_2 \leq 17\,\|\boldsymbol{e}\|_2\,.$$

For a fixed s, the bound on the number of iterations achieves its maximum value at $p = s$. Since $\log_{4/3} 5.6 < 6$, the number of iterations never exceeds $6(s + 1)$.

B.1. **Additional Notation.** Let us instate some notation that will be valuable in the proof of the theorem. We write $p = \mathrm{profile}(\boldsymbol{x})$. For each $k = 0, 1, 2, \ldots$, the signal $\boldsymbol{a}^k$ is the approximation after the kth iteration. We abbreviate $S_k = \mathrm{supp}(\boldsymbol{a}^k)$, and we define the residual signal $\boldsymbol{r}^k = \boldsymbol{x} - \boldsymbol{a}^k$. The norm of the residual can be viewed as the approximation error.

For a nonnegative integer j, we may define an auxiliary signal

$$\boldsymbol{y}^j \stackrel{\mathrm{def}}{=} \boldsymbol{x}|_{\bigcup_{i \geq j} B_i}$$

In other words, $\boldsymbol{y}^j$ is the part of $\boldsymbol{x}$ contained in the bands B_j, B_{j+1}, B_{j+2}, $\ldots$. For each $j \in J$, we have the estimate

$$\left\|\boldsymbol{y}^j\right\|_2^2 \leq \sum_{i \geq j} 2^{-i}\,\|\boldsymbol{x}\|_2^2 \cdot |B_i| \tag{B.2}$$

by definition of the bands. These auxiliary signals play a key role in the analysis.

B.2. **Proof of Theorem B.1.** The proof of the theorem involves a sequence of lemmas. The first object is to establish an alternative that holds in each iteration. One possibility is that the approximation error is small, which means that the algorithm is effectively finished. Otherwise, the approximation error is dominated by the energy in the unidentified part of the signal, and the subsequent approximation error is a constant factor smaller.

Lemma B.2. *For each iteration* $k = 0, 1, 2, \ldots$, *at least one of the following alternatives holds. Either*

$$\left\|\boldsymbol{r}^k\right\|_2 \leq 70\,\|\boldsymbol{e}\|_2 \tag{B.3}$$

or else

$$\left\|\boldsymbol{r}^k\right\|_2 \leq 2.3 \big\|\boldsymbol{x}|_{S_k^c}\big\|_2 \qquad \text{and} \tag{B.4}$$

$$\left\|\boldsymbol{r}^{k+1}\right\|_2 \leq 0.75 \left\|\boldsymbol{r}^k\right\|_2. \tag{B.5}$$

Proof. Define T_k as the merged support that occurs during iteration k. The pruning step ensures that the support S_k of the approximation at the end of the iteration is a subset of the merged support, so

$$\big\|\boldsymbol{x}|_{T_k^c}\big\|_2 \leq \big\|\boldsymbol{x}|_{S_k^c}\big\|_2 \qquad \text{for } k = 1, 2, 3, \ldots.$$

At the end of the kth iteration, the pruned vector $\boldsymbol{b}_s$ becomes the next approximation $\boldsymbol{a}^k$, so the estimation and pruning results, Lemmas 4.4 and 4.5, together imply that

$$\left\|\boldsymbol{r}^k\right\|_2 \leq 2 \cdot (1.112 \big\|\boldsymbol{x}|_{T_k^c}\big\|_2 + 1.06\,\|\boldsymbol{e}\|_2)$$
$$\leq 2.224 \big\|\boldsymbol{x}|_{S_k^c}\big\|_2 + 2.12\,\|\boldsymbol{e}\|_2 \qquad \text{for } k = 1, 2, 3, \ldots. \tag{B.6}$$

Note that the same relation holds trivially for iteration $k = 0$ because $\boldsymbol{r}^0 = \boldsymbol{x}$ and $S_0 = \emptyset$.

Suppose that there is an iteration $k \geq 0$ where

$$\big\|\boldsymbol{x}|_{S_k^c}\big\|_2 < 30\,\|\boldsymbol{e}\|_2\,.$$

We can introduce this bound directly into the inequality (B.6) to obtain the first conclusion (B.3).

