Sublinear Approximation of Signals

Anna C. Gilbert[a], Martin J. Strauss[a, b], Joel A. Tropp[a] and Roman Vershynin[c]

[a]Mathematics Department, The University of Michigan at Ann Arbor
[b]Electrical Engineering Department, The University of Michigan at Ann Arbor
[c]Mathematics Department, The University of California at Davis

ABSTRACT

It has recently been observed that sparse and compressible signals can be sketched using very few nonadaptive linear measurements in comparison with the length of the signal. This sketch can be viewed as an embedding of an entire class of compressible signals into a low-dimensional space. In particular, d-dimensional signals with m nonzero entries (m-sparse signals) can be embedded in $O(m \log d)$ dimensions. To date, most algorithms for approximating or reconstructing the signal from the sketch, such as the linear programming approach proposed by Candès–Tao and Donoho, require time polynomial in the signal length.

This paper develops a new method, called Chaining Pursuit, for sketching both m-sparse and compressible signals with $O(m \operatorname{polylog} d)$ nonadaptive linear measurements. The algorithm can reconstruct the original signal in time $O(m \operatorname{polylog} d)$ with an error proportional to the optimal m-term approximation error. In particular, m-sparse signals are recovered perfectly and compressible signals are recovered with polylogarithmic distortion. Moreover, the algorithm can operate in small space $O(m \operatorname{polylog} d)$, so it is appropriate for streaming data.

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

1. INTRODUCTION

A *compressible signal* is a long signal that can be represented with an amount of information that is small relative to the length of the signal. Many classes of d-dimensional signals are compressible, e.g.,

- The m-sparse class $B_0(m)$ consists of signals with at most m nonzero entries.
- For $0 < p < 1$, the weak ℓ_p class $B_{\text{weak-}p}(r)$ contains each signal f whose decreasing rearrangement f° satisfies $|f_i^\circ| \le r\, i^{-1/p}$.

These types of signals are pervasive in applications. Natural images are highly compressible, as are audio and speech signals. Image, music, and speech compression algorithms and coders are vital pieces of software in many technologies, from desktop computers to MP3 players. Many types of automatically generated signals are also highly redundant. For example, the distribution of bytes per source IP address in a network trace is compressible—just a few source IP addresses send the majority of the traffic.

Observe that the signals in $B_0(m)$ are determined completely by $2m$ numbers: the locations and sizes of their nonzero entries. Similarly, each signal f in $B_{\text{weak-}p}(r)$ can be approximated by an m-sparse signal f_m with error $\|f - f_m\|_1 \le C_p\, r\, m^{1-1/p}$. In other words, the essential information in these signals is captured by approximately m numbers even though the length d of these signals is considerably higher. The current paradigm for encoding and decoding compressible signals places the burden on the encoder so that the decoder can operate more efficiently. But there are emerging technologies, such as sensor networks, where it is more appropriate to reduce the burden on the encoder at the expense of greater decoding times. Because our signals inherently contain about m pieces of information and because we wish to decode these m pieces of information only (they are sufficient to approximate the signal), we hope that we can encode our signal nonadaptively with approximately m values. Moreover, we want the encoder to be a linear function of the signal, for reasons that will soon be clear.

E-mail: annacg@umich.edu, martinjs@umich.edu, jtropp@umich.edu, vershynin@math.ucdavis.edu

Let us define the problem more formally. We have as input f a compressible signal of length d which is approximated well by an m-sparse signal where $m \ll d$. We sketch f with a small number of nonadaptive linear measurements. The reconstruction algorithm uses these measurements to approximate f; that is, the algorithm returns an m-sparse signal that approximates f well. More precisely, we measure f with a matrix $\mathbf{\Phi}$ consisting of N rows and d columns and produce a vector v of N nonadaptive measurements, $\mathbf{\Phi} f = v$. This perspective raises a number of fundamental questions include the following:

1. How many measurements are necessary; how large is N?
2. How is the measurement system constructed (i.e., deterministic vs. random)?
3. What are the storage costs for the measurement system?
4. Does one system of measurements succeed for an entire class of compressible signals? Or do we need a new set of measurements for each signal?
5. What algorithms can be used to reconstruct the signal? What are their time and storage costs?
6. How does the reconstruction error compare with the optimal error in approximation by an m-sparse signal?

