Efficient Sketches for the Set Query Problem*

Eric Price†

Abstract

We develop an algorithm for estimating the values of a vector $x \in \mathbb{R}^n$ over a support S of size k from a randomized sparse binary linear sketch Ax of size $O(k)$. Given Ax and S, we can recover x' with $\|x' - x_S\|_2 \leq \epsilon \|x - x_S\|_2$ with probability at least $1 - k^{-\Omega(1)}$. The recovery takes $O(k)$ time.

While interesting in its own right, this primitive also has a number of applications. For example, we can:

1. Improve the linear k-sparse recovery of heavy hitters in Zipfian distributions with $O(k \log n)$ space from a $1+\epsilon$ approximation to a $1 + o(1)$ approximation, giving the first such approximation in $O(k \log n)$ space when $k \leq O(n^{1-\epsilon})$.
2. Recover block-sparse vectors with $O(k)$ space and a $1+\epsilon$ approximation. Previous algorithms required either $\omega(k)$ space or $\omega(1)$ approximation.

1 Introduction

In recent years, a new "linear" approach for obtaining a succinct approximate representation of n-dimensional vectors (or signals) has been discovered. For any signal x, the representation is equal to Ax, where A is an $m \times n$ matrix, or possibly a random variable chosen from some distribution over such matrices. The vector Ax is often referred to as the *measurement vector* or *linear sketch* of x. Although m is typically much smaller than n, the sketch Ax often contains plenty of useful information about the signal x.

A particularly useful and well-studied problem is that of *stable sparse recovery*. The problem is typically defined as follows: for some norm parameters p and q and an approximation factor $C > 0$, given Ax, recover a vector x' such that

$$\|x' - x\|_p \leq C \cdot \mathrm{Err}_q(x, k), \quad \text{where } \mathrm{Err}_q(x,k) = \min_{k\text{-sparse } \hat{x}} \|\hat{x} - x\|_q \tag{1}$$

where we say that $\hat{x}$ is k-sparse if it has at most k non-zero coordinates. Sparse recovery has applications to numerous areas such as data stream computing [Mut03, Ind07] and compressed sensing [CRT06, Don06], notably for constructing imaging systems that acquire images directly in compressed form (e.g., [DDT+08, Rom09]). The problem has been a subject of extensive study over the last several years, with the goal of designing schemes that enjoy good "compression rate" (i.e., low values of m) as well as good algorithmic properties (i.e., low encoding and recovery times). It is known that there exist distributions of matrices A and associated recovery algorithms that for any x with high probability produce approximations x' satisfying Equation (1) with $\ell_p = \ell_q = \ell_2$, constant approximation factor $C = 1 + \epsilon$, and sketch length $m = O(k \log(n/k))$;[1] it is also known that this sketch length is asymptotically optimal [DIPW10, FPRU10]. Similar results for other combinations of ℓ_p/ℓ_q norms are known as well.

Because it is impossible to improve on the sketch size in the general sparse recovery problem, recently there has been a large body of work on more restricted problems that are amenable to more efficient solutions. This includes *model-based compressive sensing* [BCDH10], which imposes additional constraints (or *models*) on x beyond near-sparsity. Examples of models include *block sparsity*, where the large coefficients tend to cluster together in blocks [BCDH10, EKB09]; *tree sparsity*, where the large coefficients form a rooted, connected tree structure [BCDH10, LD05]; and being *Zipfian*, where we require that the histogram of coefficient size follow a *Zipfian* (or *power law*) distribution.

A sparse recovery algorithm needs to perform two tasks: locating the large coefficients of x and estimating their value. Existing algorithms perform both tasks at the same time. In contrast, we propose decoupling these tasks. In models of interest, including Zipfian signals and block-sparse signals, existing techniques can locate the large coefficients more efficiently or accurately than they can estimate them. Prior to this work, however, estimating the large coefficients after finding them had no better solution than the general sparse recovery problem. We fill this gap by giving an optimal method for estimating the values of the large coefficients after locating them.

*This research has been supported in part by the David and Lucille Packard Fellowship, MADALGO (Center for Massive Data Algorithmics, funded by the Danish National Research Association), NSF grant CCF-0728645, a Cisco Fellowship, and the NSF Graduate Research Fellowship Program.

†MIT CSAIL

[1] In particular, a random Gaussian matrix [CD04] or a random sparse binary matrix ([GLPS09], building on [CCF02, CM04]) has this property with overwhelming probability. See [GI10] for an overview.

We refer to this task as the *Set Query Problem*[2].

Main result. (Set Query Algorithm.) We give a randomized distribution over $O(k) \times n$ binary matrices A such that, for any vector $x \in \mathbb{R}^n$ and set $S \subseteq \{1, \ldots, n\}$ with $|S| = k$, we can recover an x' from $Ax + \nu$ and S with

$$\|x' - x_S\|_2 \leq \epsilon(\|x - x_S\|_2 + \|\nu\|_2)$$

where $x_S \in \mathbb{R}^n$ equals x over S and zero elsewhere. The matrix A has $O(1)$ non-zero entries per column, recovery succeeds with probability $1 - k^{-\Omega(1)}$, and recovery takes $O(k)$ time. This can be achieved for arbitrarily small $\epsilon > 0$, using $O(k/\epsilon^2)$ rows. We achieve a similar result in the ℓ_1 norm.

The set query problem is useful in scenarios when, given a sketch of x, we have some alternative methods for discovering a "good" support of an approximation to x. This is the case, e.g., in block-sparse recovery, where (as we show in this paper) it is possible to identify "heavy" blocks using other methods. It is also a natural problem in itself. In particular, it generalizes the well-studied *point query problem* [CM04], which considers the case that S is a singleton. We note that, although the set query problem for sets of size k can be reduced to k instances of the point query problem, this reduction is less space-efficient than the algorithm we propose, as elaborated below.

Techniques. Our method is related to existing sparse recovery algorithms, including Count-Sketch [CCF02] and Count-Min [CM04]. In fact, our sketch matrix A is almost identical to the one used in Count-Sketch—each column of A has d random locations out of $O(kd)$ each independently set to ± 1, and the columns are independently generated. We can view such a matrix as "hashing" each coordinate to d "buckets" out of $O(kd)$. The difference is that the previous algorithms require $O(k \log k)$ measurements to achieve our error bound (and $d = O(\log k)$), while we only need $O(k)$ measurements and $d = O(1)$.

We overcome two obstacles to bring d down to $O(1)$ and still achieve the error bound with high probability[3]. First, in order to estimate the coordinates x_i, we need a more elaborate method than, say, taking the median of the buckets that i was hashed into. This is because, with constant probability, all such buckets might contain some other elements from S (be "heavy") and therefore using *any* of them as an estimator for y_i would result in too much error. Since, for super-constant values of $|S|$, it is highly likely that such an event will occur for at least one $i \in S$, it follows that this type of estimation results in large error.

We solve this issue by using our knowledge of S. We know when a bucket is "corrupted" (that is, contains more than one element of S), so we only estimate coordinates that lie in a large number of uncorrupted buckets. Once we estimate a coordinate, we subtract our estimation of its value from the buckets it is contained in. This potentially decreases the number of corrupted buckets, allowing us to estimate more coordinates. We show that, with high probability, this procedure can continue until it estimates every coordinate in S.

The other issue with the previous algorithms is that their analysis of their probability of success does not depend on k. This means that, even if the "head" did not interfere, their chance of success would be a constant (like $1 - 2^{-\Omega(d)}$) rather than high probability in k (meaning $1 - k^{-\Omega(d)}$). We show that the errors in our estimates of coordinates have low covariance, which allows us to apply Chebyshev's inequality to get that the total error is concentrated around the mean with high probability.

Related work. A similar recovery algorithm (with $d = 2$) has been analyzed and applied in a streaming context in [EG07]. However, in that paper the authors only consider the case where the vector y is k-sparse. In that case, the termination property alone suffices, since there is no error to bound. Furthermore, because $d = 2$ they only achieve a constant probability of success. In this paper we consider general vectors y so we need to make sure the error remains bounded, and we achieve a high probability of success.

The recovery procedure also has similarities to recovering LDPCs using belief propagation, especially over the binary erasure channel. The similarities are strongest for exact recovery of k-sparse y; our method for bounding the error from noise is quite different.

Applications. Our efficient solution to the set query problem can be combined with existing techniques to achieve sparse recovery under several models.

We say that a vector x follows a *Zipfian* or *power law* distribution with parameter α if $|x_{r(i)}| = \Theta(|x_{r(1)}| \, i^{-\alpha})$ where $r(i)$ is the location of the ith largest coefficient in x. When $\alpha > 1/2$, x is well approximated in the ℓ_2 norm by its sparse approximation. Because a wide variety of real world signals follow power law distributions ([Mit04, BKM+00]), this notion (related to "compressibility"[4]) is often considered to be much of the reason why sparse recovery is interesting [CT06, Cev08]. Prior to this work, sparse recovery of power law distributions has only been solved via general sparse recovery methods: $(1 + \epsilon)\mathrm{Err}_2(x, k)$ error in $O(k \log(n/k))$ measurements.

However, locating the large coefficients in a power law

[2]The term "set query" is in contrast to "point query," used in e.g. [CM04] for estimation of a single coordinate.

[3]In this paper, "high probability" means probability at least $1 - 1/k^c$ for some constant $c > 0$.

[4]A signal is "compressible" when $|x_{r(i)}| = O(|x_{r(1)}| \, i^{-\alpha})$ rather than $\Theta(|x_{r(1)}| \, i^{-\alpha})$ [CT06]. This allows it to decay very quickly then stop decaying for a while; we require that the decay be continuous.

distribution has long been easier than in a general distribution. Using $O(k \log n)$ measurements, the CountSketch algorithm [CCF02] can produce a candidate set $S \subseteq \{1, \ldots, b\}$ with $|S| = O(k)$ that includes all of the top k positions in a power law distribution with high probability (if $\alpha > 1/2$). We can then apply our set query algorithm to recover an approximation x' to x_S. Because we already are using $O(k \log n)$ measurements on CountSketch, we use $O(k \log n)$ rather than $O(k)$ measurements in the set query algorithm to get an $\epsilon/\sqrt{\log n}$ rather than ϵ approximation. This lets us recover a k-sparse x' with $O(k \log n)$ measurements with

$$\|x' - x\|_2 \leq \left(1 + \frac{\epsilon}{\sqrt{\log n}}\right) \mathrm{Err}_2(x, k).$$

This is especially interesting in the common regime where $k < n^{1-c}$ for some constant $c > 0$. Then no previous algorithms achieve better than a $(1+\epsilon)$ approximation with $O(k \log n)$ measurements, and the lower bound in [DIPW10] shows that any $O(1)$ approximation requires $\Omega(k \log n)$ measurements[5]. This means at $\Theta(k \log n)$ measurements, the best approximation changes from $\omega(1)$ to $1 + o(1)$.