Suppose on the contrary that in iteration k we have

$$\big\|\boldsymbol{x}|_{S_k^c}\big\|_2 \geq 30\,\|\boldsymbol{e}\|_2\,.$$

Introducing this relation into the inequality (B.6) leads quickly to the conclusion (B.4). We also have the chain of relations

$$\left\|\boldsymbol{r}^k\right\|_2 \geq \left\|\boldsymbol{r}^k|_{S_k^c}\right\|_2 = \left\|(\boldsymbol{x} - \boldsymbol{a}^k)|_{S_k^c}\right\|_2 = \left\|\boldsymbol{x}|_{S_k^c}\right\|_2 \geq 30 \left\|\boldsymbol{e}\right\|_2.$$

Therefore, the sparse iteration invariant, Theorem 4.1 ensures that (B.5) holds. $\qquad\square$

The next lemma contains the critical part of the argument. Under the second alternative in the previous lemma, we show that the algorithm completely identifies the support of the signal, and we bound the number of iterations required to do so.

Lemma B.3. *Fix* $K = \lfloor p \log_{4/3}(1 + 4.6\sqrt{s/p}) \rfloor$. *Assume that* (B.4) *and* (B.5) *are in force for each iteration* $k = 0, 1, 2, \ldots, K$. *Then* $\operatorname{supp}(\boldsymbol{a}^K) = \operatorname{supp}(\boldsymbol{x})$.

Proof. First, we check that, once the norm of the residual is smaller than each element of a band, the components in that band persist in the support of each subsequent approximation. Define J to be the set of nonempty bands, and fix a band $j \in J$. Suppose that, for some iteration k, the norm of the residual satisfies

$$\left\|\boldsymbol{r}^k\right\|_2 \leq 2^{-(j+1)/2} \left\|\boldsymbol{x}\right\|_2. \tag{B.7}$$

Then it must be the case that $B_j \subset \operatorname{supp}(\boldsymbol{a}^k)$. If not, then some component $i \in B_j$ appears in the residual: $r_i^k = x_i$. This supposition implies that

$$\left\|\boldsymbol{r}^k\right\|_2 \geq |x_i| > 2^{-(j+1)/2} \left\|\boldsymbol{x}\right\|_2,$$

an evident contradiction. Since (B.5) guarantees that the norm of the residual declines in each iteration, (B.7) ensures that the support of each subsequent approximation contains B_j.

Next, we bound the number of iterations required to find the next nonempty band B_j, given that we have already identified the bands B_i where $i < j$. Formally, assume that the support S_k of the current approximation contains B_i for each $i < j$. In particular, the set of missing components $S_k^c \subset \operatorname{supp}(\boldsymbol{y}^j)$. It follows from relation (B.4) that

$$\left\|\boldsymbol{r}^k\right\|_2 \leq 2.3 \left\|\boldsymbol{y}^j\right\|_2.$$

We can conclude that we have identified the band B_j in iteration $k + \ell$ if

$$\left\|\boldsymbol{r}^{k+\ell}\right\|_2 \leq 2^{-(j+1)/2} \left\|\boldsymbol{x}\right\|_2.$$

According to (B.5), we reduce the error by a factor of $\beta^{-1} = 0.75$ during each iteration. Therefore, the number ℓ of iterations required to identify B_j is at most

$$\log_\beta \left\lceil \frac{2.3 \left\|\boldsymbol{y}^j\right\|_2}{2^{-(j+1)/2} \left\|\boldsymbol{x}\right\|_2} \right\rceil$$

We discover that the total number of iterations required to identify all the (nonempty) bands is at most

$$k_\star \stackrel{\text{def}}{=} \sum_{j \in J} \log_\beta \left\lceil 2.3 \cdot \frac{2^{(j+1)/2} \left\|\boldsymbol{y}^j\right\|_2}{\left\|\boldsymbol{x}\right\|_2} \right\rceil.$$

For each iteration $k \geq \lfloor k_\star \rfloor$, it follows that $\operatorname{supp}(\boldsymbol{a}^k) = \operatorname{supp}(\boldsymbol{x})$.