This problem is a natural generalization of the *heavy hitters*[1] problem in theoretical computer science and we may view the quest for such measurement matrices and efficient recovery algorithms as a search for a streaming algorithm to recover compressible signals. To understand the details of this connection, notice that because our measurements are linear, we can update a set of measurements in the face of a stream of updates to our input signal; that is, $\mathbf{\Phi} f_1 + \mathbf{\Phi} f_2 = \mathbf{\Phi}(f_1 + f_2)$. If we seek measurement matrices which can be constructed in small space with $\operatorname{poly}(m, \log d)$ rows and if we also stipulate that our reconstruction algorithms run in time $\operatorname{poly}(m, \log d)$, then we are in the setting of streaming algorithms. This confluence of problems, from approximation theory and from theoretical computer science, is a fortunate one. It highlights a common fundamental problem—how to efficiently recover essential information in a signal.

1.1. Results

This paper describes an approach called *Chaining Pursuit* for sketching compressible signals and calculating m-term approximations from the sketch. This technique has some distinct advantages over other methods described in the literature.

- **Uniformity:** One system of measurements works for all signals in the class.
- **Superefficient Decoding:** The reconstruction time is sublinear in the length d of the signal and proportional to m.
- **Instance-optimal:** The reconstruction error for each signal is on the order of the optimal m-term approximation error. More precisely, let f be a signal, f_m its best m-sparse approximation, and $\widehat{f}$ the computed reconstruction. Then
$$\left\| f - \widehat{f} \right\|_1 \leq (1 + C \log m) \left\| f - f_m \right\|_1$$
Note that the algorithm recovers m-sparse signals exactly.
- **Small space:** The measurement system for Chaining Pursuit can be modified to require space $O(m \operatorname{polylog} d)$, although the version described here is not space-efficient.

The following table summarizes our results for this algorithm.

	Chaining Pursuit
Signal class	ℓ_1
Uniform	Yes
Error bound	Instance
Approx. scheme?	No
Construction	Random
Storage cost	$O(m \operatorname{polylog} d)$
# Measurements	$O(m \log^2 d)$
Encode time	$O(d \log^2 d)$
Decode time	$O(m \log^2 d)$

Note: The big-O notation suppresses factors of $\log m$.

1.2. Related Work

The problem of sketching and reconstructing m-sparse and compressible signals has several precedents in the theoretical computer sciences (TCS) literature, especially the paper of Cormode and Muthukrishnan on detecting heavy hitters in nonnegative data streams[1] and the works of Gilbert et al. on Fourier sampling.[2,3] There are several other recent papers from the TCS literature on this subject.[4,5] Sparked by the papers of Donoho[6] and Candès–Tao,[7] the computational harmonic analysis and geometric functional analysis communities have produced an enormous amount of additional work.[8–14]

Most of the previous work has focused on a reconstruction algorithm that involves linear programming or second-order cone programming.[5–7] Computation times have not been reported, but they are expected to be cubic in the length d of the signal. This cost is absurd, since we are seeking an approximation that involves $O(m)$ terms. Tropp and Gilbert describe another algorithm with running time of order $O(m^2 d \log d)$, which can be reduced to $O(md \log d)$ in certain circumstances.[13] None of these approaches are competitive with the sublinear algorithms described here.

There are a few sublinear algorithms available in the literature. The Fourier sampling paper can be viewed as a small space, sublinear algorithm for signal reconstruction.[3] Its primary shortcoming is that the measurements are not uniformly good for the entire signal class. The recent work of Cormode and Muthukrishnan proposes some other sublinear algorithms for reconstructing compressible signals[4] . Few of these algorithms offer a uniform guarantee. The ones that do require an enormous number of measurements—$O(m^2 \log d)$ or worse—which means that they are not summarizing the signal as efficiently as possible.