Another application is that of finding *block-sparse* approximations. In this application, the coordinate set $\{1 \ldots n\}$ is partitioned into n/b blocks, each of length b. We define a (k, b)-block-sparse vector to be a vector where all non-zero elements are contained in at most k/b blocks. An example of block-sparse data is time series data from n/b locations over b time steps, where only k/b locations are "active". We can define

$$\mathrm{Err}_2(x, k, b) = \min_{(k,b)-\text{block-sparse } \hat{x}} \|x - \hat{x}\|_2 .$$

The block-sparse recovery problem can now be formulated analogously to Equation 1. Since the formulation imposes restrictions on the sparsity patterns, it is natural to expect that one can perform sparse recovery from fewer than $O(k \log(n/k))$ measurements needed in the general case. Because of that reason and the prevalence of approximately block-sparse signals, the problem of stable recovery of variants of block-sparse approximations has been recently a subject of extensive research (e.g., see [EB09, SPH09, BCDH10, CIHB09]). The state of the art algorithm has been given in [BCDH10], who gave a probabilistic construction of a single $m \times n$ matrix A, with $m = O(k + \frac{k}{b} \log n)$, and an $n \log^{O(1)} n$-time algorithm for performing the block-sparse recovery in the ℓ_1 norm (as well as other variants). If the blocks have size $\Omega(\log n)$, the algorithm uses only $O(k)$ measurements, which is a substantial improvement over the general bound. However, the approximation factor C guaranteed by that algorithm was super-constant.

In this paper, we provide a distribution over matrices A, with $m = O(k + \frac{k}{b} \log n)$, which enables solving this problem with a *constant* approximation factor and in the ℓ_2 norm, with high probability. As with Zipfian distributions, first one algorithm tells us where to find the heavy hitters and then the set query algorithm estimates their values. In this case, we modify the algorithm of [ABI08] to find *block heavy hitters*, which enables us to find the support of the $\frac{k}{b}$ "most significant blocks" using $O(\frac{k}{b} \log n)$ measurements. The essence is to perform dimensionality reduction of each block from b to $O(\log n)$ dimensions, then estimate the result with a linear hash table. For each block, most of the projections are estimated pretty well, so the median is a good estimator of the block's norm. Once the support is identified, we can recover the coefficients using the set query algorithm.

2 Preliminaries

2.1 Notation

For $n \in \mathbb{Z}^+$, we denote $\{1, \ldots, n\}$ by $[n]$. Suppose $x \in \mathbb{R}^n$. Then for $i \in [n]$, $x_i \in \mathbb{R}$ denotes the value of the i-th coordinate in x. As an exception, $e_i \in \mathbb{R}^n$ denotes the elementary unit vector with a one at position i. For $S \subseteq [n]$, x_S denotes the vector $x' \in R^n$ given by $x'_i = x_i$ if $i \in S$, and $x'_i = 0$ otherwise. We use $\mathrm{supp}(x)$ to denote the support of x. We use upper case letters to denote sets, matrices, and random distributions. We use lower case letters for scalars and vectors.

2.2 Negative Association

This paper would like to make a claim of the form "We have k observations each of whose error has small expectation and variance. Therefore the average error is small with high probability in k." If the errors were independent this would be immediate from Chebyshev's inequality, but our errors depend on each other. Fortunately, our errors have some tendency to behave even better than if they were independent: the more noise that appears in one coordinate, the less remains to land in other coordinates. We use *negative dependence* to refer to this general class of behavior. The specific forms of negative dependence we use are *negative association* and *approximate negative correlation*; see Appendix A for details on these notions.

[5]The lower bound only applies to geometric distributions, not Zipfian ones. However, our algorithm applies to more general *sub-Zipfian* distributions (defined in Section 4.1), which includes both.

3 Set-Query Algorithm

Theorem 3.1. *There is a randomized sparse binary sketch matrix A and recovery algorithm $\mathscr{A}$, such that for any $x \in \mathbb{R}^n$, $S \subseteq [n]$ with $|S| = k$, $x' = \mathscr{A}(Ax + \nu, S) \in \mathbb{R}^n$ has $\mathrm{supp}(x') \subseteq S$ and*

$$\|x' - x_S\|_2 \leq \epsilon(\|x - x_S\|_2 + \|\nu\|_2)$$

with probability at least $1 - 1/k^c$. A has $O(\frac{c}{\epsilon^2}k)$ rows and $O(c)$ non-zero entries per column, and $\mathscr{A}$ runs in $O(ck)$ time.

One can achieve $\|x' - x_S\|_1 \leq \epsilon(\|x - x_S\|_1 + \|\nu\|_1)$ under the same conditions, but with only $O(\frac{c}{\epsilon}k)$ rows.

We will first show Theorem 3.1 for a constant $c = 1/3$ rather than for general c. Parallel repetition gives the theorem for general c, as described in Section 3.7. We will also only show it with entries of A being in $\{0, 1, -1\}$. By splitting each row in two, one for the positive and one for the negative entries, we get a binary matrix with the same properties. The paper focuses on the more difficult ℓ_2 result; see Appendix B for details on the ℓ_1 result.

3.1 Intuition

We call x_S the "head" and $x - x_S$ the "tail." The head probably contains the heavy hitters, with much more mass than the tail of the distribution. We would like to estimate x_S with zero error from the head and small error from the tail with high probability.

Our algorithm is related to the standard Count-Sketch [CCF02] and Count-Min [CM04] algorithms. In order to point out the differences, let us examine how they perform on this task. These algorithms show that hashing into a single $w = O(k)$ sized hash table is good in the sense that each point x_i has:

1. Zero error from the head with constant probability (namely $1 - \frac{k}{w}$).
2. A small amount of error from the tail in expectation (and hence with constant probability).

They then iterate this procedure d times and take the median, so that each estimate has small error with probability $1 - 2^{-\Omega(d)}$. With $d = O(\log k)$, we get that all k estimates in S are good with $O(k \log k)$ measurements with high probability in k. With fewer measurements, however, some x_i will probably have error from the head. If the head is much larger than the tail (such as when the tail is zero), this is a major problem. Furthermore, with $O(k)$ measurements the error from the tail would be small only in expectation, not with high probability.

We make three observations that allow us to use only $O(k)$ measurements to estimate x_S with error relative to the tail with high probability in k.

1. The total error from the tail over a support of size k is concentrated more strongly than the error at a single point: the error probability drops as $k^{-\Omega(d)}$ rather than $2^{-\Omega(d)}$.
2. The error from the head can be avoided if one knows where the head is, by modifying the recovery algorithm.
3. The error from the tail remains concentrated after modifying the recovery algorithm.

For simplicity this paper does not directly show (1), only (2) and (3). The modification to the algorithm to achieve (2) is quite natural, and described in detail and illustrated in Section 3.2. Rather than estimate every coordinate in S immediately, we only estimate those coordinates which mostly do not overlap with other coordinates in S. In particular, we only estimate x_i as the median of at least $d - 2$ positions that are not in the image of $S \setminus \{i\}$. Once we learn x_i, we can subtract Ax_ie_i from the observed Ax and repeat on $A(x - x_ie_i)$ and $S \setminus \{i\}$. Because we only look at positions that are in the image of only one remaining element of S, this avoids any error from the head. We show in Section 3.3 that this algorithm never gets stuck; we can always find some position that mostly doesn't overlap with the image of the rest of the remaining support.

We then show that the error from the tail has low expectation, and that it is strongly concentrated. We think of the tail as noise located in each "cell" (coordinate in the image space). We decompose the error of our result into two parts: the "point error" and the "propagation". The point error is error introduced in our estimate of some x_i based on noise in the cells that we estimate x_i from, and equals the median of the noise in those cells. The "propagation" is the error that comes from point error in estimating other coordinates in the same connected component; these errors propagate through the component as we subtract off incorrect estimates of each x_i.

Section 3.4 shows how to decompose the total error in terms of point errors and the component sizes. The two following sections bound the expectation and variance of these two quantities and show that they obey some notions of negative dependence. We combine these errors in Section 3.7 to get Theorem 3.1 with a specific c (namely $c = 1/3$). We then use parallel repetition to achieve Theorem 3.1 for arbitrary c.

3.2 Algorithm

We describe the sketch matrix A and recovery procedure in Algorithm 3.1. Unlike Count-Sketch [CCF02] or Count-Min [CM04], our A is not split into d hash tables

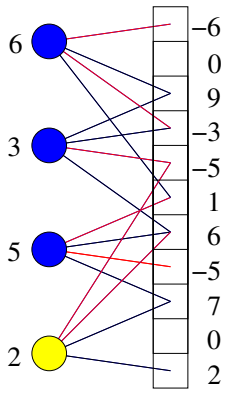

Figure 1: An instance of the set query problem. There are n vertices on the left, corresponding to x, and the table on the right represents Ax. Each vertex i on the left maps to d cells on the right, randomly increasing or decreasing the value in each cell by x_i. We represent addition by black lines, and subtraction by red lines. We are told the locations of the heavy hitters, which we represent by blue circles; the rest is represented by yellow circles.

of size $O(k)$. Instead, it has a single $w = O(d^2k/\epsilon^2)$ sized hash table where each coordinate is hashed into d unique positions. We can think of A as a random d-uniform hypergraph, where the non-zero entries in each column correspond to the terminals of a hyperedge. We say that A is drawn from $\mathbb{G}^d(w, n)$ with random signs associated with each (hyperedge, terminal) pair. We do this so we will be able to apply existing theorems on random hypergraphs.

Figure 1 shows an example Ax for a given x, and Figure 2 demonstrates running the recovery procedure on this instance.