It remains to bound $k_\star$ in terms of the profile p of the signal. For convenience, we focus on a slightly different quantity. First, observe that $p = |J|$. Using the geometric mean–arithmetic mean

inequality, we discover that

$$\exp\left\{\frac{1}{p}\sum_{j\in J}\log\left[2.3\cdot\frac{2^{(j+1)/2}\left\|\boldsymbol{y}^j\right\|_2}{\left\|\boldsymbol{x}\right\|_2}\right]\right\}\leq\exp\left\{\frac{1}{p}\sum_{j\in J}\log\left(1+2.3\cdot\frac{2^{(j+1)/2}\left\|\boldsymbol{y}^j\right\|_2}{\left\|\boldsymbol{x}\right\|_2}\right)\right\}$$

$$\leq\frac{1}{p}\sum_{j\in J}\left(1+2.3\cdot\frac{2^{(j+1)/2}\left\|\boldsymbol{y}^j\right\|_2}{\left\|\boldsymbol{x}\right\|_2}\right)$$

$$=1+\frac{2.3}{p}\sum_{j\in J}\left(\frac{2^{j+1}\left\|\boldsymbol{y}^j\right\|_2^2}{\left\|\boldsymbol{x}\right\|_2^2}\right)^{1/2}.$$

To bound the remaining sum, we recall the relation (B.2). Then we invoke Jensen's inequality and simplify the result.

$$\frac{1}{p}\sum_{j\in J}\left(\frac{2^{j+1}\left\|\boldsymbol{y}^j\right\|_2^2}{\left\|\boldsymbol{x}\right\|_2^2}\right)^{1/2}\leq\frac{1}{p}\sum_{j\in J}\left(2^{j+1}\sum_{i\geq j}2^{-i}\left|B_i\right|\right)^{1/2}\leq\left(\frac{1}{p}\sum_{j\in J}2^{j+1}\sum_{i\geq j}2^{-i}\left|B_i\right|\right)^{1/2}$$

$$\leq\left(\frac{1}{p}\sum_{i\geq0}\left|B_i\right|\sum_{j\leq i}2^{j-i+1}\right)^{1/2}\leq\left(\frac{4}{p}\sum_{i\geq0}\left|B_i\right|\right)^{1/2}$$

$$=2\sqrt{s/p}.$$

The final equality holds because the total number of elements in all the bands equals the signal sparsity s. Combining these bounds, we reach

$$\exp\left\{\frac{1}{p}\sum_{j\in J}\log\left[2.3\cdot\frac{2^{(j+1)/2}\left\|\boldsymbol{y}^j\right\|_2}{\left\|\boldsymbol{x}\right\|_2}\right]\right\}\leq1+4.6\sqrt{s/p}.$$

Take logarithms, multiply by p, and divide through by $\log\beta$. We conclude that the required number of iterations $k_\star$ is bounded as

$$k_\star\leq p\log_\beta(1+4.6\sqrt{s/p}).$$

This is the advertised conclusion. $\square$

Finally, we check that the algorithm produces a small approximation error within a reasonable number of iterations.

Proof of Theorem B.1. Abbreviate $K=\lfloor p\log(1+4.6\sqrt{s/p})\rfloor$. Suppose that (B.3) never holds during the first K iterations of the algorithm. Under this circumstance, Lemma B.2 implies that both (B.4) and (B.5) hold during each of these K iterations. It follows from Lemma B.3 that the support S_K of the Kth approximation equals the support of $\boldsymbol{x}$. Since S_K is contained in the merged support T_K, we see that the vector $\boldsymbol{x}|_{T_K^c}=\boldsymbol{0}$. Therefore, the estimation and pruning results, Lemmas 4.4 and 4.5, show that

$$\left\|\boldsymbol{r}^K\right\|_2\leq2\cdot\left(1.112\left\|\boldsymbol{x}|_{T_K^c}\right\|_2+1.06\left\|\boldsymbol{e}\right\|_2\right)=2.12\left\|\boldsymbol{e}\right\|_2.$$

This estimate contradicts (B.3).

