The Dissertation Committee for Akshay Devdas Kamath
certifies that this is the approved version of the following dissertation:

Lower Bounds for Sparse Recovery Problems

Committee:

Eric Price, Supervisor

Anna Gál

Greg Plaxton

David Woodruff

Lower Bounds for Sparse Recovery Problems

by

Akshay Devdas Kamath

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

August 2020

In memory of my brother, Nikhil D Kamath.

Acknowledgments

This thesis is a collection of the results that I produced over six years at the University of Texas at Austin. I would like to thank everyone who made the experience enjoyable.

I express gratitude to my advisor Eric Price for his guidance and support over the course of my PhD. He encouraged me to remain persistent when doing research and was patient when I was slow to make progress. It has been a great pleasure to have worked with him over the years.

I would like to thank my coauthors Sushrut Karmalkar and David Woodruff with whom I had fruitful collaborations which are featured in this thesis.

I am grateful to the students in the CS theory group who were good sounding boards and even better friends.

Finally, I thank my parents for their love and unconditional support.

Lower Bounds for Sparse Recovery Problems

Publication No. _____

Akshay Devdas Kamath, Ph.D.
The University of Texas at Austin, 2020

Supervisor: Eric Price

Sparse recovery or compressed sensing is the problem of estimating a signal from noisy linear measurements of that signal. Sparse recovery has traditionally been used in areas like image acquisition, streaming algorithms, genetic testing, and, more recently, for image recovery tasks.

Over the last decade many techniques have been developed for sparse recovery under various guarantees. We develop new lower bound techniques and show the tightness of existing results for the following variants of the sparse recovery problem:

- **Adaptive Sparse Recovery:** We present a lower bound and an upper bound for a constrained version of the adaptive sparse recovery problem where the algorithm is allowed a constant number of adaptive rounds.
- **Sparse Recovery under High SNR:** We present algorithms for sparse recovery when the signal is very close to being sparse. Our results show that existing lower bounds are tight.

- **Deterministic ℓ_2 Heavy Hitters:** We prove a new and simple lower bound on the space complexity for the heavy hitters problem in the insertion-only streaming model. Our bounds match the best known upper bound up to a logarithmic factor.

- **Compressed Sensing with Generative Models:** We prove tight lower bounds on compressed sensing algorithms that use "generative models" as a form of structure instead of sparsity.

Table of Contents

List of Tables

Chapter 1

Introduction

Compressed sensing is a class of problems where the goal is to estimate a "structured" vector from a low-dimensional linear sketch of the vector. Most literature in compressed sensing focuses on approximate sparsity as a notion of structure. This problem of recovering an approximately sparse vector from a low-dimensional sketch is called *sparse recovery*.

This has a variety of practical applications in fields such as image acquisition [DDT+08], genetic testing [ECG+09], streaming algorithms [CM06] and image reconstruction [BJPD17]. In the streaming model, compressed sensing techniques may be used to solve problems like the frequent elements problem or the *heavy hitters* problem [CCF02, GGI+02].

For image reconstruction tasks, a new form of structure known as a *generative model* is used and has achieved great practical results along with theoretical guarantees on performance [BJPD17].

In this thesis, we focus on proving lower bounds which establish the hardness of certain tasks in compressed sensing. We also prove some upper bounds with the goal of proving the tightness of known lower bounds.

1.1 Overview of Sparse Recovery

The problem of compressed sensing involves observing a linear sketch $Ax \in \mathbb{R}^m$ of a vector $x \in \mathbb{R}^n$ where the matrix A is called the *measurement matrix*. The goal is to robustly recover x while minimizing the total number of measurements m.

1.1.1 Sparse Recovery Guarantees

We say that an algorithm performs (k, C)-approximate ℓ_2/ℓ_2-sparse recovery if it recovers a vector x^* such that

$$\|x - x^*\|_2^2 \leq C \min_{k\text{-sparse } x'} \|x - x'\|_2^2 . \tag{1.1}$$

While we could also consider recovery in other norms such as ℓ_1 [CM04, CRT06a], ℓ_2 is the strongest ℓ_p-norm for which efficient sparse recovery is possible [CCF02, BJKS04].

Remark 1.1. *Due to the fact that ℓ_2/ℓ_2-sparse recovery is more studied than any other guarantee, we sometimes refer to it plainly as "sparse recovery". When we refer to sparse recovery under other norms (say p and q) in this thesis we will explicitly refer to it as ℓ_p/ℓ_q-sparse recovery.*

A somewhat stronger guarantee than (1.1), is the (k, C)-approximate ℓ_∞/ℓ_2-sparse recovery (also known as *heavy hitters*) guarantee where the goal is to accurately recover *every* coordinate of x i.e. we wish to recover x^* such

that:

$$\|x - x^*\|_\infty^2 \leq \frac{C^2}{k} \min_{k\text{-sparse } x'} \|x - x'\|_2^2 . \tag{1.2}$$

1.1.2 Adaptivity in Sparse Recovery

In some applications of sparse recovery, the goal of reducing the number of measurements far outweighs other considerations. Consider the case of genetic testing where the goal is to determine the k members of a population of size n who are susceptible to a particular genetic disease. Instead of testing n different blood samples, we could mix together blood samples in different ratios (which is a linear operation) and use compressed sensing techniques to identify the people who carry the recessive gene. Since the main goal here is to minimize the number of tests, we could attempt to reduce the total number of measurements by using adaptivity.

