

CoSaMP: Iterative Signal Recovery from Incomplete and Inaccurate Samples

By Deanna Needell and Joel A. Tropp

Abstract

Compressive sampling (CoSa) is a new paradigm for developing data sampling technologies. It is based on the principle that many types of vector-space data are *compressible*, which is a term of art in mathematical signal processing. The key ideas are that randomized dimension reduction preserves the information in a compressible signal and that it is possible to develop hardware devices that implement this dimension reduction efficiently. The main computational challenge in CoSa is to reconstruct a compressible signal from the reduced representation acquired by the sampling device. This extended abstract describes a recent algorithm, called CoSaMP, that accomplishes the data recovery task. It was the first known method to offer near-optimal guarantees on resource usage.

1. WHAT IS COMPRESSIVE SAMPLING?

In many applications, the ambient dimension of a data vector does not reflect the actual number of degrees of freedom in that vector. In mathematical signal processing, this property is captured by the notion of a *compressible signal*. Natural images provide a concrete example of compressible signals because we can approximate them accurately just by summarizing the solid areas (local averages) and the edges (local differences). This representation allows us to compress images dramatically without a substantial loss of visual quality—an idea implemented in the JPEG image coders that we use every day.

Sampling is a generic term for the process of acquiring data about an object, either in the physical or the virtual world. There are many different technologies that we use to collect samples. A ubiquitous piece of sampling hardware is the CCD array inside a digital camera, which measures the average light intensity at each small square in the focal plane of the camera. Other sampling equipment includes X-ray machines, magnetic resonance imaging (MRI), and analog-to-digital converters.

In most cases, we use sampling to acquire as much information as possible about an object of interest. Indeed, when purchasing a digital camera, we often think that more pixels are better. But as soon as the image is captured, the camera invokes compression algorithms to reduce the amount of information it needs to store. This fact suggests an awkward question: if there is a limited amount of information in the object, why are we taking so many samples? Is there some way to obtain measurements that automatically sieve out

the information in the object?

One way to exploit the gap between nominal dimension and degrees of freedom is to perform *dimension reduction*. This idea has a long tradition in computer science. Indeed, to solve geometric problems involving high-dimensional data, we often map the data to a lower dimension while approximately preserving the geometric properties that are relevant to the computation. One standard method for achieving this goal is to use a random projection to the data.

The main idea in *compressive sampling* (CoSa) is to develop sampling technologies based on dimension reduction. The mathematical foundation for this approach is the fact that an appropriate random projection of a compressible signal retains most of the information in that signal. The coordinates of this randomly projected signal are interpreted as samples, and the number of samples we need is comparable with the information content of the object we are sampling. Because we have a lot of flexibility when designing projections, we have flexibility in engineering the sampling hardware.

An intriguing example of a CoSa sensing device is the Rice University single-pixel camera. This camera uses a digital micromirror device (DMD) to selectively black out a random subset of pixels from an image and to measure the total intensity of the remaining pixels. By applying many random masks in sequence, we obtain a collection of samples that captures the information in the image.

Another key fact about CoSa is that we cannot use simple linear techniques to reconstruct the data vector from the random samples. Instead, we must develop more sophisticated algorithms. One approach that has received a lot of attention is convex programming based on ℓ_1 minimization. This abstract describes an algorithm, called CoSaMP, that is based on the greedy pursuit schema.^{18, 19} CoSaMP was the first computational technique that provably achieved near-optimal guarantees on resource usage. Using the techniques from our paper, researchers later demonstrated that other algorithms offer comparable performance guarantees. Another algorithm similar to CoSaMP, called Subspace Pursuit,⁹ was published at the same time, but the accompanying analysis was less complete.

The original version of this paper appeared in *Applied and Computational Harmonic Analysis* 26, 3 (2008), 301–321.

The remainder of the article is organized as follows. Section 2 describes the mathematical framework for CoSa as well as the theoretical results for the CoSaMP algorithm. Section 3 presents a description of the algorithm and its implementation. Section 4 discusses related work and a comparison to our method. We conclude in Section 5 with the analysis of the algorithm and sketch the proof of the main result.

2. THE MATHEMATICS OF CoSa

We continue with a description of the mathematical model for signals, sampling technologies, and signal reconstruction algorithms. The main theoretical results for the CoSaMP algorithm appear at the end of the section.

2.1. Notation

Let us instate some notation that is used throughout the paper. A *signal* is a vector $\mathbf{x} \in \mathbb{C}^N$. For $p \in [1, \infty]$, we write $\|\cdot\|_p$ for the usual ℓ_p vector norm: $\|\mathbf{x}\|_p = (\sum x_i^p)^{1/p}$. We reserve $\|\cdot\|$ for the spectral norm. For $\mathbf{x} \in \mathbb{C}^N$, we write \mathbf{x}_r for the signal in \mathbb{C}^N that is formed by restricting \mathbf{x} to its r largest-magnitude components (with ties broken lexicographically).

For $T \subset \{1, 2, \dots, N\}$, we denote the restriction of the signal to the set T as $\mathbf{x}|_T$, which is the vector equal to \mathbf{x} on T and 0 elsewhere. We occasionally abuse the notation and treat $\mathbf{x}|_T$ as an element of \mathbb{C}^T . We also define the restriction Φ_T of the sampling matrix Φ to be the column submatrix whose columns are listed in the index set T . Finally, we define the pseudoinverse of a tall, full-rank matrix \mathbf{A} by the formula $\mathbf{A}^\dagger = (\mathbf{A}^* \mathbf{A})^{-1} \mathbf{A}^*$.