Lemma 3.1. *Algorithm 3.1 runs in time $O(dk)$.*

Proof. A has d entries per column. For each of the at most dk rows q in the image of S, we can store the preimages $P(q)$. We also keep track of the sets of possible next hyperedges, $J_i = \{j \mid |L_j| \geq d - i\}$ for $i \in \{1, 2\}$. We can compute these in an initial pass in $O(dk)$. Then in each iteration, we remove an element $j \in J_1$ or J_2 and update x'_j, b, and T in $O(d)$ time. We then look at the two or fewer non-isolated vertices q in hyperedge j, and remove j from the associated $P(q)$. If this makes $|P(q)| = 1$, we check whether to insert the element in $P(q)$ into the J_i. Hence the inner loop takes $O(d)$ time, for $O(dk)$ total. □

3.3 Exact Recovery

The random hypergraph $\mathbb{G}^d(w, k)$ of k random d-uniform hyperedges on w vertices is well studied in [KŁ02]. We use their results to show that the algorithm successfully

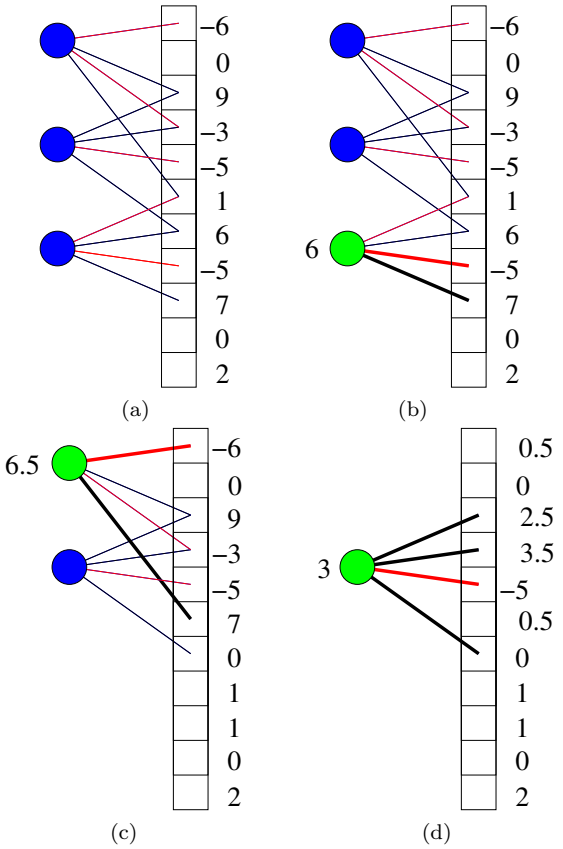

Figure 2: Example run of the algorithm. Part (a) shows the state as considered by the algorithm: Ax and the graph structure corresponding to the given support. In part (b), the algorithm chooses a hyperedge with at least $d - 2$ isolated vertices and estimates the value as the median of those isolated vertices multiplied by the sign of the corresponding edge. In part (c), the image of the first vertex has been removed from Ax and we repeat on the smaller graph. We continue until the entire support has been estimated, as in part (d).

Definition of sketch matrix A. For a constant d, let A be a $w \times n = O(\frac{d^2}{\epsilon^2}k) \times n$ matrix where each column is chosen independently uniformly at random over all exactly d-sparse columns with entries in $\{-1, 0, 1\}$. We can think of A as the incidence matrix of a random d-uniform hypergraph with random signs.

Recovery procedure.

procedure SETQUERY(A, S, b) ▷ Recover approximation x' to x_S from $b = Ax + \nu$
  $T \leftarrow S$
  while $|T| > 0$ **do**
    Define $P(q) = \{j \mid A_{qj} \neq 0, j \in T\}$ as the set of hyperedges in T that contain q.
    Define $L_j = \{q \mid A_{qj} \neq 0, |P(q)| = 1\}$ as the set of isolated vertices in hyperedge j.
    Choose a random $j \in T$ such that $|L_j| \geq d-1$. If this is not possible, find a random $j \in T$ such that $|L_j| \geq d-2$. If neither is possible, abort.
    $x'_j \leftarrow \text{median}_{q \in L_j} A_{qj} b_q$
    $b \leftarrow b - x'_j A e_j$
    $T \leftarrow T \setminus \{j\}$
  end while
  return x'
end procedure

Algorithm 3.1: Recovering a signal given its support.

terminates with high probability, and that most hyperedges are chosen with at least $d-1$ isolated vertices:

Lemma 3.2. *With probability at least $1 - O(1/k)$, Algorithm 3.1 terminates without aborting. Furthermore, in each component at most one hyperedge is chosen with only $d-2$ isolated vertices.*

We will show this by building up a couple lemmas. We define a connected hypergraph H with r vertices on s hyperedges to be a *hypertree* if $r = s(d-1)+1$ and to be *unicyclic* if $r = s(d-1)$. Then Theorem 4 of [KŁ02] shows that, if the graph is sufficiently sparse, $\mathbb{G}^d(w,k)$ is probably composed entirely of hypertrees and unicyclic components. The precise statement is as follows[6]:

Lemma 3.3 (Theorem 4 of [KŁ02])**.** *Let $m = w/d(d-1) - k$. Then with probability $1 - O(d^5 w^2/m^3)$, $\mathbb{G}^d(w,k)$ is composed entirely of hypertrees and unicyclic components.*

We use a simple consequence:

Corollary 3.1. *If $d = O(1)$ and $w \geq 2d(d-1)k$, then with probability $1 - O(1/k)$, $\mathbb{G}^d(w,k)$ is composed entirely of hypertrees and unicyclic*

[6]Their statement of the theorem is slightly different. This is the last equation in their proof of the theorem.

We now prove some basic facts about hypertrees and unicyclic components:

Lemma 3.4. *Every hypertree has a hyperedge incident on at least $d-1$ isolated vertices. Every unicyclic component either has a hyperedge incident on $d-1$ isolated vertices or has a hyperedge incident on $d-2$ isolated vertices, the removal of which turns the unicyclic component into a hypertree.*

Proof. Let H be a connected component of s hyperedges and r vertices.

If H is a hypertree, $r = (d-1)s + 1$. Because H has only ds total (hyperedge, incident vertex) pairs, at most $2(s-1)$ of these pairs can involve vertices that appear in two or more hyperedges. Thus at least one of the s edges is incident on at most one vertex that is not isolated, so some edge has $d-1$ isolated vertices.

If H is unicyclic, $r = (d-1)s$ and so at most $2s$ of the (hyperedge, incident vertex) pairs involve non-isolated vertices. Therefore on average, each edge has $d-2$ isolated vertices. If no edge is incident on at least $d-1$ isolated vertices, every edge must be incident on exactly $d-2$ isolated vertices. In that case, each edge is incident on exactly two non-isolated vertices and each non-isolated vertex is in exactly two edges. Hence we can perform an Eulerian tour of all the edges, so removing any edge does not disconnect the graph. After removing the edge, the graph has $s' = s-1$ edges and $r' = r-d+2$ vertices; therefore $r' = (d-1)s'+1$ so the graph is a hypertree. □

Corollary 3.1 and Lemma 3.4 combine to show Lemma 3.2.

3.4 Total error in terms of point error and component size

Define $C_{i,j}$ to be the event that hyperedges i and j are in the same component, and $D_i = \sum_j C_{i,j}$ to be the number of hyperedges in the same component as i. Define L_i to be the cells that are used to estimate i; so $L_i = \{q \mid A_{qj} \neq 0, |P(q)| = 1\}$ at the round of the algorithm when i is estimated. Define $Y_i = \text{median}_{q \in L_i} A_{qi}(b - Ax_S)_q$ to be the "point error" for hyperedge i, and x' to be the output of the algorithm. Then the deviation of the output at any coordinate i is at most twice the sum of the point errors in the component containing i:

Lemma 3.5.

$$|(x' - x_S)_i| \leq 2 \sum_{j \in S} |Y_j| \, C_{i,j}.$$

Proof. Let $T_i = (x' - x_S)_i$, and define $R_i = \{j \mid j \neq i, \exists q \in L_i \text{ s.t. } A_{qj} \neq 0\}$ to be the set of hyperedges that

overlap with the cells used to estimate i. Then from the description of the algorithm, it follows that

$$T_i = \underset{q \in L_i}{\text{median}} A_{qi}((b - Ax_S)_q - \sum_j A_{qj} T_j)$$

$$|T_i| \leq |Y_i| + \sum_{j \in R_i} |T_j| .$$

We can think of the R_i as a directed acyclic graph (DAG), where there is an edge from j to i if $j \in R_i$. Then if $p(i,j)$ is the number of paths from i to j,

$$|T_i| \leq \sum_j p(j,i) |Y_i| .$$

Let $r(i) = |\{j \mid i \in R_j\}|$ be the outdegree of the DAG. Because the L_i are disjoint, $r(i) \leq d - |L_i|$. From Lemma 3.2, $r(i) \leq 1$ for all but one hyperedge in the component, and $r(i) \leq 2$ for that one. Hence $p(i,j) \leq 2$ for any i and j, giving the result. □

We use the following corollary:

Corollary 3.2.

$$\|x' - x_S\|_2^2 \leq 4 \sum_{i \in S} D_i^2 Y_i^2$$

Proof.

$$\|x' - x_S\|_2^2 = \sum_{i \in S} (x' - x_S)_i^2 \leq 4 \sum_{i \in S} (\sum_{\substack{j \in S \\ C_{i,j}=1}} |Y_j|)^2$$

$$\leq 4 \sum_{i \in S} D_i \sum_{\substack{j \in S \\ C_{i,j}=1}} |Y_j|^2 = 4 \sum_{i \in S} D_i^2 Y_i^2$$

where the second inequality is the power means inequality. □

The D_j and Y_j are independent from each other, since one depends only on A over S and one only on A over $[n] \setminus S$. Therefore we can analyze them separately; the next two sections show bounds and negative dependence results for Y_j and D_j, respectively.

3.5 Bound on point error

Recall from Section 3.4 that based entirely on the set S and the columns of A corresponding to S, we can identify the positions L_i used to estimate x_i. We then defined the "point error"

$$Y_i = \underset{q \in L_i}{\text{median}} A_{qi}(b - Ax_S)_q = \underset{q \in L_i}{\text{median}} A_{qi}(A(x - x_S) + \nu)_q$$

and showed how to relate the total error to the point error. Here we would like to show that the Y_i have bounded moments and are negatively dependent. Unfortunately, it turns out that the Y_i are not negatively associated so it is unclear how to show negative dependence directly. Instead, we will define some other variables Z_i that are always larger than the corresponding Y_i. We will then show that the Z_i have bounded moments and negative association.

We use the term "NA" throughout the proof to denote negative association. For the definition of negative association and relevant properties, see Appendix A.