In adaptive sparse recovery, the algorithm is allowed to choose a particular row A_i of the measurement matrix A after observing the measurements $\langle A_j, x \rangle$ corresponding to the previous rows $j < i$.

1.1.3 Sparse Recovery under High SNR

While sparse recovery has been studied extensively over the last decade, most work has focused on algorithms and lower bounds when $C = (1+\epsilon)$ where $\epsilon \in (0, 1)$. A natural question that arises is: what happens when $C \gg 1$?

For large C bounds to be meaningful (in that $x^* = 0$ is not a valid

answer), x must be in a "high SNR" regime where the sparse "signal" is C times larger than the dense "noise".

1.1.4 Streaming Algorithms for Heavy Hitters

Sparse recovery can be applied to solve problems in streaming algorithms. Specifically, for the heavy hitters or frequent elements problem (where we wish to find the most frequently occurring elements in a stream), we can use a sparse recovery matrix to maintain a sketch of the frequency vector of elements in a stream and recover the frequent elements using the sparse recovery algorithm.

1.2 Adaptive Sparse Recovery with Limited Adaptivity

The most common goal in sparse recovery is to achieve (1.1) for $C = O(1)$ with 90% probability over the choice of matrix $A \in \mathbb{R}^{m \times n}$, with as few "measurements" m as possible. If A is chosen independently of x, it is known that $m = \Theta(k \log n)$ is necessary [DIPW10] and sufficient [CRT06a, GLPS10]. However, this sample complexity can be improved if A is chosen *adaptively*.

In adaptive sparse recovery, the algorithm picks $A_1 \in \mathbb{R}^{m_1 \times n}$, observes $A_1 x$, then picks $A_2 \in \mathbb{R}^{m_2 \times n}$ and observes $A_2 x$, and continues until $A_R x$ for some number of rounds R. The goal is still to minimize the total number of measurements $m = \sum_i m_i$. With $O(\log \log n)$ rounds of adaptivity, it is possible to achieve (1.1) with $m = O(k \log \log n)$ [IPW11, NSWZ18]. On the other hand, we know that $\Omega(k + \log \log n)$ measurements are necessary with

unlimited adaptivity [ACD13, PW13].

We consider sparse recovery with a small constant number of rounds of adaptivity. For example, what is possible with $R = 2$? This is an important question for applications, where adaptivity is typically costly. The number of rounds of adaptivity corresponds to the number of passes of a streaming algorithm, or the number of rounds of MapReduce; thus the overall communication (which is usually the speed bottleneck in such applications) is proportional to R. In other applications such as imaging or genetic testing, parallelism and latency in setting up the measurements can make it difficult to perform many rounds of adaptivity.

For $k = 1$ and $R = O(1)$, it is known that $m = \Theta(\log^{1/R} n)$ is necessary and sufficient [IPW11, PW13]. Thus one expects that the answer for $k \gg 1$ should probably be $k \log^{1/R} n$. However, the best prior algorithm (a variant of [NSWZ18]) uses three "extra" rounds, giving only $O(k \log^{1/(R-3)} n)$. This does not benefit from anything less than five rounds of adaptivity. On the lower bound side, existing work shows that $m = \Omega(k+\log^{1/R} n)$ [ACD13, PW13], but cannot connect k and n. For $C = 1 + \epsilon$, one can get an algorithm separating the dependence on n and ϵ [NSWZ18]; perhaps the same could hold for n and k?

We show upper and lower bounds that almost entirely address the problem. First, we show that $\Omega(k \log^{1/R} n)$ samples are necessary, for any k with $k < 2^{\log^{1/R} n}$. This settles the sample complexity for smallish k; for larger k, up to $n^{o(1)}$, we can still show that $\omega(k)$ samples are necessary. Second, we give

an algorithm that uses $O(k \log^{1/R} n)$ samples for any sparsity parameter k.

1.3 Sparse Recovery under High SNR

Information-theoretic arguments show that for sparse recovery when $C \gg 1$ in (1.1) $\Omega(k \log_C(n/k))$ measurements are necessary [PW11, PW12]. As mentioned earlier, for a bound to be meaningful the input vector X must be in the high-SNR regime. For such high-SNR x, we can hope to learn $\log C$ bits per measurement; is this actually achievable?

We show that the answer is yes, and in fact ℓ_2/ℓ_2 recovery is possible with the optimal $O(k \log_C(n/k))$ linear measurements.

We also show that the stronger ℓ_∞/ℓ_2 guarantee can be achieved with $O(k \log_C n)$ measurements. The best known algorithm prior to our algorithm was Count-Sketch [CCF02] which achieves the same guarantee when $C \gg 1$ by using $O(k \log n)$ linear measurements.

While these results are not in line with the stated goal of the thesis i.e. proving lower bounds for sparse recovery problems, they establish the tightness of existing lower bounds and close the problem in the ℓ_2/ℓ_2 case.

