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. 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. Several research communities are interested in techniques for measuring and recovering such signals and a variety of approaches have been proposed. 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, $m \log(d/m)$.

To date, algorithms for signal recovery that provide a uniform measurement matrix with approximately the optimal number of measurements, such as first proposed by Donoho and his collaborators, and, separately, by Candès and Tao, are based on linear programming and require time $\operatorname{poly}(d)$ instead of $m \operatorname{polylog}(d)$. On the other hand, fast decoding algorithms to date from the Theoretical Computer Science and Database communities fail with probability at least $1/\operatorname{poly}(d)$, whereas we need failure probability no more than around $1/d^m$ to achieve a uniform failure guarantee.

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.

1. 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 $B/A \leq D$ and a map $\mathbf{\Phi} : X \to Y$ such that

$$A \, d_X(x, y) \leq d_Y(\mathbf{\Phi}(x), \mathbf{\Phi}(y)) \leq B \, d_X(x, y) \quad \text{for all } x, y \in X. \tag{1.1}$$

Date: February 1, 2008.

Key words and phrases. Approximation, embedding, dimension reduction, sketching, sparse approximation, sublinear algorithms.

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. E-mail: {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. E-mail: martinjs@umich.edu. Vershynin is with the Department of Mathematics, The University of California at Davis, Davis, CA 95616. E-mail: vershynin@math.ucdavis.edu. ACG is an Alfred P. Sloan Research Fellow and has been supported in part by NSF DMS 0354600. MJS has been supported in part by NSF DMS 0354600 and NSF DMS 0510203. JAT 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 the NSF grant DMS 0401032.

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 [CM03, GGI$^+$02a] and approximate searching for nearest neighbors [IN05] (to cite just a few). The dimension reduction result of Johnson and Lindenstrauss [JL84] 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 [BC03] solved it in the negative (see another example in [NL04]). 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 [Bou85], then does dimension reduction in ℓ_2 using Johnson-Lindenstrauss result [JL84], and finally embeds ℓ_2^n into ℓ_1^{2n} with constant distortion using Kashin's theorem ([Kas77], see Corollary 2.4 in [Pis89]). For linear embeddings $\mathbf{\Phi}$, even distortions of polylogarithmic order are not achievable. Indeed, Charikar and Sahai [CS02] 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 [CS02] 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 [CS02]). 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 coordinates 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 [CS02] 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 $|f|_{(i)} \leq r\, i^{-1/p}$.

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 manyof 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*

$$\|f - \widehat{f}\|_1 \leq C(1 + \log m)\big\|f - f_m\big\|_1. \tag{1.2}$$

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

Corollary 3. *The factor $\log m$ is intrinsic to this approach. However, the proof gives a stronger statement—the approximation in the weak-1 norm* without *that factor: $\|f - \widehat{f}\|_{\text{weak}-1} \leq C\|f - f_m\|_1$. This follows directly from the definition of the weak norm and our proof, below.*

Corollary 4. *Our argument shows that the reconstruction $\widehat{f}$ is not only stable with respect to noise in the signal, as Equation* (1.2) *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*

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

1.1. **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 [CM03] on detecting heavy hitters in nonnegative data streams and the works [GGI$^+$02b, GMS05] on Fourier sampling. More recent papers from Theoretical Computer Science include [CM05, CRTV05]. Sparked by the papers [Don04] and [CT04], the computational harmonic analysis and geometric functional analysis communities have produced an enormous amount of work, including [CRT04, Don05, DT05, CT05, RV05, TG05, MPTJ05].

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 [Don04, CT04, CRTV05]. 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 [TG05] 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 [GMS05] 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 [CM05] 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.

Table 1 compares the major algorithmic contributions. 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 $E_{\text{opt}} = E_{\text{opt}}(f) = f - f_m$ denote the error vector for the optimal output. Also, let $C_{\text{opt}} = C_{\text{opt}}(f)$ denote $\max_g E_{\text{opt}}(g)$, where g is the worst possible signal in the class where f lives.

1.2. **Organization.** In Section 2, we provide an overview of determining a sketch of the signal f. In Section 3, we give an explicit construction of a distribution from which the random linear map $\mathbf{\Phi}$ is drawn. In Section 4, we detail the reconstruction algorithm, Chaining Pursuit, and in Section 5 we give an analysis of the algorithm, proving our main result. In Section 6 we use our algorithmic analysis to derive a dimension reduction in the ℓ_1 norm for sparse vectors.

2. 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.

2.1. **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 [CM05].

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.

Approach, References	Signal Class	Uniform	Error bd.	# Measurements	Storage	Decode time
OMP + Gauss [TG05]	m-sparse	No	No error	$m \log d$	$md \log d$	$m^2 d \log d$
Group testing [CM06]	m-sparse	No	No error	$m \log^2 d$	$\log d$	$m \log^2 d$
ℓ_1 min. + Gauss [Don06, DT05, DT06]	m-sparse	Yes	No error	No closed form	$\Omega(md)$	LP(md)
Group testing [CM06]	m-sparse	Yes	No error	$m^2 \log^2 d$	$m \log(d/m)$	$m^2 \log^2 d$
Group testing [CM06]	weak ℓ_p	Yes	$\lVert E \rVert_2 \le C \lVert C_{\text{opt}} \rVert_p$	$m^{\frac{3-p}{1-p}} \log^2 d$	$m^{\frac{2-p}{1-p}} \log d$	$m^{\frac{4-2p}{1-p}} \log^3 d$
ℓ_1 min. + Gauss [CT04, CDD06]	Arbitrary	Yes	$\lVert E \rVert_2 \le m^{-1/2} \lVert E_{\text{opt}} \rVert_1$	$m \log(d/m)$	$d \log(d/m)$	LP(md)
ℓ_1 min. + Fourier [CT04, RV06, CDD06]	Arbitrary	Yes	$\lVert E \rVert_2 \le m^{-1/2} \lVert E_{\text{opt}} \rVert_1$	$m \log^4 d$	$m \log^5 d$	$d \log d$ (empirical)
Chaining Pursuit [GSTV06]	Arbitrary	Yes	$\lVert E \rVert_{\text{weak}-1} \le \lVert E_{\text{opt}} \rVert_1$ $\lVert E \rVert_1 \le \log(m) \lVert E_{\text{opt}} \rVert_1$	$m \log^2 d$	$d \log^2 d$	$m \log^2 d$
Fourier sampling [GGI+02b, GMS05]	Arbitrary	No	$\lVert E \rVert_2 \le \lVert E_{\text{opt}} \rVert_2$	m polylog d	m polylog d	m polylog d
Group testing [CM06]	Arbitrary	No	$\lVert E \rVert_2 \le \lVert E_{\text{opt}} \rVert_2$	$m \log^{5/2} d$	$\log^2 d$	$m \log^{5/2} d$

Notes: Above, LP(md) denotes resources needed to solve a linear program with $\Theta(md)$ variables, plus minor overhead. We suppress big-O notation for legibility.

Table 1. Comparison of algorithmic results for compressed sensing

2.2. **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.

2.3. **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

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

2.4. **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 3, 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 $m \log^2(d)$.

2.5. **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 3, 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.

3. 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 2.2, the matrix A consists of 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 2.2. We will see from the analysis in Section 5, 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 $\{0, \ldots, r-1\}$ 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 $m \operatorname{polylog}(d)$. As in Theorem 22 below, 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 [AHU83, CLRS01]. For completeness, we sketch details.

3.1. Requirements.

To ease notation, we consider only the case of $m_k = m$. Our goal is to construct a function $f_s : \{0, \ldots, d-1\} \to \{0, \ldots, r-1\}$, 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 $0 \leq i_1, \ldots, i_m < d$ and any m targets $t_1, \ldots, t_m$, 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 $m \operatorname{polylog}(d)$.