Lemma 3.6. *Suppose $d \geq 7$ and define $\mu = O(\frac{\epsilon^2}{k}(\|x - x_S\|_2^2 + \|\nu\|_2^2))$. There exist random variables Z_i such that the variables Y_i^2 are stochastically dominated by Z_i, the Z_i are negatively associated, $\mathrm{E}[Z_i] = \mu$, and $\mathrm{E}[Z_i^2] = O(\mu^2)$.*

Proof. The choice of the L_i depends only on the values of A over S; hence conditioned on knowing L_i we still have $A(x - x_S)$ distributed randomly over the space. Furthermore the distribution of A and the reconstruction algorithm are invariant under permutation, so we can pretend that ν is permuted randomly before being added to Ax. Define $B_{i,q}$ to be the event that $q \in \text{supp}(Ae_i)$, and define $D_{i,q} \in \{-1, 1\}$ independently at random. Then define the random variable

$$V_q = (b - Ax_S)_q = \nu_q + \sum_{i \in [n] \setminus S} x_i B_{i,q} D_{i,q}.$$

Because we want to show concentration of measure, we would like to show negative association (NA) of the $Y_i = \text{median}_{q \in L_i} A_{qi} V_q$. We know ν is a permutation distribution, so it is NA [JP83]. The $B_{i,q}$ for each i as a function of q are chosen from a Fermi-Dirac model, so they are NA [DR96]. The $B_{i,q}$ for different i are independent, so all the $B_{i,q}$ variables are NA. Unfortunately, the $D_{i,q}$ can be negative, which means the V_q are not necessarily NA. Instead we will find some NA variables that dominate the V_q. We do this by considering V_q as a distribution over D.

Let $W_q = \mathrm{E}_D[V_q^2] = \nu_q^2 + \sum_{i \in [n] \setminus S} x_i^2 B_{i,q}$. As increasing functions of NA variables, the W_q are NA. By Markov's inequality $\Pr_D[V_q^2 \geq cW_q] \leq \frac{1}{c}$, so after choosing the $B_{i,q}$ and as a distribution over D, V_q^2 is dominated by the random variable $U_q = W_q F_q$ where F_q is, independently for each q, given by the p.d.f. $f(c) = 1/c^2$ for $c \geq 1$ and $f(c) = 0$ otherwise. Because the distribution of V_q over D is independent for each q, the U_q jointly dominate the V_q^2.

The U_q are the componentwise product of the W_q with independent positive random variables, so they too are NA. Then define

$$Z_i = \underset{q \in L_i}{\text{median}} U_q.$$

As an increasing function of disjoint subsets of NA variables, the Z_i are NA. We also have that

$$\begin{aligned} Y_i^2 &= (\operatorname*{median}_{q\in L_i} A_{qi}V_q)^2 \le (\operatorname*{median}_{q\in L_i} |V_q|)^2 \\ &= \operatorname*{median}_{q\in L_i} V_q^2 \le \operatorname*{median}_{q\in L_i} U_q = Z_i \end{aligned}$$

so the Z_i stochastically dominate Y_i^2. We now will bound $\mathrm{E}[Z_i^2]$. Define

$$\begin{aligned} \mu = E[W_q] &= \mathrm{E}[\nu_q^2] + \sum_{i\in[n]\setminus S} x_i^2 E[B_{i,q}] \\ &= \frac{d}{w}\|x - x_S\|_2^2 + \frac{1}{w}\|\nu\|_2^2 \\ &\le \frac{\epsilon^2}{k}(\|x - x_S\|_2^2 + \|\nu\|_2^2). \end{aligned}$$

Then we have

$$\begin{aligned} \Pr[W_q \ge c\mu] &\le \frac{1}{c} \\ \Pr[U_q \ge c\mu] &= \int_0^\infty f(x)\Pr[W_q \ge c\mu/x]dx \\ &\le \int_1^c \frac{1}{x^2}\frac{x}{c}dx + \int_c^\infty \frac{1}{x^2}dx = \frac{1+\ln c}{c} \end{aligned}$$

Because the U_q are NA, they satisfy marginal probability bounds [DR96]:

$$\Pr[U_q \ge t_q, q\in[w]] \le \prod_{i\in[n]} \Pr[U_q \ge t_q]$$

for any t_q. Therefore

$$\begin{aligned} \Pr[Z_i \ge c\mu] &\le \sum_{\substack{T\subset L_i \\ |T|=|L_i|/2}} \prod_{q\in T} Pr[U_q \ge c\mu] \\ &\le 2^{|L_i|}\left(\frac{1+\ln c}{c}\right)^{|L_i|/2} \\ \Pr[Z_i \ge c\mu] &\le \left(4\frac{1+\ln c}{c}\right)^{d/2-1} \qquad (2) \end{aligned}$$

If $d \ge 7$, this makes $\mathrm{E}[Z_i] = O(\mu)$ and $\mathrm{E}[Z_i^2] = O(\mu^2)$. □

3.6 Bound on component size

Lemma 3.7. *Let D_i be the number of hyperedges in the same component as hyperedge i. Then for any $i \ne j$,*

$$Cov(D_i^2, D_j^2) = \mathrm{E}[D_i^2 D_j^2] - \mathrm{E}[D_i^2]^2 \le O(\frac{\log^6 k}{\sqrt{k}}).$$

Furthermore, $\mathrm{E}[D_i^2] = O(1)$ *and* $\mathrm{E}[D_i^4] = O(1)$.

Proof. The intuition is that if one component gets larger, other components tend to get smaller. Also the graph is very sparse, so component size is geometrically distributed. There is a small probability that i and j are connected, in which case D_i and D_j are positively correlated, but otherwise D_i and D_j should be negatively correlated. However analyzing this directly is rather difficult, because as one component gets larger, the remaining components have a lower average size but higher variance. Our analysis instead takes a detour through the hypergraph where each hyperedge is picked independently with a probability that gives the same expected number of hyperedges. This distribution is easier to analyze, and only differs in a relatively small $\tilde{O}(\sqrt{k})$ hyperedges from our actual distribution. This allows us to move between the regimes with only a loss of $\tilde{O}(\frac{1}{\sqrt{k}})$, giving our result.

Suppose instead of choosing our hypergraph from $\mathbb{G}^d(w,k)$ we chose it from $\mathbb{G}^d(w, \frac{k}{\binom{w}{d}})$; that is, each hyperedge appeared independently with the appropriate probability to get k hyperedges in expectation. This model is somewhat simpler, and yields a very similar hypergraph $\overline{G}$. One can then modify $\overline{G}$ by adding or removing an appropriate number of random hyperedges I to get exactly k hyperedges, forming a uniform $G \in \mathbb{G}^d(w,k)$. By the Chernoff bound, $|I| \le O(\sqrt{k}\log k)$ with probability $1 - \frac{1}{k^{\Omega(1)}}$.

Let $\overline{D}_i$ be the size of the component containing i in $\overline{G}$, and $H_i = D_i^2 - \overline{D}_i^2$. Let E denote the event that any of the D_i or $\overline{D}_i$ is more than $C\log k$, or that more than $C\sqrt{k}\log k$ hyperedges lie in I, for some constant C. Then E happens with probability less than $\frac{1}{k^5}$ for some C, so it has negligible influence on $\mathrm{E}[D_i^2 D_j^2]$. Hence the rest of this proof will assume E does not happen.

Therefore $H_i = 0$ if none of the $O(\sqrt{k}\log k)$ random hyperedges in I touch the $O(\log k)$ hyperedges in the components containing i in $\overline{G}$, so $H_i = 0$ with probability at least $1 - O(\frac{\log^2 k}{\sqrt{k}})$. Even if $H_i \ne 0$, we still have $|H_i| \le (D_i^2 + D_j^2) \le O(\log^2 k)$.

Also, we show that the $\overline{D}_i^2$ are negatively correlated, when conditioned on being in separate components. Let $\overline{D}(n,p)$ denote the distribution of the component size of a random hyperedge on $\mathbb{G}^d(n,p)$, where p is the probability an hyperedge appears. Then $\overline{D}(n,p)$ dominates $\overline{D}(n',p)$ whenever $n > n'$ — the latter hypergraph is contained within the former. If $\overline{C}_{i,j}$ is the event that i and j are connected in $\overline{G}$, this means

$$\mathrm{E}[\overline{D}_i^2 \mid \overline{D}_j = t, \overline{C}_{i,j} = 0]$$

is a decreasing function in t, so we have negative corre-

lation:

$$\begin{aligned}\mathrm{E}[\overline{D}_i^2\overline{D}_j^2 \mid \overline{C}_{i,j}=0] &\le \mathrm{E}[\overline{D}_i^2 \mid \overline{C}_{i,j}=0]\,\mathrm{E}[\overline{D}_j^2 \mid \overline{C}_{i,j}=0]\\ &\le \mathrm{E}[\overline{D}_i^2]\,\mathrm{E}[\overline{D}_j^2].\end{aligned}$$

Furthermore for $i \neq j$, $\Pr[\overline{C}_{i,j}=1] = \mathrm{E}[\overline{C}_{i,j}] = \frac{1}{k-1}\sum_{l\neq i}\mathrm{E}[\overline{C}_{i,l}] = \frac{\mathrm{E}[\overline{D}_i]-1}{k-1} = O(1/k)$. Hence

$$\begin{aligned}\mathrm{E}[\overline{D}_i^2\overline{D}_j^2] =& \mathrm{E}[\overline{D}_i^2\overline{D}_j^2 \mid \overline{C}_{i,j}=0]\Pr[\overline{C}_{i,j}=0]+\\ & \mathrm{E}[\overline{D}_i^2\overline{D}_j^2 \mid \overline{C}_{i,j}=1]\Pr[\overline{C}_{i,j}=1]\\ \le& \mathrm{E}[\overline{D}_i^2]\,\mathrm{E}[\overline{D}_j^2] + O(\frac{\log^4 k}{k}).\end{aligned}$$

Therefore

$$\begin{aligned}&\mathrm{E}[D_i^2D_j^2]\\ =&\mathrm{E}[(\overline{D}_i^2+H_i)(\overline{D}_j^2+H_j)]\\ =&\mathrm{E}[\overline{D}_i^2\overline{D}_j^2] + 2\,\mathrm{E}[H_i\overline{D}_j^2] + \mathrm{E}[H_iH_j]\\ \le&\mathrm{E}[\overline{D}_i^2]\,\mathrm{E}[\overline{D}_j^2] + O(2\frac{\log^2 k}{\sqrt{k}}\log^4 k + \frac{\log^2 k}{\sqrt{k}}\log^2 k)\\ =&\mathrm{E}[D_i^2-H_i]^2 + O(\frac{\log^6 k}{\sqrt{k}})\\ =&\mathrm{E}[D_i^2]^2 - 2\,\mathrm{E}[H_i]\,\mathrm{E}[D_i^2] + \mathrm{E}[H_i]^2 + O(\frac{\log^6 k}{\sqrt{k}})\\ \le&\mathrm{E}[D_i^2]^2 + O(\frac{\log^6 k}{\sqrt{k}})\end{aligned}$$

Now to bound $\mathrm{E}[D_i^4]$ in expectation. Because our hypergraph is exceedingly sparse, the size of a component can be bounded by a branching process that dies out with constant probability at each step. Using this method, Equations 71 and 72 of [COMS07] state that $\Pr[\overline{D} \ge k] \le e^{-\Omega(k)}$. Hence $\mathrm{E}[\overline{D}_i^2] = O(1)$ and $\mathrm{E}[\overline{D}_i^4] = O(1)$. Because H_i is 0 with high probability and $O(\log^2 k)$ otherwise, this immediately gives $\mathrm{E}[D_i^2] = O(1)$ and $\mathrm{E}[D_i^4] = O(1)$. □

3.7 Wrapping it up

Recall from Corollary 3.2 that our total error

$$\|x' - x_S\|_2^2 \le 4\sum_i Y_i^2D_i^2 \le 4\sum_i Z_iD_i^2.$$

The previous sections show that Z_i and D_i^2 each have small expectation and covariance. This allows us to apply Chebyshev's inequality to concentrate $4\sum_i Z_iD_i^2$ about its expectation, bounding $\|x' - x_S\|_2$ with high probability:

Lemma 3.8. *We can recover x' from $Ax+\nu$ and S with*

$$\|x' - x_S\|_2 \le \epsilon(\|x - x_S\|_2 + \|\nu\|_2)$$

with probability at least $1 - \frac{1}{c^2k^{1/3}}$ in $O(k)$ recovery time. Our A has $O(\frac{c}{\epsilon^2}k)$ rows and sparsity $O(1)$ per column.