1.4 Deterministic L2 Heavy Hitters in the Insertion-Only Model

Heavy hitters or frequent elements is a fundamental problem in streaming algorithms. In this problem, we wish to parse a sequence of items $a_1, \ldots, a_m$

from a set $\mathcal{U} = [n]$ and identify the frequently occurring elements (or heavy hitters).

Suppose f_i is the number of occurrences of $i \in \mathcal{U}$ in the stream, all elements h such that:

$$|f_h|^2 \geq \epsilon^2 \sum_{j \in \mathcal{U}} f_j^2$$

are known as ϵ-ℓ_2-heavy hitters in the stream. In streams where insertions and deletions are allowed, ℓ_∞/ℓ_2-sparse recovery algorithms may be used to identify *all heavy hitters* in a stream by storing a linear sketch of the frequency vector f. It is known that any algorithm that utilizes a linear sketch to solve the deterministic ℓ_2-heavy hitters problem must use an $\Omega(n)$ dimensional sketch[CDD09].

When we restrict ourselves to insertion-only streams, the algorithm of Misra and Gries [MG82] can deterministically identify all ϵ-ℓ_2-heavy hitters using $O(\frac{\sqrt{n}}{\epsilon} \log m)$ measurements. This algorithm does not store a linear sketch and hence does not need to store $\Omega(n)$ bits.

We prove a lower bound of $\Omega(\frac{\sqrt{n}}{\epsilon})$ on the space complexity of any algorithm that identifies the ϵ-ℓ_2-heavy hitters in an insertion-only stream. This matches the upper bound of [MG82] up to a $\log m$ factor.

Our lower bound uses a reduction from multi-party communication complexity problem called Mostly Set Disjointness which we define in Chapter 4. We prove a communication complexity lower bound using a simple inductive argument and describe a reduction from this problem to ℓ_2-heavy hitters to

obtain a space complexity lower bound for streaming algorithms.

1.5 Compressed Sensing with Generative Models

When performing compressed sensing, sparsity is chosen as a form of structure because it is a commonly occurring form of structure.

In recent years, deep convolutional neural networks have had great success in producing rich models for representing the manifold of images, notably with generative adversarial networks (GANs) [GPAM$^+$14] and variational autoencoders (VAEs) [KW14]. These methods produce generative models $G : \mathbb{R}^k \to \mathbb{R}^n$ that allow approximate sampling from the distribution of images. So a natural question is whether these generative models can be used for compressed sensing.

In [BJPD17] it was shown how to use generative models to achieve a guarantee analogous to (1.1): for any L-Lipschitz $G : \mathbb{R}^k \to \mathbb{R}^n$, one can achieve

$$\|x - \hat{x}\|_2 \leq C \min_{z' \in B_k^2(r)} \|x - G(z')\|_2 + \delta, \tag{1.3}$$

where $r, \delta > 0$ are parameters, $B_k^2(r)$ denotes the radius-r ℓ_2 ball in $\mathbb{R}^k$ and Lipschitzness is defined with respect to the ℓ_2-norms, using only $m = O(k + k \log \frac{Lr}{\delta})$ measurements.

Thus, the recovered vector is almost as good as the nearest point in the *range of the generative model*, rather than in the set of k-sparse vectors. We

will refer to the problem of achieving the guarantee (1.3) as "generative-model recovery".

We prove two theorems that further our understanding of this new notion of structure and establish a connection between sparse recovery and generative model recovery. Our first theorem shows that the [BJPD17] result is tight: for any setting of parameters n, k, L, r, δ, there exists an L-Lipschitz function $G : \mathbb{R}^k \to \mathbb{R}^n$ such that the [BJPD17] measurements bound is necessary in order to achieving (1.3).

Our second result directly relates the two notions of structure: sparsity and generative models. We produce a simple ReLU-based neural network whose image is precisely the set of all k-sparse vectors.

1.6 Organization

This thesis is divided into 4 different chapters each corresponding to a different paper. Each chapter is written such that it is self contained but we have ensured that notation is consistent across chapters.

Chapter 2 covers the results of [KP19] where we proved improved lower bounds and an almost tight upper bound for adaptive sparse recovery under limited adaptivity. Our upper bound in that paper used sparse recovery under high SNR as a black-box algorithm. In Chapter 3, we present the results of [KP20] where we give a tight upper bound for sparse recovery under high SNR and thereby close the gap in [KP19]. In Chapters 4 and 5, we return

to our overarching goal of proving lower bounds. In Chapter 4 we study the deterministic ℓ_2-heavy hitters problem which is the streaming analog of the sparse recovery problem. We present a new (and almost tight) lower bound from [KPW20] where we studied this problem in the insertion-only model.

Chapter 5 presents two results from [KKP20]. The first establishes an information theoretic lower bound on the measurement complexity for compressed sensing with generative models. Our second result shows that compressed sensing with generative models is a generalization of sparse recovery.

Chapter 2

Adaptive Sparse Recovery with Limited Adaptivity

In this chapter we present a lower bound and a matching upper bound for adaptive sparse recovery with $O(1)$ rounds of adaptivity[1].