3.2. Construction.

Let $s = (s_0, s_1, \ldots, s_K)$ 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 $h : \{0, \ldots, r\lfloor p/r \rfloor - 1\} \to \{0, \ldots, r-1\}$ 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 $\{0, \ldots, r\lfloor p/r \rfloor - 1\}$ 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 and we will write $g_s^k(A)$ for the list $\{g_s^k(j) | j \in A\}$. Figure 1 gives a formal algorithm.

FIGURE 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 $\{0, \ldots, r-1\}$, representing $\langle f_s(j) : j \in A \rangle$.
Compute $g_s^k(A)$ for $k = 0, 1, 2, \ldots, K-1$.
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 $j \in A$
    $k_j = \min\{k : g_s^k(j) < r\lfloor p/r \rfloor\}$
    $f_s(j) = g_s^{k_j}(j)$.
```

3.3. **Correctness.**

Lemma 6. *Our construction of $f_s : \{0, \ldots, d-1\} \to \{0, \ldots, r-1\}$ produces a uniform m-wise independent partition with probability at least $1 - d^{-3}$.*

Proof. The proof of the correctness of our construction is a standard argument, which we sketch for completeness. First, there is a prime p with $d \leq p \leq \text{poly}(d)$. Next, let us consider the construction of g_s^k. Because the definition of g_s^k is independent of k, we drop the s and k and write g for simplicity. Because g is a random polynomial of degree $m-1$ over the field of p elements, we can view the construction of g as the multiplication of a vector c of length m (the coefficients of g) by the Vandermonde matrix

$$V = \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 0 & 1 & 2 & 3 & \cdots & p-1 \\ 0 & 1 & 2^2 & 3^2 & \cdots & (p-1)^2 \\ \vdots & & & & & \end{pmatrix}.$$

Thus we obtain $cV = (g(0), g(1), \ldots, g(p-1))$. If A is a list of m positions, then $g(A)$ is cV_A, where V_A is the submatrix of V gotten by selecting columns according to A. Since V is a square vandermonde matrix over a field, it is invertible. It follows that, as c varies, cV_A varies over all of $\mathbb{Z}_p^m$, hitting each element exactly once.

Next, $g(j) \geq r\lfloor p/r \rfloor$ with probability at most $r/p \leq 1/2$. It follows that, for some $k < K$, we have, with probability at least $1 - 2^{-K}$, that $g(j) < r\lfloor p/r \rfloor$. For sufficiently large $K \leq O(\log d)$, the probability is at least $1 - 1/d^4$. Taking a union bound over all d possible j's, the construction succeeds with probability at least $1 - 1/d^3$.

It is easy to check that, by construction, $f_s(A)$ is uniform on $\{0, \ldots, r-1\}^m$ conditioned on the construction succeeding. □

3.4. **Efficiency.**

Lemma 7. *Given an arbitrary set A of m positions in $\mathbb{Z}_p$ and a degree $m-1$ polynomial g, we can evaluate g on A in time $O(m \,\text{poly}\log(d))$.*

Proof. Evaluating g on the set A is known as the multipoint polynomial evaluation (MPE) problem. We recall that the MPE problem can be reduced to $\text{polylog}(m)$ polynomial multiplications [AHU83] and that we can multiply polynomials efficiently using the FFT algorithm. We observe that the time to multiply polynomials in $m\,\text{polylog}(d)$ as we may multiply polynomials by convolving their coefficients (via the FFT algorithm) over $\mathbb{C}$ and then quantizing and reducing modulo p the result. We note that arithmetic modulo p take time at most $\text{polylog}(d)$.

Let us now review the MPE problem. Recall that we wish to evaluate g on the set A, $g(A)$. The evaluation of $g(x)$ at some point $x = t$ is equivalent to finding $g \bmod (x - t)$, since we can write $g(x) = q(x)(x - t) + r$ by the division theorem. To compute the quotients $g \mod (x - a$ for each $a \in A$, let us assume that $|A|$ is a power of 2 (padding if necessary), then we order $A = \{a_i\}$ arbitrarily and form a binary tree in which the k'th node at depth j corresponds to the subset $A_{j,k} = \{a_i : km/2^j \le i < (k + 1)m/2^j\} \subseteq A$. Once we have formed the binary tree, we compute the polynomials $p_{j,k}(x) = \prod_{i \in A_{j,k}}(x - a_i)$ at each node. We also define $g_{j,k}$ at each node by $g_{j,k} = g \bmod p_{j,k}$. Our goal is to compute $g_{\lg m,k} = g \bmod p_{\lg m,k}$ for all k, *i.e.*, reduce g modulo each polynomial in a leaf of the tree. To do this, we start with $g = g_{0,0} = g \bmod p_{0,0}$, *i.e.*, g mod the root polynomial. From $g_{j,k}$ we form the two children, $g_{j+1,2k} = g_{j,k} \bmod p_{j+1,2k}$ and $g_{j+1,2k+1} = g_{j,k} \bmod p_{j+1,2k+1}$. Note that, at depth j, we have 2^j polynomials $g_{j,k}$ of degree $m/2^j - 1$ and $p_{j,k}$ of degree $m/2^j$.

We form the tree of $p_{j,k}$'s in a straightforward fashion, using the FFT algorithm to multiply polynomials. Multiplying a pair of polynomials at depth j takes time $m/2^j\,\text{polylog}(d)$ and there are $O(2^j)$ such problems, for total time $m\,\text{polylog}(d)$ at depth j, and total time $m\,\text{polylog}(d)$ in aggregrate over all $O(\log m)$ levels.

It remains to show how to reduce a polynomial g of degree $2n - 1$ by a polynomial q of degree n in time $n\,\text{polylog}(d)$. First, we reduce x^{n-1+2^k} for all $k = 0, 1, 2, \ldots, \lg(n)$. Suppose we have done the reduction for for $x^n, x^{n+1}, x^{n+3}, x^{n+7} \ldots, x^{n-1+2^{k-1}}$. We claim that we can then reduce any polynomial h of degree $n - 1 + 2^k$ by q in time $n\,\text{polylog}(d)$. To see this, write

$$h(x) = x^{n-1+2^{k-1}} h'(x) + h''(x),$$

where h' has degree 2^{k-1} and h'' has degree $n - 1 + 2^{k-1}$. Then, multiply $x^{n+2^{k-1}-1} \bmod q$ by h' and obtain a polynomial h of degree $n - 1 + 2^{k-1}$ which we add to h''. This reduces the problem for a polynomial of degree $n - 1 + 2^k$ to a polynomial of degree $n - 1 + 2^{k-1}$, in time $n\,\text{polylog}(d)$. Let us perform this reduction $k \le \lg(n)$ times so that we have a polynomial h of degree n. Once we obtain h, we can reduce this polynomial directly, by writing $h(x) = x^n h' + h''(x)$, where h' is constant and h'' has degree $n - 1$, and then adding h' times x^n mod q to h''.

The above discussion for a polynomial h of degree $n - 1 + 2^k$ holds in particular if $h(x) = x^{n-1+2^k}$; it follows by induction that we can reduce x^{n+2^k-1} for all $k = 0, 1, 2, \ldots, \lg(n) - 1$ in time $n\,\text{polylog}(d)$. Finally, we apply the above again to reduce our arbitrary polynomial g. □

We note that we can find a suitable prime p in time $\text{poly}(d)$ by testing all numbers from d to $\text{poly}(d)$. This is a preprocessing step and the time does not count against the claimed measurement time of $d\,\text{polylog}(d)$ or claimed decoding time of $m\,\text{polylog}(d)$ our algorithm. (In fact, time $\text{polylog}(d)$ suffices to find a prime.) We omit details.

From the preceding lemmas, we conclude:

Theorem 8. *There is an implementation of the Chaining Pursuit algorithm that runs in time* $m\,\text{polylog}(d)$ *and requires total storage* $m \log(d)$ *numbers bounded by* $\text{poly}(d)$ *(*i.e., $O(\log d)$ *bits).*

Figure 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 identify the spike position
             Use a bit test to estimate the spike magnitude
         Retain $m_k$ 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 sketch
Return the signal consisting of the $m$ largest retained spikes.
```

4. 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*; we focus on that until Section 5.4.