2.2. Sparse and compressible signals

We say that a signal is *s-sparse* when it has $s \ll N$ nonzero entries:

$$\|\mathbf{x}\|_0 := |\text{supp}(\mathbf{x})| := |\{j : x_j \neq 0\}| \leq s.$$

This definition encapsulates the idea that the signal contains far less information than its ambient dimension suggests. Observe that it takes roughly $\log_2 \binom{N}{s} = O(s \log(N/s))$ bits to write down the locations of the nonzero entries in an s -sparse signal in \mathbb{C}^N .

While sparse signals are important for theory, they rarely arise in practice, so we consider the larger class of *compressible signals*. For $p \in (0, 1)$, we say that \mathbf{x} is *p-compressible* with magnitude R if its components taken in sorted order obey

$$|x|_{(i)} \leq R \cdot i^{-1/p} \quad \text{for } i = 1, 2, 3, \dots \quad (1)$$

Compressible signals are well approximated by sparse signals. The smaller the number p , the better the approximation:

$$\begin{aligned} \|\mathbf{x} - \mathbf{x}_s\|_1 &\leq C_p \cdot R \cdot s^{1-1/p} \\ \|\mathbf{x} - \mathbf{x}_s\|_2 &\leq D_p \cdot R \cdot s^{1/2-1/p}. \end{aligned} \quad (2)$$

Let us emphasize that we do not know in advance *which* coordinates of a compressible signal are significant—only that few coordinates matter.

We limit our attention to signals that are sparse or compressible in the standard coordinate basis, although the

ideas apply equally well to signals that are sparse or compressible with respect to other orthonormal bases. The generalization is important in applications. For instance, natural images are usually not compressible in the standard basis but they are compressible in an appropriate wavelet basis.

2.3. Modeling the sampling operation

We can model each of the sampling technologies mentioned in the introduction as a linear map from a signal of interest to a collection of samples. We use an $m \times N$ matrix Φ to describe the sampling operation. Thus, we write $\mathbf{u} = \Phi \mathbf{x}$ to summarize the process of collecting a vector \mathbf{u} of m samples from an N -dimensional signal \mathbf{x} .

Restricted Isometry Constants: In this section, to build intuition, we focus on the class of s -sparse signals. It is important to understand what properties of the sampling matrix Φ ensure that s -sparse signals can be distinguished based on their samples. In other words, we need to make sure that Φ is a linear embedding (i.e., a bijective structure-preserving map) of the set of s -sparse signals into \mathbb{C}^m . A necessary and sufficient condition is that each $2s$ -column submatrix of the sampling matrix Φ forms a linearly independent set. Certain Vandermonde matrices with $m = 2s$ rows, arising from Reed–Solomon codes, have this property. Unfortunately, these matrices contain nearly singular $2s$ -column submatrices, so the sampling matrix is not stably invertible on the image of the set of s -sparse signals. Noise and signal perturbations become a serious problem.

To avoid this difficulty, Candès and Tao⁴ insist that each $2s$ -column submatrix of the measurement matrix should be well conditioned. More formally, they define the r th *restricted isometry constant* of the matrix Φ to be the least number δ_r such that

$$(1 - \delta_r) \|\mathbf{x}\|_2^2 \leq \|\Phi \mathbf{x}\|_2^2 \leq (1 + \delta_r) \|\mathbf{x}\|_2^2 \quad \text{for } \|\mathbf{x}\|_0 \leq r. \quad (3)$$

The condition $\delta_{2s} \ll 1$ implies that the measurement matrix Φ preserves the geometry (with respect to the Euclidean metric) of the set of all s -sparse signals. This condition is sufficient to ensure that sampling is stably invertible. It has become standard practice in CoSa to assume that the measurement matrix satisfies a condition of this type.

How Many Samples?: The next question is *how many* samples m do we need to ensure that the sampling map embeds the set of s -sparse signals into \mathbb{C}^m . Because an s -sparse signal contains only about $s \log(N/s)$ bits of information, we hope that $m = O(s \log(N/s))$ samples suffice for the embedding. In fact, many random sampling schemes allow us to achieve this sampling rate, or one that is just slightly worse. (It is *necessary* in some sense to take at least this many samples, so these sampling schemes are optimal.) Furthermore, there are technologies that implement these sampling schemes. Here are two examples.

A *random sign matrix* has independent, identically distributed entries that take the values ± 1 with equal probability. Its restricted isometry constant⁵ satisfies

$$\delta_{2s} \leq \varepsilon \quad \text{whenever } m \gtrsim \frac{s \log(N/s)}{\varepsilon^2}.$$

In words, we can embed the set of s -sparse signals into m dimensions, where m is proportional to the sparsity and

logarithmic in the ambient dimension. A random sign matrix can be used to construct the random masks required by the Rice single-pixel camera.

The *subsampling Fourier matrix* consists of a collection of m rows chosen uniformly at random from the discrete Fourier transform matrix. The best theoretical bound for the restricted isometry constant²³ is

$$\delta_{2s} \leq \varepsilon \quad \text{whenever} \quad m \gtrsim \frac{s \log^5(N) \cdot \log(\varepsilon^{-1})}{\varepsilon^2}$$

It is conjectured that the actual scaling is $m \gtrsim s \log(N)$. The subsampled Fourier matrix describes measurements that can be acquired by an MRI machine. Note that a subsampled Fourier matrix can be applied to a vector using $O(N \log N)$ arithmetic operations, and it requires only $O(m \log N)$ bits of storage. These computational properties are very important in practice, as we will see.

