One Sketch for All: Fast Algorithms for Compressed Sensing

A. C. Gilbert, M. J. Strauss, J. A. Tropp
Department of Mathematics
University of Michigan
Ann Arbor, MI 48109
{annacg,martinjs,jtropp}@umich.edu

R. Vershynin
Department of Mathematics
University of California at Davis
Davis, CA 95616
vershynin@math.ucdavis.edu

ABSTRACT

Compressed Sensing is a new paradigm for acquiring the compressible signals that arise in many applications. These signals can be approximated using an amount of information much smaller than the nominal dimension of the signal. Traditional approaches acquire the entire signal and process it to extract the information. The new approach acquires a small number of nonadaptive linear measurements of the signal and uses sophisticated algorithms to determine its information content. Emerging technologies can compute these general linear measurements of a signal at unit cost per measurement.

This paper exhibits a randomized measurement ensemble and a signal reconstruction algorithm that satisfy four requirements:

1. The measurement ensemble succeeds for all signals, with high probability over the random choices in its construction.
2. The number of measurements of the signal is optimal, except for a factor polylogarithmic in the signal length.
3. The running time of the algorithm is polynomial in the amount of information in the signal and polylogarithmic in the signal length.
4. The recovery algorithm offers the strongest possible type of error guarantee. Moreover, it is a fully polynomial approximation scheme with respect to this type of error bound.

Emerging applications demand this level of performance. Yet no other algorithm in the literature simultaneously achieves all four of these desiderata.

Categories and Subject Descriptors

G.4 [**Mathematical Software**]: Algorithm design and analysis

STOC'07, June 11–13, 2007, San Diego, California, USA.

General Terms

Algorithms, Theory

Keywords

Approximation, embedding, group testing, sketching, sparse approximation, sublinear algorithms

1. INTRODUCTION

Compressed Sensing is a new paradigm for acquiring signals, images, and other types of compressible data. These data have the property that they can be approximated using much less information than their nominal dimension would suggest. At present, the standard approach to signal acquisition is to measure a complete copy of the signal and then process it to extract the important information. For example, one typically measures an image in the pixel basis and then applies JPEG compression to obtain a more efficient representation. Instead, the new approach collects a small number of carefully chosen (but nonadaptive) linear measurements that condense the information in the signal. Sophisticated algorithms are used to approximately reconstruct the signal from these measurements.

Some exciting new technological applications are driving the theoretical work on Compressed Sensing. In these applications, it is possible to compute general linear measurements of the signal with unit cost per measurement. Therefore, the acquisition cost is proportional to the number of signal measurements that we take. (This setting stands in contrast with the digital computation of a dot product component by component.)

In traditional signal acquisition models, measurements of the signal have a straightforward interpretation. On the other hand, Compressed Sensing uses measurements that have no real meaning. In particular, there is no simple map from measurement data back to the signal domain. As a result, we are also very concerned about the time it takes to reconstruct signals from measurements.

Scientists and engineers are developing technologies where the computational model of Compressed Sensing applies. They are building cameras [15, 18], analog-to-digital converters [8, 13, 12], and other sensing devices [20, 19] that can obtain a general linear measurement of a signal at unit cost. Compressive imaging cameras use a digital micro-mirror ar-

ray to optically compute inner products of the image with pseudorandom binary patterns. The image is digitally reconstructed from the projections. Popular media have showcased this application. See *Business Week* (Oct. 16, 2006) and *The Economist* (Oct. 28, 2006).

In fact, certain types of Compressed Sensing devices are already widespread, namely CT and MRI scanners. The detector in a Computed Tomography (CT) scanner takes a number of snapshots or profiles of an attenuated X-ray beam as it passes through a patient. The profiles are used to reconstruct a two-dimensional image. Each snapshot of the X-ray beam is, in essence, the line integral of the X-ray beam through the patient (i.e., an inner product).

1.1 Desiderata for Compressed Sensing

Our premise is that, if one measures a highly compressible signal, it is pointless to reconstruct a full-length copy of the signal because it will include a huge number of small, noisy components that bear no information. Instead, a recovery algorithm should directly identify those few components of the signal that are significant. The algorithm should output this compressed representation directly, and its runtime should be roughly proportional to the size of the representation.

Let us be more formal. We are interested in acquiring signals in $\mathbb{R}^d$ that are well approximated by sparse signals with m nonzero components, where $m \ll d$. The measurement process can be represented by an $n \times d$ matrix $\boldsymbol{\Psi}$, where n is roughly proportional to m rather than d. Each signal $\boldsymbol{f}$ yields a sketch $\boldsymbol{v} = \boldsymbol{\Psi}\boldsymbol{f}$. The recovery algorithm uses the sketch and a description of the measurement matrix to construct a signal approximation $\widehat{\boldsymbol{f}}$ that has only $\mathrm{O}(m)$ nonzero components. We want the measurements and the algorithm to satisfy the following properties:

1. One (randomly generated) measurement matrix $\boldsymbol{\Psi}$ is used to measure all signals. With high probability over the random choices in its construction, it must succeed for all signals.
2. The number of measurements is nearly optimal, namely $n = m \operatorname{polylog}(d)$.
3. The algorithm must run in time $\operatorname{poly}(m, \log d)$.
4. Given the sketch of an arbitrary input signal, the algorithm must return a nearly optimal m-term approximation of that signal.

1.2 Our results

We present a linear measurement procedure that takes a near-optimal number of measurements of a signal. We also present HHS Pursuit,[1] a fully polynomial approximation scheme that uses these measurements to construct a sparse estimate of the signal with an optimal error bound. Moreover, this algorithm is exponentially faster than known recovery algorithms that offer equivalent guarantees.

This section states our major theorem and two important corollaries. We establish these results in Section 4 (and the appendices). We discuss the error bounds in Sections 1.3 and 1.4. Section 1.5 provides a comparison with related work.

[1]The initials HHS stand for "Heavy Hitters on Steroids," which reflects the strong demands on the algorithm.

Given a signal $\boldsymbol{f}$, we write $\boldsymbol{f}_m$ to denote the signal obtained by zeroing all the components of $\boldsymbol{f}$ except the m components with largest magnitude. (Break ties lexicographically.) We refer to $\boldsymbol{f}_m$ as the *head* of the signal; it is the best approximation of the signal using at most m terms with respect to any monotonic norm (such as ℓ_p). The vector $\boldsymbol{f} - \boldsymbol{f}_m$ is called the *tail* of the signal since it contains the entries with small magnitude.

THEOREM 1. *Fix an integer m and a number $\varepsilon \in (0, 1)$. With probability at least 0.99, the random measurement matrix $\boldsymbol{\Psi}$ has the following property. Suppose that $\boldsymbol{f}$ is a d-dimensional signal, and let $\boldsymbol{v} = \boldsymbol{\Psi}\boldsymbol{f}$ be the signal sketch. Given m, ε, and $\boldsymbol{v}$, the HHS Pursuit algorithm produces a signal approximation $\widehat{\boldsymbol{f}}$ with $\mathrm{O}(m/\varepsilon^2)$ nonzero entries. The approximation satisfies*

$$\left\|\boldsymbol{f} - \widehat{\boldsymbol{f}}\right\|_2 \leq \frac{\varepsilon}{\sqrt{m}} \left\|\boldsymbol{f} - \boldsymbol{f}_m\right\|_1.$$

The signal sketch has size $(m/\varepsilon^2)\operatorname{polylog}(d/\varepsilon)$, and HHS Pursuit runs in time $(m^2/\varepsilon^4)\operatorname{polylog}(d/\varepsilon)$. The algorithm uses working space $(m/\varepsilon^2)\operatorname{polylog}(d/\varepsilon)$, including storage of the matrix $\boldsymbol{\Psi}$.

In particular, note that the algorithm recovers every m-term signal without error.

The first corollary shows that we can construct an m-term signal approximation whose ℓ_2 error is within an an additive ℓ_1 term of the optimal ℓ_2 error. One can show that this corollary is equivalent with the theorem.