Recall that in Chapter 1 (1.1), we said that an algorithm achieves the (k, C)-approximate sparse recovery guarantee for a vector $x \in \mathbb{R}^n$ if it recovers a vector $x^* \in \mathbb{R}^n$ such that

$$\|x - x^*\|_2^2 \leq C \min_{k\text{-sparse } x'} \|x - x'\|_2^2 . \tag{2.1}$$

An algorithm performs (k, C)-sparse recovery with R adaptive rounds of linear measurements if it picks $A_1 \in \mathbb{R}^{m_1 \times n}$, observes $A_1 x$, then picks $A_2 \in \mathbb{R}^{m_2 \times n}$ and observes $A_2 x$, and continues until $A_R x$ and then uses these observations to recover the vector x^*.

It is known from previous results [IPW11, PW13] on adaptive sparse recovery that for $k = 1$ and $R = O(1)$, the number of measurements that is necessary and sufficient is $\Theta(\log^{\frac{1}{R}} n)$. Simple attempts at extending the lower bound to apply for arbitrary k yield a lower bound of $\Omega(k + \log^{\frac{1}{R}} n)$. The

[1]The results presented in this chapter appeared in [KP19].

	Paper	Measurements	Rounds	Comment
Upper	[IPW11]	$\frac{k}{\epsilon} r \log^{\frac{1}{r}} n$	$O(r \log^* k)$	
	[NSWZ18]	$\frac{1}{\epsilon} kr \log^{\frac{1}{r}} \frac{1}{\epsilon} + kr \log^{\frac{1}{r}} n$	$O(r \log^* k)$	
		$\frac{k}{\epsilon} r \log^{\frac{1}{r}} n$	$r+3$	
	Corollary 2.11	$k \log^{\frac{1}{r}} n \cdot 5^r \log^* k$	r	$\epsilon = O(1)$
Lower	[PW13]	$r \log^{1/r} n$	r	
	[ACD13]	k/ϵ	r	
	Corollary 2.5	$\frac{1}{r} \cdot k \log^{\frac{1}{r}} n$	r	$\log k < \log^{\frac{1}{r}} n$
	Theorem 2.4	$\omega(k)$	r	$k = n^{o(1)}, r = O(1)$

Table 2.1: Results for adaptive $(k, 1+\epsilon)$-sparse recovery. The measurements column drops constant factors. The upper bounds above are not explicit in previous papers, which only state the bounds for $r = O(\log \log n)$. However, all previous algorithms reduce to 1-sparse recovery as a black box, and plugging in r-round $O(r \log^{1/r} n)$-sample 1-sparse recovery gives the above.

result of [IPW11] also gives a $O(k \log^{\frac{1}{R}} n)$ upper bound on the measurement complexity with $O(r \log^* k)$ rounds of adaptivity. In this chapter, we attempt to bridge this gap between the upper bound and the lower bound for arbitrary k.

2.1 Prior Work on Limited Adaptivity

The adaptive measurement model has been explored in many papers, starting with empirical results [MSW08, JXC08, CHNR08] and theoretical results for $k = 1$ [CHNR08]. Results from the compressed sensing side of the literature have focused on signal approximation accuracy, which corresponds to the behavior for $C = 1 + \epsilon$ as $\epsilon \to 0$. With Gaussian noise, nonadaptive algorithms take $m = O(\frac{1}{\epsilon} k \log n)$, while [HCN11, HBCN12] improve this

to $O\left(k \log n + \frac{1}{\epsilon} k(\log k + \log\log\log n)\right)$; a corresponding $\Omega(k/\epsilon)$ lower bound appeared in [ACD13]. On the sparse recovery side of the literature, [IPW11] gave a fully adaptive algorithm using $O(\frac{1}{\epsilon} k \log\log n)$ measurements performed in $R = O(\log\log n \log^* k)$ rounds. This was improved by [NSWZ18] in two incomparable ways: either R can be improved to $O(\log\log n)$ or the sample complexity can be improved to $O(\frac{\log\log\frac{1}{\epsilon}}{\epsilon} k + k \log\log n)$, splitting n and ϵ in the sample complexity.

The algorithms in [IPW11] and [NSWZ18] can easily be adapted to use fewer rounds of adaptivity. Each algorithm's round complexity is dominated by black-box applications of the $O(\log\log n)$-round $O(\log\log n)$-sample $O(1)$-approximate 1-sparse recovery algorithm of [IPW11]. By changing this to an r-round $O(\log^{1/r} n)$-sample version, the algorithms can be performed with fewer rounds; see Figure 2.1. Most relevantly, one of the algorithms in [NSWZ18] would use $O(k \log^{1/r} n)$ samples in $r + 3$ rounds. It seems likely that a more careful analysis could reduce this to $r+2$ rounds, but no further: the approach requires an initial round to find the important subproblem instances, and a final round to clean up missing elements.

2.2 Our Results and an Overview of Techniques

We give a simple explanation of the techniques used in our lower bound and upper bound in this section. In both cases, the reader will benefit from knowledge of the previous results. We provide a simple explanation of previous results from which we borrow techniques or derive inspiration.