Certain types of deterministic matrices also satisfy bounds on their restricted isometry constants. At present, all known examples require that the number m of samples satisfy $m \gtrsim s^2$. Randomness seems to provide a significant advantage when designing schemes for CoSa.

2.4. Signal reconstruction via CoSaMP

When the sampling matrix Φ stably embeds the set of s -sparse signals, it is possible in principle to recover an arbitrary s -sparse signal from its samples. The major computational question in CoSa is to determine how to reconstruct the sparse signal *using a tractable algorithm*. For practical applications, we also need to understand how the reconstruction is affected when the signal is merely compressible and the samples are contaminated by noise. The following issues are crucial:

1. Can the algorithm recover the signal under a variety of sampling schemes? In other words, is the class of measurement maps for which the algorithm can succeed large?
2. Can the algorithm reconstruct the signal from a minimal number m of samples?
3. Is signal recovery robust when samples are contaminated with noise or the signal is not exactly sparse? Is the error bound optimal for each target signal?
4. Does the algorithm offer provably efficient computational cost and storage?

This abstract discusses a reconstruction algorithm called Compressive Sampling Matching Pursuit (CoSaMP) that satisfies all of the foregoing criteria. Our original publications^{18,19} were the first to prove that a CoSa signal reconstruction algorithm could achieve essentially optimal resource usage.

Properties of the CoSaMP Algorithm: Let us state and explain our main result. We postpone the details of the algorithm and the analysis until later.

THEOREM A (CoSaMP). *Suppose that Φ is an $m \times N$ sampling matrix with restricted isometry constant $\delta_{2s} \leq c$. Let $\mathbf{u} = \Phi \mathbf{x} + \mathbf{e}$ be a vector of samples of an arbitrary signal, contaminated with arbitrary noise.*

The CoSaMP algorithm takes as input a routine to multiply Φ and Φ^* by a vector, the vector \mathbf{u} of samples, the sparsity parameter s , and a precision parameter η . The output is an s -sparse signal approximation \mathbf{a} that satisfies

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

The running time is $O(\mathcal{L} \cdot \log(\|\mathbf{x}\|_2/\eta))$, where \mathcal{L} bounds the cost of a matrix–vector multiply with Φ or Φ^* . Working storage is $O(N)$. The constants $c < 1$ and $C > 1$ are absolute.

How does this result deliver the desired performance?

1. The algorithm requires only that the matrix satisfy a bound on the restricted isometry constant. Many types of random matrices (and certain deterministic matrices) have this property. As a consequence, the algorithm can be used with many measurement technologies.
2. The number m of samples required to obtain the bound $\delta_{2s} \leq c$ depends on the type of sampling matrix. For many types of random matrices, $m = O(s \log^\alpha N)$ for a small integer α . Since this value is optimal up to the constants, the algorithm can recover signals from a minimal number of samples.
3. The error bound demonstrates that the algorithm succeeds for all signals, even when there is noise in the samples. As we expect, the algorithm performs best for sparse and compressible signals. Indeed, Equation 2 shows that CoSaMP produces an approximation \mathbf{a} of a p -compressible signal \mathbf{x} that satisfies

$$\|\mathbf{x} - \mathbf{a}\|_2 \leq C \cdot \max \left\{ \eta, R \cdot s^{1/2-1/p} + \|\mathbf{e}\|_2 \right\},$$

which is the best possible approximation error we could hope for in general.⁶

4. The computational cost of the algorithm depends on the type of signals we are approximating. For a compressible signal, we may take the precision parameter $\eta = R \cdot s^{1/2-1/p}$ to see that the number of iterations required to produce a minimal approximation error is at most $O(\log s)$. See Section 1.2 of Needell and Tropp¹⁹ for details.

The running time also depends on the type of sampling matrix we use. In particular, we can apply a subsampled Fourier matrix to a vector using $\mathcal{L} = O(N \log N)$ operations.

It follows that if we acquire a compressible signal with a structured sampling matrix, we can perform signal recovery in time $O(N \log^2 N)$. This runtime bound is essentially optimal when the sparsity level is roughly the same order as the dimension (which is the setting in most practical problems).

3. THE CoSaMP ALGORITHM

The CoSaMP algorithm is, at heart, a greedy iterative method for reconstructing a signal from compressive samples. This section provides an overview of the algorithm and its implementation. We also explain the basic idea behind the analysis of the algorithm, which demonstrates that each iteration

reduces the error in the current signal approximation.

3.1. Description of algorithm

The input to the algorithm consists of (access to) the sampling operator Φ , the samples \mathbf{u} , the target sparsity level s , and a halting criterion. We impose the following hypotheses:

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

The algorithm is initialized with a trivial signal estimate, meaning that the initial residual is the entire unknown target signal. Each iteration then consists of five major steps.

1. **Identification.** Using the current samples, the algorithm computes a vector that is highly correlated with the signal, called the signal proxy. From the proxy, components of the signal that carry a lot of energy are located.
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 five steps are repeated until the halting criterion is satisfied. In our analysis, we assume that the method uses a fixed number of iterations, but Needell and Tropp^{18,19} discuss other alternatives. The CoSaMP algorithm can be summarized by the pseudocode presented as Algorithm 1.