COROLLARY 2. *Let $\widehat{\boldsymbol{f}}_m$ be the best m-term approximation to the output $\widehat{\boldsymbol{f}}$ of HHS Pursuit. Then*

$$\left\|\boldsymbol{f} - \widehat{\boldsymbol{f}}_m\right\|_2 \leq \left\|\boldsymbol{f} - \boldsymbol{f}_m\right\|_2 + \frac{2\varepsilon}{\sqrt{m}} \left\|\boldsymbol{f} - \boldsymbol{f}_m\right\|_1.$$

This result should be compared with Theorem 2 of [2], which gives an analogous bound for the (superlinear) ℓ_1 minimization algorithm. A second corollary provides an ℓ_1 error estimate.

COROLLARY 3. *Let $\widehat{\boldsymbol{f}}_m$ be the best m-term approximation to the output $\widehat{\boldsymbol{f}}$ of HHS Pursuit. Then*

$$\left\|\boldsymbol{f} - \widehat{\boldsymbol{f}}_m\right\|_1 \leq (1 + 3\varepsilon) \left\|\boldsymbol{f} - \boldsymbol{f}_m\right\|_1.$$

The error bound in Corollary 3 is more intuitive but substantially weaker than the bound in Theorem 1. One may check this point by considering a signal whose first component equals $m^{-1/4}$ and whose remaining components equal d^{-1}. The ℓ_1 error bound holds even if an algorithm fails to identify the first signal component, but the mixed-norm error bounds do not.

1.3 Compressible signals

A *compressible signal* has the property that its components decay when sorted by magnitude. These signals arise in numerous applications because one can compress the wavelet and Fourier expansions of certain classes of natural signals [7]. A common measure of compressibility is the weak-ℓ_p norm, which is defined for $0 < p < \infty$ as

$$\|\boldsymbol{f}\|_{w\ell_p} \stackrel{\text{def}}{=} \inf\{r : |f|_{(k)} \leq r \cdot k^{-1/p} \quad \text{for } k = 1, 2, \ldots, d\}.$$

The notation $|f|_{(k)}$ indicates the kth largest magnitude of a signal component. When the weak-ℓ_p norm is small for some $p < 2$, the signal can be approximated efficiently by a sparse signal because

$$\|\boldsymbol{f} - \boldsymbol{f}_m\|_1 \leq m^{1-1/p} \, \|\boldsymbol{f}\|_{w\ell_p}$$

Theorem 1 shows that the computed approximation $\widehat{\boldsymbol{f}}$ satisfies the error bound

$$\left\|\boldsymbol{f} - \widehat{\boldsymbol{f}}\right\|_2 \leq \varepsilon \, m^{1/2-1/p} \, \|\boldsymbol{f}\|_{w\ell_p} .$$

In particular, when $p = 1$, the error decays like $m^{-1/2}$.

1.4 Optimality of error bounds

The error guarantees may look strange at first view. Indeed, one might hope to take $(m/\varepsilon^2)\,\mathrm{polylog}(d)$ measurements of a signal $\boldsymbol{f}$ and produce an m-sparse approximation $\widehat{\boldsymbol{f}}$ that satisfies the error bound

$$\left\|\boldsymbol{f} - \widehat{\boldsymbol{f}}\right\|_2 \leq (1+\varepsilon) \, \|\boldsymbol{f} - \boldsymbol{f}_m\|_2 .$$

It has been established [9] that this guarantee is possible if we construct a random measurement matrix *for each signal*. On the other hand, Cohen, Dahmen, and DeVore have shown [4] that it is impossible to obtain this error bound simultaneously *for all* signals unless the number of measurements is $\Omega(d)$.

The same authors also proved a more general lower bound [4]. For each p in the range $[1, 2)$, it requires $\Omega(m(d/m)^{2-2/p})$ measurements to achieve

$$\left\|\boldsymbol{f} - \widehat{\boldsymbol{f}}\right\|_2 \leq \mathrm{C}_p \, m^{1/2-1/p} \, \|\boldsymbol{f} - \boldsymbol{f}_m\|_p \qquad (1.1)$$

simultaneously for all signals. This result holds for all possible recovery algorithms. It becomes vacuous when $p = 1$, which is precisely the case delineated in Theorem 1.

It is not hard to check that the number of measurements required by our algorithm is within a polylogarithmic factor of the lower bound.

Proposition 4. *Fix p in the range $[1, 2)$. With $m(d/m)^{2-2/p}\,\mathrm{polylog}(d)$ measurements, the HHS Pursuit algorithm produces for every signal $\boldsymbol{f}$ an m-term estimate $\widehat{\boldsymbol{f}}$ such that* (1.1) *holds.*

1.5 Related work

The major difference between our work and other algorithms for Compressed Sensing is that we simultaneously provide (i) a uniform guarantee for all signals, (ii) an optimal error bound, (iii) a near-optimal number of measurements, and (iv) a sublinear running time. We discuss these points in turn and summarize this discussion in Table 1. Some additional comments on this table may help clarify the situation. If the signal is f and the output is $\widehat{f}$, let $E = E(f) = f - \widehat{f}$ denote the error vector of the output and let $E_{\mathrm{opt}} = E_{\mathrm{opt}}(f) = f - f_m$ denote the error vector for the optimal output. Also, let $C_{\mathrm{opt},p}$ denote $\max_g \left\|E_{\mathrm{opt}}(g)\right\|_p$, where g is the worst possible signal in the class where f lives. The two results in [5] refer to the two deterministic constructions which are uniform on a class of functions (noted in the result). $\mathrm{LP}(md)$ denotes resources needed to solve a linear program with $\Theta(md)$ variables, plus minor overhead. We suppress big-O notation for legibility.

First, we focus on the algorithm's guarantees, both a uniform guarantee for all signals and an optimal error bound. Typically, randomized sketches guarantee that "on each signal, with high probability, the algorithm succeeds." When the application involves adaptiveness or iteration, it is much better to have a uniform guarantees of the form "with high probability, on all signals, the algorithm succeeds." Most approaches to Compressed Sensing yield uniform guarantees—exceptions include work on Orthogonal Matching Pursuit (OMP) due to Tropp–Gilbert [16] and the randomized algorithm of Cormode–Muthukrishnan [5] which achieves the strongest error bounds. Our algorithm achieves a uniform bound because, unlike "for each" algorithms, HHS uses a stronger estimation matrix and a combination of sifting and noise reduction matrices (see below) tailored to the mixed-norm bound of Theorem 1. (We include in Table 1 uniform results only.)

Chaining Pursuit is the only algorithm in the literature that achieves the first three desiderata [10]. The error bound in Chaining Pursuit, however, is less than optimal. Not only is this error bound worse than the HHS error bound, but also Chaining Pursuit is not an approximation scheme. Our algoirthm achieves a mixed-norm approximation scheme because, unlike Chaining, HHS uses separate matrices for estimation, sifting, and noise reduction.

Next, we examine the number of measurements. A major selling point for Compressed Sensing is that it uses only $m\,\mathrm{polylog}(d)$ measurements to recover an entire class of compressible signals. Candès–Romberg–Tao [2] and Donoho [6] have shown that a linear programming algorithm achieves this goal. The Chaining Pursuit algorithm of the current authors [10] also has this property. On the other hand, the algorithms of Cormode–Muthukrishnan that yield a uniform guarantee require $\Omega(m^2)$ measurements [5]. The determinism in [5] is an important desideratum not achieved by HHS. Our algorithm manages with only $m\,\mathrm{polylog}(d)$ measurements because, unlike [5], HHS recovers only a fraction of spikes at a time (see below).

Finally, we discuss the running time of the different Compressed Sensing algorithms. The major advantage of our work is that most recovery algorithms for Compressed Sensing have runtimes that are at least linear in the length of the input signal. In particular, the linear programming technique has cost $\Omega(d^{3/2})$. Cormode–Muthukrishnan have developed some sublinear algorithms whose runtimes are comparable with HHS Pursuit [5]. The Chaining Pursuit algorithm has running time $m\,\mathrm{polylog}(d)$, so it is even faster than HHS Pursuit.

1.6 Roadmap

The next three sections give an overview of our approach. Section 2 provides a detailed description of the measurement matrix required by HHS Pursuit. Section 3 states the HHS algorithm, along with implementation details and pseudocode. Section 4 shows how to draw the corollaries from the main theorem, and it explains how the analysis of the algorithm breaks into two cases. The bulk of the proof is deferred to the journal version of this extended abstract.

