Algorithmic linear dimension reduction in the ℓ_1 norm for sparse vectors

A. C. Gilbert, M. J. Strauss, J. A. Tropp, and R. Vershynin

***Abstract*— Using a number of different algorithms, we can recover approximately a sparse signal with limited noise, *i.e*, a vector of length d with at least $d-m$ zeros or near-zeros, using little more than $m\log(d)$ nonadaptive linear measurements rather than the d measurements needed to recover an arbitrary signal of length d. We focus on two important properties of such algorithms.**

- **Uniformity. A single measurement matrix should work simultaneously for all signals.**
- **Computational Efficiency. The time to recover such an m-sparse signal should be close to the obvious lower bound, $m\log(d/m)$.**

This paper develops a new method for recovering m-sparse signals that is simultaneously uniform and quick. We present a reconstruction algorithm whose run time, $O(m\log^2(m)\log^2(d))$, is *sublinear* in the length d of the signal. The reconstruction error is within a logarithmic factor (in m) of the optimal m-term approximation error in ℓ_1. In particular, the algorithm recovers m-sparse signals perfectly and noisy signals are recovered with polylogarithmic distortion. Our algorithm makes $O(m\log^2(d))$ measurements, which is within a logarithmic factor of optimal. We also present a small-space implementation of the algorithm.

These sketching techniques and the corresponding reconstruction algorithms provide an algorithmic dimension reduction in the ℓ_1 norm. In particular, vectors of support m in dimension d can be linearly embedded into $O(m\log^2 d)$ dimensions with polylogarithmic distortion. We can reconstruct a vector from its low-dimensional sketch in time $O(m\log^2(m)\log^2(d))$. Furthermore, this reconstruction is stable and robust under small perturbations.

I. INTRODUCTION

We say that a metric space (X, d_X) embeds into a metric space (Y, d_Y) with distortion D if there are positive numbers A, B such that $B/A \le D$ and a map $\mathbf{\Phi} : X \to Y$ such that

$$A\, d_X(x,y) \le d_Y(\mathbf{\Phi}(x), \mathbf{\Phi}(y)) \le B\, d_X(x,y) \quad \text{for all } x, y \in X. \tag{I.1}$$

A fundamental problem is to understand when a finite metric space, which is isometrically embedded in some normed space X, admits a dimension reduction; *i.e.*, when we can embed it in an appropriate normed space Y of low dimension. Dimension reduction techniques enjoy a wide variety of algorithmic applications, including data stream computations [1], [2] and approximate searching for nearest neighbors [3] (to cite just a few). The dimension reduction result of Johnson and Lindenstrauss [4] is a fundamental one. It states that any set of N points in ℓ_2 can be embedded in ℓ_2^n with distortion $(1+\epsilon)$ and where the dimension $n = O(\log(N)/\epsilon^2)$.

A similar problem in the ℓ_1 space had been a longstanding open problem; Brinkman and Charikar [5] solved it in the negative (see another example in [6]). There exists a set of N points in ℓ_1 such that any embedding of it into ℓ_1^n with distortion D requires $n = N^{\Omega(1/D^2)}$ dimensions. Thus, a dimension reduction in ℓ_1 norm with constant distortion is not possible. However, it is well known how to do such a dimension reduction with a logarithmic distortion. One first embeds any N-point metric space into ℓ_2 with distortion $O(\log N)$ using Bourgain's theorem [7], then does dimension reduction in ℓ_2 using Johnson-Lindenstrauss result [4], and finally embeds ℓ_2^n into ℓ_1^{2n} with constant distortion using Kashin's theorem ([8], see Corollary 2.4 in [9]). For linear embeddings $\mathbf{\Phi}$, even distortions of polylogarithmic order are not achievable. Indeed, Charikar and Sahai [10] give an example for which any linear embedding into ℓ_1^n incurs a distortion $\Omega(\sqrt{N/n})$.

Two fundamental questions arise from the previous discussion.

1) What are spaces for which a dimension reduction in the ℓ_1 norm is possible with constant distortion?
2) What are spaces for which a linear dimension reduction in the ℓ_1 norm is possible with constant or polylogarithmic distortion?

One important space which addresses question (2) positively consists of all vectors of small support. Charikar and Sahai [10] prove that the space of vectors of support m in dimension d can be linearly embedded into ℓ_1^n with distortion $1+\epsilon$ with respect to the ℓ_1 norm, where $n = O((m/\epsilon)^2 \log d)$ (Lemma 1 in [10]). They do not, however, give a reconstruction algorithm for such signals and their particular embedding does not lend itself to an efficient algorithm.

The main result of our paper in an *algorithmic* linear dimension reduction for the space of vectors of small support. The algorithm runs in sublinear time and is stable.

Theorem 1: Let Y be a set of points in $\mathbb{R}^d$ endowed with the ℓ_1 norm. Assume that each point has non-zero coordi-

ACG is an Alfred P. Sloan Research Fellow and has been supported in party by NSF DMS 0354600. MJS has been supported in party by NSF DMS 0354600 and NSF DMS 0510203. JAS has been supported by NSF DMS 0503299. ACG, MJS, and JAT have been partially supported by DARPA ONR N66001-06-1-2011. RV is an Alfred P. Sloan Research Fellow. He was also partially supported by NSF DMS 0401032.