REMARK 1. *We emphasize that the method we present is a specific example from a more general framework. The articles^{18,19} discuss a number of variations. In particular, note that the term $2s$ in the identification step can be replaced by αs for other values of α . This type of tuning is valuable in practice, but it makes the proof more complicated.*

REMARK 2. *Some knowledge about the sparsity level is required as input to the algorithm. There are several strategies for estimating this parameter if it is not known a priori. One such method would be to use the number m of measurements to deduce the sparsity level. Since $m \approx 2s \log N$ is often sufficient, the estimate $s \approx m/2 \log N$ is reasonable. A second approach would be to run the CoSaMP algorithm using a variety of sparsity levels and select the best approximation \mathbf{a} obtained using the empirical error $\|\Phi\mathbf{a} - \mathbf{u}\|_2$. If one varies s according to a geometric progression (e.g., $s = 1, 2, 4, 8, \dots, m$), this variation increases the runtime by at most $O(\log m)$. See Appendix A of*

Algorithm 1: CoSaMP Recovery Algorithm

Input: Sampling matrix Φ , noisy sample vector \mathbf{u} , sparsity level s

Output: An s -sparse approximation \mathbf{a} of the target signal

$\mathbf{a}^0 \leftarrow \mathbf{0}, \mathbf{v} \leftarrow \mathbf{u}, k \leftarrow 0$ { Initialization }

repeat

$k \leftarrow k + 1$

$\mathbf{y} \leftarrow \Phi^* \mathbf{v}$ { Form signal proxy }

$W \leftarrow \text{supp}(\mathbf{y}_{2s})$ { Identify large components }

$T \leftarrow W \cup \text{supp}(\mathbf{a}^{k-1})$ { Merge supports }

$\mathbf{b}|_T \leftarrow \Phi_T^\dagger \mathbf{u}$ { Signal estimation }

$\mathbf{b}|_{T^c} \leftarrow \mathbf{0}$

$\mathbf{a}^k \leftarrow \mathbf{b}_s$ { Prune approximation }

$\mathbf{v} \leftarrow \mathbf{u} - \Phi \mathbf{a}^k$ { Update current samples }

until halting criterion *true*

Needell and Tropp¹⁹ for more details.

3.2. Implementation

CoSaMP was designed to be a practical algorithm for signal recovery, so it is worth spending some time on implementation issues. The least-squares step in the algorithm presents the largest contribution to the runtime. Fortunately, we have constructed Φ_T to ensure it is well conditioned, so we can apply its pseudoinverse quickly using an iterative method. Section 5 of Needell and Tropp¹⁹ contains a thorough analysis of iterative least-squares methods as modules in CoSaMP. This analysis shows that the cost of solving the least-squares problem is $O(\mathcal{L})$, where \mathcal{L} bounds the cost of a matrix–vector multiply with Φ_T or Φ_T^* .

The remaining steps of the algorithm involve standard techniques, and we can estimate the operation counts as follows.

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

Identification. We can locate the largest $2s$ entries of a vector in time $O(N)$ using the approach in Cormen et al.^{7, Ch. 9}. In practice, it may be faster to use an $O(N \log N)$ sorting algorithm (e.g., quicksort, mergesort, heapsort^{7, Sec. 11}) on the entries of the signal and then select the first $2s$ of them.

Support Merger. We can merge two support sets of size $O(s)$ in expected time $O(s)$ via randomized hashing methods,^{7, Ch. 11} or by sorting both sets first and using the elementary merge procedure^{7, p. 29} for a total cost $O(s \log s)$.

LS estimation. We can use the conjugate gradient method or the LSQR algorithm (see e.g. Paige and Saunders²²) to compute $\Phi_T^\dagger \mathbf{u}$. Initializing the least-squares algorithm requires a matrix–vector multiply with Φ_T^* , and each iteration requires one matrix–vector multiply each with Φ_T and Φ_T^* . Since Φ_T is a submatrix of Φ , the matrix–vector multiplies can also be obtained from multiplication with the full matrix. We show in Section 5 of Needell and Tropp¹⁹ that a constant number of least-squares iterations suffices for our results to hold.

Pruning. As in the identification step, pruning can be

Table 1. Operation count for CoSaMP

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

implemented in time $O(s)$, but it may be preferable to sort the components of the vector and then select the first s at a cost of $O(s \log s)$.

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

Table 1 summarizes the cost of each step for the cases in which the standard and fast multiply is used.

Finally, we may analyze the storage requirements of the algorithm. Aside from the sampling matrix storage requirement, the algorithm constructs only one vector of length N , the signal proxy. The sample vectors \mathbf{u} and \mathbf{v} have length m , and thus require $O(m)$ storage. Since the signal approximations are sparse, they can be stored using sparse data structures which require at most $O(s \log N)$ storage. Similarly, the index sets that appear require only $O(s \log N)$ storage. The total working storage is therefore $O(N)$.

The following result summarizes these observations.

THEOREM 1 (RESOURCE REQUIREMENTS). *Each iteration of CoSaMP requires $O(\mathcal{L})$ time, where \mathcal{L} bounds the cost of a multiplication with the matrix Φ or Φ^* . The algorithm uses working storage $O(N)$.*

3.3. Error reduction bound

The theoretical analysis of the CoSaMP algorithm is based on an estimate for how much the algorithm reduces the approximation error in each iteration. This result demonstrates that the algorithm makes significant progress whenever the error is substantially larger than a certain baseline value. We call this baseline the *unrecoverable energy* v in the signal:

$$v \stackrel{\text{def}}{=} \|\mathbf{x} - \mathbf{x}_s\|_2 + \frac{1}{\sqrt{s}} \|\mathbf{x} - \mathbf{x}_s\|_1 + \|\mathbf{e}\|_2. \quad (4)$$

The unrecoverable energy is due to the noise in the samples and the fact that the signal is not exactly sparse. For a detailed discussion, see Section 2.5 of Needell and Tropp.¹⁹

The main technical result is the following iteration invariant, which we establish in Section 5.