The following table describes the best algorithmic contributions on some of the same axes as before. A very similar table appears in the paper of Cormode and Muthukrishnan.[4]

Ref.	Signal Class	Error bd.	Uniform	Storage	# Measurements	Decode time
[CRT04][8]	m-sparse	No error	Yes	$O(m \log d)$	$O(m \log^6 d)$	$\Omega(md)$ LP
[RV05][12]	m-sparse	No error	Yes	$O(md \log d/m)$	$O(m \log d/m)$	$\Omega(md)$ LP
[TG05][13]	m-sparse	No error	No	$O(md \log d)$	$O(m \log d)$	$O(m^2 d \log d)$
[CM05][4]	m-sparse	No error	No	$O(\log d)$	$O(m \log^2 d)$	$O(m \log^2 d)$
[CM05][4]	m-sparse	No error	Yes	$O(m \log d/m)$	$O(m^2 \log^2 d)$	$O(m \log d/m)$
[CT04][7]	weak ℓ_p, ℓ_1	Minimax	Yes	$O(m \log d)$	$O(m \log^4 d)$	$\Omega(md)$ LP
[CM05][4]	weak ℓ_p	Instance	Yes	$O(m^{\frac{2-p}{1-p}} \log d)$	$O(m^{\frac{3-p}{1-p}} \log^2 d)$	$O(m^{\frac{4-2p}{1-p}} \log^3 d)$
[GMS05][3]	ℓ_2	Instance	No	$O(m\varepsilon^{-2} \log^2 d)$	$O(m\varepsilon^{-2} \log^2 d)$	$O(m\varepsilon^{-2} \log^2 d)$
[CM05][4]	ℓ_2	Instance	No	$O(\log^2 d)$	$O(m\varepsilon^{-1} \log^{5/2} d)$	$O(m\varepsilon^{-1} \log^{5/2} d)$

Notes: $\Omega(md)$ LP denotes the time to solve a linear program with $\Omega(md)$ variables.

2. ENCODING FOR CHAINING PURSUIT

Our algorithm for approximating a compressible signal is called Chaining Pursuit. This section describes a linear measurement process that can be used to summarize a d-dimensional signal f for recovery via Chaining Pursuit. Suppose that f is well approximated by an m-sparse function (that is, by m "spikes"). The measurement process used for Chaining Pursuit consists of two steps, which summarize the values and locations of the spikes. First a sequence of bit masks is applied to the signal. Then an isolation matrix is applied to the masked signals to obtain a set of measurements. Intuitively, the goal of this process is to isolate the m spikes from each other so that each measurement involves at most one spike. Then the bit test will correctly identify the location of the spike and estimate its value, provided that other components of the signal ("noise") do not accumulate in that measurement.

Measurement Process: Bit Tests on Random Partitions. The bit-test is the following measurement of a signal. For every bit $b = 1, \dots, \log_2 d$, the bit-test records the sums $s_0(b)$ and $s_1(b)$ of values of the signal corresponding to positions where bit b is 0 and 1 respectively.

The Chaining Pursuit algorithm will make $K = O(\log m)$ passes over the signal, and it will use different set of measurements for each pass. So, for for each pass $k = 0, \dots, K-1$, we repeat the following set of measurements independently $O(k \log d)$ times (*trials*):

1. randomly partition off the set of d signal positions into $O(m/2^k)$ subsets;
2. apply the bit-test to the signal restricted to each subset.

Formally, the measurement operator $\boldsymbol{\Phi}$ is a linear operator that acts as

$$\boldsymbol{\Phi} f = \boldsymbol{A}(\operatorname{diag} f)\boldsymbol{B}$$

where $\boldsymbol{A}$ is the *isolation matrix* and $\boldsymbol{B}$ is the *bit-test matrix*. Applying the measurement operator to a signal f yields a data matrix $\boldsymbol{V}$ with dimensions $O(m \log d) \times O(\log d)$. Each row of the data matrix contains the result of the bit-test applied to the signal restricted to some subset. We will refer to each row as a *measurement* of the signal.