2. THE MEASUREMENTS

This section describes a random construction of a measurement matrix $\boldsymbol{\Psi}$. Afterward, we explain how to store and apply the matrix efficiently. For clarity, we focus on the case $\varepsilon = 1$. To obtain an approximation scheme, we substitute m/ε^2 for m, which increases the costs by $(1/\varepsilon)^{\mathrm{O}(1)}$.

Approach, Refs	Error bd.	# Meas.	Time
ℓ_1 min. + Gauss [3]	$\|E\|_2 \le m^{-1/2}\|E_{\text{opt}}\|_1$	$m \log(d/m)$	LP(md)
ℓ_1 min. + Fourier	$\|E\|_2 \le m^{-1/2}\|E_{\text{opt}}\|_1$	$m \log^4 d$	$d \log d$ (empirical)
Combinatorial [5]	$\|E\|_2 \le CC_{\text{opt},p}$ $0 < p < 1$	$m^{\frac{3-p}{1-p}} \log^2 d$	$m^{\frac{4-2p}{1-p}} \log^3 d$
Combinatorial [5]	$\|E\|_2 \le CC_{\text{opt},2}$ exp. decay	m^2 polylog d	m^2 polylog d
Chaining Pursuit [10]	$\|E\|_{\text{weak}-1} \le \|E_{\text{opt}}\|_1$	$m \log^2 d$	$m \log^2 d$
HHS (this result)	$\|E\|_2 \le \epsilon \cdot m^{-1/2}\|E_{\text{opt}}\|_1$	(m/ϵ^2) polylog(d/ϵ)	(m^2/ϵ^4) polylog(d/ϵ)

Table 1: Comparison of algorithmic results for compressed sensing

The matrix $\boldsymbol{\Psi}$ consists of two pieces: an identification matrix $\boldsymbol{\Omega}$ and an estimation matrix $\boldsymbol{\Phi}$. We view the matrix as a linear map that acts on a signal $\boldsymbol{f}$ in $\mathbb{R}^d$ to produce a two-part sketch.

$$\boldsymbol{\Psi f} = \begin{bmatrix} \boldsymbol{\Omega} \\ \hline \boldsymbol{\Phi} \end{bmatrix} \boldsymbol{f} = \begin{bmatrix} \boldsymbol{v}_{\text{id}} \\ \hline \boldsymbol{v}_{\text{est}} \end{bmatrix}.$$

The first part of the sketch, $\boldsymbol{v}_{\text{id}} = \boldsymbol{\Omega f}$, is used to identify large components of the signal quickly. The second part, $\boldsymbol{v}_{\text{est}} = \boldsymbol{\Phi f}$, is used to estimate the size of the identified components. Decoupling the identification and estimation steps allows us to produce strong error guarantees.

2.1 Row tensor products

The identification matrix zeroes out many different subsets of the signal components to isolate large components from each other, and then it computes inner products between these restricted signals and a group testing matrix. We construct this restriction map by applying several different restrictions in sequence. We introduce notation for this operation.

If $\boldsymbol{q}$ and $\boldsymbol{r}$ are 0–1 vectors, we can view them as masks that determine which entries of a signal appear and which ones are zeroed out. For example, the signal $\boldsymbol{q} \circ \boldsymbol{f}$ is the signal $\boldsymbol{f}$ restricted the components in $\boldsymbol{q}$ that equal one. (The notation $\circ$ indicates the Hadamard, or componentwise, product.) The sequential restriction by $\boldsymbol{q}$ and $\boldsymbol{r}$ can be written as $(\boldsymbol{r} \circ \boldsymbol{q}) \circ \boldsymbol{f}$. Given 0–1 matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$, we can form a matrix that encodes sequential restrictions by all pairs of their rows. We express this matrix using the *row tensor product*, as in [10, 5].

DEFINITION 5. *Let $\boldsymbol{Q}$ be a $q \times d$ matrix and $\boldsymbol{R}$ an $r \times d$ matrix with rows $\{\boldsymbol{q}_i : 0 \le i < q\}$ and $\{\boldsymbol{r}_k : 0 \le k < r\}$, respectively. The* row tensor product $\boldsymbol{A} = \boldsymbol{Q} \otimes_{\text{r}} \boldsymbol{R}$ *is a $qr \times d$ matrix whose rows are $\{\boldsymbol{q}_i \circ \boldsymbol{r}_k : 0 \le i < q, 0 \le k < r\}$.*

2.2 The identification operator

The identification matrix $\boldsymbol{\Omega}$ is a 0–1 matrix with dimensions $\text{O}(m \log^2(m) \log(d/m) \log^2(d)) \times d$. It consists of a combination of a structured deterministic matrix and ensembles of simple random matrices. Formally, $\boldsymbol{\Omega}$ is the row tensor product $\boldsymbol{\Omega} = \boldsymbol{B} \otimes_{\text{r}} \boldsymbol{A}$. The *bit-test matrix* $\boldsymbol{B}$ has dimensions $\text{O}(\log d) \times d$, and the *isolation matrix* $\boldsymbol{A}$ has dimensions $\text{O}(m \log^2(m) \log(d/m) \log d) \times d$.

2.2.1 The bit-test matrix

The matrix $\boldsymbol{B}$ is a deterministic matrix that contains a row of 1s appended to a 0–1 matrix $\boldsymbol{B}_0$. The matrix $\boldsymbol{B}_0$ has dimensions $\log_2 \lceil d \rceil \times d$. Its kth column contains the binary expansion of k. Therefore, the inner product of the ith row of $\boldsymbol{B}_0$ with a signal $\boldsymbol{f}^T$ sums the components of $\boldsymbol{f}$ that have bit i equal to one. The bit-test matrix with $d = 8$ is

$$\boldsymbol{B} = \left[\begin{array}{cccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{array}\right].$$

In coding theory, $\boldsymbol{B}$ is called the parity check matrix for the extended Hamming code.

2.2.2 The isolation matrix

The isolation matrix $\boldsymbol{A}$ is a randomly constructed 0–1 matrix with a hierarchical structure. It consists of $\text{O}(\log_2 m)$ blocks $\boldsymbol{A}^{(j)}$ labeled by $j = 1, 2, 4, 8, \ldots, J$, where $J = \text{O}(m)$. See Figure 1.

Each block, in turn, has further substructure as a row tensor product of two 0–1 matrices: $\boldsymbol{A}^{(j)} = \boldsymbol{R}^{(j)} \otimes_{\text{r}} \boldsymbol{S}^{(j)}$. The second matrix $\boldsymbol{S}^{(j)}$ is called the *sifting matrix*, and its dimensions are $\text{O}(j \log(d/j)) \times d$. The first matrix $\boldsymbol{R}^{(j)}$ is called the *noise reduction matrix*, and its dimensions are $\text{O}((m/j) \log(m) \log d) \times d$.

2.2.3 The sifting matrix

The purpose of the sifting matrix $\boldsymbol{S}^{(j)}$ is to isolate about j distinguished signal positions from each other. It is a random 0–1 block matrix, as shown in Figure 1. Each submatrix of $\boldsymbol{S}^{(j)}$ has dimensions $\text{O}(j) \times d$, and the number of submatrices is $T_j = \text{O}(\log(d/j))$.

The T_j submatrices are fully independent from each other. Each of the submatrices encodes a $\text{O}(j)$-wise independent random assignment of each signal position to a row. The (i, k) entry of the matrix equals one when the kth signal component is assigned to the ith row. Therefore, with high probability, the componentwise product of the ith row of the matrix with $\boldsymbol{f}$ generates a copy of the signal with $d/\text{O}(j)$ components selected and the others zeroed out. For example,

$$\boldsymbol{S}_t^{(j)} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

This submatrix can also viewed as a random linear hash function from the space of d keys onto a set of $\text{O}(j)$ buckets.

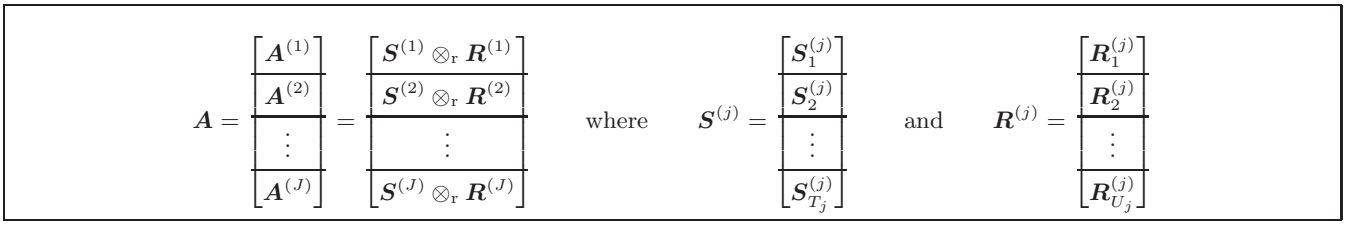

Figure 1: The structure of the isolation matrix $\boldsymbol{A}$. See Section 2.2.2 for details.