2.2.1 Overview of Our Lower Bound

Prior Work ($k = 1$). We begin by giving an overview of the lower bound for $k = 1$ from [PW13]. The lower bound instance consists of the signal $e_X + w$, where $X \in [n]$ is a uniform random index and $w \sim \mathcal{N}(0, I_n/n)$ is Gaussian. This signal is such that successful 1.1-approximate 1-sparse recovery must return a vector that is close to e_X, and in particular reveals the identity of X. Hence

$$I(X; Y_1, \ldots, Y_R) = \Omega(\log n).$$

On the other hand, [PW13] shows that after learning b bits about X, each measurement in the next round reveals only $O(b+1)$ bits. That is, for any set of observations $y_1, \ldots, y_{r-1}$ seen so far, if we define

$$b = H(X) - H(X \mid Y_1 = y_1, \ldots, Y_{r-1} = y_{r-1}) \tag{2.2}$$

to be the information revealed so far about X, then it can be shown that the next round has

$$I(X; Y_r \mid Y_1 = y_1, \ldots, Y_{r-1} = y_{r-1}) = m_r \cdot O(b+1) \tag{2.3}$$

where m_r is the number of measurements in round r.

It follows that $I(X; Y_1, \ldots, Y_R) \leq C^R \prod_{i=1}^{R} m_r$. Then, an application of AM-GM shows $(O(m/R))^R = \Omega(\log n)$, or $m = \Omega(R \log^{1/R} n)$. Thus the key step is to show (2.3).

The intuition for why (2.3) should hold is as follows. For any single

measurement vector v of unit norm, the corresponding observation is

$$y = \langle v, e_X + w \rangle = v_X + w'$$

where $w' \sim N(0, 1/n)$. This is an additive white gaussian noise channel, so the Shannon-Hartley Theorem (Theorem A.1) may be applied here to bound the information capacity in terms of the signal-to-noise ratio:

$$I(X; y) \leq \frac{1}{2} \log(1 + n\,\mathbb{E}[v_X^2]).$$

This holds even conditioned on $Y_1 = y_1, \ldots, Y_{r-1} = y_{r-1}$, so we want to bound $\mathbb{E}[v_X^2 \mid Y_1 = y_1, \ldots, Y_{r-1} = y_{r-1}]$. Let $p : [n] \to \mathbb{R}$ denote the probability distribution of $(X \mid Y_1 = y_1, \ldots, Y_{r-1} = y_{r-1})$, so $b = \log n - H(p)$. If p happens to be uniform over its support, then its value is $2^b/n$ at $n/2^b$ locations; then any unit norm v has

$$n \mathop{\mathbb{E}}_{X \sim p}[v_X^2] \leq n \cdot \sum_{i=1}^{n} \frac{2^b}{n} v_i^2 = 2^b$$

or $I(X; y \mid Y_1 = y_1, \ldots, Y_{r-1} = y_{r-1}) \leq \frac{1}{2} \log(1 + 2^b) \lesssim (b + 1)$.

However, p is not necessarily uniform over its support, which necessitates care. For example, consider if $p(1) = 1/\log n$ and p is uniform otherwise. Then $b = O(1)$, yet by setting $v = e_1$ we have

$$n \mathop{\mathbb{E}}_{X \sim p}[v_X^2] = n/\log n$$

so Shannon-Hartley would only show $O(\log n)$ bits per measurement. The problem is that Shannon-Hartley is only a good bound if the signal – in this

case v_X – is at a consistent scale. The fix is to partition the indices of X by the scale of $p(X)$; we define $T_j = \{i \mid np(i) \in [2^j, 2^{j+1})\}$ for $j > 0$, and T_0 to have the rest. Let J be the random variable denoting the j such that $X \in T_j$. We can decompose (with implicit conditioning on $y_1, \ldots, y_{r-1}$)

$$I(X; y) \le I(X; (y, J)) = I(X; y \mid J) + I(X; J). \tag{2.4}$$

Then $I(X; J) \le H(J) \lesssim b + 1$ by simple algebra, and since $(X \mid J)$ is roughly uniform over its support the Shannon-Hartley bound can give $I(X; y \mid J) \lesssim b+1$. This bounds the information content in any single measurement; summing over all m_r measurements in Y_r yields (2.3).

We now describe how to adapt these techniques to prove a result for $k > 1$.

Problem instance for general k. We use the natural extension of the problem instance, which is to concatenate k copies of the hard instance; that is, for $N = nk$, we draw $X_1, \ldots, X_k \in [n]$, and set the vector to

$$x = \left(\sum_{i=1}^{k} e_{(i-1)n + X_i} \right) + w$$

where $w = N(0, \frac{k}{N} I_N)$. Then successful 1.1-approximate sparse recovery must recover most coordinates X_i, so

$$I(X_1, \ldots, X_k; Y_1, \ldots, Y_R) = \Omega(k \log n).$$

Defining the per-round goal. The first difficulty is how best to define the goal (2.3). While (2.3) is true as stated, this is not enough: it would give a lower bound of $(k \log n)^{1/R}$ not $k \log^{1/R} n$. Yet (2.3) is also tight; given b bits of information about the first coordinate, a single measurement *can* learn $\Omega(b)$ bits about that coordinate.