Bit-test matrix. The bit-test matrix $\boldsymbol{B} = [\boldsymbol{B}_1 \ \boldsymbol{B}_2]$ is a zero–one matrix with dimensions $d \times O(\log_2 d)$. Each of the two submatrices $\boldsymbol{B}_0$ and $\boldsymbol{B}_1$ has dimensions $d \times \log_2 \lceil d \rceil$. The ith column of $\boldsymbol{B}_0$ has a one in each row whose index has a zero in the ith digit of its binary expansion. Likewise, the ith column of $\boldsymbol{B}_1$ has a one in each row whose index has a one in the ith digit of its binary expansion. An example of a bit-test matrix $\boldsymbol{B}$ for dimension $d = 8$ is

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

Isolation matrix. The isolation matrix $\boldsymbol{A}$ is a zero–one matrix with dimensions $O(m \log d) \times d$ and a hierarchical structure. Let a be a sufficiently large constant, to be discussed in the next section. The Chaining Pursuit algorithm makes $K = 1 + \log_a m$ passes over the signal, and the isolation matrix involves a different set of measurements for each pass. The measurements for the kth pass are contained in the $O(mk/2^k) \times d$ submatrix $\boldsymbol{A}^{(k)}$. During each pass, the algorithm performs $T_k = O(k \log d)$ trials. Each trial t is associated with a further submatrix $\boldsymbol{A}_t^{(k)}$, which has dimensions $O(m/2^k) \times d$. The submatrix for each trial encodes a random partition of the d signal positions into $O(m/2^k)$ disjoint parts. In other words, we assign each of the signal positions independently at random to one of the $O(m/2^k)$ measurements. Here is a picture of the isolation matrix:

$$\boldsymbol{A} = \left[\begin{array}{c} \boldsymbol{A}^{(1)} \\ \hline \boldsymbol{A}^{(2)} \\ \hline \vdots \\ \hline \boldsymbol{A}^{(K)} \end{array}\right] \qquad \text{where} \qquad \boldsymbol{A}^{(k)} = \left[\begin{array}{c} \boldsymbol{A}_1^{(k)} \\ \hline \boldsymbol{A}_2^{(k)} \\ \hline \vdots \\ \hline \boldsymbol{A}_{T_k}^{(k)} \end{array}\right].$$

An example of $\boldsymbol{A}_t^{(k)}$ with dimensions 3×8 is

$$\boldsymbol{A}_t^{(k)} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Note that each of the d signal positions (corresponding to columns) appears in exactly one row.

Storage cost. The bit test matrix requires no storage. The total storage for the isolation matrix $\boldsymbol{A}$ is $O(d \log^2 m \log d)$ bits. This cost is calculating by observing that the submatrix $\boldsymbol{A}_t^{(k)}$ can be stored using $d \log_2(m/2^k)$ bits and summing over all trials and all passes.

Encoding time. The total time cost to apply the measurement operator $\boldsymbol{\Phi}$ to a signal is $O(d \log^2 m \log^2 d)$. This follows because the number of nonzero entries in $\boldsymbol{A}$ is proportional to $d \log^2 m \log d$, and we must apply $\boldsymbol{A}$ to each of the $O(\log d)$ masked signals.

Properties of isolation matrix. The proof that Chaining Pursuit is correct relies on certain qualities of the isolation matrix. The matrix has these qualities with high probability over the choice of random partitions. It is conceivable that the properties could also be attained with a deterministic or small-space construction. For more details, please see Section 4.

3. DECODING WITH CHAINING PURSUIT

The Chaining Pursuit algorithm takes as input a data matrix $\boldsymbol{V}$, the isolation matrix $\boldsymbol{A}$, and a target number of spikes m. Its goal is to produce an approximation of the original signal f using at most $O(m)$ spikes. Let a be a sufficiently large number. The basic idea is to recover all except a $1/a$ fraction of the remaining spikes in each pass. After $O(\log m)$ passes, there are no spikes left. The reason for the name "Chaining Pursuit" is that this process decomposes the signal into pieces with supports of geometrically decreasing size. It resembles an approach in analysis and probability, also called chaining, that is used to control the size of a function by decomposing it into pieces with geometrically decreasing sizes. A famous example of chaining in probability theory is to prove bounds on the expected supremum of an empirical process.[15] For an example of chaining in TCS, see the results of Indyk and Naor on embeddings.[16]