THEOREM 2 (ERROR REDUCTION). *For each iteration $k \geq 0$, the signal approximation \mathbf{a}^k is s -sparse and*

$$\|\mathbf{x} - \mathbf{a}^{k+1}\|_2 \leq 0.5 \|\mathbf{x} - \mathbf{a}^k\|_2 + 10v.$$

In particular,

$$\|\mathbf{x} - \mathbf{a}^k\|_2 \leq 2^{-k} \|\mathbf{x}\|_2 + 20v.$$

The main consequence of Theorem 2 is the fact that, after $\log_2(\|\mathbf{x}\|_2/\eta)$ iterations, the approximation error is no greater than $\eta + 20v$. See Needell and Tropp^{18,19} for more discussion.

The statement of Theorem A depends on a simple upper bound for the ℓ_2 term in the unrecoverable energy:

$$v \leq \frac{1.71}{\sqrt{s}} \|\mathbf{x} - \mathbf{x}_{s/2}\|_1 + \|\mathbf{e}\|_2. \quad (5)$$

This estimate is a consequence of Lemma 7 of Gilbert.¹⁵

Finally, we can replace the assumption that $\delta_{4s} \leq 0.1$ by a stronger bound on δ_{2s} because of Corollary 3.4 in Needell and Tropp,¹⁹ which states that $\delta_{cr} \leq c \cdot \delta_{2r}$ for any positive integers c and r .

4. RELATED WORK

CoSaMP is inspired by algorithmic ideas and analytic techniques that have appeared previously. We briefly summarize the major algorithmic approaches and compare them with CoSaMP. This work can be classified in three rough categories: convex relaxation,^{4,11} greedy pursuits,^{12,20,24} and combinatorial algorithms.^{8,14-16}

The convex optimization methods^{1,10} attempt to reconstruct signals by solving the mathematical program

$$\min \|\mathbf{y}\|_1 \quad \text{subject to} \quad \|\Phi \mathbf{y} - \mathbf{u}\|_2 \leq \varepsilon, \quad (6)$$

where we assume that $\|\mathbf{e}\|_2 \leq \varepsilon$. The intuition behind minimizing the ℓ_1 norm is that this norm promotes sparsity, and so the solution to this program should approximate a compressible signal well. Candès et al.³ establish that each minimizer \mathbf{a} of Equation 6 satisfies

$$\|\mathbf{x} - \mathbf{a}\|_2 \leq C \left[\frac{1}{\sqrt{s}} \|\mathbf{x} - \mathbf{x}_s\|_1 + \varepsilon \right] \quad (7)$$

whenever Φ has restricted isometry constant $\delta_{4s} \leq 0.2$. These restricted isometry constant requirements continue to be improved.^{2,13} The error bound for CoSaMP is equivalent with Equation 7, up to the precise value of the constants.

Many algorithms have been proposed to optimize Equation 6. In particular, interior-point methods^{1,17} are guaranteed to solve the problem to a fixed precision in $O(m^2N^{1.5})$ time.²¹ The runtime for CoSaMP is much better than these interior-point methods.

Greedy algorithms such as OMP,²⁴ StOMP,¹² and ROMP²⁰ are iterative methods for approximating a signal from compressive samples. In each iteration, they use a greedy rule to identify new elements in the support of the signal. These methods provide fast runtimes. On the other hand, to the extent that their behavior is understood, greedy algorithms require more measurements or produce worse errors than the convex optimization Equation 6.

Tropp and Gilbert proposed the greedy algorithm *orthogonal matching pursuit* (OMP) for signal recovery.²⁴ OMP is similar to CoSaMP in that it uses a signal proxy to select large components of the target signal, but it selects one such component at each iteration. However, it does not provide the same uniform guarantees as convex relaxation, and it is unknown whether it succeeds in the noisy setting.

Other algorithms based on OMP have been formulated, such as *regularized OMP*, or ROMP,²⁰ developed by Needell and Vershynin. ROMP is noteworthy because the authors establish that under the restricted isometry property, the algorithm can approximately recover compressible signals from noisy samples. The error bound and measurement requirements provided by these results are analogous with those of the convex optimization method, modulo parasitic logarithmic terms. Our results for CoSaMP remove all the extraneous factors, so the performance of CoSaMP is essentially optimal.

Finally, we mention that there is a class of algorithms for signal recovery that have sublinear (in the signal dimension) runtime. Some examples are the Fourier sampling algorithm (FS) of Gilbert et al.,¹⁶ chaining pursuit,¹⁴ and HHS pursuit.¹⁵ These methods are very fast, but they require highly structured measurements.

Table 2 summarizes the behavior of the above algorithms in terms of the following five criteria.

General samples. Does the algorithm succeed for a variety of sampling schemes, or does it require structured samples? The designation “RIP” implies that a bound on a restricted isometry constant suffices. “Gauss” means that the algorithm succeeds when the sampling matrix Φ has iid sub-gaussian entries.

Optimal number of samples. Can the algorithm reconstruct s -sparse signals from a near-optimal number of measurements, $m = O(s \log N)$? Or are its sampling requirements higher (by logarithmic factors)?

Uniformity. Given a fixed sampling matrix, does the algorithm recover all signals? Or do the results hold with high probability only for each fixed signal?

Stability. Does the algorithm succeed when the signal is compressible (but not sparse) and the samples are noisy?

Running time. In the worst case (without fast multiplies), how long does it take the algorithm to recover a real-valued s -sparse signal within a fixed relative precision, given a sampling matrix with no special structure? The designation LP(N, m) represents the cost of solving a linear program with N variables and m constraints ($O(m^2 N^{1.5})$ using an interior-point method).