Gilbert and Tropp are with the Department of Mathematics, The University of Michigan at Ann Arbor, 2074 East Hall, 530 Church St., Ann Arbor, MI 48109-1043. Email: {annacg, jtropp}@umich.edu

Strauss is with the Department of Mathematics and the Department of Electrical Engineering and Computer Science, The University of Michigan at Ann Arbor. Email: martinjs@umich.edu

Vershynin is with the Department of Mathematics, The University of California at Davis, Davis, CA, 95616. Email: veryshnin@math.ucdavis.edu

nates in at most m dimensions. Then these points can be linearly embedded into ℓ_1 with distortion $O(\log^2(d)\log^3(m))$, using only $O(m\log^2 d)$ dimensions. Moreover, we can reconstruct a point from its low-dimensional sketch in time $O(m\log^2(m)\log^2(d))$.

This dimension reduction reduces the quadratic order of m in [10] to a linear order. Our embedding does, however, incur a distortion of polylogarithmic order. In return for this polylogarithmic distortion, we gain an *algorithmic linear dimension reduction*—there exists a sublinear time algorithm that can reconstruct every vector of small support from its low-dimensional sketch.

The space of vectors of support m in dimension d is a natural and important space as it models closely the space of compressible signals. 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 entries, sorted by decaying magnitude, satisfy $|f|_{(i)} \le r\, i^{-1/p}$.

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.

One important algorithmic application of our dimension reduction is the reconstruction of compressible signals. This paper describes a method for constructing a random linear operator $\mathbf{\Phi}$ that maps each signal f of length d to a sketch of size $O(m\log^2 d)$. We exhibit an algorithm called Chaining Pursuit that, given this sketch and the matrix $\mathbf{\Phi}$, constructs an m-term approximation of the signal with an error that is within a logarithmic factor (in m) of the optimal m-term approximation error. A compressible signal is well-approximated by an m-sparse signal so the output of Chaining Pursuit is a good approximation to the original signal, in addition to being a compressed represention of the original signal. Moreover, this measurement operator succeeds simultaneously for all signals with high probability. In many of the above application settings, we have resource-poor encoders which can compute a few random dot products with the signal but cannot store the entire signal nor take many measurements of the signal. The major innovation of this result is to combine sublinear reconstruction time with stable and robust linear dimension reduction of all compressible signals.

Let f_m denote the best m-term representation for f; *i.e.*, f_m consists of f restricted to the m positions that have largest-magnitude coefficients.

Theorem 2: With probability at least $(1 - O(d^{-3}))$, the random measurement operator $\mathbf{\Phi}$ has the following property. Suppose that f is a d-dimensional signal whose best m-term approximation with respect to ℓ_1 norm is f_m. Given the sketch $V = \mathbf{\Phi} f$ of size $O(m\log^2(d))$ and the measurement matrix $\mathbf{\Phi}$, the Chaining Pursuit algorithm produces a signal $\widehat{f}$ with at most m nonzero entries. The output $\widehat{f}$ satisfies

$$\|f - \widehat{f}\|_1 \le C(1+\log m)\big\|f - f_m\big\|_1. \qquad \text{(I.2)}$$

In particular, if $f_m = f$, then also $\widehat{f} = f$. The time cost of the algorithm is $O(m\log^2(m)\log^2(d))$.

The factor $\log m$ is intrinsic to this approach. However, the proof gives a stronger statement.

Corollary 3: The approximation in the weak-1 norm does *not* include the factor: $\|f - \widehat{f}\|_{\text{weak}-1} \le C\|f - f_m\|_1$.

This follows directly from the definition of the weak norm and our proof of the main theorem.

Corollary 4: Our argument shows that the reconstruction $\widehat{f}$ is not only stable with respect to noise in the signal, as Equation (I.2) shows, but also with respect to inaccuracy in the measurements. Indeed, a stronger inequality holds. For every V (not necessarily the sketch $\mathbf{\Phi} f$ of f) if $\widehat{f}$ is the reconstruction from V (not necessarily from $\mathbf{\Phi} f$), we have

$$\|f_m - \widehat{f}\|_1 \le C(1+\log m)\Big(\|f - f_m\|_1 + \|\mathbf{\Phi} f - V\|_1\Big).$$

A. Related Work

The problem of sketching and reconstructing m-sparse and compressible signals has several precedents in the Theoretical Computer Science literature, especially the paper [1] on detecting heavy hitters in nonnegative data streams and the works [11], [12] on Fourier sampling. More recent papers from Theoretical Computer Science include [13], [14]. Sparked by the papers [15] and [16], the computational harmonic analysis and geometric functional analysis communities have produced an enormous amount of work, including [17], [18], [19], [20], [21], [22], [23].

Most of the previous work has focused on a reconstruction algorithm that involves linear programming (as first investigated and promoted by Donoho and his collaborators) or second-order cone programming [15], [16], [14]. The authors of these papers do not report computation times, but they are expected to be cubic in the length d of the signal. This cost is high, since we are seeking an approximation that involves $O(m)$ terms. The paper [22] describes 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. None of these approaches is comparable with the sublinear algorithms described here.