Proof. Our total error is

$$\|x' - x_S\|_2^2 \le 4\sum_i Y_i^2D_i^2 \le 4\sum_i Z_iD_i^2.$$

Then by Lemma 3.6 and Lemma 3.7,

$$\mathrm{E}[4\sum_i Z_iD_i^2] = 4\sum_i \mathrm{E}[Z_i]\,\mathrm{E}[D_i^2] = k\mu$$

where $\mu = O(\frac{\epsilon^2}{k}(\|x - x_S\|_2^2 + \|\nu\|_2^2))$. Furthermore,

$$\begin{aligned}\mathrm{E}[(\sum_i Z_iD_i^2)^2] &= \sum_i \mathrm{E}[Z_i^2D_i^4] + \sum_{i\neq j}\mathrm{E}[Z_iZ_jD_i^2D_j^2]\\ &= \sum_i \mathrm{E}[Z_i^2]\,\mathrm{E}[D_i^4] + \sum_{i\neq j}\mathrm{E}[Z_iZ_j]\,\mathrm{E}[D_i^2D_j^2]\\ &\le \sum_i O(\mu^2) + \sum_{i\neq j}\mathrm{E}[Z_i]\,\mathrm{E}[Z_j](\mathrm{E}[D_i^2]^2 + O(\frac{\log^6 k}{\sqrt{k}}))\\ &= O(\mu^2k\sqrt{k}\log^6 k) + k(k-1)\,\mathrm{E}[Z_iD_i^2]^2\\ \mathrm{Var}(\sum_i Z_iD_i^2) &= \mathrm{E}[(\sum_i Z_iD_i^2)^2] - k^2\,\mathrm{E}[Z_iD_i^2]^2\\ &\le O(\mu^2k\sqrt{k}\log^6 k)\end{aligned}$$

By Chebyshev's inequality, this means

$$\Pr[4\sum_i Z_iD_i^2 \ge (1+c)\mu k] \le O(\frac{\log^6 k}{c^2\sqrt{k}})$$

$$\Pr[\|x' - x_S\|_2^2 \ge (1+c)C\epsilon^2(\|x - x_S\|_2^2 + \|\nu\|_2^2)] \le O(\frac{1}{c^2k^{1/3}})$$

for some constant C. Rescaling ϵ down by $\sqrt{C(1+c)}$, we can get

$$\|x' - x_S\|_2 \le \epsilon(\|x - x_S\|_2 + \|\nu\|_2)$$

with probability at least $1 - \frac{1}{c^2k^{1/3}}$: □

Now we shall go from $k^{-1/3}$ probability of error to k^{-c} error for arbitrary c, with $O(c)$ multiplicative cost in time and space. We simply perform Lemma 3.8 $O(c)$ times in parallel, and output the pointwise median of the results. By a standard parallel repetition argument, this gives our main result:

Theorem 3.1. *We can recover x' from $Ax + \nu$ and S with*

$$\|x' - x_S\|_2 \le \epsilon(\|x - x_S\|_2 + \|\nu\|_2)$$

with probability at least $1 - \frac{1}{k^c}$ in $O(ck)$ recovery time. Our A has $O(\frac{c}{\epsilon^2}k)$ rows and sparsity $O(c)$ per column.

Proof. Lemma 3.8 gives an algorithm that achieves $O(k^{-1/3})$ probability of error. We will show here how to achieve k^{-c} probability of error with a linear cost in c, via a standard parallel repetition argument.

Suppose our algorithm gives an x' such that $\|x' - x_S\|_2 \leq \mu$ with probability at least $1-p$, and that we run this algorithm m times independently in parallel to get output vectors $x^1, \ldots, x^m$. We output y given by $y_i = \text{median}_{j\in[m]}(x^j)_i$, and claim that with high probability $\|y - x_S\|_2 \leq \mu\sqrt{3}$.

Let $J = \{j \in [m] \mid \left\|x^j - x_S\right\|_2 \leq \mu\}$. Each $j \in [m]$ lies in J with probability at least $1-p$, so the chance that $|J| \leq 3m/4$ is less than $\binom{m}{m/4}p^{m/4} \leq (4ep)^{m/4}$. Suppose that $|J| \geq 3m/4$. Then for all $i \in S$, $\left|\{j \in J \mid (x^j)_i \leq y_i\}\right| \geq |J| - \frac{m}{2} \geq |J|/3$ and similarly $\left|\{j \in J \mid (x^j)_i \geq y_i\}\right| \geq |J|/3$. Hence for all $i \in S$, $|y_i - x_i|$ is smaller than at least $|J|/3$ of the $\left|(x^j)_i - x_i\right|$ for $j \in J$. Hence

$$|J|\,\mu^2 \geq \sum_{i\in S}\sum_{j\in J}((x^j)_i - x_i)^2 \geq \sum_{i\in S}\frac{|J|}{3}(y_i - x_i)^2$$
$$= \frac{|J|}{3}\|y - x\|_2^2$$

or

$$\|y - x\|_2 \leq \sqrt{3}\mu$$

with probability at least $1 - (4ep)^{m/4}$.

Using Lemma 3.8 to get $p = \frac{1}{16k^{1/3}}$ and $\mu = \epsilon(\|x - x_S\|_2 + \|\nu\|_2)$, with $m = 12c$ repetitions we get Theorem 3.1. □

4 Applications

We give two applications where the set query algorithm is a useful primitive.

4.1 Heavy Hitters of sub-Zipfian distributions

For a vector x, let r_i be the index of the ith largest element, so $|x_{r_i}|$ is non-increasing in i. We say that x is *Zipfian with parameter* α if $|x_{r_i}| = \Theta(|x_{r_1}|\, i^{-\alpha})$. We say that x is *sub-Zipfian with parameters (k, α)* if there exists a non-increasing function f with $|x_{r_i}| = \Theta(f(i)i^{-\alpha})$ for all $i \geq k$. A Zipfian with parameter α is a sub-Zipfian with parameter (k, α) for all k, using $f(i) = |x_{r_1}|$.

The Zipfian heavy hitters problem is, given a linear sketch Ax of a Zipfian x and a parameter k, to find a k-sparse x' with minimal $\|x - x'\|_2$ (up to some approximation factor). We require that x' be k-sparse (and no more) because we want to find the heavy hitters themselves, not to find them as a proxy for approximating x.

Zipfian distributions are common in real-world data sets, and finding heavy hitters is one of the most important problems in data streams. Therefore this is a very natural problem to try to improve; indeed, the original paper on Count-Sketch discussed it [CCF02]. They show a result complementary to our work, namely that one can find the support efficiently:

Lemma 4.1 (Section 4.1 of [CCF02])**.** *If x is sub-Zipfian with parameter (k, α) and $\alpha > 1/2$, one can recover a candidate support set S with $|S| = O(k)$ from Ax such that $\{r_1, \ldots, r_k\} \subseteq S$. A has $O(k \log n)$ rows and recovery succeeds with high probability in n.*

Proof sketch. Let $S_k = \{r_1, \ldots, r_k\}$. With $O(\frac{1}{\epsilon^2}k\log n)$ measurements, Count-Sketch identifies each x_i to within $\frac{\epsilon}{k}\|x - x_{S_k}\|_2$ with high probability. If $\alpha > 1/2$, this is less than $|x_{r_k}|/3$ for appropriate ϵ. But $|x_{r_{9k}}| \leq |x_{r_k}|/3$. Hence only the largest $9k$ elements of x could be estimated as larger than anything in x_{S_k}, so the locations of the largest $9k$ estimated values must contain S_k. □

It is observed in [CCF02] that a two-pass algorithm could identify the heavy hitters exactly. However, with a single pass, no better method has been known for Zipfian distributions than for arbitrary distributions; in fact, the lower bound [DIPW10] on linear sparse recovery uses a geometric (and hence sub-Zipfian) distribution.

As discussed in [CCF02], using Count-Sketch[7] with $O(\frac{k}{\epsilon^2}\log n)$ rows gets a k-sparse x' with

$$\|x' - x\|_2 \leq (1+\epsilon)\text{Err}_2(x,k) = \Theta(\frac{|x_{r_1}|}{\sqrt{2\alpha-1}}k^{1/2-\alpha}).$$

where, as in Section 1,

$$\text{Err}_2(x,k) = \min_{k\text{-sparse }\hat{x}}\|\hat{x} - x\|_2.$$

The set query algorithm lets us improve from a $1+\epsilon$ approximation to a $1+o(1)$ approximation. This is not useful for approximating x, since increasing k is much more effective than decreasing ϵ. Instead, it is useful for finding k elements that are quite close to being the actual k heavy hitters of x.

Naïve application of the set query algorithm to the output set of Lemma 4.1 would only get a close $O(k)$-sparse vector, not a k-sparse vector. To get a k-sparse vector, we must show a lemma that generalizes one used in the proof of sparse recovery of Count-Sketch (first in [CM06], but our description is more similar to [GI10]).

[7] Another analysis ([CM05]) uses Count-Min to achieve a better polynomial dependence on ϵ, but at the cost of using the ℓ_1 norm. Our result is an improvement over this as well.