2.2.4 The noise reduction matrix

The purpose of the noise reduction matrix $\boldsymbol{R}^{(j)}$ is to attenuate the noise in a signal that has a single large component. It is also a random 0–1 block matrix, as seen in Figure 1. Each submatrix $\boldsymbol{R}_u^{(j)}$ has dimensions $\mathrm{O}(\sqrt{(m/j)\log m}) \times d$, and the total number of submatrices $U_j = \mathrm{O}(\sqrt{(m/j)\log m}\log d)$.

The submatrices are fully independent from each other. Each one encodes a pairwise independent assignment of each signal positions to a row. The (i, k) entry of the matrix equals one when the kth signal component is assigned to the ith row, as in the sifting matrix. Each submatrix can be viewed as a random linear hash function from d keys to $\mathrm{O}(\sqrt{(m/j)\log m})$ buckets.

2.3 The estimation matrix

The estimation matrix $\boldsymbol{\Phi}$ is a randomly constructed matrix with complex entries. Let $\lambda = O(\log^4 d)$ and $L = \mathrm{O}(m\sqrt{\log m})$. Choose $q \geq \lambda L$. The estimation matrix consists of q rows drawn independently at random from the $d \times d$ discrete Fourier transform (DFT) matrix. The matrix $\boldsymbol{\Phi}$ is scaled by $q^{-1/2}$ so its columns have unit ℓ_2 norm.

2.4 Storage costs

The bit test matrix requires no storage as it is straightforward to generate as needed. The isolation and estimation matrices can be generated from short pseudorandom seeds, as needed.

The total storage for the estimation matrix is $q \log(d) = \mathrm{O}(m\sqrt{\log m}\log^5 d)$ because it takes $\log d$ bits to store the index of each of the q rows drawn from the DFT matrix.

The total storage for the isolation matrix $\boldsymbol{A}$ is $\mathrm{O}(m \log^3 d)$, which is negligible compared with the cost of the estimation matrix. To obtain the bound for the isolation matrix, we examine the sifting matrices and the noise reduction matrices separately.

First, observe that each block $\boldsymbol{S}_t^{(j)}$ of the sifting matrix requires $\mathrm{O}(j \log d)$ bits.[2] Since there are $T_j = \mathrm{O}(\log(d/j))$ independent blocks for each j, we have a space bound $\mathrm{O}(j \log^2 d)$ for $\boldsymbol{S}_t^{(j)}$. Summing over $j = 1, 2, 4, \ldots, \mathrm{C}m$, we find that the sifting matrices require $\mathrm{O}(m \log^2 d)$ bits.

Meanwhile, each block $\boldsymbol{R}_u^{(j)}$ of the noise reduction matrix requires $\mathrm{O}(\log d)$ bits. There are $U_j = \mathrm{O}(\sqrt{m/j \log m}\log d)$ blocks, giving a space bound $\mathrm{O}(\sqrt{m/j \log m}\log^2 d)$ for $\boldsymbol{R}^{(j)}$. Summing on j, we see that the noise reduction matrices require space $\mathrm{O}(m^{1/2+\mathrm{o}(1)} \log^2 d)$.

[2]We generate the random variables using a polynomial of degree j over a field of size around d. Without loss of generality, we may assume that m and d are powers of two.

2.5 Encoding time

The encoding time depends on the technology for computing measurements. If we can compute inner products in constant time, the encoding time is proportional to the number of measurements. This section focuses on the case where the cost is proportional to the minimal sparsity of the vectors that appear in the inner product. This analysis plays a role in determining the runtime of the algorithm.

We show that the time required to measure a signal $\boldsymbol{f}$ that has exactly one nonzero component is $\mathrm{O}(m\sqrt{\log m}\log^4(d))$ word operations. This analysis implies a time bound for measuring a vector with more nonzeros. The time (in word operations) to generate the estimation matrix $\boldsymbol{\Phi}$ and to multiply $\boldsymbol{\Phi}$ by $\boldsymbol{f}$ dominates the time (in bit operations) to generate the identification matrix $\boldsymbol{\Omega}$ and to multiply $\boldsymbol{\Omega}$ by $\boldsymbol{f}$.

The time cost for measuring a nonzero component ℓ of a signal with the estimation operator $\boldsymbol{\Phi}$ is $q = \mathrm{O}(m\sqrt{\log m}\log^4 d)$, assuming column ℓ of $\boldsymbol{\Phi}$ has been computed. The (k, ℓ) entry of $\boldsymbol{\Phi}$ is simply $q^{-1/2}\exp\{-2\pi \mathrm{i}\, t_k \omega_\ell / d\}$, which is computed from t_k and ω_ℓ in a constant number of word operations.

We now turn to the identification matrix $\boldsymbol{\Omega}$. Each of its columns contains roughly $\sqrt{m}$ nonzeros, so we can ignore the cost to *apply* the matrix. To *generate* a column of $\boldsymbol{\Omega}$, we form the sifting matrices and noise reduction matrices and then compute their row tensor product. Afterward, we form the row tensor product with the bit-test matrix. The total cost is $\mathrm{O}(m \log^4 d)$ bit operations, as follows.

To construct $\boldsymbol{S}_t^{(j)}$, one can use a polynomial of degree $\mathrm{O}(j)$ over a field of size approximately d to obtain $\mathrm{O}(j)$-wise independent random variables. This step must be repeated $T_j = O(\log(d/j))$ times, for a total of $\mathrm{O}(j \log(d/j))$ field operations over a field of size around d. Therefore, to generate $\boldsymbol{S}^{(j)}$ costs $\mathrm{O}(j \log^3 d)$ bit operations. To generate each $\boldsymbol{R}_u^{(j)}$ requires $\log d$ operations, so the cost to generate $\boldsymbol{R}^{(j)}$ is roughly $\sqrt{m}$, which is negligible. To build $\boldsymbol{R}^{(j)} \otimes_{\mathrm{r}} \boldsymbol{S}^{(j)}$ from its factors costs $T_j \cdot U_j = \mathrm{O}(\sqrt{m}\log^3 d)$, so it can also be ignored. Summing on j, we find that the bit time to construct $\boldsymbol{A}$ is $\mathrm{O}(m \log^3 d)$ bit operations. The row tensor product with the bit-test matrix gives an additional factor of $\mathrm{O}(\log d)$ bit operations.

To summarize, the total time cost to use $\boldsymbol{\Psi}$ to measure a signal with k nonzero entries is $O(km\,\mathrm{polylog}(d))$. During the HHS Pursuit algorithm, we must encode a list L of $\mathrm{O}(m\sqrt{\log m})$ signals with one nonzero component each. The time cost for this encoding procedure is $m^2\,\mathrm{polylog}(d)$ if we use the straightforward algorithm for matrix–vector multiplication.[3]

[3]Note that encoding requires us to compute a partial discrete Fourier transform with unequally-spaced points on the domain and codomain of the transform. We are not aware

3. THE HHS ALGORITHM

The HHS algorithm is an iterative procedure, where each iteration has several stages. The first stage is designed to identify a small set of signal positions that carry a constant proportion of the remaining energy. The second stage estimates the values of the signal on these components. The third stage adds the new approximation to the old approximation and prunes it so that it contains only $\mathrm{O}(m)$ nonzero components. The fourth stage encodes the new approximation using the measurement matrix and subtracts it from the initial sketch to obtain a sketch of the current residual signal. See Figure 2 for pseudocode. For brevity, we use the term *spike* to refer to the location and size of a single signal component.

The algorithm also employs a preprocessing phase. The preprocessing step encodes the signal with the Chaining Pursuit measurement matrix $\boldsymbol{C}$ and executes the Chaining Pursuit algorithm to produce a good initial approximation $\boldsymbol{a}^{\text{init}}$ of the input signal. This initial approximation has at most m nonzero components. We encode $\boldsymbol{a}^{\text{init}}$ with the HHS measurement matrix and subtract it from the original chaining sketch $\boldsymbol{\Psi f}$ to obtain a sketch $\boldsymbol{s}$ of the initial residual $\boldsymbol{f} - \boldsymbol{a}^{\text{init}}$. See Figure 3 for pseudocode.