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 [Tal05]. For an example of chaining in Theoretical Computer Science, see [IN05].

4.1. Overview of Algorithm.

The structure of the Chaining algorithm is similar to other sublinear approximation methods described in the literature [GGI$^+$02b]. 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 5 and the quantity m_k is m/a^k. Pseudocode is given in Figure 2.

4.2. **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:

$$\left[\begin{array}{cccc|c} b(0) & b(1) & \ldots & b(\log_2\lceil d\rceil - 1) & c \end{array}\right].$$

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.

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

4.4. **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.

5. 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 30 in the sequel and let $\boldsymbol{\Phi} = \boldsymbol{A} \otimes_r \boldsymbol{B}$, where $\boldsymbol{B}$ is a bit test matrix.

Theorem 9 (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 $\widehat{f} = f$.

5.1. **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.*, [GGI$^+$02b]. 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 9 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 30, which summarizes the needed properties of the isolation matrix.

The proof of Theorem 9 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 9. The following is our central loop invariant:

Invariant 10. *In pass k, the signal has the form*

$$f^{(k)} = s_k + w + \sum_{j=0}^{k-1} \nu_j \tag{5.1}$$

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 9.

The proof that Invariant 10 is maintained will only use two properties of an isolation matrix, given in Condition 30. While we only know how to construct such matrices using randomness, any matrix satisfying these properties is acceptable. Section 5.2 will prove that Invariant 10 holds for any matrix $\mathbf{\Phi}$ having the properties in Condition 30; Section 5.3 proves that most matrices (according to the definition implicit in Section 2.2) satisfy these properties. 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 11 (Chaining Recovery Conditions for Isolation Matrices)**.** *A 0-1 matrix with pass/trial hierarchical structure described in Section 2.2 (*i.e.*, any matrix from the sample space described in Section 2.2) is said to satisfy the* Chaining Recovery Conditions *if for any signal of the form in Invariant 10 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 $\frac{1}{1000}m_k^{-1}$.*

5.2. **Deterministic Part.**

In this section, we consider only matrices satisfying Condition 30. Proposition 12 considers the performance of the algorithm in one of the 99/100 non-exceptional trials under an artificial assumption that will be removed in Proposition 17. Following that, we consider the performance of the combination of trials, prove that Invariant 10 is maintained, and conclude about the overall performance of Chaining Pursuit Proper.

Proposition 12 (One Trial, No Inaccuracies)**.** *Suppose that a 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.

Proof. To prove this proposition, we outline a series of lemmas. We begin with a simple observation about the performance of the bit-tests.

Lemma 13. *Assume that a measurement contains a position i of value p and that the ℓ_1 norm of the other positions in the measurement is at most ϵ. Then*

1. *The estimated value p_{est} is bounded by the total measurement; that is, $|p_{\text{est}}| \leq |p| + \epsilon$.*

(2) *If $|p| > 2\epsilon$, then the estimated position is i (*i.e.*, the bit-test locates the position correctly) and the estimated value is within ϵ from p, $|p_{\text{est}} - p| \leq \epsilon$.*

Proof. Follows immediately from the definitions of the bit-tests and the estimation procedures. □

Let us set $\epsilon = \frac{1}{1000} m_k^{-1}$ and observe that $\frac{1}{m_{k-1}} < \frac{1}{1000 m_k} = \epsilon$. Definition 31 establishes two criteria that most measurements satisfy. We refer to these two types as *good measurements* and define them precisely.

Definition 14. *A* good measurement *satisfies one of the following two criteria:*

(1) *The measurement is empty; that is, it contains positions with values $\left|f^{(k)}(i)\right| \leq \epsilon$ and the total ℓ_1 norm of the positions in the measurement is less than 1.5ϵ.*
(2) *The measurement contains one spike at position i with $\left|f^{(k)}(i)\right| > \epsilon$ and the ℓ_1 norm of all other positions in this measurement is less than 0.5ϵ.*

The next lemma states that for good measurements, the bit-tests return reasonably accurate estimates.

Lemma 15. *Assume that a measurement is a good one. If the measurement is empty, then the estimated value of $f^{(k)}$ in that measurement is no more than 1.5ϵ. If the measurement contains one spike, then the estimated position is the position of the spike and its estimated value is within 0.5ϵ of the true value of the spike.*

Proof. Follows from the definitions of good measurements and Lemma 13. □

The next lemma follows from the previous argument and demonstrates that if the bit-tests identify a spike position, they do so reasonably accurately and precisely.

Lemma 16. *Assume that in a single measurement the bit-tests identify and estimate a spike at position i and that the estimated value p_{est} is greater than 1.5ϵ, $|p_{\text{est}}| > 1.5\epsilon$. Then the measurement contains a spike at position i and the true value of p is within 0.5ϵ of p_{est}.*

Strictly speaking, we perform multiple trials at each round k. We obtain, in a single trial, an estimate position and its estimated value, which we call the preliminary estimated value for that position. If, after performing all the trials, we have more than one preliminary estimated value for an estimated position, we simply use the preliminary value with the largest absolute value as the estimate assigned to this position. We then identify the m_k positions with the largest assigned estimates. A simple argument (which we omit here for brevity) demonstrates that the true value p of a spike at position i with $|p| > 2\epsilon$ is assigned an estimate p_{est} within 0.5ϵ of p. Furthermore, if the assigned estimate p_{est} of a spike at position i satisfies $|p_{\text{est}}| > 1.5\epsilon$, then the true value p is within 0.5ϵ of p_{est}. To simplify our arguments in what follows, we simply refer to the estimated values in one trial as the assigned estimate values, $\widetilde{f}^{(k)}(i)$.

With the above lemmas, we are able to complete the proof of the proposition. Our previous discussion shows that those positions i with $\left|f^{(k)}(i)\right| > \epsilon$ include the positions with estimated values larger than 1.5ϵ; *i.e.*,

$$\left\{i \,\middle|\, |\widetilde{f}^{(k)}(i)| > 1.5\epsilon\right\} \subseteq \left\{i \,\middle|\, |f^{(k)}(i)| > \epsilon\right\}.$$

Our inductive hypothesis assumes that there are at most m_k positions in the right set above, so there are at most m_k positions in the left set as well. Our algorithm (for one trial) identifies all of these positions and reports estimated values that are within 0.5ϵ of the true values. Hence, our list of identified positions includes those i with $\left|f^{(k)}(i)\right| > 2\epsilon$. If a position i is identified and if $\left|f^{(k)}(i)\right| \leq 4\epsilon$, the its estimated value is at most 5ϵ in magnitude by the previous lemmas. This proves the proposition. □

The next proposition removes the artificial assumption on the spikes and noise.

Proposition 17 (One Trial)**.** *Suppose that the trial is not an exceptional trial. In this trial, suppose each measurement is a good one, except for at most $\frac{1}{50}m_{k+1}$. 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. We begin the proof with a lemma that shows the list produced by the algorithm is stable with respect to changes in a few measurements.