Under all of these metrics, CoSaMP achieves the best performance out of the linear and superlinear methods. Of

Table 2. Comparison of several signal recovery algorithms

	CoSaMP	ROMP	Convex Optimization
General samples	RIP	RIP	RIP
Opt. # samples	Yes	No	Yes
Uniformity	Yes	Yes	Yes
Stability	Yes	Yes	Yes
Running time	$O(mN)$	$O(smN)$	LP(N, m)

	OMP	Fourier Sampling	HHS Pursuit
General samples	Gauss	No	No
Opt. # samples	Yes	No	No
Uniformity	No	No	Yes
Stability	?	Yes	Yes
Running time	$O(smN)$	$s \text{ polylog}(N)$	$\text{poly}(s \log N)$

course, CoSaMP is slower than the sublinear algorithms, but it allows more general sampling matrices and demands fewer samples, which makes it more adaptable to practical applications. CoSaMP delivers the same guarantees as the best convex optimization approaches as well as rigorous bounds on computational cost and storage that are comparable with faster greedy methods. Thus, CoSaMP is essentially optimal in every important respect.

5. PROOF OF RESULTS

The CoSaMP algorithm uses an approach inspired by the restricted isometry property to identify the largest s components of the target signal. Owing to the restricted isometry property, each set of s components of the proxy vector $\mathbf{y} = \Phi^* \Phi \mathbf{x}$ approximates the energy in the corresponding s components of \mathbf{x} . Since the samples are of the form $\mathbf{u} = \Phi \mathbf{x}$, we can obtain the proxy by applying the matrix Φ^* to the samples \mathbf{u} . Once the set T of significant locations is identified, the signal coefficients can be recovered using Φ_T^\dagger .

The algorithm repeats this idea at each iteration, updating the samples to reflect the residual (the part of the signal yet to be approximated). We use least squares to estimate the signal on this support set and repeat this process until the recoverable energy in the signal has been found.

We now outline the proof of Theorem 2.

5.1. Consequences of the RIP

We begin with some important consequences of the restricted isometry property. Omitted proofs appear in Needell and Tropp.¹⁹

PROPOSITION 1 (CONSEQUENCES). *Suppose Φ has restricted isometry constant δ_r . Let T be a set of r indices or fewer. Then*

$$\begin{aligned} \|\Phi_T^* \mathbf{u}\|_2 &\leq \sqrt{1 + \delta_r} \|\mathbf{u}\|_2 \\ \|\Phi_T^\dagger \mathbf{u}\|_2 &\leq \frac{1}{\sqrt{1 - \delta_r}} \|\mathbf{u}\|_2 \\ \|\Phi_T^* \Phi_T \mathbf{x}\|_2 &\preceq (1 \pm \delta_r) \|\mathbf{x}\|_2 \\ \|(\Phi_T^* \Phi_T)^{-1} \mathbf{x}\|_2 &\preceq \frac{1}{1 - \delta_r} \|\mathbf{x}\|_2. \end{aligned}$$

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

A corollary of these bounds is the fact that disjoint sets of columns from the sampling matrix span nearly orthogonal subspaces.

PROPOSITION 2 (ORTHOGONALITY). *Suppose Φ has restricted isometry constant δ_r . Let S and T be disjoint sets of indices whose combined cardinality does not exceed r . Then*

$$\|\Phi_S^* \Phi_T\| \leq \delta_r.$$

We apply this result through a corollary.

COROLLARY 1. *Suppose Φ has restricted isometry constant δ_r . Let T be a set of indices, and let \mathbf{x} be a vector. Provided that*

$$r \geq |T \cup \text{supp}(\mathbf{x})|,$$

$$\|\Phi_r^* \Phi \cdot \mathbf{x}|_{T^c}\|_2 \leq \delta_r \|\mathbf{x}|_{T^c}\|_2.$$

The last consequence of the restricted isometry property that we will need measures how much the sampling matrix inflates non-sparse vectors. Its proof uses arguments from functional analysis.

PROPOSITION 3 (ENERGY BOUND). *Suppose Φ verifies the upper inequality of (3), viz.*

$$\|\Phi \mathbf{x}\|_2 \leq \sqrt{1 + \delta_r} \|\mathbf{x}\|_2 \quad \text{when} \quad \|\mathbf{x}\|_0 \leq r.$$

Then, for every signal \mathbf{x} ,

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

5.2. Iteration invariant: sparse case

In proving Theorem 2, we first operate under the assumption that the signal \mathbf{x} is exactly s -sparse. We remove this assumption later.

THEOREM 3 (SPARSE ERROR REDUCTION). *Assume \mathbf{x} is s -sparse. For each $k \geq 0$, the signal approximation \mathbf{a}^k is s -sparse, and*

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

In particular,

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

REMARK 3. *This bound assumes the least-squares step in the algorithm is performed without error. In Section 5 of Needell and Tropp,¹⁹ we study how many iterations of a least-squares solver are needed to make the resulting error negligible. We show that, if we use conjugate gradient for the estimation step of CoSaMP, then initializing the least-squares algorithm with the current approximation \mathbf{a}^{k-1} , then Theorem 3 still holds.*

The proof of the iteration invariant, Theorem 3 involves a sequence of short lemmas, one for each step in the algorithm. We fix an iteration k , and let $\mathbf{a} = \mathbf{a}^{k-1}$ be the signal approximation at the beginning of the iteration. We define the residual as $\mathbf{r} = \mathbf{x} - \mathbf{a}$, which must be $2s$ -sparse since both \mathbf{a} and \mathbf{x} are s -sparse. We also observe that the vector \mathbf{v} of updated samples can be interpreted as noisy samples of the residual:

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

The first step in the algorithm is the identification step, in which a set of components is found corresponding to locations where the residual signal has a lot of energy. This is summarized by the following lemma which is proven in Needell and Tropp.¹⁹

LEMMA 1 (IDENTIFICATION). *The set $W = \text{supp}(\mathbf{y}_{2s})$, where $\mathbf{y} = \Phi^* \mathbf{v}$ is the signal proxy, contains at most $2s$ indices, and*

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

Next, the algorithm merges the support of the current estimate \mathbf{a} with the new selection of components. The following simple lemma shows that components of the signal \mathbf{x} outside this set have small energy.