Lemma 4.2. *Let $x, x' \in \mathbb{R}^n$. Let S and S' be the locations of the largest k elements (in magnitude) of x and x', respectively. Then if*

$$(*) \qquad \|(x' - x)_{S \cup S'}\|_2 \leq \epsilon \mathrm{Err}_2(x, k),$$

for $\epsilon \leq 1$, we have

$$\|x'_{S'} - x\|_2 \leq (1 + 3\epsilon)\mathrm{Err}_2(x, k).$$

Previous proofs have shown the following weaker form:

Corollary 4.1. *If we change the condition (*) to $\|x' - x\|_\infty \leq \frac{\epsilon}{\sqrt{2k}}\mathrm{Err}_2(x, k)$, the same result holds.*

The corollary is immediate from Lemma 4.2 and $\|(x' - x)_{S \cup S'}\|_2 \leq \sqrt{|S \cup S'|}\,\|(x' - x)_{S \cup S'}\|_\infty$.

Proof of Lemma 4.2. We have

(3)
$$\|x'_{S'} - x\|_2^2 = \|(x' - x)_{S'}\|_2^2 + \left\|x_{S \setminus S'}\right\|_2^2 + \left\|x_{[n] \setminus (S \cup S')}\right\|_2^2$$

The tricky bit is to bound the middle term $\left\|x_{S \setminus S'}\right\|_2^2$. We will show that it is not much larger than $\left\|x_{S' \setminus S}\right\|_2^2$.

Let $d = |S \setminus S'|$, and let a be the d-dimensional vector corresponding to the absolute values of the coefficients of x over $S \setminus S'$. That is, if $S \setminus S' = \{j_1, \ldots, j_d\}$, then $a_i = |x_{j_i}|$ for $i \in [d]$. Let a' be analogous for x' over $S \setminus S'$, and let b and b' be analogous for x and x' over $S' \setminus S$, respectively.

Let $E = \mathrm{Err}_2(x, k) = \|x - x_S\|_2$. We have

$$\begin{aligned}
\left\|x_{S \setminus S'}\right\|_2^2 - \left\|x_{S' \setminus S}\right\|_2^2 &= \|a\|_2^2 - \|b\|_2^2 \\
&= (a - b) \cdot (a + b) \\
&\leq \|a - b\|_2 \|a + b\|_2 \\
&\leq \|a - b\|_2 (2\|b\|_2 + \|a - b\|_2) \\
&\leq \|a - b\|_2 (2E + \|a - b\|_2)
\end{aligned}$$

So we should bound $\|a - b\|_2$. We know that $||p| - |q|| \leq |p - q|$ for all p and q, so

$$\begin{aligned}
\|a - a'\|_2^2 + \|b - b'\|_2^2 &\leq \left\|(x - x')_{S \setminus S'}\right\|_2^2 + \left\|(x - x')_{S' \setminus S}\right\|_2^2 \\
&\leq \|(x - x')_{S \cup S'}\|_2^2 \leq \epsilon^2 E^2.
\end{aligned}$$

We also know that $a - b$ and $b' - a'$ both contain all nonnegative coefficients. Hence

$$\begin{aligned}
\|a - b\|_2^2 &\leq \|a - b + b' - a'\|_2^2 \\
&\leq (\|a - a'\|_2 + \|b' - b\|_2)^2 \\
&\leq 2\|a - a'\|_2^2 + 2\|b - b'\|_2^2 \\
&\leq 2\epsilon^2 E^2 \\
\|a - b\|_2 &\leq \sqrt{2}\epsilon E.
\end{aligned}$$

Therefore

$$\begin{aligned}
\left\|x_{S \setminus S'}\right\|_2^2 - \left\|x_{S' \setminus S}\right\|_2^2 &\leq \sqrt{2}\epsilon E(2E + \sqrt{2}\epsilon E) \\
&\leq (2\sqrt{2} + 2)\epsilon E^2 \\
&\leq 5\epsilon E^2.
\end{aligned}$$

Plugging into Equation 3, and using $\|(x' - x)_{S'}\|_2^2 \leq \epsilon^2 E^2$,

$$\begin{aligned}
\|x'_{S'} - x\|_2^2 &\leq \epsilon^2 E^2 + 5\epsilon E^2 + \left\|x_{S' \setminus S}\right\|_2^2 + \left\|x_{[n] \setminus (S \cup S')}\right\|_2^2 \\
&\leq 6\epsilon E^2 + \left\|x_{[n] \setminus S}\right\|_2^2 \\
&= (1 + 6\epsilon)E^2 \\
\|x'_{S'} - x\|_2 &\leq (1 + 3\epsilon)E.
\end{aligned}$$

□

With this lemma in hand, on Zipfian distributions we can get a k-sparse x' with a $1+o(1)$ approximation factor.

Theorem 4.1. *Suppose x comes from a sub-Zipfian distribution with parameter $\alpha > 1/2$. Then we can recover a k-sparse x' from Ax with*

$$\|x' - x\|_2 \leq \frac{\epsilon}{\sqrt{\log n}}\mathrm{Err}_2(x, k).$$

with $O(\frac{c}{\epsilon^2} k \log n)$ rows and $O(n \log n)$ recovery time, with probability at least $1 - \frac{1}{k^c}$.

Proof. By Lemma 4.1 we can identify a set S of size $O(k)$ that contains all the heavy hitters. We then run the set query algorithm of Theorem 3.1 with $\frac{\epsilon}{3\sqrt{\log n}}$ substituted for ϵ. This gives an $\hat{x}$ with

$$\|\hat{x} - x_S\|_2 \leq \frac{\epsilon}{3\sqrt{\log n}}\mathrm{Err}_2(x, k).$$

Let x' contain the largest k coefficients of $\hat{x}$. By Lemma 4.2 we have

$$\|x' - x\|_2 \leq (1 + \frac{\epsilon}{\sqrt{\log n}})\mathrm{Err}_2(x, k).$$

□

4.2 Block-sparse vectors

In this section we consider the problem of finding *block-sparse* approximations. In this case, the coordinate set $\{1 \ldots n\}$ is partitioned into n/b blocks, each of length b. We define a (k, b)-block-sparse vector to be a vector where all non-zero elements are contained in at most k/b blocks. That is, we partition $\{1, \ldots, n\}$ into $T_i = \{(i - 1)b + 1, \ldots, ib\}$. A vector x is (k, b)-block-sparse if there exist $S_1, \ldots, S_{k/b} \in \{T_1, \ldots, T_{n/b}\}$ with $\mathrm{supp}(x) \subseteq \bigcup S_i$. Define

$$\mathrm{Err}_2(x, k, b) = \min_{(k,b)-\text{block-sparse } \hat{x}} \|x - \hat{x}\|_2.$$

Finding the support of block-sparse vectors is closely related to finding block heavy hitters, which is studied for the ℓ_1 norm in [ABI08]. The idea is to perform dimensionality reduction of each block into $\log n$ dimensions, then perform sparse recovery on the resulting $\frac{k \log n}{b}$-sparse vector. The differences from previous work are minor, so we relegate the details to Appendix C.

Lemma 4.3. *For any b and k, there exists a family of matrices A with $O(\frac{k}{\epsilon^5 b} \log n)$ rows and column sparsity $O(\frac{1}{\epsilon^2} \log n)$ such that we can recover a support S from Ax in $O(\frac{n}{\epsilon^2 b} \log n)$ time with*

$$\|x - x_S\|_2 \leq (1+\epsilon)\mathrm{Err}_2(x, k, b)$$

with probability at least $1 - n^{-\Omega(1)}$.

Once we know a good support S, we can run Algorithm 3.1 to estimate x_S:

Theorem 4.2. *For any b and k, there exists a family of binary matrices A with $O(\frac{1}{\epsilon^2}k + \frac{k}{\epsilon^5 b} \log n)$ rows such that we can recover a (k, b)-block-sparse x' in $O(k + \frac{n}{\epsilon^2 b} \log n)$ time with*

$$\|x' - x\|_2 \leq (1+\epsilon)\mathrm{Err}_2(x, k, b)$$

with probability at least $1 - \frac{1}{k^{\Omega(1)}}$.

Proof. Let S be the result of Lemma 4.3 with approximation $\epsilon/3$, so

$$\|x - x_S\|_2 \leq (1 + \frac{\epsilon}{3})\mathrm{Err}_2(x, k, b).$$

Then the set query algorithm on x and S uses $O(k/\epsilon^2)$ rows to return an x' with

$$\|x' - x_S\|_2 \leq \frac{\epsilon}{3} \|x - x_S\|_2 .$$

Therefore

$$\begin{aligned}
\|x' - x\|_2 &\leq \|x' - x_S\|_2 + \|x - x_S\|_2 \\
&\leq (1 + \frac{\epsilon}{3}) \|x - x_S\|_2 \\
&\leq (1 + \frac{\epsilon}{3})^2 \mathrm{Err}_2(x, k, b) \\
&\leq (1+\epsilon)\mathrm{Err}_2(x, k, b)
\end{aligned}$$

as desired. □

If the block size b is at least $\log n$ and ϵ is constant, this gives an optimal bound of $O(k)$ rows.

5 Conclusion and Future Work

We show efficient recovery of vectors conforming to Zipfian or block sparse models, but leave open extending this to other models. Our framework decomposes the task into first locating the heavy hitters and then estimating them, and our set query algorithm is an efficient general solution for estimating the heavy hitters once found. The remaining task is to efficiently locate heavy hitters in other models.

Our analysis assumes that the columns of A are fully independent. It would be valuable to reduce the independence needed, and hence the space required to store A.

We show k-sparse recovery of Zipfian distributions with $1 + o(1)$ approximation in $O(k \log n)$ space. Can the $o(1)$ be made smaller, or a lower bound shown, for this problem?

Acknowledgments

I would like to thank my advisor Piotr Indyk for much helpful advice, Anna Gilbert for some preliminary discussions, and Joseph O'Rourke for pointing me to [KŁ02].

References

[ABI08] A. Andoni, K. Do Ba, and P. Indyk. Block heavy hitters. *MIT Technical Report TR-2008-024*, 2008.

[BCDH10] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde. Model-based compressive sensing. *IEEE Transactions on Information Theory*, 56, No. 4:1982–2001, 2010.

[BKM+00] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. *Comput. Netw.*, 33(1-6):309–320, 2000.

[CCF02] M. Charikar, K. Chen, and M. Farach-Colton. Finding frequent items in data streams. *ICALP*, 2002.

[CD04] Z. Chen and J. J. Dongarra. Condition numbers of gaussian random matrices. *SIAM Journal on Matrix Analysis and Applications*, 27:603–620, 2004.

[Cev08] V. Cevher. Learning with compressible priors. In *NIPS*, Vancouver, B.C., Canada, 7–12 December 2008.