Lemma 18. *Assume that we perform one trial of the algorithm with two different signals and that their measurements (in this one trial) are identical except for b measurements. Then the estimated signals are identical except in $2b$ positions.*

Proof sketch. Prove this for $b = 2$ and then proceed by induction. □

Let us now consider the set of fewer than $\frac{1}{50}m_{k+1}$ bad measurements and set to zero the signal positions that fall into these measurements. This procedure creates two signals: the original signal and the restricted signal (with zeroed out positions). The restricted signal satisfies the conditions in Proposition 12 so all its measurements are good ones and agree with those of the original signal except for $\frac{1}{50}m_{k+1}$ measurements. The previous lemma guarantees that the estimated signals for the original and restricted signals are identical in all but $\frac{1}{25}m_{k+1}$ positions. By our inductive hypothesis, there are at most $\frac{1}{50}m_{k+1}$ positions of the original signal with value greater than 2ϵ in the exceptional measurements. Let us gather these $\frac{1}{25}m_{k+1}$ and $\frac{1}{50}m_{k+1}$ exceptional positions into one set of $\frac{3}{50}m_{k+1}$ exceptions. It is straightforward to show that the positions not in this exceptional set are good positions and, if they are identified, they are identified accurately. □

We combine results from all trials. The algorithm considers positions identified in at least $\frac{9}{10}$ of the total trials T. It then takes the median (over all trials) to estimate the values of these positions.

Lemma 19 (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 $\frac{10}{9}m_k$.*

Proof. We prove the first part of the lemma with a simple counting argument. Let T denote the total number of trials. We have to bound b where

$$b = \#\{\text{positions bad in} \geq \tfrac{T}{10} \text{ trials}\} \leq \#\{\text{positions bad in} \geq \tfrac{T}{11} \text{ good trials}\}.$$

Let

$$\Sigma = \sum_{j \in \text{ good trials}} \#\{\text{positions bad in trial } j\}.$$

We have $\Sigma \geq \frac{bT}{11}$. Let T' be the number of good trials. Then Proposition 17 tells us that $\Sigma \leq \frac{3}{50}m_{k+1}T' \leq \frac{3 \cdot T}{50}m_{k+1}$. Therefore, $\frac{bT}{11} \leq \frac{3T}{50}m_{k+1}$ and, hence, $b \leq \frac{33}{50}m_{k+1}$.

To prove the second part, recall that the algorithm updates a position if and only if the position is identified in at least 9/10 of the trials. Let ℓ denote the number of such positions. In every trial, m_k positions are identified. Hence,

$$m_k T = \sum_{t=1}^{T} \#\{\text{positions identified in trial } t\} \geq \frac{9}{10}T\ell$$

and thus $\ell \leq \frac{10}{9}m_k$. □

Now we are ready to prove the induction step. Recall that after round k, the new signal is the difference between the current signal and its estimate: $f^{(k)} = f^{(k-1)} - \tilde{f}^{(k-1)}$, with the convention that if a signal position is not considered and not changed by the algorithm, its estimated value is zero.

Lemma 20 (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. Recall that $4.5\epsilon < m_k^{-1}$. It suffices to prove that for the non-exceptional positions i that satisfy the conclusions of Lemma 19, the value $\left|f^{(k)}(i)\right| \leq 4.5\epsilon < \frac{2}{m_k}$. Let us fix such a position i which is good in at least $9/10T$ trials and show that

$$\left|f^{(k-1)}(i) - \tilde{f}^{(k-1)}(i)\right| \leq 4.5\epsilon.$$

If $\left|f^{(k-1)}(i)\right| > 2\epsilon$, then the goodness of i in 9/10 trials implies that i is identified in these trials and hence i is considered by the Algorithm. In each of these 9/10 trials, the goodness of i also means that the assigned estimated value of i is within 0.5ϵ from its true value $f^{(k-1)}(i)$. Since $\tilde{f}^{(k-1)}(i)$ is the median of the assigned estimated values of i in each trial, it follows that it is also within 0.5ϵ from the true value $f^{(k-1)}(i)$.

Suppose that $\left|f^{(k-1)}(i)\right| \leq 2\epsilon$. If i is not considered by the algorithm, then the value of the signal at this position is not changed, so $|f^{(k)}(i)| = |f^{(k-1)}(i)| \leq 2\epsilon$. We can assume that i is considered by the algorithm. There are $(9/10)T$ trials in which i is identified and there are $(9/10)T$ trials in which i is good. Hence, there are at least $8/10T$ trials in which i is both good and identified. By the definition of goodness, this means that in each of these trials, the assigned estimated value $|\tilde{f}^{(k-1)}(i)|$ is at most 2.5ϵ. Since the estimated value is the median of the assigned estimated values of i in each trial, it follows that $|\tilde{f}^{(k-1)}(i)| \leq 2.5\epsilon$. Then

$$|f^{(k-1)}(i) - \tilde{f}^{(k-1)}(i)| \leq |f^{(k-1)}(i)| + |\tilde{f}^{(k-1)}(i)| \leq 2\epsilon + 2.5\epsilon = 4.5\epsilon.$$

This completes the proof of the first part of the lemma.

The first part of this lemma shows formally that the difference between the spikes in the signal $f^{(k)}$ and the large entries in the update signal (*i.e.*, those with absolute values greater than m_k^{-1}) contains at most m_{k+1} terms. By the inductive step, the same holds for the previous rounds. In addition, Lemma 19 tells us that the algorithm updates at most $\frac{10}{9}m_k$ positions in the signal. By the triangle inequality it follows that the difference contains at most $m_{k+1} + m_k + \frac{10}{9}m_k \leq 3m_k$ terms. The maximal absolute value of the difference signal is

$$\frac{1}{m_k} + \frac{1}{m_{k-1}} \leq \frac{2}{m_k}.$$

□

The previous lemma proves the induction hypothesis. The next and final lemma of this section controls the recovery error and completes the proof of Theorem 22.

Lemma 21 (Total Spikes and Recovery Error)**.** *Chaining Pursuit Proper recovers at most $O(m)$ spikes. The total recovery error is at most $(1 + C\log m)\left\|w\right\|_1$.*

Proof Sketch. After pass $K = \log_a m$, there are no more spikes remaining since $m_k = m/a^k < 1$. At most $\frac{10}{9}m_k$ spikes are recovered in pass k. Since m_k decays geometrically, the total number of spikes is $O(m)$. The error after the last pass is the ℓ_1 norm of the signal $f^{(K+1)}$. This signal consists of the external noise, which has norm $\left\|w\right\|_1$, and the internal noise, which satisfies

$$\left\|\sum_{j=0}^{K} \nu_j\right\|_1 \leq \sum_{j=1}^{K} 6m_j m_j^{-1} = 6\log_a m.$$

Since a is a constant and $\left\|w\right\|_1$ was normalized to be constant, the overall error is at most $(1 + C\log m)\left\|w\right\|_1$ for some constant C. □

5.3. **Probabilistic Part.** Here we prove that a random isolation matrix $\boldsymbol{A}$ indeed satisfies the CRC with high probability.

Theorem 22. *With probability at least $(1 - O(d^{-3}))$, a matrix $\boldsymbol{A}$ drawn from the distribution described in Section 2.2 satisfies the Chaining Recovery Conditions (Conditions 30).*

The main lemmas of this section are as follows. First, Lemma 23 is an abstract technical Lemma about putting balls into buckets and the number of isolated balls that likely result. Lemma 24 is a corollary for our context. We then show, in Lemma 26, that Condition 30 holds for most matrices.

Lemma 23 (Balls and Bins)**.** *Put n balls randomly and independently into $N > C(M)n$ buckets. Then, with probability $1 - 2e^{-9n}$, all except n/M balls are isolated in their buckets.*

Proof. One complication in the proof comes from the absence of independence among buckets. We would rather let buckets choose balls. However, the contents of different buckets is dependent because the total number of balls is limited. So we will replace the original n-ball model with an *independent* model. The independent model will be easier to handle by the standard large deviation technique; the independent model reduces the original model from it by conditioning on the number of balls.