The overall structure of the algorithm is similar to other sublinear approximation algorithms described in the literature. It involves three steps:

1. Identify spikes
2. Estimate their values
3. Iterate on the residual

An overview of the algorithm appears below; the details follow.

```
                    Algorithm:  Chaining Pursuit

Inputs:  Number $m$ of spikes, data matrix $\boldsymbol{V}$, isolation matrix $\boldsymbol{A}$
Output:  A list of $O(m)$ spike locations and values

For each pass $k = 0, 1, \dots, \log_a m$:
     For each trial $t = 1, 2, \dots, O(k \log d)$:
          For each measurement $n = 1, \dots, O(m/2^k)$
               Use bit tests to identify the spike position
               Use a bit test to estimate the spike magnitude
          Retain $m/a^k$ distinct spikes with values largest in magnitude
     Retain spike positions that appear in more than 9/10 of trials
     Estimate final spike sizes using medians
     Encode the spikes using the measurement operator
     Subtract the encoded spikes from the data matrix
```

Implementation. Most of the steps of this algorithm are straightforward to implement using standard abstract data structures. One point that may require comment is the application of bit tests to identify spike positions and values.

A measurement is a row of the data matrix, consisting of $2 \log_2 \lceil d \rceil$ numbers:

$$\left[\begin{array}{cccc|cccc} b_0(1) & b_0(2) & \dots & b_0(\log_2 d) & b_1(1) & b_1(2) & \dots & b_1(\log_2 d) \end{array} \right].$$

We omit the ceiling for legibility. These numbers allow us to obtain the binary representation of a spike location. If $|b_0(i)| \geq |b_1(i)|$ then the ith bit of the location is zero. If $|b_1(i)| > |b_1(i)|$ then the ith bit of the location is one. Suppose that the measurement contains one spike and the ℓ_1 norm of the other signal positions assigned to that measurement is less than the magnitude of the spike. It is easy to check that this bit test correctly identifies the spike location.

To obtain an estimated value for the spike from the measurements, we just use the least significant bit. If $|b_0(1)| \geq |b_1(1)|$, then the estimated size of the spike is $b_0(1)$. Otherwise, the estimated size is $b_1(1)$. Any other bit would work just as well. A median over all the bits might be more robust, but it costs an extra factor of $O(\log d)$. It is easy to check that, if the measurement contains one spike, then the estimated value is equal to the actual value plus or minus the ℓ_1 norm of the other positions assigned to the measurement.

In pass k, the number of recovered spikes is at most $O(m/a^k)$, so the cost of encoding these spikes is $O(ma^{-k} \log^2 m \log^2 d)$. Updating the data matrix requires the same amount of time. Note that this cost assumes random access to the isolation matrix!

Storage costs. The primary storage cost derives from the isolation matrix $\boldsymbol{A}$. Otherwise, the algorithm requires only $O(m \log d)$ working space.

Time costs. During pass k, the primary cost of the algorithm occurs when we encode the recovered spikes. This operation requires $O(ma^{-k} \log^2 m \log^2 d)$ time in pass k. Summing over all passes, we obtain $O(m \log^2 m \log^2 d)$ total running time.

4. ANALYSIS OF CHAINING PURSUIT

This section summarizes our analysis of Chaining Pursuit, which yields the following theorem. Fix an isolation matrix $\boldsymbol{A}$ that satisfies the conclusions of Theorem 2 of the sequel.

Theorem 1 (Chaining Pursuit). *Suppose that f is a d-dimensional signal whose best m-term approximation with respect to ℓ_1 norm is f_m. Given the data matrix $\boldsymbol{V} = \boldsymbol{\Phi} f$ and the number m, Chaining Pursuit produces a signal $\widehat{f}$ consisting of at most $O(m)$ spikes. This signal estimate satisfies*

$$\left\| f - \widehat{f} \right\|_1 \leq (1 + C \log m) \left\| f - f_m \right\|_1.$$

In particular, if $f_m = f$, then also $\widehat{f} = f$.