The ℓ_2 norm of the residual after preprocessing is proportional with m and the optimal ℓ_1 error with m terms. This fact ensures the algorithm recovers sparse signals exactly and that it requires only $\mathrm{O}(\log m)$ iterations to reduce the error by a polynomial factor in m. The correctness of the preprocessing phase follows from our previous work [10].

If we do not run the optional preprocessing step, then the initial residual sketch $\boldsymbol{s}$ and list L of spike locations and values are $\boldsymbol{\Psi f}$ and empty, respectively. In that case, one can run the algorithm for $\mathrm{O}(\log \Delta)$ iterations, where $\Delta = \|\boldsymbol{f}\|_1 / \|\boldsymbol{f} - \boldsymbol{f}_m\|_1$ is the *dynamic range* of the problem which we assume is known.

3.1 Implementation

The HHS Pursuit algorithm is easily implemented with standard data structures. There are a few steps that require a short discussion. The Jacobi iteration is a standard algorithm from numerical analysis.

The bit tests also require explanation. Each row of the isolation matrix $\boldsymbol{A}$ effectively generates a copy of the input signal with many locations zeroed out. The bit-test matrix calculates inner products between its rows and the restricted signal. The bit tests attempt to use these numbers to find the location of the largest entry in the restricted signal.

Suppose that the bit tests yield the following $\log_2 \lceil d \rceil + 1$ numbers:

$$c, \quad b(0), \quad b(1), \quad \ldots, \quad b(\log_2 \lceil d \rceil - 1).$$

The number c arises from the top row of the bit test matrix, so it is the sum of the components of the restricted signal. We estimate a spike location as follows. If $|b(i)| \geq |c - b(i)|$, then the ith bit of the estimated location is zero. Otherwise, the ith bit of the estimated location is one. It is clear that that the estimated location is correct if the restricted signal contains one large component, and the remaining components have ℓ_1 norm smaller than the magnitude of the large component.

of any nontrivial algorithm for this problem, despite the existence of faster algorithms [1] for problems that are superficially similar.

We encode the recovered spikes by accessing the columns of the identification and estimation matrices corresponding to the locations of these spikes and then re-scaling these columns by the spike values. Note that this step requires us to generate arbitrary columns.

3.2 Resource Requirements

In Section 2.4, we showed that we need space $m \operatorname{polylog}(d)$ to store pseudorandom seeds from which columns of the measurement operator can be generated as needed, in time $m \operatorname{polylog}(d)$ each. We recall that we can apply $\boldsymbol{\Phi}_{L'}^{\dagger} \boldsymbol{s}_{\text{est}}$ via Jacobi iteration in time $m^2 \operatorname{polylog}(d)$. It follows that our algorithm requires just $m \operatorname{polylog}(d)$ working space and $m^2 \operatorname{polylog}(d)$ time. Our algorithm becomes an approximation scheme by substituting m/ε^2 for m. This increases the space to $(m/\varepsilon^2) \operatorname{polylog}(d/\varepsilon)$ and the overall time cost to $(m^2/\varepsilon^4) \operatorname{polylog}(d/\varepsilon)$.

4. ANALYSIS OF THE ALGORITHM

This section describes, at the highest level, why HHS works. We establish the following result.

THEOREM 6. *Fix m. Assume that $\boldsymbol{\Psi}$ is a measurement matrix that satisfies the conclusions of Lemmas 9 and 15. Suppose that $\boldsymbol{f}$ is a d-dimensional signal. Given the sketch $\boldsymbol{v} = \boldsymbol{\Psi f}$, the HHS Pursuit algorithm produces a signal $\widehat{\boldsymbol{f}}$ with at most $8m$ nonzero entries. This signal estimate satisfies*

$$\left\|\boldsymbol{f} - \widehat{\boldsymbol{f}}\right\|_2 \leq \frac{20}{\sqrt{m}} \left\|\boldsymbol{f} - \boldsymbol{f}_m\right\|_1.$$

Let m and ε be fixed. Observe that we can apply the theorem with $m' = m/\varepsilon^2$ to obtain a signal estimate $\widehat{\boldsymbol{f}}$ with $8m'$ terms that satisfies the error bound

$$\left\|\boldsymbol{f} - \widehat{\boldsymbol{f}}\right\|_2 \leq \frac{20\varepsilon}{\sqrt{m}} \left\|\boldsymbol{f} - \boldsymbol{f}_{m'}\right\|_1.$$

The running time increases by a factor of $(1/\varepsilon^4) \operatorname{polylog}(1/\epsilon)$. This leads to Theorem 1. We give an overview of the proof in the next subsections.

The goal of the algorithm is to identify a small set of signal components that carry most of the energy in the signal and to estimate the magnitudes of those components well. We argue that, when our signal estimate is poor, the algorithm makes substantial progress toward this goal. When our estimate is already good, the algorithm does not make it much worse. We focus on the analysis of the algorithm in the case when our signal estimate is poor as this is the critical case. The most important portion of the analysis is the identification of the energetic signal components, and, as such, we concentrate on this section of the analysis in the Section 4.2. We omit many of the details of the rest of the analysis from this extended abstract.

4.1 Preliminaries

In a given iteration, the performance of the algorithm depends on the size of the residual. We establish that if the approximation is poor then the algorithm improves it substantially. More precisely, assume that the current approxi-

```
                                Algorithm:  HHS Pursuit

Inputs:  The number m of spikes, the HHS measurement matrix Ψ,
      the initial sketch v = Ψf, the initial list L of m spikes,
      the initial residual sketch s
Output:  A list L of O(m) spikes

For each iteration k = 0, 1, ..., O(log m) {
      For each scale j = 1, 2, 4, ..., O(m) {
           Initialize L' = ∅.
           For each row of A^(j) {
                Use the O(log d) bit tests to identify one spike location
           }
           Retain a list L'_j of the spike locations
                that appear Ω(√(m/j log m) log(d/j) log d) times each
           Update L' ⟵ L' ∪ L'_j
      }
      Estimate values for the spikes in L' by forming Φ†_{L'} s_est
           with Jacobi iteration
      Update L by adding the spikes in L'
           If a spike is duplicated, add the two values together
      Prune L to retain the O(m) largest spikes
      Encode these spikes with measurement matrix Ψ
      Subtract encoded spikes from original sketch v to form
           a new residual sketch s
}
```

Figure 2: Pseudocode for the HHS Pursuit algorithm

```
                    Algorithm:  (Optional) Chaining Pursuit Preprocessing

Inputs:  The number m of spikes, the Chaining measurement matrix C,
      the Chaining sketch w = Cf, the HHS measurement matrix Ψ,
      the HHS sketch v = Ψf
Outputs:  A list L of m spikes, the residual sketch s

Run ChainingPursuit(m, w, C) to obtain a list L of m spikes
Encode the spikes in L using the HHS measurement matrix Ψ.
Subtract the encoded spikes from v to form the residual sketch s.
```

Figure 3: Pseudocode for Chaining Pursuit Preprocessing

mation $\boldsymbol{a}$ satisfies

$$\|\boldsymbol{f} - \boldsymbol{a}\|_2 > \frac{1}{\sqrt{m}} \|\boldsymbol{f} - \boldsymbol{f}_m\|_1 . \qquad \text{(Case 1)}$$

Then one iteration produces a new approximation $\boldsymbol{a}^{\text{new}}$ for which $\|\boldsymbol{f} - \boldsymbol{a}^{\text{new}}\|_2 \le \frac{1}{2} \|\boldsymbol{f} - \boldsymbol{a}\|_2$.

On the other hand, when the approximation is good, then the algorithm produces a new approximation that is not too bad. Suppose that $\boldsymbol{a}$ satisfies

$$\|\boldsymbol{f} - \boldsymbol{a}\|_2 \le \frac{1}{\sqrt{m}} \|\boldsymbol{f} - \boldsymbol{f}_m\|_1 . \qquad \text{(Case 2)}$$

Then the next approximation $\boldsymbol{a}^{\text{new}}$ satisfies $\|\boldsymbol{f} - \boldsymbol{a}^{\text{new}}\|_2 \le \frac{20}{\sqrt{m}} \|\boldsymbol{f} - \boldsymbol{f}_m\|_1$.