The independent model is the following assignment. We divide each bucket into n sub-buckets, and let δ_{ki} be independent $0, 1$ valued random variables with expectation $\mathbb{E}\,\delta_{ki} = 1/N$, for all buckets $k = 1, \dots, N$ and sub-buckets $i = 1, \dots, n$. The independent random variables $X_k = \sum_{i=1}^{n} \delta_{ki}$ will be called the number of balls in bucket k in the independent model. If we condition on the total number of such "balls", we obtain the distribution of the numbers of true balls X'_k in bucket k in the original model:

$$(X'_1, \dots, X'_N) \equiv \Big(X_1, \dots, X_N \;\Big|\; \sum_{k=1}^{N} X_k = n \Big).$$

To prove the Lemma, we have to show that the number of non-isolated balls is small. The number of non-isolated balls in bucket k is $Y'_k = X'_k \cdot 1_{\{X'_k > 1\}}$ so the conclusion of the Lemma is that

$$\mathbb{P}\Big\{ \sum_{k=1}^{N} Y'_k > n/M \Big\} \le 2e^{-9n}. \tag{5.2}$$

We will now transfer this problem to the independent model. First, without loss of generality we can change the n balls in the lemma and in the original model to $0.9n$ balls. We do not change the independent model, so the number of non-isolated balls in the independent model is $Y_k = X_k \cdot 1_{\{X_k > 1\}}$. We have to bound

$$\begin{aligned}
\mathbb{P}\Big\{ \sum_{k=1}^{N} Y'_k > n/M \Big\} &= \mathbb{P}\Big\{ \sum_{k=1}^{N} Y_k > n/M \Big| \sum_{k=1}^{N} X_k = 0.9n \Big\} \\
&\le \mathbb{P}\Big\{ \sum_{k=1}^{N} Y_k > n/M \Big| \sum_{k=1}^{N} X_k \ge 0.9n \Big\} \\
&\le \frac{\mathbb{P}\Big\{ \sum_{k=1}^{N} Y_k > n/M \Big\}}{\mathbb{P}\Big\{ \sum_{k=1}^{N} X_k \ge 0.9n \Big\}}.
\end{aligned}$$

By Prokhorov-Bennett inequality,

$$\mathbb{P}\Big\{ \sum_{k=1}^{N} X_k \ge 0.9n \Big\} = \mathbb{P}\Big\{ \sum_{k=1}^{N} \sum_{i=1}^{n} \delta_{ki} \ge 0.9n \Big\} \ge 1/2.$$

Therefore, proving Equation (5.2) in the original model reduces to proving that

$$\mathbb{P}\Big\{\sum_{k=1}^{N} Y_k > n/M\Big\} \le e^{-9n} \tag{5.3}$$

in the independent model.

A standard way to prove deviation inequalities such as Equation (5.3) is through the moment generating function. By Markov's inequality and independence of the variables Y_k, we have

$$\begin{aligned}\mathbb{P}\Big\{\sum_{k=1}^{N} Y_k > n/M\Big\} &= \mathbb{P}\Big\{e^{10M\sum_{k=1}^{N} Y_k} > e^{10n}\Big\} \le e^{-10n}\cdot \mathbb{E}\Big[e^{10M\sum_{k=1}^{N} Y_k}\Big] \\ &= e^{-10n}\cdot \Big(\mathbb{E}\Big[e^{10MY_1}\Big]\Big)^{N}\end{aligned}$$

To complete the proof of Equation (5.3) it remains to show that for $Y = (\sum_{i=1}^{n}\delta_i)\cdot 1_{\{\sum_{i=1}^{n}\delta_i>1\}}$, its moment generating function satisfies

$$\Big(\mathbb{E}\Big[e^{10MY}\Big]\Big)^{N} \le e^{n} \tag{5.4}$$

where δ_i are $0,1$ valued independent random variables with $\mathbb{E}\,\delta_i = 1/N$. To estimate the moment generating function $\mathbb{E}[e^{MY}]$ in Equation (5.4), it suffices to know the tail probability $\mathbb{P}\{Y > t\}$ for large t. For large t, we estimate this tail probability by removing the restriction onto non-isolated bucket in the definition of Y and applying Chernoff's inequality for independent random variables. The tail probability is, however, much smaller if we do restrict onto non-isolated buckets. We take this into account for small t by computing the expectation of Y (which is straightforward).

Let us start with the first moment of Y. We claim that

$$\mathbb{E}[Y] \le C\Big(\frac{n}{N}\Big)^{2}. \tag{5.5}$$

Compare this with the average number of balls without conditioning on being non-isolated, $\mathbb{E}[X_k] = \frac{n}{N}$. Indeed, by the linearity of expectation,

$$\begin{aligned}\mathbb{E}[Y] &= n\cdot\mathbb{E}\Big[\delta_1\cdot 1_{\{\sum_{i=1}^{n}\delta_i>1\}}\Big] \\ &= n\cdot\mathbb{P}\Big\{\delta_1 = 1 \text{ and there exists } i\in\{2,\dots,n\}:\ \delta_i = 1\Big\} \\ &= \mathbb{P}\Big\{\delta_1 = 1\Big\}\cdot\big(1-\mathbb{P}\Big\{\forall i\in\{2,\dots,n\}:\delta_i = 0\Big\}\big) \\ &= \frac{n}{N}\cdot\left(1-\Big(1-\frac{1}{N}\Big)^{n-1}\right) \le \frac{n}{N}\cdot\Big(1-e^{-n/N}\Big) \le C\Big(\frac{n}{N}\Big)^{2}.\end{aligned}$$

Next, by the Chernoff inequality, for $s > 2$, we have

$$\mathbb{P}\{Y > s\} \le \mathbb{P}\{\sum_{i=1}^{n}\delta_i > s\} \le (s\frac{N}{n})^{-s}. \tag{5.6}$$

Now we are ready to bound the moment generating function. Let $K = 10M$ and change variables $t = e^{Ks}$, so that

$$\begin{aligned}\mathbb{E}\Big[e^{KY}\Big] &= \int_0^\infty \mathbb{P}\Big\{e^{KY} > t\Big\}\,dt = 1 + \int_1^\infty \mathbb{P}\Big\{e^{KY} > t\Big\}\,dt \\ &= 1 + K\int_0^\infty \mathbb{P}\Big\{Y > s\Big\}\,e^{Ks}\,ds.\end{aligned}$$

We split the integral in two parts. We use Equation (5.5) to estimate the integral near zero as

$$\begin{aligned}\int_0^3 \mathbb{P}\Big\{Y > s\Big\}\, e^{Ks}\, ds &\le e^{3K} \int_0^\infty \mathbb{P}\Big\{Y > s\Big\}\, ds \\ &= e^{3K}\, \mathbb{E}[Y] \le Ce^{3K}\Big(\frac{n}{N}\Big)^2,\end{aligned}$$

and we use Equation (5.6) to estimate the integral near infinity as

$$\begin{aligned}\int_3^\infty \mathbb{P}\Big\{Y > s\Big\}\, e^{Ks}\, ds \le \int_3^\infty \Big(\frac{N}{n}s\Big)^{-s} e^{Ks}\, ds &\le \int_3^\infty \Big(\frac{N}{n}e^{-K}\Big)^{-s}\, ds \\ &= \frac{1}{\ln(\frac{N}{n}e^{-K})}\Big(\frac{N}{n}e^{-K}\Big)^{-3} \le e^{2K}\Big(\frac{n}{N}\Big)^2.\end{aligned}$$

Combining these, we conclude that

$$\mathbb{E}\Big[e^{KY}\Big] \le 1 + CKe^{3K}\Big(\frac{n}{N}\Big)^2 \le 1 + \frac{n}{10N}.$$

Hence we obtain Equation (5.4)

$$\Big(\mathbb{E}\Big[e^{KY}\Big]\Big)^N \le \Big(1 + \frac{n}{10N}\Big)^N \le e^{\frac{n}{5N}\cdot N} \le e^n.$$

This proves the lemma. □

Lemma 24 (Isolations)**.** *Fix a round k. 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 sketch. In the hypothesis of Invariant 10, the signal has at most $n := m_k$ positions of value larger than $1/m_{k-1}$. We put these in $N := \overline{m}_k = C'(a)m/2^k$ buckets. We can choose the function $C'(a)$ so that, for C of Lemma 21, we have $C'(a) = C(\frac{1}{100a})$. Apply Lemma 23 which states that all except $\frac{1}{100}m_{k+1}$ positions are isolated with probability $1-\delta$, where $\delta = 2e^{-9m_k}$.