Remark 1. *(i) The number C is a constant that depends only on the constant a; it does not appear possible to make C arbitrarily small. (ii) The number $O(m)$ spikes in the output signal $\widehat{f}$ can be improved to $(1+\varepsilon)m$ for any fixed $\varepsilon \in (0,1)$ with small modifications of the algorithm; the current version of gives* $1.105m$ *spikes. (iii) The factor of* $\log m$ *is intrinsic to this approach. However, the proof gives a stronger statement – the approximation in the weak-1 norm* without *that factor:* $\| f - \widehat{f} \|_{\text{weak}-1} \le C \| f - f_m \|_1$. In the remainder of this section, we will abbreviate $m_k = m/a^k$.

4.1. Challenges of the analysis

Chaining Pursuit is an iterative algorithm. The major difficulty of its analysis is to control the approximation error from blowing up in a geometric progression from pass to pass. In pass $k = 0$, the algorithm is working with measurements of the original signal f. This signal can be decomposed as $f = s_0 + w$, where s_0 is the best m-term approximation of f (*spikes*) and w is the remainder of the signal, called *external noise*. If $w = 0$, the analysis becomes quite simple. Indeed, in that case we exactly recover a constant fraction of spikes in each pass; so we will exactly recover the signal f in $O(\log m)$ passes.

In presence of the external noise $w \neq 0$, we can still recover a constant fraction of spikes in the first pass, however with error whose ℓ_1 norm is proportional to the ℓ_1 norm of the noise w. This error forms the "internal noise", which will add to the external noise in the next round. So, *the total noise doubles at every round*. After the $\log_a m$ rounds (needed to recover all spikes), the error of recovery will become polynomial in m. This is clearly unacceptable: Theorem 1 claims the error to be logarithmic in m.

This calls for a more delicate analysis of the error. Instead of adding the internal noise as a whole to the original noise, we will show that the internal noise spreads out over the subsets of the random partitions. So, most of the measurements will contain a small fraction of the internal noise, which will yield a small error of recovery in the current round. The major difficulty is to prove that this spreading phenomenon is *uniform* – one isolation matrix spreads the internal noise for all signals f at once, with high probability. This is a quite delicate problem. Indeed, in the last passes a constant number of spikes remain in the signal, and we have to find them correctly. So, the spreading phenomenon must hold for all but a constant number of measurements. Allowing so few exceptional measurements would naturally involve a very weak probability of such phenomenon to hold. On the other hand, in the last passes the internal noise is very big (having accumulated in all previous passes). Yet we need the spreading phenomenon to be uniform in all possible choices of the internal noise. It may seem that the weak probability estimates would not be sufficient to control a big internal noise in the last passes.

We will resolve this difficulty by doing a "surgery" on the internal noise, decomposing it in pieces corresonding to the previous passes, proving corresponding uniform probability estimates for each of these pieces, and uniting them in the end. This yields the Isolation Theorem 2, which summarizes the properties of the isolation matrix.

4.2. Deterministic Part

The proof of Theorem 1 is by induction on the pass k. We will normalize the signal so that $\|w\|_1 = 1/(400000a)$.

We will actually prove a result stronger than Theorem 1. We will show that, in pass k, the signal has the form

$$f^{(k)} = s_k + w + \sum_{j=0}^{k-1} \nu_k \tag{1}$$

where s_k contains at most m_k spikes. The vector ν_j is the *internal noise* from pass j, which consists of $3m_j$ or fewer nonzero components with magnitudes at most $2m^{-j}$. When we have done with all passes, that is when $k = 1 + \log_a m$, we will have no more spikes in the signal ($m_k = 0$ thus $s_k = 0$). This at once implies Theorem 1.

The proof of (1) will only use the two *deterministic* properties of the isolation matrix. They are summarized in the following theorem, whose proof we will sketch in the next subsection.