LEMMA 2 (SUPPORT MERGER). *Let W be a set of at most $2s$ indices. The set $T = W \cup \text{supp}(\mathbf{a})$ contains at most $3s$ indices, and*

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

The estimation step in the algorithm obtains values for coefficients in the set T by solving a least-squares problem. The next result bounds the error of this approximation.

LEMMA 3 (ESTIMATION¹⁹). *Let T be a set of at most $3s$ indices, and define the least-squares signal estimate \mathbf{b} by the formulae*

$$\mathbf{b}|_T = \Phi_T^\dagger \mathbf{u} \quad \text{and} \quad \mathbf{b}|_{T^c} = \mathbf{0},$$

where $\mathbf{u} = \Phi \mathbf{x} + \mathbf{e}$. Then

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

Proof. Note first that

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

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

$$\begin{aligned} \|\mathbf{x}|_T - \mathbf{b}|_T\|_2 &= \|\mathbf{x}|_T - \Phi_T^\dagger (\Phi \cdot \mathbf{x}|_T + \Phi \cdot \mathbf{x}|_{T^c} + \mathbf{e})\|_2 \\ &\leq \|(\Phi_T^* \Phi_T)^{-1} \Phi_T^* \Phi \cdot \mathbf{x}|_{T^c}\|_2 + \|\Phi_T^\dagger \mathbf{e}\|_2. \end{aligned}$$

The cardinality of T is at most $3s$, and \mathbf{x} is s -sparse, so Proposition 1 and Corollary 1 imply that

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

Combine the bounds to reach

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

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

Lastly, the algorithm prunes the intermediate estimation to its largest s terms. The triangle inequality bounds the error in this pruned approximation.

LEMMA 4 (PRUNING¹⁹). *The pruned approximation \mathbf{b}_s satisfies*

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

At the end of the iteration, the new approximation is formed by setting $\mathbf{a}^k = \mathbf{b}_s$. The sequence of lemmas above allows us to

prove the iteration invariant for the sparse case, Theorem 3. Indeed, we have:

$$\begin{aligned} \| \mathbf{x} - \mathbf{a}^k \|_2 &= \| \mathbf{x} - \mathbf{b}_s \|_2 \\ &\leq 2 \| \mathbf{x} - \mathbf{b} \|_2 \\ &\leq 2 \cdot (1.112 \| \mathbf{x} \big|_{7^c} \|_2 + 1.06 \| \mathbf{e} \|_2) \\ &\leq 2.224 \| \mathbf{r} \big|_{\Omega^c} \|_2 + 2.12 \| \mathbf{e} \|_2 \\ &\leq 2.224 \cdot (0.2223 \| \mathbf{r} \|_2 + 2.34 \| \mathbf{e} \|_2) + 2.12 \| \mathbf{e} \|_2 \\ &< 0.5 \| \mathbf{r} \|_2 + 7.5 \| \mathbf{e} \|_2 \\ &= 0.5 \| \mathbf{x} - \mathbf{a}^{k-1} \|_2 + 7.5 \| \mathbf{e} \|_2. \end{aligned}$$

To obtain the second bound in Theorem 3, we solve the error recursion and observe that

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

This point completes the argument.

5.3. Extension to general signals

We now turn to the proof of the main result for CoSaMP, Theorem A. We must remove the assumption that the signal \mathbf{x} is sparse. Although this job seems difficult at first, it turns out to have an elegant solution. We can view the noisy samples of a generic signal as samples of a sparse signal corrupted by additional noise that reflects the tail of the signal. We have the following reduction to the sparse case, of which Theorem 2 is a consequence.

LEMMA 5 (REDUCTION TO SPARSE CASE¹⁹). *Let \mathbf{x} be an arbitrary vector in \mathbb{C}^N . The sample vector $\mathbf{u} = \Phi \mathbf{x} + \mathbf{e}$ can also be expressed as $\mathbf{u} = \Phi \mathbf{x}_s + \tilde{\mathbf{e}}$ where*

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

This reduction will now allow us to prove the iteration invariant for general signals, Theorem 2. Let \mathbf{x} be a general signal. By Lemma 5, we can write the noisy vector of samples as $\mathbf{u} = \Phi \mathbf{x}_s + \tilde{\mathbf{e}}$. Applying the sparse iteration invariant, Theorem 3, we obtain

$$\| \mathbf{x}_s - \mathbf{a}^{k+1} \|_2 \leq 0.5 \| \mathbf{x}_s - \mathbf{a}^k \|_2 + 7.5 \| \tilde{\mathbf{e}} \|_2.$$

We then invoke the lower and upper triangle inequalities to obtain

$$\| \mathbf{x} - \mathbf{a}^{k+1} \|_2 \leq 0.5 \| \mathbf{x} - \mathbf{a}^k \|_2 + 7.5 \| \tilde{\mathbf{e}} \|_2 + 1.5 \| \mathbf{x} - \mathbf{x}_s \|_2.$$