Let I be the event that all but $\frac{1}{100}m_k$ of the spikes are isolated. Let us repeat the random assignment above independently T times (for T trials) and let δ_t be independent Bernoulli random variables, $\mathbb{E}\,\delta_t = \delta$. We see by Chernoff's inequality that

$$\begin{aligned}\mathbb{P}\{I \text{ fails in more than } \frac{1}{100}T \text{ trials}\} \le \mathbb{P}\Big\{\sum_{t=1}^T \delta_t > \frac{1}{100}T\Big\} &\le (100e\delta)^{\frac{1}{100}T} \\ &= \exp\big(-m_k \cdot \frac{1}{100}T\big) \le \exp(-4m_k \log d).\end{aligned}$$

□

This concludes the proof of Lemma 24.

We now proceed to prove that Invariant 10 is maintained by most isolation matrices $\boldsymbol{A}$. That is, we need to show that, for most $\boldsymbol{A}$, when the Chaining Pursuit algorithm uses $\boldsymbol{A}$ on a signal satisfying Invariant 10 for round k, the algorithm produces a signal satisfying Invariant 10 for round $k+1$. So, in the remainder of this section, we may fix a signal satisfying Invariant 10 for round k. Lemma 25 controls the external noise and Lemma 26 controls the internal noise.

Lemma 25 (External noise)**.** *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 $\frac{1}{200}m_{k+1}$.*

Proof sketch. This is an easy part of the argument. The $(1,1)$ operator norm of each matrix $\boldsymbol{A}_t^{(k)}$ equals one, so it does not inflate the norm of the external noise. We use Markov's inequality to bound the number of measurements with too much noise. □

Lemma 26 (Internal noise)**.** *Fix a round k. With probability at least $1 - \exp\{-4m_k \log d\}$, the following is true. In pass k, in at least 99/100 of the 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. Let us recall the Invariant 10. This lemma is a statement is about the signal $f^{(k-1)}$ in the positions with values smaller than $1/m_{k-1}$. Lemma 25 gives us a proof for $k = 0$.

Let $k \geq 1$. We may assume that the external noise w is 0 and we may absorb the spikes in Equation 5.1 into the first term; *i.e.* we can assume that our signal has the form

$$f^{(k)} = \sum_{j=0}^{k-1} \nu_j \tag{5.7}$$

where ν_j consists of $4m_j$ or fewer nonzero components with magnitudes at most $\frac{3}{m_j}$.

To prove this result, we introduce positive parameters λ_j, ϵ_j, $j = 0, \dots, k-1$, which satisfy

$$\sum_{j=0}^{k-1} \lambda_j \leq \frac{1}{C'} \tag{5.8}$$

and

$$\sum_{j=0}^{k-1} \epsilon_j \leq \frac{1}{C'a} \tag{5.9}$$

where C' is a positive absolute constant to be chosen later. Next, we will prove the following separate claim about the internal noise ν_j.

Claim 27. *Assume that a signal satisfies Equation* (5.7)*. Let $j \in \{0, \dots, k-1\}$. Then*

$$\sum_{t=1}^{T} \#\Big(\textit{measurements in trial } t \textit{ s.t. } \|\nu_j\|_1 > \frac{\lambda_j}{m_k}\Big) < \epsilon_j m_k T \tag{5.10}$$

with probability

$$1 - e^{-\gamma m_j T}, \tag{5.11}$$

where γ is some positive number such that

$$\gamma T \geq 10 \log d. \tag{5.12}$$

Proof. **Claim implies Lemma 26.** We will first show that this Claim implies Lemma 26. Assume the claim holds. By the definition of T, the exceptional probability is

$$e^{-\gamma m_j T} \leq e^{-10 m_j \log d} \leq \binom{d}{4m_j}^{-4},$$

while the number of choices of round j signal is $\binom{d}{4m_j}$. Hence, with probability $1 - \binom{d}{4m_j}^{-3}$, the inequality in Equation (5.10) holds uniformly for all choices of internal noise ν_j. Summing up these exceptional probabilities for all rounds $j = 0, \dots, k-1$, we conclude that:

> *with probability $1 - \binom{d}{4m_j}^{-2}$, the system of measurements is such that the inequality in Equation* (5.10) *holds uniformly for all choices of the signal satisfying Equation* (5.7)*.*

Fix a system of measurements which meets these requirements so that we can dismiss the probability issues. Let us also fix a measurement v and a trial t. We refer to $\nu_{k,v}^t$ as the signal in measurement v

and trial t. By Equation (5.8), for a fixed trial and a fixed measurement v, we have the containment of events:

$$\Big\{\|\nu_{k,v}^t\|_1 > \frac{1}{2000m_k}\Big\} \subseteq \bigcup_{j=0}^{k-1} \Big\{\|\nu_{j,v}^t\|_1 > \frac{\lambda_j}{m_k}\Big\}.$$

Counting the measurements that satisfy each side of this containment, then summing over the trials, we obtain:

$$\begin{aligned}\sum_{t=1}^{T} \#\Big\{\|\nu_{k,v}^t\|_1 > \frac{1}{2000m_k}\Big\} &\le \sum_{t=1}^{T}\sum_{j=0}^{k-1} \#\Big\{\|\nu_{j,v}^t\|_1 > \frac{\lambda_j}{m_k}\Big\} \\ &\le \sum_{j=0}^{k-1} \epsilon_j m_k T \quad \text{by (5.10)} \\ &\le \frac{1}{C'a} m_k T \quad \text{by (5.9)} \\ &= \frac{1}{C'} m_{k+1} T.\end{aligned}$$

By Markov's inequality, this implies (provided C' is chosen large enough) that in at most $\frac{1}{100}T$ trials t, the number of measurements where the ℓ_1 norm of the internal noise exceeds $\frac{1}{2000m_k}$ is greater than $\frac{1}{200}m_{k+1}$. This implies the conclusion of Lemma 26.

Proof of the claim. Now we prove the claim itself. This is a purely probabilistic problem. We call the nonzero positions of ν_j "balls" and we informally refer to the measurements as "buckets". By the definition of ν_j, the 1-norm of ν_j in a measurement (or bucket) v is large, $\|\nu_{j,v}\|_1 > \frac{\lambda_j}{m_k}$ if and only if the measurement contains at least $\frac{1}{3}\lambda_j\frac{m_j}{m_k}$ balls. Then Equation (5.10) is equivalent to

$$\sum_{t=1}^{T} \#(\text{measurement in trial } t \text{ which contain} > \frac{1}{3}\lambda_j\frac{m_j}{m_k} \text{ balls}) \le \epsilon_j m_k T. \tag{5.13}$$

To prove this with required probability (5.11), we will transfer the problem to an independent model—similar to the proof of Lemma 23.

Recall that the *original model* with n balls and T trials, in which we want to prove Equation (5.13), is to put $n = 4m_j$ balls into $N = \overline{m}_k$ buckets (or measurements) independently, and repeat this T times (trials) independently.