Suppose that (Case 1) is in force at the beginning of an iteration. We describe some generic properties of signals that are important in the analysis. We show that the ℓ_2 norm of the tail of a signal is much smaller than the ℓ_1 norm of the entire signal. The pruning step of the algorithm ensures that we have the loop invariant $\|\boldsymbol{a}\|_0 \le 8m$. We abbreviate $p = 8m$ to make the argument clearer.

LEMMA 7. *For any signal $\boldsymbol{g}$, it holds that $\|\boldsymbol{g} - \boldsymbol{g}_t\|_2 \le \frac{1}{2\sqrt{t}} \|\boldsymbol{g}\|_1$. In particular, for $\boldsymbol{g} = \boldsymbol{f} - \boldsymbol{f}_m$, we have $\|\boldsymbol{f} - \boldsymbol{f}_p\|_2 \le \frac{1}{2\sqrt{7m}} \|\boldsymbol{f} - \boldsymbol{f}_m\|_1$ since $p - m = 7m$.*

When the condition (Case 1) holds, most of the energy in the signal is concentrated in its largest components. Let $\boldsymbol{r} = \boldsymbol{f} - \boldsymbol{a}$ denote the residual signal. The number α is this lemma in a constant that will be fixed later.

LEMMA 8 (HEADS AND TAILS). *Suppose that* (Case 1) *is in force. Fix a number $\alpha \ge 1$, and let M be the smallest power of two that exceeds $16\alpha^2 m + 9m$. Then the following bounds hold.*

$$\|\boldsymbol{r}\|_2 \ge \alpha \|\boldsymbol{r} - \boldsymbol{r}_M\|_2 \quad \textit{and} \quad \|\boldsymbol{r}\|_2 \ge \frac{\alpha}{\sqrt{M}} \|\boldsymbol{r}\|_1 .$$

4.2 Identification

As the bulk of the innovation of our result is in the identification of significant signal components for Case 1, we

explain this portion of the analysis in more detail (at the expense of the other portions). The identification matrix $\boldsymbol{\Omega} = \boldsymbol{B} \otimes_{\mathrm{r}} \boldsymbol{A}$ is a complicated thing and in this object lies the majority of the analysis of the algorithm. We can best understand its behavior by studying its pieces separately.

The isolation matrix $\boldsymbol{A}$ consists of $\log_2 M$ blocks $\boldsymbol{A}^{(j)}$, where $j = 1, 2, 4, \ldots, M/2$. Each block $\boldsymbol{A}^{(j)} = \boldsymbol{R}^{(j)} \otimes_{\mathrm{r}} \boldsymbol{S}^{(j)}$ where $\boldsymbol{S}^{(j)}$ is the sifting matrix and $\boldsymbol{R}^{(j)}$ is the noise reduction matrix. It is best to think about the action of the isolation matrix $\boldsymbol{A}^{(j)}$ in two phases.

1. First $\boldsymbol{S}^{(j)}$ takes an input signal and generates a collection of output signals of the same length by zeroing out different collections of components. The idea is that most of the distinguished components will appear in an output signal that contains no other distinguished component.
2. Then $\boldsymbol{R}^{(j)}$ takes each of these signals and generates a further collection of output signals by zeroing out additional subsets of components. The idea is that, in many of the output signals, a distinguished component will survive, but the ℓ_1 norm of the other components will be substantially reduced.

Afterward, the bit-test matrix $\boldsymbol{B}$ forms the inner product between each of its rows and each of the numerous output signals. Whenever an output signal contains a distinguished component and a small amount of noise, the $\log d$ bit tests allow us to determine the location of the distinguished component correctly. The bit test process can always identify the largest component of a signal, provided that the ℓ_1 norm of the remaining components is not too large.

Let us consider a fixed collection I of components in a signal $\boldsymbol{g}$, where $j \le |I| < 2j$. The next result shows that $\boldsymbol{A}^{(j)}$ succeeds in generating a lot of output signals where a large proportion of the components in $|I|$ are isolated from each other. Moreover, the ℓ_1 norm of the other components in these signals is small in comparison with the total norm of the signal. The number ρ in this lemma is a constant (depending only on α) that will be determined shortly.

LEMMA 9. *Except with probability* $\mathrm{O}(d^{-1} \log m)$*, the random isolation operator* $\boldsymbol{A}^{(j)}$ *satisfies the following property. Let* $\boldsymbol{g}$ *be a signal, and let* I *be an arbitrary subset of* $\{1, 2, \ldots, d\}$ *with* $j \le |I| < 2j$*. For at least* $(1 - \rho)\,|I|$ *of the components* $i \in I$*, the operator* $\boldsymbol{A}^{(j)}$ *generates at least*

$$\mathrm{O}(\sqrt{(M/j) \log M} \log(d/j) \log d)$$

signals of the form $g_i \mathbf{e}_i + \boldsymbol{\nu}$ *where*

$$\|\boldsymbol{\nu}\|_1 \le \frac{1}{4\,|I|} \sqrt{\frac{j}{M \log_2 M}}\, \|\boldsymbol{g}\|_1 .$$

The proof of this result takes several long steps and we focus on the sifting and the noise reduction matrics to highlight the necessary lemmas which form the significant steps in the proof.

We can think about the action of one submatrix $\boldsymbol{S}_t^{(j)}$ as

$$\boldsymbol{S}_t^{(j)} : \boldsymbol{g} \mapsto \begin{bmatrix} \boldsymbol{h}^1 & \boldsymbol{h}^2 & \ldots & \boldsymbol{h}^N \end{bmatrix},$$

mapping each input signal to a collection of output signals. The first result shows that one trial of sifting is very likely to isolate all but a constant proportion of the distinguished indices.

LEMMA 10 (SIFTING: ONE TRIAL). *Let* $\boldsymbol{g}$ *be a signal, and let* $I \subset \{1, 2, \ldots, d\}$ *with* $j \le |I| < 2j$*. Write* $k = |I|$*. Suppose we apply the random operator* $\boldsymbol{S}_t^{(j)}$ *to* $\boldsymbol{g}$*. Except with probability* $\mathrm{e}^{-1.7 - \rho k/5}$*, for at least* $(1-\rho)k$ *of the indices* $i \in I$*, there is an output signal* $\boldsymbol{h}$ *of the form*

$$\boldsymbol{h} = g_i \mathbf{e}_i + \boldsymbol{\nu} \quad \text{and} \quad \|\boldsymbol{\nu}\|_1 \le \frac{2}{\rho k} \|\boldsymbol{g}\|_1 .$$

PROOF. We can think of the sifting operator as assigning each of the k distinguished positions (balls) to one of the N output signals (bins) uniformly at random. We hope that the balls are isolated from one another. We will see that if the number of bins satisfies $N \ge \max\{10k\rho^{-1}, 850\rho^{-1}\}$, then the result holds.

For $n = 1, 2, \ldots, N$, let X_n be the indicator variable for the event that the nth bin is empty, and write $X = \sum X_n$ for the total number of empty bins. The symbols μ and σ^2 will denote the expectation and variance of X. To understand large deviations of X requires some effort because the set of indicators $\{X_n\}$ is not stochastically independent. Nevertheless, X satisfies a rather strong tail bound.

FACT 11 (THEOREM 6, [11]).

$$\mathbb{P}\{X \ge \mathbb{E}\, X + a\} \le \exp\left\{ -\left(\sigma^2 + a\right) \log\left(1 + \frac{a}{\sigma^2}\right) - a \right\}.$$

This result is based on the surprising fact, due to Vatutin and Mikhailov [17], that X *can be expressed as a sum of* independent *indicators.*

The content of our argument is to develop explicit bounds on the expectation and variance of X, which will allow us to apply Janson's result. By calculating the means and covariances of the variables X_n, we determine that

$$\mu = N\left(1 - \frac{1}{N}\right)^k < (N - k) + \frac{k^2}{2N}$$

and that

$$\sigma^2 = N\left(1 - \frac{1}{N}\right)^k + N(N-1)\left(1 - \frac{2}{N}\right)^k - N^2\left(1 - \frac{1}{N}\right)^{2k}.$$

The variance bound takes some work. First, regroup terms and factor and then apply Bernoulli's inequality $(1 + x)^k \ge 1 + kx$, which is valid for $x \ge -1$. Finally, we obtain the bound

$$\sigma^2 \le k \cdot h(1/N) \le \frac{k(k-1)}{N} < \frac{k^2}{N}$$

provided that $N \ge 2$.