However, with b bits of information overall, most coordinates will only have $O(b/k)$ of information "about them." Each such coordinate can only be observed with signal-to-noise ratio $2^{O(b/k)}$. Thus we can hope to say that there exists a large set of coordinates, $W \subset [k]$ of size $|W| > 0.99k$, such that

$$I(\{X_i\}_{i \in W}; Y_r \mid Y_1 = y_1, \ldots, Y_{r-1} = y_{r-1}) = m_r \cdot O(\frac{b}{k} + 1).$$

Unfortunately, this is false. Suppose we have learned the parity of $X_i \oplus X_1$ for all i; this is only $b = k-1$ bits of information. Then the measurement vector v which matches all the parities will have signal-to-noise-ratio k; with a variation on this example[2], the information learned in a single measurement can be $\Omega(\log k)$ bits for every large W even though $b = k$. Thus, the replacement for (2.3) that we can show is

$$I(\{X_i\}_{i \in W}; Y_r \mid Y_1 = y_1, \ldots, Y_{r-1} = y_{r-1}) = m_r \cdot O(\frac{b}{k} + \log k) + O(b + k). \tag{2.5}$$

The extra $O(b + k)$ term comes from a term analogous to $I(X; J)$ in (2.4).

[2]Partition $[k]$ into $\log k$ pieces, and the prior information reveals the relative parities within each partition.

Implications for sample complexity. In the first round we can replace the bound (2.5) by the straightforward nonadaptive bound

$$I(\{X_i\}_{i\in[k]}; Y_1) \leq O(m_1).$$

Now, for simplicity of exposition suppose each $m_i = m/R = \Theta(m)$. If $m > k \log k$, then after the first round the dominant term in (2.5) will be $O(b \cdot \frac{m}{k})$. Hence chaining (2.5) gives a set W_R such that

$$k \log n \lesssim I(\{X_i\}_{i\in W_R}; Y_1, \ldots, Y_R) \leq m \cdot \left(O(\frac{m}{k})\right)^{R-1} = k \left(O(\frac{m}{k})\right)^{R}.$$

Thus $m = \Omega(k \log^{1/R} n)$, as long as this is more than $k \log k$.

Analog of J for general k. The proof of (2.5) is analogous to that of (2.3), where we partition the X by "scale", and bound the mutual information conditioned on the scale by Shannon-Hartley. However, the new partition is subtle so we describe it here.

The first coordinate X_1 is partitioned the same way as its marginal would be in the $k = 1$ case: the set T_{j_1} has $\{i \in [n] \mid np(X_1 = i) \in [2^{j_1}, 2^{j_1+1})\}$ for $j_1 > 0$, T_0 has everything else, and J_1 denotes the $j_1 \geq 0$ with $X_1 \in T_{j_1}$. The second coordinate is partitioned as its marginal *conditioned on* J_1. That is, we have sets

$$T_{j_1,j_2} = \{i \in [n] \mid np(X_2 = i \mid X_1 \in T_{j_1}) \in [2^{j_2}, 2^{j_2+1})\}$$

and the random variable J_2 is such that $x_2 \in T_{J_1,J_2}$. This naturally extends to $x_i \in T_{J_1,\ldots,J_i}$.

We show that this partitioning $J = (J_1, \ldots, J_k)$ of the coordinates $X_1, \ldots, X_k$ has the following properties: First,$H(J) = O(b)$ so conditioning on J does not reveal too much information. Second, the "signal power" $Z_{i,J}$ that any measurement has about X_i conditioned on J obeys

$$\mathop{\mathbb{E}}_{i \in [k]} \mathop{\mathbb{E}}_{J} \log(1 + Z_{i,J}) \lesssim \frac{b}{k}. \tag{2.6}$$

Since the Shannon-Hartley theorem implies

$$I(X_1, \ldots, X_k; Y_r \mid J, Y_1 = y_1, \ldots, Y_{r-1} = y_{r-1}) \lesssim m_r \cdot \mathop{\mathbb{E}}_{J} \log(1 + \sum_{i=1}^{k} Z_{i,J})$$

one would get—if (2.6) held for all i not just on average—that

$$\begin{aligned} I(X_1, \ldots, X_k; Y_r \mid J, Y_1 = y_1, \ldots, Y_{r-1} = y_{r-1}) &\lesssim m_r \cdot \log(1 + k2^{b/k}) \\ &\approx m_r(\frac{b}{k} + \log k) \end{aligned}$$

as desired. Using Markov's inequality to choose for each J a large set W of i where (2.6) is not too far off, we can get (2.5) and complete the proof.

2.2.2 Overview of Our Upper Bound

Prior work for $k = 1$. The high-level intuition for our algorithm is based on the intuition for $k = 1$ from the upper bound in [IPW11] and corresponding lower bound in [PW13]. Suppose the vector x has one large coordinate i^*, of value 1. For $O(1)$-approximate sparse recovery to be nontrivial, the amount of "noise" in other coordinates, i.e. $\left\|x_{[n]\setminus\{i\}}\right\|_2^2$, will be at most a small constant.