We want to replace this by the following *independent model*, where the contents of buckets are independent. There are N buckets in each of T trials. Divide each bucket into $S = nT$ sub-buckets. Let δ_{tli} be independent $0,1$ valued random variables with expectation $\mathbb{E}\,\delta_{tli} = 1/NT$, for all trials $t = 1,\dots,T$, buckets $l = 1,\dots,N$ and sub-buckets $i = 1,\dots,S$. The independent random variables

$$X_{tl} = \sum_{i=1}^{S} \delta_{tli} \tag{5.14}$$

will be called the number of balls in bucket l, trial t, in the independent model. Note that

$$\mathbb{E}\,X_{tl} = \frac{S}{NT} = \frac{n}{N}.$$

Thus the average total number of balls in buckets in one trial is n. Let E_{tot} be the event that in each of at least $T/2$ trials, the total number of balls in buckets is at least $n/2$. It is then easy to deduce by Chernoff and Prokhorov-Bennett's inequalities that with probability at least 1/2 (actually $1 - e^{-nT}$), E_{tot} holds; that is, $\mathbb{P}\{E_{\text{tot}}\} \geq 1/2$. We have to bound above the probability of the event

$$E := \{\text{ Equation (5.13) does not hold}\}$$

in the original model. We reduce it to the independent model as follows:

$$\begin{aligned}\mathbb{P}\{\text{E in independent model}\} &\geq \mathbb{P}\{\text{E in independent model} \bigm| E_{\text{tot}}\} \cdot \mathbb{P}\{E_{\text{tot}}\} \\ &\geq \frac{1}{2}\ \mathbb{P}\{\text{E in independent model} \bigm| E_{\text{tot}}\}\end{aligned}$$

The probability can only decrease when we consecutively do the following changes:

(1) remove both occurences of "at least" in E_{tot}, resulting in exactly $T/2$ trials and exactly $n/2$ balls;
(2) restrict the sum in Equation (5.13) to the $T/2$ trials included in the new (exact) version of E_{tot}; and
(3) fix the set of $T/2$ trials in E_{tot}—say, require that these be the first $T/2$ trials.

After doing this, the law becomes the original model with $n/2$ balls and T trials. Hence,

$$\mathbb{P}\{\text{E in independent model}\} \geq \frac{1}{2}\ \mathbb{P}\{\text{E in original model with } n/2 \text{ balls and } T/2 \text{ trials}\}.$$

Therefore, it suffices to prove that Equation (5.13) holds in the independent model, with probability as in (5.11), *i.e.* with probability $1 - \frac{1}{2}e^{-\gamma m_j T}$, where γ is as in Equation (5.12).

In order to prove Equation (5.13) with the requisite probability, we first estimate the number of balls X_{tl} in one bucket, see Equation (5.14). It is a sum of $S = nT$ independent Bernoulli random variables with expectations $\frac{1}{NT}$. Then by the Chernoff inequality,

$$\mathbb{P}\Big\{X_{tl} > \frac{1}{3}\lambda_j \frac{m_j}{m_k}\Big\} \leq \Big(\frac{1}{12e}\lambda_j \frac{\overline{m}_k}{m_k}\Big)^{-\frac{1}{3}\lambda_j \frac{m_j}{m_k}}. \tag{5.15}$$

Let us call this probability η. We have to estimate the sum in Equation (5.13) which equals $\sum_{t=1}^{T}\sum_{l=1}^{N}\delta_{tl}$ where

$$\delta_{tl} = 1_{\{X_{tl} > \frac{1}{3}\lambda_j \frac{m_j}{m_k}\}}$$

are independent Bernoulli random variables whose expectations are $\mathbb{E}\,\delta_{tl} \leq \eta$ by Equation (5.15). Then by the Chernoff inequality, the probability that Equation (5.13) fails to hold is

$$\mathbb{P}\Big\{\sum_{t=1}^{T}\sum_{l=1}^{N}\delta_{tl} > \epsilon_j m_k T\Big\} \leq (\beta/e)^{-\epsilon_j m_k T},$$

where

$$\beta = \frac{\epsilon_j m_k}{\eta N} = \epsilon_j \cdot \frac{m_k}{\overline{m}_k} \cdot \frac{1}{\eta} = \epsilon_j \cdot \frac{m_k}{\overline{m}_k} \cdot \Big(\frac{1}{12e}\lambda_j \frac{\overline{m}_k}{m_k}\Big)^{\frac{1}{3}\lambda_j \frac{m_j}{m_k}} \tag{5.16}$$

To complete the proof, we need to show that

$$(\beta/e)^{-\epsilon_j m_k T} \leq \frac{1}{2}e^{-\gamma m_j T},$$

which would follow from

$$(\beta/e)^{\epsilon_j \cdot \frac{m_k}{m_j}} \geq e^{2\gamma}, \tag{5.17}$$

where γ must satisfy Equation (5.12).

Now we specify our choice of λ_j and ϵ_j. Set

$$\lambda_j := \max\Big\{12e\Big(\frac{m_k}{\overline{m}_k}\Big)^{1/2},\ 24\frac{m_k}{m_j},\ \frac{1}{C'k}\Big\},$$
$$\epsilon_j := \max\Big\{\frac{m_k}{\overline{m}_k},\ \frac{1}{C'ak}\Big\}.$$

Then Equations (5.8) and (5.9) are clearly satisfied (recall that $j < k$). Next, the base in Equation (5.16) is estimated as

$$\frac{1}{12e}\lambda_j\frac{\overline{m}_k}{m_k} \ge \Big(\frac{\overline{m}_k}{m_k}\Big)^{1/2}$$

and the exponent can be estimated using

$$\lambda_j\frac{m_j}{m_k} \ge 24.$$

Also, the first factor of Equation (5.16) is estimated as

$$\epsilon_j \cdot \frac{m_k}{\overline{m}_k} \ge \Big(\frac{m_k}{\overline{m}_k}\Big)^2.$$

Combining these three estimates, we obtain

$$\beta \ge \Big(\frac{m_k}{\overline{m}_k}\Big)^2 \cdot \Big(\frac{\overline{m}_k}{m_k}\Big)^{\frac{1}{6}\lambda_j\frac{m_j}{m_k}} \ge \Big(\frac{\overline{m}_k}{m_k}\Big)^{\frac{1}{8}\lambda_j\frac{m_j}{m_k}}.$$

Now we can check Equation (5.17).

$$\begin{aligned}(\beta/e)^{\epsilon_j\cdot\frac{m_k}{m_j}} &\ge \Big(\frac{\overline{m}_k}{em_k}\Big)^{\frac{1}{8}\lambda_j\epsilon_j}\\ &\ge (a/2)^{\frac{\lambda_j\epsilon_j}{8}\cdot k} \quad \text{by the definition of } m_k,\, \overline{m}_k\\ &\ge \exp\Big(\frac{\ln(a/2)}{(C')^2a}\cdot\frac{k}{k^2}\Big) \quad \text{by the definition of } \lambda_j \text{ and } \epsilon_j\\ &= e^{c(a)/k}.\end{aligned}$$

Therefore Equation (5.17) holds for $\gamma = c(a)/k$, and this choice of γ satisfies the required condition in Equation (5.12), since $T = C(a)(k+1)\log(d)$. □

This completes the proof of Lemma 26. □

5.4. **Pruning.** To this point, we have shown that the Chaining Pursuit Proper algorithm produces an approximation $\widehat{f}$ of at most $O(m)$ terms with

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

We now show that pruning produces $\widehat{f}_m$ with

$$\big\|f - \widehat{f}_m\big\|_1 \le 3(1 + C\log m)\big\|f - f_m\big\|_1.$$

That is, we reduce the number of terms to exactly m while increasing the error by a small constant factor. This result applies to any approximation, not just an approximation produced by Chaining Pursuit Proper. This is our top-level result.