Depending on the size of k, we need to choose a different number N of bins to obtain the required probabilities. First, assume that $0.2\rho k > 1.7$. In this case, we select $N \ge 10k/\rho$, which yields the following estimates on the mean and variance of X:

$$\mu \le (N - k) + 0.05\rho k \quad \text{and} \quad \sigma^2 < 0.1\rho k.$$

We invoke Fact 11 with the value $a = 0.2\rho k$ to reach

$$\mathbb{P}\{X > (N - k) + 0.25\rho k\} < \mathrm{e}^{-0.4\rho k} < \mathrm{e}^{-1.7 - 0.2\rho k} \tag{4.1}$$

using $0.2\rho k > 1.7$.

This estimate allows us to bound the number Y of balls that fail to be isolated. It takes at least $(N - X)$ balls to fill the nonempty bins. The remaining $(k - (N - X))$ balls can be placed in no more than $(k - (N - X))$ bins, where they will

result in no more than $Y = 2(k-(N-X)) = 2(X-(N-k))$ collisions. Using the deviation bound (4.1), we conclude that

$$\mathbb{P}\left\{Y > \frac{\rho k}{2}\right\} < \mathrm{e}^{-1.7-0.2\rho k}.$$

Second, we assume that k is small. Precisely, consider the case where $0.2\rho k \le 1.7$. Now, select $N \ge 100k^2$. The mean and variance of X satisfy

$$\mu \le (N-k) + 0.005 \quad \text{and} \quad \sigma^2 \le 0.01$$

Apply Fact 11 with the value $a = 0.995$ to reach

$$\mathbb{P}\{X \ge (N-k)+1\} < \mathrm{e}^{-3.6} < \mathrm{e}^{-1.7-0.2\rho k}$$

using $0.2\rho k \le 1.7$. When $X < (N-k)+1$, the maximum number of bins are empty, and so all k of the balls are isolated. Furthermore, we observe that the number N of bins required here satisfies $N \le 850\rho^{-1}$.

Finally, we need to argue that few of the output signals have a lot of noise. Let u_n denote the ℓ_1 norm of the nth output signal. Since each position in the input signal $\boldsymbol{g}$ is assigned to exactly one output signal, $\sum_n u_n = \|\boldsymbol{g}\|_1$. By Markov's inequality,

$$\#\left\{n : u_n \ge \frac{2}{\rho k}\|\boldsymbol{g}\|_1\right\} \le \frac{\rho k}{2\|\boldsymbol{g}\|_1}\sum\nolimits_n u_n = \frac{\rho k}{2}.$$

In particular, no more than $\rho k/2$ of the isolated balls can appear in a bin whose ℓ_1 norm exceeds $(2/\rho k)\|\boldsymbol{g}\|_1$. Therefore, the total number of positions in I lost to collisions or noise is at most ρk except with probability $\mathrm{e}^{-1.7-0.2\rho k}$. □

The failure probability for one trial is not small enough to take a union bound over all possible sets I. We perform $T_j = \mathrm{O}(\log(d/j))$ repeated trials to drive down the failure probability.

Lemma 12 (Sifting: All Trials). *Except with probability* $\exp(-j\log d)$*, the sifting operator* $\boldsymbol{S}^{(j)}$ *has the following property. Let* $\boldsymbol{g}$ *be an arbitrary signal, and let* $I \subset \{1,2,\dots,d\}$ *satisfy* $j \le |I| < 2j$*. For some set of* $(1-\rho)|I|$ *indices* $i \in I$*, there are at least* $0.5T_j$ *output signals* $\boldsymbol{h}$ *of the form* $\boldsymbol{h} = g_i\mathbf{e}_i + \boldsymbol{\nu}$ *and* $\|\boldsymbol{\nu}\|_1 \le \frac{2}{\rho k}\|\boldsymbol{g}\|_1$.

Proof. Let $\boldsymbol{g}$ be a signal, and fix a set $I \subset \{1,2,\dots,d\}$ that contains k or more indices. Lemma 10 shows that each isolation trial succeeds for at least $(1-\rho)|I|$ of the distinguished indices, except with probability $p = \mathrm{e}^{-1.7-0.2\rho k}$. We repeat this experiment T_j times. Let X_t be the indicator variable for the event that trial t fails, so $\mathbb{E}\,X_t \le p$. Then the random variable $X = \sum X_t$ counts the total number of trials in which ρk balls fail to be isolated, and its mean satisfies $\mu \le pT_j$. Chernoff's bound shows that

$$\mathbb{P}\{X > 0.5T_j\} < \left[\frac{\mathrm{e}}{0.5T_j/(pT_j)}\right]^{T_j} < \mathrm{e}^{-0.2\rho k T_j}$$

Choose $T_j = 15\rho^{-1}\log(\mathrm{e}d/j)$, and use the fact that $k \ge j$ to obtain

$$\mathbb{P}\{X > T_j\} < \mathrm{e}^{-3j\log(\mathrm{e}d/j)}.$$

Next, we must count the total number of subsets whose size is between j and $(2j-1)$. We bound

$$\sum_{r=j}^{2j-1}\binom{d}{r} \le \sum_{r=j}^{2j-1}\mathrm{e}^{r\log(\mathrm{e}d/r)} \le \int_j^{2j}\mathrm{e}^{x\log(\mathrm{e}d/x)}\,\mathrm{d}x \le \mathrm{e}^{2j\log(\mathrm{e}d/j)}.$$

Finally, we take a union bound over all sets I with size between j and $(2j-1)$ to obtain a failure probability of

$$\mathrm{e}^{2j\log(\mathrm{e}d/j)}\cdot\mathrm{e}^{-3j\log(\mathrm{e}d/j)} = \mathrm{e}^{-j\log(\mathrm{e}d/j)}.$$

In other words, for every such I, the sifting matrix $\boldsymbol{S}^{(j)}$ isolates at least $(1-\rho)|I|$ of the distinguished indices in at least half the trials. □

We establish that, if $\boldsymbol{g}$ contains a single distinguished component (a spike) plus noise, then a large number of the output signals contain that spike along with a reduced amount of noise.

Lemma 13 (Noise Reduction). *Except with probability* d^{-1}*, the noise reduction matrix* $\boldsymbol{R}^{(j)}$ *has the following property. Let* $\boldsymbol{g}$ *be an input signal that satisfies (i)* $\boldsymbol{g} = \boldsymbol{\delta}+\boldsymbol{\nu}$*, (ii)* $\|\boldsymbol{\delta}\|_0 = 1$*, and (iii)* $\operatorname{supp}(\boldsymbol{\delta}) \cap \operatorname{supp}(\boldsymbol{\nu}) = \emptyset$*. Then there are at least* $0.5Cr\log d$ *output signals* $\boldsymbol{h}$ *of the form* $\boldsymbol{h} = \boldsymbol{\delta}+\boldsymbol{\mu}$ *where*

$$\|\boldsymbol{\mu}\|_1 \le \frac{10}{Cr}\|\boldsymbol{\nu}\|_1 = \frac{\sqrt{j}}{8\rho^{-1}\sqrt{M\log_2 M}}\|\boldsymbol{\nu}\|_1.$$

Proof. Let $\boldsymbol{g}$ be a signal of the form $\boldsymbol{g} = \boldsymbol{\delta}+\boldsymbol{\nu}$, where $\boldsymbol{\delta}$ is a spike at position i. Consider the submatrix $\boldsymbol{X}$ of $\boldsymbol{R}^{(j)}$ constructed by extracting the rows of $\boldsymbol{R}^{(j)}$ where the index i appears and then removing the ith column. This submatrix contains exactly $Cr\log d$ rows and $(d-1)$ columns. Let $\boldsymbol{\nu}'$ be the vector $\boldsymbol{\nu}$ without its ith component (which equals zero by hypothesis). Note that $\boldsymbol{\nu}'$ has the same ℓ_1 norm as $\boldsymbol{\nu}$.