[CIHB09] V. Cevher, P. Indyk, C. Hegde, and R. G. Baraniuk. Recovery of clustered sparse signals from compressive measurements. *SAMPTA*, 2009.

[CM04] G. Cormode and S. Muthukrishnan. Improved data stream summaries: The count-min sketch and its applications. *Latin*, 2004.

[CM05] Graham Cormode and S. Muthukrishnan. Summarizing and mining skewed data streams. In *SDM*, 2005.

[CM06] G. Cormode and S. Muthukrishnan. Combinatorial algorithms for compressed sensing. *Sirocco*, 2006.

[COMS07] A. Coja-Oghlan, C. Moore, and V. Sanwalani. Counting connected graphs and hypergraphs via the probabilistic method. *Random Struct. Algorithms*, 31(3):288–329, 2007.

[CRT06] E. J. Candès, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Comm. Pure Appl. Math.*, 59(8):1208–1223, 2006.

[CT06] E.J. Candès and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *Information Theory, IEEE Transactions on*, 52(12):5406 – 5425, dec. 2006.

[DDT+08] M. Duarte, M. Davenport, D. Takhar, J. Laska, T. Sun, K. Kelly, and R. Baraniuk. Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine*, 2008.

[DIPW10] K. Do Ba, P. Indyk, E. Price, and D. Woodruff. Lower bounds for sparse recovery. *SODA*, 2010.

[Don06] D. L. Donoho. Compressed Sensing. *IEEE Trans. Info. Theory*, 52(4):1289–1306, Apr. 2006.

[DPR96] D. Dubhashi, V. Priebe, and D. Ranjan. Negative dependence through the FKG inequality. In *Research Report MPI-I-96-1-020, Max-Planck-Institut fur Informatik, Saarbrucken*, 1996.

[DR96] D. Dubhashi and D. Ranjan. Balls and bins: A study in negative dependence. *Random Structures & Algorithms*, 13:99–124, 1996.

[EB09] Y.C. Eldar and H. Bolcskei. Block-sparsity: Coherence and efficient recovery. *IEEE Int. Conf. Acoustics, Speech and Signal Processing*, 2009.

[EG07] D. Eppstein and M. T. Goodrich. Space-efficient straggler identification in round-trip data streams via Newton's identitities and invertible Bloom filters. *WADS*, 2007.

[EKB09] Y. C. Eldar, P. Kuppinger, and H. Bölcskei. Compressed sensing of block-sparse signals: Uncertainty relations and efficient recovery. *CoRR*, abs/0906.3173, 2009.

[FPRU10] S. Foucart, A. Pajor, H. Rauhut, and T. Ullrich. The Gelfand widths of lp-balls for $0 < p \leq 1$. *preprint*, 2010.

[GI10] A. Gilbert and P. Indyk. Sparse recovery using sparse matrices. *Proceedings of IEEE*, 2010.

[GLPS09] A. C. Gilbert, Y. Li, E. Porat, and M. J. Strauss. Approximate sparse recovery: Optimizing time and measurements. *CoRR*, abs/0912.0229, 2009.

[Ind07] P. Indyk. Sketching, streaming and sublinear-space algorithms. *Graduate course notes, available at* `http://stellar.mit.edu/S/course/6/fa07/6.895/`, 2007.

[JP83] K. Joag-Dev and F. Proschan. Negative association of random variables with applications. *The Annals of Statistics*, 11(1):286–295, 1983.

[KŁ02] M. Karoński and T. Łuczak. The phase transition in a random hypergraph. *J. Comput. Appl. Math.*, 142(1):125–135, 2002.

[LD05] C. La and M. N. Do. Signal reconstruction using sparse tree representation. In *in Proc. Wavelets XI at SPIE Optics and Photonics*, 2005.

[Mit04] M. Mitzenmacher. A brief history of generative models for power law and lognormal distributions. *Internet Mathematics*, 1:226–251, 2004.

[Mut03] S. Muthukrishnan. Data streams: Algorithms and applications (invited talk at SODA'03). *Available at* `http://athos.rutgers.edu/£\sim£muthu/stream-1-1.ps`, 2003.

[Rom09] J. Romberg. Compressive sampling by random convolution. *SIAM Journal on Imaging Science*, 2009.

[SPH09] M. Stojnic, F. Parvaresh, and B. Hassibi. On the reconstruction of block-sparse signals with an optimal number of measurements. *IEEE Trans. Signal Processing*, 2009.

A Negative Dependence

Negative dependence is a fairly common property in balls-and-bins types of problems, and can often cleanly be analyzed using the framework of *negative association* ([DR96, DPR96, JP83]).

Definition 1 (Negative Association)**.** *Let $(X_1, \ldots, X_n)$ be a vector of random variables. Then $(X_1, \ldots, X_n)$ are* negatively associated *if for every two disjoint index sets, $I, J \subseteq [n]$,*

$$\begin{aligned}&\mathrm{E}[f(X_i, i \in I) g(X_j, j \in J)]\\ \leq &\mathrm{E}[f(X_i, i \in I)] E[g(X_j, j \in J)]\end{aligned}$$

for all functions $f\colon \mathbb{R}^{|I|} \to \mathbb{R}$ and $g\colon \mathbb{R}^{|J|} \to \mathbb{R}$ that are both non-decreasing or both non-increasing.

If random variables are negatively associated then one can apply most standard concentration of measure arguments, such as Chebyshev's inequality and the Chernoff bound. This means it is a fairly strong property, which makes it hard to prove directly. What makes it so useful is that it remains true under two composition rules:

Lemma A.1 ([DR96], Proposition 7)**.**

1. *If $(X_1, \ldots, X_n)$ and $(Y_1, \ldots, Y_m)$ are each negatively associated and mutually independent, then $(X_1, \ldots, X_n, Y_1, \ldots, Y_m)$ is negatively associated.*
2. *Suppose $(X_1, \ldots, X_n)$ is negatively associated. Let $I_1, \ldots, I_k \subseteq [n]$ be disjoint index sets, for some positive integer k. For $j \in [k]$, let $h_j\colon \mathbb{R}^{|I_j|} \to \mathbb{R}$ be functions that are all non-decreasing or all non-increasing, and define $Y_j = h_j(X_i, i \in I_j)$. Then $(Y_1, \ldots, Y_k)$ is also negatively associated.*

Lemma A.1 allows us to relatively easily show that one component of our error (the point error) is negatively associated without performing any computation. Unfortunately, the other component of our error (the component size) is not easily built up by repeated applications of Lemma A.1[8]. Therefore we show something much weaker for this error, namely *approximate negative correlation*:

$$\mathrm{E}[X_i X_j] - \mathrm{E}[X_i] E[X_j] \leq \frac{1}{k^{\Omega(1)}} \mathrm{E}[X_i] \mathrm{E}[X_j]$$

for all $i \neq j$. This is still strong enough to use Chebyshev's inequality.

[8]This paper considers the component size of each hyperedge, which clearly is not negatively associated: if one hyperedge is in a component of size k than so is every other hyperedge. But one can consider variants that just consider the distribution of component sizes, which seems plausibly negatively associated. However, this is hard to prove.

B Set Query in the ℓ_1 norm

This section works through all the changes to prove the set query algorithm works in the ℓ_1 norm with $w = O(\frac{1}{\epsilon}k)$ measurements.

We use Lemma 3.5 to get an ℓ_1 analog of Corollary 3.2:

$$\begin{aligned}(4)\qquad \|x' - x_S\|_1 &= \sum_{i \in S} |(x' - x_S)_i| \\ &\leq \sum_{i \in S} 2 \sum_{j \in S} C_{i,j} |Y_j| = 2 \sum_{i \in S} D_i |Y_i| .\end{aligned}$$

Then we bound the expectation, variance, and covariance of D_i and $|Y_i|$. The bound on D_i works the same as in Section 3.6: $\mathrm{E}[D_i] = O(1)$, $\mathrm{E}[D_i^2] = O(1)$, $\mathrm{E}[D_i D_j] - \mathrm{E}[D_i]^2 \leq O(\log^4 k/\sqrt{k})$.

The bound on $|Y_i|$ is slightly different. We define

$$U'_q = |\nu_q| + \sum_{i \in [n] \setminus S} |x_i| B_{i,q}$$

and observe that $U'_q \geq |V_q|$, and U'_q is NA. Hence

$$Z'_i = \operatorname*{median}_{q \in L_i} U'_q$$

is NA, and $|Y_i| \leq Z'_i$. Define

$$\begin{aligned}\mu = \mathrm{E}[U'_q] &= \frac{d}{w} \|x - x_S\|_1 + \frac{1}{w} \|\nu\|_1 \\ &\leq \frac{\epsilon}{k} (\|x - x_S\|_1 + \|\nu\|_1)\end{aligned}$$

then

$$\Pr[Z'_i \geq c\mu] \leq 2^{|L_i|} (\frac{1}{c})^{|L_i|/2} \leq \left(\frac{4}{c}\right)^{d-2}$$

so $\mathrm{E}[Z'_i] = O(\mu)$ and $\mathrm{E}[Z'^2_i] = O(\mu^2)$.

Now we will show the analog of Section 3.7. We know

$$\|x' - x_S\|_2 \leq 2 \sum_i D_i Z'_i$$

and

$$\mathrm{E}[2 \sum_i D_i Z'_i] = 2 \sum_i \mathrm{E}[D_i] \mathrm{E}[Z'_i] = k\mu'$$

for some $\mu' = O(\frac{\epsilon}{k}(\|x - x_S\|_1 + \|\nu\|_1))$. Then

$$\begin{aligned}\mathrm{E}[(\sum D_i Z'_i)^2] &= \sum_i \mathrm{E}[D_i^2] \mathrm{E}[Z'^2_i] + \sum_{i \neq j} \mathrm{E}[D_i D_j] \mathrm{E}[Z'_i Z'_j] \\ &\leq \sum_i O(\mu'^2) + \sum_{i \neq j} (\mathrm{E}[D_i]^2 + O(\log^4 k/\sqrt{k})) \mathrm{E}[Z'_i]^2 \\ &= O(\mu'^2 k \sqrt{k} \log^4 k) + k(k-1) \mathrm{E}[D_i Z'_i]^2\end{aligned}$$

$$\mathrm{Var}(2 \sum_i Z'_i D_i) \leq O(\mu'^2 k \sqrt{k} \log^4 k).$$

By Chebyshev's inequality, we get

$$\Pr[\|x' - x_S\|_1 \geq (1+\alpha)k\mu'] \leq O(\frac{\log^4 k}{\alpha^2\sqrt{k}})$$

and the main theorem (for constant $c = 1/3$) follows. The parallel repetition method of Section 3.7 works the same as in the ℓ_2 case to support arbitrary c.