There are a few sublinear algorithms available in the literature. The Fourier sampling paper [12] can be viewed as a small space, sublinear algorithm for signal reconstruction. Its primary shortcoming is that the measurements are not uniformly good for the entire signal class. The recent work [13] proposes some other sublinear algorithms for reconstructing compressible signals. Few of these algorithms offer a uniform guarantee. The ones that do require more

measurements—$O(m^2 \log d)$ or worse—which means that they are not sketching the signal as efficiently as possible.

B. Organization

In Section II, we provide an overview of determining a sketch of the signal f. In Section III, we give an explicit construction of a distribution from which the random linear map $\mathbf{\Phi}$ is drawn. In Section IV, we detail the reconstruction algorithm, Chaining Pursuit, and in Section V we give an overview of the analysis of the algorithm and sketch the proof of our main result. In Section VI we use our algorithmic analysis to derive a dimension reduction in the ℓ_1 norm for sparse vectors. The complete detailed analysis of our algorithm is in a companion journal paper and we encourage the interested reader to look there.

II. Sketching the Signal

This section describes a linear process for determining a sketch V of a signal f. Linearity is essential for supporting additive updates to the signal. Not only is this property important for applications, but it arises during the iterative algorithm for reconstructing the signal from the sketch. Linearity also makes the computation of the sketch straightforward, which may be important for modern applications that involve novel measurement technologies.

A. Overview of sketching process

We will construct our measurement matrix by combining simple matrices and ensembles of matrices. Specifically, we will be interested in restricting a signal f to a subset A of its d positions and then restricting to a smaller subset $B \subseteq A$, and it will be convenient to analyze separately the two stages of restriction. If P and Q are 0-1 matrices, then each row P_i of P and each row Q_j of Q restricts f to a subset by multiplicative action, $P_i f$ and $Q_j f$, and sequential restrictions are given by $P_i Q_j f = Q_j P_i f$. We use the following notation, similar to [13].

Definition 5: Let P be a p-by-d matrix and Q a q-by-d matrix, with rows $\{P_i : 0 \le i < p\}$ and $\{Q_j : 0 \le j < q\}$, respectively. The *row tensor product* $S = P \otimes_r Q$ of P and Q is a pq-by-d matrix whose rows are $\{P_i Q_j : 0 \le i < p,\ 0 \le j < q\}$, where $P_i Q_j$ denotes the componentwise product of two vectors of length d.

The order of the rows in $P \otimes_r Q$ will not be important in this paper. We will sometimes index the rows by the pair (i, j), where i indexes P and j indexes Q, so that $P \otimes_r Q$ applied to a vector x yields a $q \times p$ matrix.

Formally, the measurement operator $\mathbf{\Phi}$ is a row tensor product $\mathbf{\Phi} = \boldsymbol{B} \otimes_r \boldsymbol{A}$. Here, $\boldsymbol{A}$ is a $O(m \log d) \times d$ matrix called the *isolation matrix* and $\boldsymbol{B}$ is a $O(\log d) \times d$ matrix called the *bit test matrix*. The measurement operator applied to a signal f produces a sketch $V = \mathbf{\Phi} f$, which we can regard as a matrix with dimensions $O(m \log d) \times O(\log d)$. Each row of V as a matrix contains the result of the bit tests applied to a restriction of the signal f by a row of A. We will refer to each row of the data matrix as a *measurement* of the signal.

B. The isolation matrix

The isolation matrix $\boldsymbol{A}$ is a 0-1 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 two sections. The Chaining Pursuit algorithm makes $K = 1 + \log_a m$ passes (or "rounds") over the signal, and it requires a different set of measurements for each pass. The measurements for the kth pass are contained in the $O(mk \log(d)/2^k) \times d$ submatrix $\boldsymbol{A}^{(k)}$. During the kth 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 $O(m/2^k) \times d$.

In summary,

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

Each trial submatrix $\boldsymbol{A}_t^{(k)}$ encodes a random partition of the d signal positions into $O(m/2^k)$ subsets. That is, each signal position is assigned uniformly at random to one of $O(m/2^k)$ subsets. So the matrix contains a 1 in the (i, j) position if the jth component of the signal is assigned to subset i. Therefore, the submatrix $\boldsymbol{A}_t^{(k)}$ is a 0-1 matrix in which each column has exactly one 1, *e.g.*,

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

The trial submatrix can also be viewed as a random linear hash function from a space of d keys onto a set of $O(m/2^k)$ buckets.

C. The bit test matrix

Formally, the matrix $\boldsymbol{B}$ consists of a row $\mathbf{e}$ of 1's and other rows given by a 0-1 matrix $\boldsymbol{B_0}$, which we now describe. The matrix $\boldsymbol{B}_0$ has dimensions $\log_2 \lceil d \rceil \times d$. The ith column of B_0 is the binary expansion of i. Therefore, the componentwise product of the ith row of B_0 with f yields a copy of the signal f with the components that have bit i equal to one selected and the others zeroed out.