Finally, recalling the estimate for $\| \tilde{\mathbf{e}} \|_2$ from Lemma 5 and simplifying, we reach

$$\begin{aligned} \| \mathbf{x} - \mathbf{a}^{k+1} \|_2 &\leq 0.5 \| \mathbf{x} - \mathbf{a}^k \|_2 + 9.375 \| \mathbf{x} - \mathbf{x}_s \|_2 \\ &\quad + \frac{7.875}{\sqrt{s}} \| \mathbf{x} - \mathbf{x}_s \|_1 + 7.5 \| \mathbf{e} \|_2 \\ &< 0.5 \| \mathbf{x} - \mathbf{a}^k \|_2 + 10\nu. \end{aligned}$$

where ν is the unrecoverable energy (Equation 4). □

References

1. Candès E, Romberg J., Tao T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete Fourier information. *IEEE Trans. Info. Theory* 52, 2 (Feb. 2006), 489–509.
2. Candès, E.J. The restricted isometry property and its implications for compressed sensing. *C. R. Math. Acad. Sci. Paris, Serie I* 346 (2008), U589–U592.
3. Candès, E.J., Romberg, J., Tao, T. Stable signal recovery from incomplete and inaccurate measurements. *Commun. Pur. Appl. Math.* 59, 8 (2006), 1207–1223.
4. Candès, E.J., Tao, T. Decoding by linear programming. *IEEE Trans. Info. Theory* 51 (2005), 4203–4215.
5. Candès, E.J., Tao, T. Near optimal signal recovery from random projections: Universal encoding strategies? *IEEE Trans. Info. Theory* 52, 12 (Dec. 2006), 5406–5425.
6. Cohen, A., Dahmen, W., DeVore, R. Compressed sensing and best k-term approximation. *J. Am. Math. Soc.* 22, 1 (2009) 211–231.
7. Cormen, T., Lesierson, C., Rivest, L., Stein, C. *Introduction to Algorithms*, 2nd edition, MIT Press, Cambridge, MA, 2001.
8. Cormode, G., Muthukrishnan, S. Combinatorial algorithms for compressed sensing. Technical report, DIMACS, 2005.
9. Dai, W., Milenkovic, O. Subspace pursuit for compressive sensing signal reconstruction. *IEEE Trans. Info. Theory* 55, 5 (2009).
10. Donoho, D.L. Compressed sensing. *IEEE Trans. Info. Theory* 52, 4 (Apr. 2006), 1289–1306.
11. Donoho, D.L., Stark, P.B. Uncertainty principles and signal recovery. *SIAM J. Appl. Math.* 49, 3 (June 1989)906–931.
12. Donoho, D.L., Tsai, Y., Drori, I., Starck, J.-L. Sparse solution of underdetermined linear equations by stagewise Orthogonal Matching Pursuit (StOMP). Submitted for publication (2007)
13. Foucart, S. A note on guaranteed sparse recovery via ℓ_1 -minimization. *Appl. Comput. Harmon. Anal.* (2010). To appear.
14. Gilbert, A., Strauss M., Tropp J., Vershynin R. Algorithmic linear dimension reduction in the ℓ_1 norm for sparse vectors. In *Proceedings of the 44th Annual Allerton Conference on Communication, Control and Computing* (Sept. 2006).
15. Gilbert, A., Strauss, M., Tropp, J., Vershynin, R. One sketch for all: fast algorithms for compressed sensing. In *Proceedings of the 39th ACM Symposium. Theory of Computing* (San Diego, June 2007).
16. Gilbert, A.C., Guha, S., Indyk, P., Muthukrishnan, S., Strauss, M.J. Near-optimal sparse Fourier representations via sampling. In *Proceedings of the 2002 ACM Symposium on Theory of Computing STOC* (2002), 152–161.
17. Kim, S.-J., Koh, K., Lustig, M., Boyd, S., Gorinevsky, D. An interior-point method for large-scale ℓ_1 -regularized least squares. *IEEE J. Sel. Top. Signa.* 1, 4 (2007) 606–617.
18. Needell, D., Tropp, J.A. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. ACM Technical Report 2008–01, California Institute of Technology, Pasadena, July 2008.
19. Needell, D., Tropp, J.A. CoSaMP: Iterative signal recovery from noisy samples. *Appl. Comput. Harmon. Anal.* 26, 3 (2008), 301–321.
20. Needell, D., Vershynin, R. Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit. *IEEE J. Sel. Top. Signa.* (2007). To appear.
21. Nesterov, Y.E., Nemirovski, A.S. *Interior Point Polynomial Algorithms in Convex Programming*. SIAM, Philadelphia, 1994.
22. Paige, C.C., Saunders, M.A. LSQR: An algorithm for sparse linear equations and sparse least squares. *TOMS* 8, 1 (1982), 43–71.
23. Rudelson, M., Vershynin, R. Sparse reconstruction by convex relaxation: Fourier and Gaussian measurements. In *Proceedings of the 40th Annual Conference on Info. Sciences and Systems*, Princeton, Mar. 2006.
24. Tropp, J.A., Gilbert, A.C. Signal recovery from random measurements via Orthogonal Matching Pursuit. *IEEE Trans. Info. Theory* 53, 12 (2007), 4655–4666.

Deanna Needell (dneedell@stanford.edu), Stanford University, Stanford, CA.

Joel A. Tropp (jtropp@acm.caltech.edu), California Institute of Technology, Pasadena, CA.