It follows that there is an iteration $k\leq K$ where (B.3) is in force. Repeated applications of the iteration invariant, Theorem 4.1, allow us to conclude that

$$\left\|\boldsymbol{r}^{K+6}\right\|_2<17\left\|\boldsymbol{e}\right\|_2.$$

This point completes the argument. $\square$

B.3. **Proof of Theorem 2.2.** Finally, we extend the sparse iteration count result to the general case.

Theorem B.4 (Iteration Count). *Let $\boldsymbol{x}$ be an arbitrary signal, and define $p = \mathrm{profile}(\boldsymbol{x}_s)$. After at most*

$$p \log_{4/3}(1 + 4.6\sqrt{s/p}) + 6$$

iterations, CoSaMP *produces an approximation $\boldsymbol{a}$ that satisfies*

$$\|\boldsymbol{x} - \boldsymbol{a}\|_2 \le 20 \|\boldsymbol{e}\|_2 .$$

Proof. Let $\boldsymbol{x}$ be a general signal, and let $p = \mathrm{profile}(\boldsymbol{x}_s)$. Lemma 6.1 allows us to write the noisy vector of samples $\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{x}_s + \widetilde{\boldsymbol{e}}$. The sparse iteration count result, Theorem B.1, states that after at most

$$p \log_{4/3}(1 + 4.6\sqrt{s/p}) + 6$$

iterations, the algorithm produces an approximation $\boldsymbol{a}$ that satisfies

$$\|\boldsymbol{x}_s - \boldsymbol{a}\|_2 \le 17 \|\widetilde{\boldsymbol{e}}\|_2 .$$

Apply the lower triangle inequality to the left-hand side. Then recall the estimate for the noise in Lemma 6.1, and simplify to reach

$$\begin{aligned}
\|\boldsymbol{x} - \boldsymbol{a}\|_2 &\le 17 \|\widetilde{\boldsymbol{e}}\|_2 + \|\boldsymbol{x} - \boldsymbol{x}_s\|_2 \\
&\le 18.9 \|\boldsymbol{x} - \boldsymbol{x}_s\|_2 + \frac{17.9}{\sqrt{s}} \|\boldsymbol{x} - \boldsymbol{x}_s\|_1 + 17 \|\boldsymbol{e}\|_2 \\
&< 20\nu,
\end{aligned}$$

where ν is the unrecoverable energy. $\square$

Proof of Theorem 2.2. Invoke Theorem B.4. Recall that the estimate for the number of iterations is maximized with $p = s$, which gives an upper bound of $6(s + 1)$ iterations, independent of the signal. $\square$

References

[1] Å. Björck. *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia, 1996.

[2] E. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete Fourier information. *IEEE Trans. Info. Theory*, 52(2):489–509, Feb. 2006.

[3] E. Candès, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, 2006.

[4] E. J. Candès. The restricted isometry property and its implications for compressed sensing. Submitted for publication, Feb. 2008.

[5] E. J. Candès and T. Tao. Decoding by linear programming. *IEEE Trans. Info. Theory*, 51(12):4203–4215, Dec. 2005.

[6] E. J. Candès and T. Tao. Near optimal signal recovery from random projections: Universal encoding strategies? *IEEE Trans. Info. Theory*, 52(12):5406–5425, Dec. 2006.

[7] A. Cohen, W. Dahmen, and R. DeVore. Compressed sensing and best k-term approximation. Submitted for publication, 2006.

[8] T. Cormen, C. Lesierson, L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, Cambridge, MA, 2nd edition, 2001.

[9] G. Cormode and S. Muthukrishnan. Combinatorial algorithms for compressed sensing. Technical report, DI-MACS, 2005.

[10] W. Dai and O. Milenkovic. Subspace pursuit for compressive sensing: Closing the gap between performance and complexity. Submitted for publication, Mar. 2008.

[11] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm. Pure Appl. Math.*, 57:1413–1457, 2004.