An example of a bit test matrix with $d = 8$ is

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

D. Storage costs

The bit test matrix requires no storage. The total storage for the isolation matrix is $O(d \log d)$. The space required for the isolation matrix is large, but this space can conceivably be shared among several instances of the problem. In Section III, we give an alternate construction in which a pseudorandom isolation matrix is regenerated as needed from

a seed of size $m \log^2(d)$; in that construction only the seed needs to be stored, so the total storage cost is $m \log^2(d)$.

E. Encoding time

The time cost for measuring a signal is $O(\log^2(m) \log^2(d))$ *per nonzero component*. This claim follows by observing that a single column of $\boldsymbol{A}$ contains $O(\log^2(m) \log(d))$ nonzero entries, and we must apply $\boldsymbol{A}$ to each of $O(\log d)$ restrictions of the signal—one for each row of B. Note that this argument assumes random access to the columns of the isolation matrix. We will use this encoding time calculation when we determine the time costs of the Chaining Pursuit algorithm. In Section III, we give an alternative construction for A that reduces the storage requirements at the cost of slightly increased time requirements. Nevertheless, in that construction, any m columns of A can be computed in time $m^{o(1)}$ each, where $o(1)$ denotes a quantity that tends to 0 as both m and d get large. This gives measurement time $m^{o(1)}$ per nonzero component.

III. Small Space Construction

We now discuss a small space construction of the isolation matrix, A. The goal is to specify a pseudorandom matrix A from a small random seed, to avoid the $\Omega(d \log d)$ cost of storing A explicitly. We then construct entries of A as needed, from the seed. If we were to use a standard pseudorandom number generator without further thought, however, the time to construct an entry of A might be $\Omega(m)$, compared with $O(1)$ for a matrix that is fully random and explicitly stored. We will give a construction that addresses both of these concerns.

As discussed in Section II-B, the matrix A consists of $\operatorname{polylog}(d)$ submatrices that are random partitions of the d signal positions into $O(m_k)$ subsets. In this section, we give a new construction for each submatrix; the submatrices fit together to form A in the same way as in Section II-B. We will see from the analysis in Section V, the partition map of each random submatrix need only be m_k-wise independent; full independence is not needed as we need only control the allocation of m_k spikes into measurements in each submatrix. It follows that we need only construct a family of d random variables that are m_k-wise independent and take values in $\{0, \dots, r-1\}$ for any given $r \leq d$. Our goal is to reduce the storage cost from $O(d \log d)$ to $m \operatorname{polylog}(d)$ without unduly increasing the computation time. It will require time $\Omega(m)$ to compute the value of any *single* entry in the matrix, but we will be able to compute any submatrix of m columns (which is all zeros except for one 1 per column) in *total* time $m \operatorname{polylog}(d)$. That is, the values of any m random variables can be computed in time $m \operatorname{polylog}(d)$. Our construction will be allowed to fail with probability $1/d^3$, which will be the case. (Note that success probability $1 - e^{-cm \log d}$ is not required.) Our construction combines several known constructions from [24], [25] and for the sake of brevity, we omit the details.

A. Requirements

To ease notation, we consider only the case of $m_k = m$. Our goal is to construct a function $f_s : \{0, \dots, d-1\} \to \{0, \dots, r-1\}$, where s is a random seed. The construction should "succeed" with probability at least $1 - 1/d^3$; the remaining requirements only need to hold if the construction succeeds. The function should be uniform and m-wise independent, meaning, for any m distinct positions $0 \leq i_1, \dots, i_m < d$ and any m targets $t_1, \dots, t_m$, we have

$$\mathbb{P}_s(\forall j \; f_s(i_j) = t_j) = r^{-m},$$

though the distribution on $m+1$ random variables may otherwise be arbitrary. Finally, given any list A of m positions, we need to be able to compute $\{f(j) : j \in A\}$ in time $m \operatorname{polylog}(d)$.

B. Construction

Let $s = (s_0, s_1, \dots, s_K)$ be a sequence of $K \leq O(\log d)$ independent, identically distributed random bits. Let p be a prime with $p \geq 2r$ and $d \leq p \leq \operatorname{poly}(d)$. Define the map $g_s^k : \mathbb{Z}_p \to \mathbb{Z}_p$ which uses the kth element s_k from the seed s and maps $j \in \mathbb{Z}_p$ uniformly at random to a point $g_s^k(j) \in \mathbb{Z}_p$. The map g_s^k is a random polynomial of degree $m-1$ over the field with p elements. If

$$0 \leq g_s^o(j) < r \lfloor p/r \rfloor,$$

where $r \lfloor p/r \rfloor$ represents the largest multiple of r that is at most p, then define

$$f_s(j) = \lfloor g_s^0(j) r/p \rfloor = h(g_s^0(j)).$$

The function $h : \{0, \dots, r \lfloor p/r \rfloor - 1\} \to \{0, \dots, r-1\}$ is a function that is exactly $\lfloor p/r \rfloor$-to-1. If $g_s^0(j) > r \lfloor p/r \rfloor$, then we map j to $\mathbb{Z}_p$ by $g_s^1(j)$, that is independent of g_s^0 and identically distributed. We repeat the process until j gets mapped to $\{0, \dots, r \lfloor p/r \rfloor - 1\}$ or we exhaust $K \leq O(\log d)$ repetitions. For computational reasons, for each k, we will compute g_s^k at once on all values in a list A of m values as a multipoint polynomial evaluation (MPE) and we will write $g_s^k(A)$ for the list $\{g_s^k(j) | j \in A\}$. Figure 1 gives a formal algorithm.