At any given round, if we have learned b bits of information in the previous round, we can expect to have located i^* to within a set S of size

$n/2^b$. Then our measurement matrix in this round can place zero mass on any coordinate outside S. Effectively, in this round we are trying to find i^* within x_S. This vector still has "signal" 1, but the "noise" $\left\|x_{S\setminus\{i\}}\right\|_2^2$ is likely to be much smaller: if S is random, the noise will be $O(1/2^b)$ on average. With such a high signal-to-noise ratio (SNR), we can hope to learn $\Theta(\log \mathrm{SNR}) = \Theta(b)$ bits per measurement. This will quickly reduce the size of our candidate set S, further enriching the SNR of X_S and increasing the information per measurement.

Given r rounds with t measurements each, we expect to learn t bits in the first round; $\Theta(t^2)$ bits in the second round; $\Theta(t^3)$ bits in the third round; and so on till $\Theta(t^R)$ bits in the Rth round. Setting $t = \log^{1/R} n$, we can learn the desired $\log n$ bits of information in $O(R \log^{1/R} n)$ measurements.

Algorithm for general k. Previous adaptive algorithms which achieve $m = o(k \log n)$ use the $k = 1$ algorithm as a black box [IPW11, NSWZ18]. Unfortunately, such efforts seem to require additional rounds of adaptivity to set up the subproblem instances and/or to clean up coordinates missed in the first pass. Our algorithm avoids this by opening up the $k = 1$ algorithm and extending its techniques to general k.

Our goal is to maintain a candidate set $S \subseteq [n]$ of locations that include the largest k elements of x, known as the "heavy hitters". In each round except for the last, we would like to take a number of measurements that are insufficient to identify the heavy hitters of x_S exactly, but that are sufficient

to find a small subset S' of S that contains (almost) all of the heavy hitters. If S' is also fairly random, then $x_{S'}$ will have almost all the signal while only a small fraction of the noise, so it has much higher SNR.

A first attempt for finding such a subset S' could be as follows. Suppose that the SNR is C—that is, the largest k elements of x_S have C times more ℓ_2^2 mass than the other elements. For some parameter $D \gg k$, we construct a vector $y \in \mathbb{R}^D$ by hashing x_S as per Count-Sketch [CCF02]—so each coordinate $i \in S$ is assigned a random coordinate $h(i) \in [D]$ and sign $s_i \in \{\pm 1\}$, and $y_j = \sum_{i:h(i)=j} x_i s_i$. The SNR of y will also be about C, so we can learn a lot about y by performing nonadaptive $C^{0.1}$-approximate sparse recovery of it. This takes $O(k \log_C(D/k) \cdot \log^* k)$ measurements [PW12], so we can set $D = kC^{\log^{1/R} n}$ and fit within our sample complexity budget. The top $O(k)$ elements of y will contain most of the heavy hitters of x, so we can set S' to the pre-image of those elements; this has size about $k(|S|/D) = |S|/C^{\log^{1/R} n}$. Hence the C used in the next round will be roughly a $C^{\log^{1/R} n}$ factor larger; after R rounds of this, C will grow from constant to n^{10}, at which point the problem is easy. In fact, the Rth round can estimate x_U directly to avoid needing an extra cleanup round.

This approach *mostly* works, but suffers from one major flaw: in every stage, the set S' can miss a small fraction of the heavy hitters. Even with zero noise, heavy hitters that collide in $[D]$ can cancel out when combining into y, causing them to disappear from S' and from the final reconstruction. Previous algorithms based on the Count-Sketch hashing often run into this

issue, and address it by cleaning up the residual afterward [GLPS10, IPW11, LNW18, NSWZ18]. In our context, such a solution would require more rounds of adaptivity.

Triple gaussian hashing. We introduce a new approach to hashing for sparse recovery that lets us avoid any major false negatives, based on replacing the signs s_i with gaussians $g_i \sim N(0,1)$ in the computation of y, so $y_j = \sum_{i:h(i)=j} x_i g_i$. This hash avoids the issue described above with zero noise, since if $x_i \neq 0$ then $y_{h(i)} \neq 0$ with probability 1.

To understand how this hash behaves with noise, consider the following example: $x = v + w$ where $v \in \{0,1\}^n$ is k-sparse and w is gaussian with norm 1. Successful $O(1)$-approximate recovery of x must find all but $O(1)$ elements in $\text{supp}(v)$. In the gaussian hash y of x, the image of w is still very spread out with norm about 1, but the image of v is no longer binary: each entry $\left|y_{h(i)}\right|$ has a $\Theta(\epsilon)$ chance of being less than ϵ. This means about $k^{2/3}$ positions in $h(\text{supp}(v))$ will be smaller than $1/k^{1/3}$. Since these collectively have norm 1, successful $O(1)$-approximate recovery of y could miss all $k^{2/3}$ of these positions, which would be a problem.

We avoid such false negatives by repeating the hash three times, with the same h and different g, and applying sparse recovery separately to the three different y. In the above example, where coordinates are missing from sparse recovery with probability $1/k^{1/3}$, the expected number of coordinates that are missed three times in a row is $O(1)$. In general, the chance q_i that

$h(i)$ is recovered by the sparse recovery algorithm may depend on i and x_i in a complicated fashion that we can't control, since the sparse recovery algorithm is a black box. Still, we can show that the (k, C)-approximate recovery guarantee implies that the expected mass lost all three times—$\sum_i q_i^3 x_i^2$—is bounded in terms of the noise level.