[12] D. L. Donoho. Compressed sensing. *IEEE Trans. Info. Theory*, 52(4):1289–1306, Apr. 2006.

[13] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck. Sparse solution of underdetermined linear equations by stagewise Orthogonal Matching Pursuit (StOMP). Submitted for publication, 2007.

[14] 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. Selected Topics in Signal Processing: Special Issue on Convex Optimization Methods for Signal Processing*, 1(4):586–598, 2007.

[15] A. Garnaev and E. Gluskin. On widths of the Euclidean ball. *Sov. Math. Dokl.*, 30:200–204, 1984.

[16] A. Gilbert, M. Strauss, J. Tropp, and R. Vershynin. Algorithmic linear dimension reduction in the ℓ_1 norm for sparse vectors. Submitted for publication, August 2006.

[17] A. Gilbert, M. Strauss, J. Tropp, and R. Vershynin. One sketch for all: Fast algorithms for compressed sensing. In *Proc. 39th ACM Symp. Theory of Computing*, San Diego, June 2007.

[18] A. C. Gilbert, S. Guha, P. Indyk, Y. Kotidis, S. Muthukrishnan, and M. J. Strauss. Fast, small-space algorithms for approximate histogram maintenance. In *ACM Symposium on Theoretical Computer Science*, 2002.

[19] A. C. Gilbert, S. Guha, P. Indyk, S. Muthukrishnan, and M. J. Strauss. Near-optimal sparse Fourier representations via sampling. In *Proc. of the 2002 ACM Symposium on Theory of Computing STOC*, pages 152–161, 2002.

[20] A. C. Gilbert, M. Muthukrishnan, and M. J. Strauss. Approximation of functions over redundant dictionaries using coherence. In *Proc. of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, Jan. 2003.

[21] A. C. Gilbert, S. Muthukrishnan, and M. J. Strauss. Improved time bounds for near-optimal sparse Fourier representation via sampling. In *Proceedings of SPIE Wavelets XI*, San Diego, CA, 2005.

[22] M. Iwen. A deterministic sub-linear time sparse Fourier algorithm via non-adaptive compressed sensing methods. In *Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2008.

[23] B. Kashin. The widths of certain finite dimensional sets and classes of smooth functions. *Izvestia*, 41:334–351, 1977.

[24] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. A method for large-scale ℓ_1-regularized least-squares problems with applications in signal processing and statistics. Submitted for publication, 2007.

[25] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Trans. Signal Process.*, 41(12):3397–3415, 1993.

[26] A. J. Miller. *Subset Selection in Regression*. Chapman and Hall, London, 2nd edition, 2002.

[27] D. Needell and R. Vershynin. Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit. Submitted for publication, October 2007.

[28] D. Needell and R. Vershynin. Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit. Submitted for publication, July 2007.

[29] Y. E. Nesterov and A. S. Nemirovski. *Interior Point Polynomial Algorithms in Convex Programming*. SIAM, Philadelphia, 1994.

[30] A. Pajor. Personal communication, Feb. 2008.

[31] G. Reeves and M. Gastpar. Sampling bounds for sparse support recovery in the presence of noise. Submitted for publication, Jan. 2008.

[32] M. Rudelson and R. Vershynin. Sparse reconstruction by convex relaxation: Fourier and Gaussian measurements. In *Proc. 40th Annual Conference on Information Sciences and Systems*, Princeton, Mar. 2006.

[33] V. Temlyakov. Nonlinear methods of approximation. *Foundations of Comput. Math.*, 3(1):33–107, 2003.

[34] J. A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Trans. Info. Theory*, 50(10):2231–2242, Oct. 2004.

[35] J. A. Tropp. Beyond Nyquist: Efficient sampling of sparse, bandlimited signals. Presented at SampTA, June 2007.

[36] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Trans. Info. Theory*, 53(12):4655–4666, 2007.

[37] J. A. Tropp, A. C. Gilbert, S. Muthukrishnan, and M. J. Strauss. Improved sparse approximation over quasi-incoherent dictionaries. In *Proc. 2003 IEEE International Conference on Image Processing*, Barcelona, 2003.