Using these known constructions, we conclude:

Theorem 6: There is an implementation of the Chaining Pursuit algorithm that runs in time $m \operatorname{polylog}(d)$ and requires total storage $m \log(d)$ numbers bounded by $\operatorname{poly}(d)$ (*i.e.*, $O(\log d)$ bits).

IV. Signal Approximation with Chaining Pursuit

Suppose that the original signal f is well-approximated by a signal with m nonzero entries (spikes). The goal of the Chaining Pursuit algorithm is to use a sketch of the signal to obtain a signal approximation with no more than m spikes. To do this, the algorithm first finds an intermediate approximation g with possibly more than m spikes, then returns g_m, the restriction of g to the m positions that maximize the coefficient magnitudes of g. We call the final step of the algorithm the *pruning* step. The algorithm *without* the pruning step will be called *Chaining Pursuit Proper*.

Fig. 1. Top-level description of m-wise independent random variables.

Algorithm: Hashing

Parameters: m, r, d, K
Input: List A of m values in $\mathbb{Z}_d$; pseudorandom seed s.
Output: List B of m values in $\{0, \ldots, r-1\}$, rep'ing $\langle f_s(j) : j \in A \rangle$.

For each $k = 0, 1, \ldots, K-1$:
  Compute $g_s^k(A)$ as MPE
If for some $j \in A$, for all $k < K$, we have $g_s^k(j) \geq r\lfloor p/r \rfloor$, then FAIL
  For $j \in A$
    $k_j = \min\{k : g_s^k(j) < r\lfloor p/r \rfloor\}$
    $f_s(j) = g_s^{k_j}(j)$.

The Chaining Pursuit Proper algorithm proceeds in passes. In each pass, the algorithm recovers a constant fraction of the remaining spikes. Then it sketches the recovered spikes and updates the data matrix to reflect the residual signal—the difference between the given signal and the superposition of the recovered spikes. After $O(\log m)$ passes, the residual has no significant entries remaining.

The reason for the name "Chaining Pursuit" is that this process decomposes the signal into pieces with supports of geometrically decreasing sizes. 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 is to establish bounds on the expected supremum of an empirical process [26]. For an example of chaining in Theoretical Computer Science, see [3].

A. Overview of Algorithm

The structure of the Chaining algorithm is similar to other sublinear approximation methods described in the literature [11]. First, the algorithm identifies spike locations and estimates the spike magnitudes. Then it encodes these spikes and subtracts them from the sketch to obtain an implicit sketch of the residual signal. These steps are repeated until the number of spikes is reduced to zero. The number a that appears in the statement of the algorithm is a sufficiently large constant that will be discussed further in Section V and the quantity m_k is m/a^k. Pseudocode is given in Figure 2.

B. Implementation

Most of the steps in this algorithm are straightforward to implement using standard abstract data structures. The only point that requires comment is the application of bit tests to identify spike positions and values.

Recall that a measurement is a row of the sketch matrix, which consists of $\log_2 \lceil d \rceil + 1$ numbers:

$$\begin{bmatrix} b(0) & b(1) & \ldots & b(\log_2 \lceil d \rceil - 1) & | & c \end{bmatrix}.$$

Fig. 2. Chaining Pursuit algorithm

Algorithm: Chaining Pursuit

Inputs: Number m of spikes, the sketch V, the isolation matrix $\boldsymbol{A}$
Output: A list of m spike locations and values

For each pass $k = 0, 1, \ldots, \log_a m$:
  For each trial $t = 1, 2, \ldots, O(k \log d)$:
    For each measurement $n = 1, \ldots, O(m/2^k)$
      Use bit tests to id. spike posn.
      Use a bit test to est. spike mag.
    Keep m_k dist. spikes with largest vals. (in mag.)
  Keep spike posns. in more than 9/10 trials
  Estimate final spike sizes using medians
  Encode spikes using meas. operator
  Subtract encoded spikes from sketch
Return sig. of m largest kept spikes.

The number c arises from the top row of the bit test matrix. We obtain an (estimated) spike location from these numbers as follows. If $|b(i)| \geq |c - b(i)|$, then the ith bit of the location is zero. Otherwise, the ith bit of the location is one. To estimate the value of the spike from the measurements, we use c.

Recall that each measurement arises by applying the bit test matrix to a copy of the signal restricted to a subset of its components. It is immediate that the estimated location and value are accurate if the subset contains a single large component of the signal and the other components have smaller ℓ_1 norm.

We encode the recovered spikes by accessing the columns of the isolation matrix corresponding to the locations of these spikes and then performing a sparse matrix-vector multiplication. Note that this step requires random access to the isolation matrix.

C. Storage costs

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

D. Time costs

During pass k, the primary cost of the algorithm occurs when we encode the recovered spikes. The number of recovered spikes is at most $O(m/a^k)$, so the cost of encoding these spikes is $O(ma^{-k} \log^2(m) \log^2(d))$. The cost of updating the sketch is the same. Summing over all passes, we obtain $O(m \log^2(m) \log^2(d))$ total running time.