Theorem 2 (Properties of Isolation Matrix). *The isolation matrix has the following quality with probability at least* $(1 - d^{-3})$. *If a signal has the form* (1) *for some pass* k, *then at least 99/100 of the trial submatrices have these two properties:*

1. *All but $\frac{1}{100} m_{k+1}$ spikes appear alone in a measurements, isolated from the other spikes.*
2. *Except for at most $\frac{1}{100} m_{k+1}$ of the measurements, the internal and external noise assigned to that measurement have ℓ_1 norm at most $\frac{1}{1000} m_k^{-1}$.*

The first lemma considers the performance of the algorithm in one of the 99/100 non-exceptional trials under an artificial assumption that will be removed in the second lemma.

Lemma 4.1 (One Trial, No Exceptions). *Suppose that the trial is not exceptional. Assume that each measurement contains at most one spike and that the external noise in each measurement is no greater than $\varepsilon = \frac{1}{1000} m_k^{-1}$. Then the trial constructs a list of at most m_k spikes.*

1. *If $\left|f^{(k)}(i)\right| > 2\varepsilon$ then the list contains a spike with position i and estimated value $f^{(k)}(i) \pm \varepsilon$.*
2. *If the list contains a spike with position i and $\left|f^{(k)}(i)\right| \leq 4\varepsilon$, then the estimated value of the spike is no more than 5ε in magnitude.*

We call list items that satisfy these estimates accurate.

The first lemma follows from simple observations about the performance of the bit tests and the algorithm. The second lemma removes the artificial assumption on the spikes and noise.

Lemma 4.2 (One Trial). *Suppose that the trial is not exceptional. Then the trial constructs a list of at most m_k spikes. All items in the list are accurate, except at most $\frac{3}{50} m_{k+1}$.*

Proof. [Proof sketch.] The list produced by the algorithm is stable. That is, if the noise in a measurement is too large, at most two entries in the list are ruined. If a spike is not isolated, it ruins at most one entry of the list. □

Lemma 4.3 (Combining Trials). *The number of list items that are inaccurate in more than 1/10 of the trials is at most m_{k+1}. The total number of positions that appear in 9/10 of the trials is at most $\frac{10}{9} m_k$.*

Both parts follow by simple counting arguments. The list items that are inaccurate come from the accumulation of the $\frac{3}{50} m_{k+1}$ exceptional positions in Lemma 4.2. These become the spikes in the next pass.

Lemma 4.4 (Induction Hypothesis). *After pass k, there are at most m_{k+1} spikes remaining. The contribution ν_k to the internal noise contains at most $3m_k$ components with values at most $2/m_k$.*

Proof. [Proof sketch.] The list items that are inaccurate in more than 1/10 of the become spikes in the next pass. The list items that are identified in 9/10 of the trials consist of at least 8/10 accurate estimates, so medians over all values also yield accurate estimates. Since accurate estimates are either close to the actual spike value or close to zero, the value of the residual signal in these locations is less than $9\varepsilon < m_k^{-1}$. The difference between the spikes in the signal $f^{(k)}$ and the large entries in the update signal yields contains at most $m_{k+1} + m_k + \frac{10}{9} m_k < 3m_k$ terms of size less than $m_k^{-1} + m_{k+1}^{-1} < 2m_k^{-1}$. □

An immediate consequence of Lemma 4.4 is (1) for pass $k+1$, completing the proof.

4.3. Probabilistic Part

Here we prove that the random isolation matrix indeed has the two properties described in Theorem 2.

Lemma 4.5 (Isolations). *With probability at least $1 - \exp\{-4m_k \log d\}$, the following is true. In pass k, at least 99/100 of the trial submatrices isolate all but $\frac{1}{100}$ of the m_k spikes.*

Proof. [Proof sketch.] A standard martingale argument shows that the probability that one trial fails to isolate a $\frac{1}{100}$ fraction of the m_k spikes is $\mathrm{e}^{-O(m_k)}$. Apply the Chernoff bound over trials to obtain the result. □

Lemma 4.6 (Noise control). *With probability at least $1 - \exp\{-4m_k \log d\}$, the following is true. In pass k, in every trial, the number of measurements where the ℓ_1 norm of the external noise exceeds $\frac{1}{2000} m_k^{-1}$ is at most $\frac{1}{200} m_{k+1}$. In pass k, in at least 99/100 trials, the number of measurements where the ℓ_1 norm of the internal noise exceeds $\frac{1}{2000} m_k^{-1}$ is at most $\frac{1}{200} m_{k+1}$.*

Proof. [Proof sketch.] The external noise is controlled by Markov's inequality and the fact that each trial submatrix $\boldsymbol{A}_t^{(k)}$ has (1,1) operator norm equal to one.