C Block Heavy Hitters

Lemma 4.3. *For any b and k, there exists a family of matrices A with $O(\frac{k}{\epsilon^5 b}\log n)$ rows and column sparsity $O(\frac{1}{\epsilon^2}\log n)$ such that we can recover a support S from Ax in $O(\frac{n}{\epsilon^2 b}\log n)$ time with*

$$\|x - x_S\|_2 \leq (1+\epsilon)\mathrm{Err}_2(x,k,b)$$

with probability at least $1 - n^{-\Omega(1)}$.

Proof. This proof follows the method of [ABI08], but applies to the ℓ_2 norm and is in the (slightly stronger) sparse recovery framework rather than the heavy hitters framework. The idea is to perform dimensionality reduction, then use an argument similar to those for Count-Sketch (first in [CM06], but we follow more closely the description in [GI10]).

Define $s = k/b$ and $t = n/b$, and decompose $[n]$ into equal sized blocks $T_1, \ldots, T_t$. Let $x_{(T_i)} \in \mathbb{R}^b$ denote the restriction of x_{T_i} to the coordinates T_i. Let $U \subseteq [t]$ have $|U| = s$ and contain the s largest blocks in x, so $\mathrm{Err}_2(x,k,b) = \left\|\sum_{i\notin U} x_{T_i}\right\|_2$.

Choose an i.i.d. standard Gaussian matrix $\rho \in \mathbb{R}^{m\times b}$ for $m = O(\frac{1}{\epsilon^2}\log n)$. Define $y_{q,i} = (\rho x_{(T_q)})_i$, so as a distribution over ρ, $y_{q,i}$ is a Gaussian with variance $\left\|x_{(T_q)}\right\|_2^2$.

Let $h_1, \ldots, h_m \colon [t] \to [l]$ be pairwise independent hash functions for some $l = O(\frac{1}{\epsilon^3}s)$, and $g_1, \ldots, g_m \colon [t] \to \{-1, 1\}$ also be pairwise independent. Then we make m hash tables $H^{(1)}, \ldots, H^{(m)}$ of size l each, and say that the value of the jth cell in the ith hash table $H^{(i)}$ is given by

$$H^{(i)}_j = \sum_{q:h_i(q)=j} g_i(q) y_{q,i}$$

Then the $H^{(i)}_j$ form a linear sketch of $ml = O(\frac{k}{\epsilon^5 b}\log n)$ cells. We use this sketch to estimate the mass of each block, and output the blocks that we estimate to have the highest mass. Our estimator for $\|x_{T_i}\|_2$ is

$$z'_i = \alpha \operatorname*{median}_{j\in[m]} \left|H^{(j)}_{h_j(i)}\right|$$

for some constant scaling factor $\alpha \approx 1.48$. Since we only care which blocks have the largest magnitude, we don't actually need to use α.

We first claim that for each i and j with probability $1 - O(\epsilon)$, $(H^{(j)}_{h_j(i)} - y_{i,j})^2 \leq O(\frac{\epsilon^2}{s}(\mathrm{Err}_2(x,k,b))^2)$. To prove it, note that the probability any $q \in U$ with $q \neq i$ having $h_j(q) = h_j(i)$ is at most $\frac{s}{l} \leq \epsilon^3$. If such a collision with a heavy hitter does not happen, then

$$\begin{aligned}
\mathrm{E}[(H^{(j)}_{h_j(i)} - y_{i,j})^2] &= \mathrm{E}[\sum_{p\neq i, h_j(p)=h_j(i)} y^2_{p,j}] \\
&\leq \sum_{p\notin U} \frac{1}{l}\mathrm{E}[y^2_{p,j}] \\
&= \frac{1}{l}\sum_{p\notin U} \left\|x_{T_p}\right\|_2^2 \\
&= \frac{1}{l}(\mathrm{Err}_2(x,k,b))^2
\end{aligned}$$

By Markov's inequality and the union bound, we have

$$\Pr[(H^{(j)}_{h_j(i)} - y_{i,j})^2 \geq \frac{\epsilon^2}{s}(\mathrm{Err}_2(x,k,b))^2] \leq \epsilon + \epsilon^3 = O(\epsilon)$$

Let $B_{i,j}$ be the event that $(H^{(j)}_{h_j(i)} - y_{i,j})^2 > O(\frac{\epsilon^2}{s}(\mathrm{Err}_2(x,k,b))^2)$, so $\Pr[B_{i,j}] = O(\epsilon)$. This is independent for each j, so by the Chernoff bound $\sum_{j=1}^m B_{i,j} \leq O(\epsilon m)$ with high probability in n.

Now, $|y_{i,j}|$ is distributed according to the positive half of a Gaussian, so there is some constant $\alpha \approx 1.48$ such that $\alpha\,|y_{i,j}|$ is an unbiased estimator for $\|x_{T_i}\|_2$. For any $C \geq 1$ and some $\delta = O(C\epsilon)$, we expect less than $\frac{1-C\epsilon}{2}m$ of the $\alpha\,|y_{i,j}|$ to be below $(1-\delta)\,\|x_{T_i}\|_2$, less than $\frac{1-C\epsilon}{2}m$ to be above $(1+\delta)\,\|x_{T_i}\|_2$, and more than $C\epsilon m$ to be in between. Because $m \geq \Omega(\frac{1}{\epsilon^2}\log n)$, the Chernoff bound shows that with high probability the actual number of $\alpha\,|y_{i,j}|$ in each interval is within $\frac{\epsilon}{2}m = O(\frac{1}{\epsilon}\log n)$ of its expectation. Hence

$$\left|\|x_{T_i}\|_2 - \alpha \operatorname*{median}_{j\in[m]} |y_{i,j}|\right| \leq \delta\,\|x_{T_i}\|_2 = O(C\epsilon)\,\|x_{T_i}\|_2 .$$

even if $\frac{(C-1)\epsilon}{2}m$ of the $y_{i,j}$ were adversarially modified. We can think of the events $B_{i,j}$ as being such adversarial modifications. We find that

$$\begin{aligned}
|\|x_{T_i}\|_2 - z_i| &= \left|\|x_{T_i}\|_2 - \alpha \operatorname*{median}_{j\in[m]} \left|H^{(j)}_{h_j(i)}\right|\right| \\
&\leq O(\epsilon)\,\|x_{T_i}\|_2 + O(\frac{\epsilon}{\sqrt{s}}\mathrm{Err}_2(x,k,b)).
\end{aligned}$$

$$(\|x_{T_i}\|_2 - z_i)^2 \leq O(\epsilon^2\,\|x_{T_i}\|_2^2 + \frac{\epsilon^2}{s}(\mathrm{Err}_2(x,k,b))^2)$$

Define $w_i = \|x_{T_i}\|_2$, $\mu = \mathrm{Err}_2(x,k,b)$, and $\hat{U} \subseteq [t]$ to contain the s largest coordinates in z. Since z is computed from the sketch, the recovery algorithm can compute $\hat{U}$. The output of our algorithm will be the blocks corresponding to $\hat{U}$.

We know $\mu^2 = \sum_{i \notin U} w_i^2 = \left\|w_{[t]\setminus U}\right\|_2^2$ and $|w_i - z_i| \leq O(\epsilon w_i + \frac{\epsilon}{\sqrt{s}}\mu)$ for all i. We will show that

$$\left\|w_{[t]\setminus \hat{U}}\right\|_2^2 \leq (1 + O(\epsilon))\mu^2.$$

This is analogous to the proof of Count-Sketch, or to Corollary 4.1. Note that

$$\left\|w_{[t]\setminus \hat{U}}\right\|_2^2 = \left\|w_{U\setminus \hat{U}}\right\|_2^2 + \left\|w_{[t]\setminus (U\cup \hat{U})}\right\|_2^2$$

For any $i \in U \setminus \hat{U}$ and $j \in \hat{U} \setminus U$, we have $z_j > z_i$, so

$$w_i - w_j \leq O(\frac{\epsilon}{\sqrt{s}}\mu + \epsilon w_i)$$

Let $a = \max_{i \in U\setminus \hat{U}} w_i$ and $b = \min_{j \in \hat{U}\setminus U} w_j$. Then $a \leq b + O(\frac{\epsilon}{\sqrt{s}}\mu + \epsilon a)$, and dividing by $(1 - O(\epsilon))$ we get $a \leq b(1 + O(\epsilon)) + O(\frac{\epsilon}{\sqrt{s}}\mu)$. Furthermore $\left\|w_{\hat{U}\setminus U}\right\|_2^2 \geq b^2 \left|\hat{U} \setminus U\right|$, so

$$\begin{aligned}
\left\|w_{U\setminus \hat{U}}\right\|_2^2 &\leq \left(\left\|w_{\hat{U}\setminus U}\right\|_2 \frac{1 + O(\epsilon)}{\sqrt{\left|\hat{U} \setminus U\right|}} + O(\frac{\epsilon}{\sqrt{s}}\mu)\right)^2 \left|\hat{U} \setminus U\right| \\
&\leq \left(\left\|w_{\hat{U}\setminus U}\right\|_2 (1 + O(\epsilon)) + O(\epsilon\mu)\right)^2 \\
&= \left\|w_{\hat{U}\setminus U}\right\|_2^2 (1 + O(\epsilon)) + (2 + O(\epsilon)) \left\|w_{\hat{U}\setminus U}\right\|_2 O(\epsilon\mu) \\
&\quad + O(\epsilon^2\mu^2) \\
&\leq \left\|w_{\hat{U}\setminus U}\right\|_2^2 + O(\epsilon\mu^2)
\end{aligned}$$

because $\left\|w_{\hat{U}\setminus U}\right\|_2 \leq \mu$. Thus

$$\begin{aligned}
\left\|w - w_{\hat{U}}\right\|_2 = \left\|w_{[t]\setminus \hat{U}}\right\|_2^2 &\leq O(\epsilon\mu^2) + \left\|w_{\hat{U}\setminus U}\right\|_2^2 + \left\|w_{[t]\setminus (U\cup \hat{U})}\right\|_2^2 \\
&= O(\epsilon\mu^2) + \mu^2 = (1 + O(\epsilon))\mu^2.
\end{aligned}$$

This is exactly what we want. If $S = \bigcup_{i \in \hat{U}} T_i$ contains the blocks corresponding to $\hat{U}$, then

$$\|x - x_S\|_2 = \left\|w - w_{\hat{U}}\right\|_2 \leq (1+O(\epsilon))\mu = (1+O(\epsilon))\mathrm{Err}_2(x, k, b)$$

Rescale ϵ to change $1+O(\epsilon)$ into $1+\epsilon$ and we're done. $\square$