V. Analysis of Chaining Pursuit

This section contains a detailed analysis of the Chaining Pursuit Proper algorithm (*i.e.*, Chaining Pursuit without the final pruning step), which yields the following theorem. Fix an isolation matrix $\boldsymbol{A}$ which satisfies the conclusions of Condition 10 in the sequel and let $\boldsymbol{\Phi} = \boldsymbol{A} \otimes_r \boldsymbol{B}$, where $\boldsymbol{B}$ is a bit test matrix.

Theorem 7 (Chaining Pursuit Proper): Suppose that f is a d-dimensional signal whose best m-term approximation with respect to ℓ_1 norm is f_m. Given the sketch $V = \boldsymbol{\Phi} f$ and the matrix $\boldsymbol{\Phi}$, Chaining Pursuit Proper produces a signal $\widehat{f}$ with at most $O(m)$ nonzero entries. 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 $\widehat{f} = f$.

A. Overview of the analysis

Chaining Pursuit Proper is an iterative algorithm. Intuitively, at some iteration k, we have a signal that consists of a limited number of spikes (positions whose coefficient is large) and noise (the remainder of the signal). We regard the application of the isolation matrix $\boldsymbol{A}$ as repeated trials of partitioning the d signal positions into $\Theta(m_k)$ random subsets, where m_k is approximately the number of spikes, and approximately the ratio of the 1-norm of the noise to the magnitude of spikes. There are two important phenomena:

- A measurement may have exactly one spike, which we call *isolated*.
- A measurement may get approximately its fair share of the noise—approximately the fraction $1/\mu$ if μ is the number of measurements.

If both occur in a measurement, then it is easy to see that the bit tests will allow us to recover the position of the spike and a reasonable estimate of the coefficient (that turns out to be accurate enough for our purposes). With high probability, this happens to many measurements.

Unfortunately, a measurement may get zero spikes, more than one spike, and/or too much noise. In that case, the bit tests may return a location that does not correspond to a spike and our estimate of the coefficient may have error too large to be useful. In that case, when we subtract the "recovered" spike from the signal, we actually introduce additional spikes and *internal* noise into the signal. We bound both of these phenomena. If we introduce a false spike, our algorithm has a chance to recover it in future iterations. If we introduce a false position with small magnitude, however, our algorithm may not recover it later. Thus the internal noise may accumulate and ultimately limit the performance of our algorithm—this is the ultimate source of the logarithmic factor in our accuracy guarantee.

In pass $k = 0$, the algorithm is working with measurements of the original signal f. This signal can be decomposed as $f = f_m + w$, where f_m 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 this respect, Chaining is superficially similar to, *e.g.*, [11]. An important difference is that, in the analysis of Chaining pursuit, we exploit the fact that a fraction of spikes is recovered except with probability exponentially small in the number of spikes; this lets us unite over all configurations of spike positions and, ultimately, to get a uniform failure guarantee.

The major difficulty of the analysis here concerns controlling the approximation error from blowing up in a geometric progression from pass to pass. More precisely, while it is comparatively easier to show that, for *each* signal, the error remains under control, providing a uniform guarantee—such as we need—is more challenging. In presence of the external noise $w \neq 0$, we can still recover a constant fraction of spikes in the first pass, although 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 7 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 "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 leads to Condition 10, which summarizes the needed properties of the isolation matrix.

The proof of Theorem 7 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 7. The following is our central loop invariant:

Invariant 8: In pass k, the signal has the form

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

where s_k contains at most m_k spikes, $w = f - f_m$ is the external noise, and each vector ν_j is the *internal noise* from pass j, which consists of $3m_j$ or fewer nonzero components with magnitudes at most $2/m_j$.

When we have finished 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 7.

The proof that Invariant 8 is maintained will only use two properties of an isolation matrix, given in Condition 10. While we only know how to construct such matrices using randomness, any matrix satisfying these properties is acceptable. We must show that Invariant 8 holds for any matrix $\mathbf{\Phi}$ having the properties in Condition 10. We also must show that most matrices (according to the definition implicit in Section II-B) satisfy these properties[1].

Theorem 9: With probability at least $(1 - O(d^{-3}))$, a matrix $\boldsymbol{A}$ drawn from the distribution described in Section II-B satisfies the Chaining Recovery Conditions (Conditions 10). Note that the conditions are given in terms of matrix actions upon certain kinds of signals, but the conditions are properties only of matrices.

Condition 10 (Recovery Conditions for Isolation Matrices): A 0-1 matrix with pass/trial hierarchical structure described in Section II-B (*i.e.*, any matrix from the sample space described in Section II-B) is said to satisfy the *Chaining Recovery Conditions* if for any signal of the form in Invariant 8 and for any 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 measurement, 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 each measurement has ℓ_1 norm at most $\frac{1}{1000} m_k^{-1}$.

B. Robustness

In this subsection, we prove Corollary 4. As advertised in the introduction, the Chaining Pursuit algorithm is not only stable with respect to noise in the signal but also robust to inaccuracy or errors in the measurements. Suppose that instead of using the sketch $\mathbf{\Phi} f$ of the signal f, e receive $V = \mathbf{\Phi} f + y$ and we reconstruct $\widehat{f}$ from V. We assume that once we carry out the Chaining Pursuit algorithm, there are no perturbations to the intermediate measurements, only to the original sketch $\mathbf{\Phi} f$.