Theorem 28. *Let $\widehat{f}$ be an approximation to f with $\big\|f - \widehat{f}\big\|_1 \le B\big\|f - f_m\big\|_1$. Then*

$$\big\|f - \widehat{f}_m\big\|_1 \le (2B+1)\big\|f - f_m\big\|_1.$$

Proof. We have, using the triangle inequality and optimality of $\widehat{f}_m$ for $\widehat{f}$,

$$\begin{aligned}\|f-\widehat{f}_m\|_1 &\le \|f-\widehat{f}\|_1+\|\widehat{f}-\widehat{f}_m\|_1\\ &\le \|f-\widehat{f}\|_1+\|\widehat{f}-f_m\|_1\\ &\le \|f-\widehat{f}\|_1+\|\widehat{f}-f\|_1+\|f-f_m\|_1\\ &\le (2B+1)\|f-f_m\|_1.\end{aligned}$$

□

5.5. **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 29. *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*

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

Proof. We need only make a few adjustments to the proof of the main theorem to obtain this result. For brevity, we note these changes. Let $V=\mathbf{\Phi} f+y$ and let us refer to y as the measurement error.

First, we normalize the signal so that the measurement error has ℓ_1 norm $\|y\|_1=\frac{1}{(800{,}000a)}$ and the noise $w=f-f_m$ has ℓ_1 norm $\|w\|_1=\frac{1}{(800{,}000a)}$. Next, we modify the Chaining Recovery conditions for Isolation Matrices in Condition 30 to include a third property.

Condition 30 (Chaining Recovery Conditions for Robust Isolation Matrices)**.** *A 0-1 matrix with pass/trial hierarchical structure described in Section 2.2 (*i.e., *any matrix from the sample space described in Section 2.2) is said to satisfy the* Chaining Recovery Conditions *if for any signal of the form in Invariant 10 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 $\frac{1}{1000}m_k^{-1}$.*
3. *Except for at most $\frac{1}{100}m_{k+1}$ of the measurements, the measurement error assigned to each measurement has ℓ_1 norm at most $\frac{1}{1000}m_k^{-1}$.*

To prove that a random isolation matrix satisfies this additional property with high probability, we use Markov's inequality to bound the number of measurements that are large. This is the same argument as in the second half of the proof of Lemma 25.

Next, we adjust Lemma 13 to include in the bound ϵ not just the ℓ_1 norm of the other positions but also the measurement error. We also modify the definition of a good measurement in Definition 31 to include the measurement error.

Definition 31. *A* good measurement *satisfies one of the following two criteria:*

1. *The measurement is empty; that is, it contains positions with values $\left|f^{(k)}(i)\right| \le \epsilon$ and the total ℓ_1 norm of the positions in the measurement plus the measurement error is less than 1.5ϵ.*

(2) *The measurement contains one spike at position i with $\left|f^{(k)}(i)\right| > \epsilon$ and the ℓ_1 norm of all other positions in this measurement plus the measurement error is less than 0.5ϵ.*

We conclude by noting that with the above changes, we change Lemma 21 to include the ℓ_1 norm of the measurement error $\|y\|_1$, as well as the noise $\|w\|_1$. That is, after the last pass, the error $\|f^{(K+1)}\|_1$ with

$$\|f^{(K+1)}\|_1 \leq \|w\|_1 + \Big\|\sum_{j=0}^{K} \nu_j\Big\|_1 \leq \|w\|_1 + \sum_{j=1}^{K} 6m_j m_J^{-1} = \|w\|_1 + 6\log_a m.$$

Since a is a constant and $\|w\|_1$ and $\|y\|_1$ were normalized to be constant, we have that the overall error is at most

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

□

6. Algorithmic Dimension Reduction

The following dimension reduction theorem holds for sparse vectors.

Theorem 32. *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,

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

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 33. *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 $x = f_m$, $\eta = f - f_m$, $\nu = \mathbf{\Phi} f - V$. □

7. 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 [CT04, RV06, CDD06], 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. A companion paper, in progress, addresses this as well as the logarithmic factor in the approximation error that we give here.

If the measurement matrix is a random Gaussian matrix, as in [CT04, RV06, CDD06], 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 [CT04, RV06, CDD06] 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

[AHU83] A. Aho, J. E. Hopcroft, and J. D. Ullman. *Data structures and algorithms*. Addison-Wesley, Reading, Mass., 1983.

[BC03] 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.

[Bou85] J. Bourgain. On lipschitz embedding of finite metric spaces in hilbert space. *Israel J. Math.*, 52:46–52, 1985.

[CDD06] A. Cohen, W. Dahmen, and R. DeVore. Remarks on compressed sensing. Working draft, 2006.

[CLRS01] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, second edition, 2001.

[CM03] G. Cormode and S. Muthukrishnan. What's hot and what's not: Tracking most frequent items dynamically. In *Proc. ACM Principles of Database Systems*, pages 296–306, 2003.

[CM05] G. Cormode and S. Muthukrishnan. Towards an algorithmic theory of compressed sensing. Technical report, DIMACS, July 2005.

[CM06] G. Cormode and S. Muthukrishnan. Combinatorial algorithms for compressed sensing. In *Proc. 40th IEEE Conference on Information Sciences and Systems*, Princeton, Mar. 2006.

[CRT04] E. Candès, J. Romberg, and T. Tao. Exact signal reconstruction from highly incomplete frequency information. Submitted for publication, June 2004.

[CRTV05] E. J. Candès, M. Rudelson, T. Tao, and R. Vershynin. Error correction via linear programming. In *Proc. FOCS 2005*, Pittsburgh, Oct. 2005.

[CS02] 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.

[CT04] E. J. Candès and T. Tao. Near optimal signal recovery from random projections: Universal encoding strategies? Submitted for publication, revised April 2005, Oct. 2004.

[CT05] E. J. Candès and T. Tao. Decoding by linear programming. Available from `arXiv:math.MG/0502327`, Feb. 2005.

[Don04] D. L. Donoho. Compressed sensing. Unpublished manuscript, Oct. 2004.

[Don05] D. L. Donoho. Neighborly polytopes and sparse solution of underdetermined linear equations. Dept. of Statistics TR 2005-4, Stanford Univ., 2005.

[Don06] D. L. Donoho. Neighborly polytopes and sparse solution of underdetermined linear equations. *IEEE Trans. Inform. Theory*, 2006. To appear.

[DT05] 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.

[DT06] D. L. Donoho and J. Tanner. Thresholds for the recovery of sparse solutions via l1 minimization. In *Proc. 40th IEEE Conference on Information Sciences and Systems*, Princeton, Mar. 2006.

[GGI$^+$02a] 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.

[GGI$^+$02b] 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.

[GMS05] 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.

[GSTV06] A. C. Gilbert, M. J. Strauss, J. A. Tropp, and R. Vershynin. Algorithmic dimension reduction in the ℓ_1 norm. 2006.

[IN05] P. Indyk and A. Naor. Nearest neighbor preserving embeddings. Submitted, 2005.

[JL84] W. B. Johnson and J. Lindenstrauss. Extensions of lipschitz mapping into hilbert space. *Contemporary Mathematics*, (26):189–206, 1984.

[Kas77] B. Kashin. Sections of some finite dimensional sets and classes of smooth functions. *Izv. Acad. Nauk SSSR*, 41:334–351, 1977.

[MPTJ05] S. Mendelson, A. Pajor, and N. Tomczak-Jaegermann. Reconstruction and subgaussian processes. *Comptes Rendus Acad. Sci.*, 340:885–888, 2005.

[NL04] A. Naor and J. R. Lee. Embedding the diamond graph in l_p and dimension reduction in l_1. *Geometric and Functional Analysis*, 14(4):745–747, 2004.

[Pis89] G. Pisier. *The volume of convex bodies and Banach space geometry.* Cambridge University Press, 1989.

[RV05] M. Rudelson and R. Veshynin. Geometric approach to error correcting codes and reconstruction of signals. Available from `arXiv:math.MG/0502299`, Feb. 2005.

[RV06] 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.

[Tal05] M. Talagrand. *The Generic Chaining.* Springer, Berlin, 2005.

[TG05] J. A. Tropp and A. C. Gilbert. Signal recovery from partial information via Orthogonal Matching Pursuit. Submitted to *IEEE Trans. Inform. Theory*, April 2005.