Let $\boldsymbol{x}$ be a column of $\boldsymbol{X}$. The entries of $\boldsymbol{x}$ are independent binary random variables with mutual expectation $(Cr)^{-1}$ because each one comes from a different submatrix. Therefore, $\mathbb{E}\|\boldsymbol{x}\|_1 = \frac{Cr\log d}{Cr} = \log d$. Chernoff's bound shows that $\mathbb{P}\left\{\|\boldsymbol{x}\|_1 > 5\log d\right\} \le \left[\frac{\mathrm{e}^4}{5^5}\right]^{\log d} < d^{-3}$. Applying the union bound over all $(d-1)$ columns of $\boldsymbol{X}$, $\mathbb{P}\left\{\|\boldsymbol{X}\|_{1,1} > 5\log d\right\} \le d^{-2}$.

Therefore, the number of output signals in which position i appears and where the noise is large satisfies

$$\begin{aligned}\#\{n : |(\boldsymbol{X}\boldsymbol{\nu}')_n| > \frac{10\|\boldsymbol{\nu}'\|_1}{Cr}\} &\le 0.1Cr\frac{\|\boldsymbol{X}\boldsymbol{\nu}'\|_1}{\|\boldsymbol{\nu}'\|_1}\\ &\le 0.5Cr\log d.\end{aligned}$$

Since position i appears in exactly $Cr\log d$ output signals, we discover that the number of output signals where position i appears and the noise is less than $(10/Cr)\|\boldsymbol{\nu}\|_1$ is at least $0.5Cr\log d$.

So far, we have only established the result for a single spike location i. To complete the proof, we perform a union bound over the d possible locations for the spike to obtain a final failure probability of d^{-1}. □

Combine Lemma 12 and Lemma 13 to obtain the announced Lemma 9.

Let us instantiate our fixed collection I of signal components as those which fall in a significant band B_s. Let s be a power of two between 1 and $M/2$, and note that s takes $\log_2 M$ values. We define the sth band of the residual to be the set $B_s = \left\{i : \frac{1}{2s}\|\boldsymbol{r}\|_1 < |r_i| \le \frac{1}{s}\|\boldsymbol{r}\|_1\right\}$. We say that the sth band is *significant* if $\left\|\boldsymbol{r}\big|_{B_s}\right\|_2 > \frac{\alpha^{-1}}{\sqrt{\log_2 M}}\|\boldsymbol{r}\|_2$. First, we check that B_s meets our size constraints. Next, we use our guarantee on the isolation operator $\boldsymbol{A}^{(j)}$ to conclude that

1. the identification process finds at least $(1-\rho)\,|B_s|$ of the components in the band at least $O(\sqrt{(M/j)\log M}\log(d/j)\log d)$ times each;

2. the total ℓ_2 norm of the lost components is at most $\alpha^{-1}\,\|\boldsymbol{r}\|_2$; and

3. the final list of identified components contains $\mathrm{O}(M\sqrt{\log M})$ items.

Now we argue that the signal positions in the list of identified components carry most of the energy in the residual signal. This fact ensures that the iteration is making progress toward finding significant signal positions. Moreover, it guarantees that the estimation step can accurately predict the values of the signal positions listed in L. The parts of the residual that we miss fall into several categories. The challenging piece requires some serious work which we presented above. The remaining pieces follow from the head-tail relationships and the definition of significant band.

4.3 Estimation

The estimation step produces an approximation $\boldsymbol{b}$ to the residual $\boldsymbol{r}$ that lies relatively close to the residual, even though it contains $\mathrm{O}(M\sqrt{\log M})$ nonzero entries. There are two technical results that are essential to the proof. The first appeared in the work of Rudelson and Vershynin [14].

LEMMA 14 (RESTRICTED ISOMETRY). *The estimation matrix $\boldsymbol{\Phi}$ has the property that every $|L|$-column submatrix $\boldsymbol{A}$ satisfies for every vector $\boldsymbol{x}$, $\frac{1}{2}\,\|\boldsymbol{x}\|_2 \le \|\boldsymbol{A}\boldsymbol{x}\|_2 \le \frac{3}{2}\,\|\boldsymbol{x}\|_2$.*

LEMMA 15. *The estimation matrix $\boldsymbol{\Phi}$ has the property that for every vector $\boldsymbol{x}$, $\|\boldsymbol{\Phi}\boldsymbol{x}\|_2 \le \frac{3}{2}\left[\|\boldsymbol{x}\|_2 + \frac{1}{\sqrt{M}}\,\|\boldsymbol{x}\|_1\right]$.*

Acknowledgments

We wish to thank Mark Rudelson for showing us this lovely proof of Lemma 15. We thank Howard Karloff and Piotr Indyk for many insightful discussions and, finally, we thank the anonymous referees for helping us clarify the related work. ACG is an Alfred P. Sloan Research Fellow. ACG has been supported in part by NSF DMS 0354600. JAT has been supported by NSF DMS 0503299. MJS has been supported in part by NSF DMS 0354600. RV is an Alfred P. Sloan Research Fellow. He was also partially supported by the NSF grant DMS 0401032.

5. REFERENCES

[1] G. Beylkin. On the Fast Fourier Transform of functions with singularities. *Appl. Comp. Harmonic Anal.*, 2:363–381, 1995.

[2] E. J. Candès, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Comm. Pure Appl. Math.*, 59(8):1208–1223, 2006.

[3] E. J. Candès and T. Tao. Near optimal signal recovery from random projections: Universal encoding strategies? Submitted for publication, Nov. 2004.

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

[5] G. Cormode and S. Muthukrishnan. Combinatorial algorithms for Compressed Sensing. In *Proc. of SIROCCO*, pages 280–294, 2006.

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

[7] D. L. Donoho, I. Daubechies, R. DeVore, and M. Vetterli. Data compression and harmonic analysis. *IEEE Trans. Info. Theory*, 44(6):2435–2476, 1998.

[8] M. F. Duarte, M. B. Wakin, and R. G. Baraniuk. Fast reconstruction of piecewise smooth signals from random projections. In *Proc. SPARS05*, Rennes, France, Nov. 2005.

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

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

[11] S. Janson. Large deviation inequalities for sums of indicator variables. Technical report, Mathematics Dept., Uppsala Univ., 1994.

[12] S. Kirolos, J. Laska, M. Wakin, M. Duarte, D. Baron, T. Ragheb, Y. Massoud, and R. Baraniuk. Analog-to-information conversion via random demodulation. In *IEEE Dallas Circuits and Systems Workshop (DCAS)*, Dallas, Texas, Oct. 2006.

[13] J. Laska, S. Kirolos, Y. Massoud, R. Baraniuk, A. Gilbert, M. Iwen, and M. Strauss. Random sampling for analog-to-information conversion of wideband signals. In *IEEE Dallas Circuits and Systems Workshop (DCAS)*, Dallas, Texas, Oct. 2006.

[14] M. Rudelson and R. Veshynin. Sparse reconstruction by convex relaxation: Fourier and Gaussian measurements. In *Proc. 40th Ann. Conf. Information Sciences and Systems*, Princeton, Mar. 2006.

[15] D. Takhar, J. Laska, M. B. Wakin, M. F. Duarte, D. Baron, S. Sarvotham, K. Kelly, and R. G. Baraniuk. A new compressive imaging camera architecture using optical-domain compression. In *Proc. IS&T/SPIE Symposium on Electronic Imaging*, 2006.

[16] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via Orthogonal Matching Pursuit. Submitted for publication, Apr. 2005. Revised, Nov. 2006, 2006.

[17] V. A. Vatutin and V. G. Mikhailov. Limit theorems for the numer of empty cells in an equiprobable scheme for group allocation of particles. *Theor. Probab. Appl.*, 27:734–743, 1982.

[18] M. Wakin, J. Laska, M. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. Kelly, and R. Baraniuk. Compressive imaging for video representation and coding. In *Proc. Picture Coding Symposium 2006*, Beijing, China, Apr. 2006.

[19] Y. Zheng, D. J. Brady, M. E. Sullivan, and B. D. Guenther. Fiber-optic localization by geometric space coding with a two-dimensional gray code. *Applied Optics*, 44(20):4306–4314, 2005.

[20] Y. H. Zheng, N. P. Pitsianis, and D. J. Brady. Nonadaptive group testing based fiber sensor deployment for multiperson tracking. *IEEE Sensors Journal*, 6(2):490–494, 2006.