Corollary 11: With probability at least $(1 - O(d^{-3}))$, the random measurement operater $\mathbf{\Phi}$ has the following property. Suppose that f is a d-dimensional signal whose best m-term approximation with respect to the ℓ_1 norm is f_m. Given the measurement operator $\mathbf{\Phi}$, for every V (not necessarily the sketch $\mathbf{\Phi} f$ of f), if $\widehat{f}$ is the reconstruction from V, then

$$\|f - f_m\|_1 \leq C(1 + \log(m)) \Big(\|f - f_m\|_1 + \|\mathbf{\Phi} f - V\|_1 \Big).$$

[1]For details of these proofs, please see the extended journal version of this manuscript.

VI. Algorithmic Dimension Reduction

The following dimension reduction theorem holds for sparse vectors.

Theorem 12: Let X be the union of all m-sparse signals in $\mathbb{R}^d$ and endow $\mathbb{R}^d$ with the ℓ_1 norm. The linear map $\mathbf{\Phi} : \mathbb{R}^d \to \mathbb{R}^n$ in Theorem 2 satisfies

$$A\|f - g\|_1 \leq \|\mathbf{\Phi}(f) - \mathbf{\Phi}(g)\|_1 \leq B\|f - g\|_1$$

for all f and g in X, where $1/A = C \log(m)$ and $B = C \log^2(m) \log^2(d)$ and $n = O(m \log^2 d)$.

Proof: The upper bound is equivalent to saying that the $\ell_1 \to \ell_1$ operator norm satisfies $\|\mathbf{\Phi}\|_{1 \to 1} \leq B$. This norm is attained at an extreme point of the unit ball of ℓ_1^d, which is thus at a point with support 1. Then the upper bound follows at once from the definition of $\mathbf{\Phi}$. That is, any 0-1 vector of support 1 gets mapped by $\mathbf{\Phi}$ to a 0-1 vector of support bounded by the total number of bit-tests in all trials and passes, which is $\sum_{k=0}^{\log_a m} O(k \log d) \log_2 d \leq B$.

The lower bound follows from Theorem 2. Let f and g be any d-dimensional signals with support m, so that $f = f_m$ and $g = g_m$. Let $V = \mathbf{\Phi} g$. Then the reconstruction $\widehat{f}$ from V will be exact: $\widehat{f} = g$. As proven in Corollary 4,

$$\begin{aligned} \|f - g\|_1 &= \|f - \widehat{f}\|_1 \\ &\leq C \log(m) \left(\big\|f - f_m\big\|_1 + \big\|\mathbf{\Phi} f - V\big\|_1 \right) \\ &= C \log(m) \, \big\|\mathbf{\Phi} f - \mathbf{\Phi} g\big\|_1, \end{aligned}$$

which completes the proof. ∎

We are interested not only in the distortion and dimension reduction properties of our embedding but also in the stability and robustness properties of the embedding. Our previous analysis guarantees that $\mathbf{\Phi}^{-1}\mathbf{\Phi}$ is the identity on X and that the inverse can be computed in sublinear time since Chaining Pursuit Proper perfectly recovers m-sparse signals. Our previous analysis also shows that our dimension reduction is stable and robust. In other words, our embedding and the reconstruction algorithm can tolerate errors η in the data $x \in X$, as well as errors ν in the measurements:

Theorem 13: The linear map $\mathbf{\Phi} : \mathbb{R}^d \to \mathbb{R}^n$ in Theorem 2 and the reconstruction map $\mathbf{\Psi} : \mathbb{R}^n \to \mathbb{R}^d$ given by the Chaining Pursuit Proper algorithm satisfy the following for every $\eta \in \mathbb{R}^d$ and every $\nu \in \mathbb{R}^n$ and for all m-sparse signals x in $\mathbb{R}^d$:

$$\|x - \mathbf{\Psi}(\mathbf{\Phi}(x + \eta) + \nu)\|_1 \leq (1 + \mathrm{C} \log m)(\|\eta\|_1 + \|\nu\|_1).$$

Proof: This is just a reformulation of our observations in Corollary 4 with $x = f_m$, $\eta = f - f_m$, $\nu = \mathbf{\Phi} f - V$. ∎

VII. Conclusions

We have presented the first algorithm for recovery of a noisy sparse vector from a nearly optimal number of non-adaptive linear measurements that satisfies the following two desired properties:

- A single uniform measurement matrix works simultaneously for all signals.
- The recovery time is, up to log factors, proportional to the size of the *output*, not the length of the vector.

The output of our algorithm has error with ℓ_1-norm bounded in terms of the ℓ_1-norm of the optimal output. Elsewhere in the literature, *e.g.*, in [16], [27], [28], the ℓ_2-norm of the output error is bounded in terms of the ℓ_1-norm of the optimal error, a mixed-norm guarantee that is somewhat stronger than the result we give here.