The internal noise control is a delicate statement, which forms a main technical part of the argument. First we pass from the random partition model to a model where measurements are independent, using conditioning. Each piece ν_j of the internal noise exceeds $\lambda_j m_k$ in a single measurement with probability $\mathrm{e}^{-O(m_j T_k)}$. We apply Chernoff's bound to find the probability that more than $\varepsilon_j m_k T_k$ measurements are noisy during all T_k trials. Then Markov's inequality controls the number of trials where more than $\varepsilon_j m_k$ measurements are contaminated. The numbers λ_j and ε_j are chosen so that the quantitative pigeonhole principle yields a bound on the total number of measurements with internal noise greater than $\frac{1}{2000} m_k^{-1}$. We must refer the reader to a forthcoming technical report for details. □

ACKNOWLEDGMENTS

Gilbert and Strauss have been supported in part by NSF DMS 0354600. Tropp has been supported by NSF DMS 0503299. Vershynin is an Alfred P. Sloan Research Fellow, and he was partially supported by NSF DMS 0401032.

REFERENCES

1. G. Cormode and S. Muthukrishnan, "What's hot and what's not: Tracking most frequent items dynamically," in *Proc. ACM Principles of Database Systems*, pp. 296–306, 2003.
2. A. C. Gilbert, S. Guha, P. Indyk, S. Muthukrishnan, and M. J. Strauss, "Near-optimal sparse Fourier representations via sampling," in *ACM Symposium on Theoretical Computer Science*, 2002.
3. 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.
4. G. Cormode and S. Muthukrishnan, "Towards an algorithmic theory of compressed sensing," tech. rep., DIMACS, July 2005.
5. E. J. Candès, M. Rudelson, T. Tao, and R. Vershynin, "Error correction via linear programming," in *Proc. FOCS 2005*, (Pittsburgh), Oct. 2005.
6. D. L. Donoho, "Compressed sensing." Unpublished manuscript, Oct. 2004.
7. E. J. Candès and T. Tao, "Near optimal signal recovery from random projections: Universal encoding strategies?." Submitted for publication, Nov. 2004.
8. E. Candès, J. Romberg, and T. Tao, "Exact signal reconstruction from highly incomplete frequency information." Submitted for publication, June 2004.
9. D. L. Donoho, "Neighborly polytopes and sparse solution of underdetermined linear equations," Dept. of Statistics TR 2005-4, Stanford Univ., Feb. 2005.
10. D. L. Donoho and J. Tanner, "Sparse nonnegative solution of underdetermined linear equations by linear programming," Dept. of Statistics TR 2005-6, Stanford Univ., Apr. 2005.
11. E. J. Candès and T. Tao, "Decoding by linear programming." Available from `arXiv:math.MG/0502327`, Feb. 2005.
12. M. Rudelson and R. Veshynin, "Geometric approach to error correcting codes and reconstruction of signals." Available from `arXiv:math.MG/0502299`, Feb. 2005.
13. J. A. Tropp and A. C. Gilbert, "Signal recovery from partial information via Orthogonal Matching Pursuit." Submitted to *IEEE Trans. Inform. Theory*, April 2005.
14. S. Mendelson, A. Pajor, and N. Tomczak-Jaegermann, "Reconstruction and subgaussian processes," *Comptes Rendus Acad. Sci.* **340**, pp. 885–888, 2005.
15. M. Talagrand, *The Generic Chaining*, Springer, Berlin, 2005.
16. P. Indyk and A. Naor, "Nearest neighbor preserving embeddings." Submitted, 2005.