If the measurement matrix is a random Gaussian matrix, as in [16], [27], [28], the measurement matrix distribution is invariant under unitary transformations. It follows that such algorithms support recovery of signals that are sparse in a basis *unknown at measurement time*. That is, one can measure a signal f^* as $V = \mathbf{\Phi} f^*$. Later, one can decide that f^* can be written as $f^* = Sf$, where S is an arbitrary unitary matrix independent of $\mathbf{\Phi}$ and f is a noisy sparse vector of the form discussed above. Thus $V = (\mathbf{\Phi} S)f$, where $\mathbf{\Phi} S$ is Gaussian, of the type required by the recovery algorithm. Thus, given $V, \mathbf{\Phi}$, and S, the algorithms of [16], [27], [28] can recover f.

If the matrix S is known at measurement time, our algorithm can substitute $\mathbf{\Phi} S$ for $\mathbf{\Phi}$ *at measurement time* and proceed without further changes. If S is unknown at measurement time, however, our algorithm breaks down. But note that an important point of our algorithm is to provide decoding in time $m \operatorname{polylog}(d)$, which is clearly not possible if the decoding process must first read an arbitrary unitary d-by-d matrix S. Once a proper problem has been formulated, it remains interesting and open whether sublinear-time decoding is compatible with basis of sparsity unknown at measurement time.

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*, 2003, pp. 296–306.

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

[3] P. Indyk and A. Naor, "Nearest neighbor preserving embeddings," 2005, submitted.

[4] W. B. Johnson and J. Lindenstrauss, "Extensions of lipschitz mapping into hilbert space," *Contemporary Mathematics*, no. 26, pp. 189–206, 1984.

[5] B. Brinkman and M. Charikar, "On the impossibility of dimension reduction in ℓ_1," in *Proceedings of the 44th Annual IEEE Conference on Foundations of Computer Science (2003)*, 2003.

[6] A. Naor and J. R. Lee, "Embedding the diamond graph in l_p and dimension reduction in l_1," *Geometric and Functional Analysis*, vol. 14(4), pp. 745–747, 2004.

[7] J. Bourgain, "On lipschitz embedding of finite metric spaces in hilbert space," *Israel J. Math.*, vol. 52, pp. 46–52, 1985.

[8] B. Kashin, "Sections of some finite dimensional sets and classes of smooth functions," *Izv. Acad. Nauk SSSR*, vol. 41, pp. 334–351, 1977.

[9] G. Pisier, *The volume of convex bodies and Banach space geometry*. Cambridge University Press, 1989.

[10] M. Charikar and A. Sahai, "Dimension reduction in the ℓ_1 norm," in *Proceedings of the 43rd Annual IEEE Conference on Foundations of Computer Science (2002)*. IEEE Press, 2002.

[11] 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.

[12] 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.

[13] G. Cormode and S. Muthukrishnan, "Towards an algorithmic theory of compressed sensing," DIMACS, Tech. Rep., July 2005.

[14] E. J. Candès, M. Rudelson, T. Tao, and R. Vershynin, "Error correction via linear programming," in *Proc. FOCS 2005*, Pittsburgh, Oct. 2005.

[15] D. L. Donoho, "Compressed sensing," Oct. 2004, unpublished manuscript.

[16] E. J. Candès and T. Tao, "Near optimal signal recovery from random projections: Universal encoding strategies?" Oct. 2004, submitted for publication, revised April 2005.

[17] E. Candès, J. Romberg, and T. Tao, "Exact signal reconstruction from highly incomplete frequency information," June 2004, submitted for publication.

[18] D. L. Donoho, "Neighborly polytopes and sparse solution of underdetermined linear equations," Stanford Univ., Dept. of Statistics TR 2005-4, 2005.

[19] D. L. Donoho and J. Tanner, "Sparse nonnegative solution of underdetermined linear equations by linear programming," Stanford Univ., Dept. of Statistics TR 2005-6, Apr. 2005.

[20] E. J. Candès and T. Tao, "Decoding by linear programming," Feb. 2005, available from arXiv:math.MG/0502327. [Online]. Available: arXiv:math.MG/0502327

[21] M. Rudelson and R. Veshynin, "Geometric approach to error correcting codes and reconstruction of signals," Feb. 2005, available from arXiv: math.MG/0502299. [Online]. Available: arXiv:math.MG/0502299

[22] J. A. Tropp and A. C. Gilbert, "Signal recovery from partial information via Orthogonal Matching Pursuit," April 2005, submitted to *IEEE Trans. Inform. Theory*.

[23] S. Mendelson, A. Pajor, and N. Tomczak-Jaegermann, "Reconstruction and subgaussian processes," *Comptes Rendus Acad. Sci.*, vol. 340, pp. 885–888, 2005.

[24] A. Aho, J. E. Hopcroft, and J. D. Ullman, *Data structures and algorithms*. Reading, Mass.: Addison-Wesley, 1983.

[25] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. The MIT Press, 2001.

[26] M. Talagrand, *The Generic Chaining*. Berlin: Springer, 2005.

[27] M. Rudelson and R. Veshynin, "Sparse reconstruction by convex relaxation: Fourier and Gaussian measurements," in *Proc. 40th IEEE Conference on Information Sciences and Systems*, Mar. 2006.

[28] A. Cohen, W. Dahmen, and R. DeVore, "Remarks on compressed sensing," 2006, working draft.