Our triple Gaussian hash thus gives a set of locations without any significant false negatives, so we do not need to clean up the missing coordinates. We believe that this technique is likely to have applications in other, nonadaptive, sparse recovery settings.

Decreasing the noise. So far, we have outlined how the algorithm gets a small set S' that does not lose much signal mass. Another key part of the argument is that $x_{S'}$ should have much less noise than x_S. Since S' is much smaller than S, this would be immediate if S' were random. However, since S' is the pre-image of the largest coordinates of y, it is biased towards the elements of x containing more noise.

We show that this effect is limited: after dropping $O(k)$ noise coordinates, the rest of the noise shrinks by a factor of $\sqrt{D/k}$. We tolerate the $O(k)$ large noise coordinates by increasing the sparsity k by a constant factor in each round; and the $\sqrt{D/k}$ factor, although not as good as the D/k factor decrease in $|S|$, is still $C^{\Theta(\log^{1/R} k)}$.

By choosing the parameters carefully, we can ensure the total error and total failure probability remain small over all rounds. The measurement

complexity of our algorithm for constant R is $O(k \log^{1/R}(n/k))$. In our paper [KP19], where we first published this result, the measurement complexity was $O(k \log^{1/R}(n/k) \log^*(k))$. This is because our algorithm makes black-box calls to the C-approximate nonadaptive sparse recovery algorithm whose measurement complexity at that time was $O(k \log(n/k) \log^* k)$. We have since improved the measurement complexity of that algorithm and shaved off the extra $\log^* k$ factor. We present that result in Chapter 3.

2.3 Lower Bound for Adaptive Sparse Recovery with Limited Adaptivity

In this section we present a lower bound on the total number of linear measurements for adaptive R-round $(k, O(1))$-sparse recovery.

The instance for which we show a lower bound is as follows: Alice divides the domain $[N]$ into k contiguous "bins" of size n each (indexed by $[k]$) and for every bin i chooses $x_i \in [n]$ uniformly at random. Alice then chooses i.i.d. Gaussian noise $w \in \mathbb{R}^N$ with $\mathbb{E}[\|w\|_2^2] = \sigma^2 = \Theta(k)$, then sets $x = w + \sum_{i=1}^{k} e_{(i-1)n+x_i}$. Bob performs R adaptive rounds of linear measurements on x, getting $y^r = A^r x = (y_1^r, \ldots, y_{m_r}^r)$ in each round r. Let X_i and Y^r denote the random variables from which x_i and y^r are drawn, respectively. In order for sparse recovery to succeed under an appropriate setting of constant for σ^2, at least $k/2$ of the variables $X_1, \ldots, X_k$ must be recovered.

For ease of notation, we use j_1^r to denote the tuple $(j_1, \ldots, j_r)$. Similarly, j_1^{i-1}, J_i denotes the tuple $(j_1, \ldots, j_{i-1}, J_i)$ where the distinction in the context

of this proof is that $j_1, \ldots, j_{i-1}$ are fixed and J_i is a random variable. We use $(X)_W$ for $W = \{i_1, \ldots, i_{|W|}\} \subseteq [k]$ to denote the tuple $(X_{i_1}, \ldots, X_{i_W})$.

Definition 2.3.1. Given random variables $X_1, \ldots, X_k \in [n]$ with joint probability distribution $p(l_1, \ldots l_k) = Pr[X_1 = l_1, \ldots, X_k = l_k]$, we define the **sequentially conditioned partition** of the domain of X_i as follows

1. $T_{j_1^i} = \{l \in [n] \mid 2^{j_i} < np_i(l \mid X_1 \in T_{j_1^1}, \ldots, X_{i-1} \in T_{j_1^{i-1}}) \leq 2^{j_i+1}\}$ for $j_i > 0$, and

2. $T_{j_1^i} = \{l \in [n] \mid np_i(l \mid X_1 \in T_{j_1^1}, \ldots, X_{i-1} \in T_{j_1^{i-1}}) \leq 2\}$ for $j_i = 0$.

where p_i denotes the marginal distribution of X_i. Additionally, we define the probability mass within each partition as

$$q_{j_1^i} = \sum_{l \in T_{j_1^i}} p_i(l \mid X_1 \in T_{j_1^1}, \ldots, X_{i-1} \in T_{j_1^{i-1}}).$$

So, if we fix $j_1, \ldots, j_{i-1}$, we have $\sum_{j_i=0}^{\infty} q_{j_1^i} = 1$.

Denote the event $X_1 \in T_{j_1^1}, \ldots, X_i \in T_{j_1^i}$ by $E_{j_1^i}$. These partitions are defined in such a way that $(X_i \mid E_{j_1^i})$ is close to uniform over its support. This allows us to bound the maximum conditional probability within a sequentially conditioned partition of the domain of X_i. So,

$$M_{j_1^i} \stackrel{\text{def}}{=} n \cdot \max_{l \in T_{j_1^i}} \Big(p_i(l \mid E_{j_1^i}) \Big) \leq \frac{2^{j_i+1}}{q_{j_1^i}}. \tag{2.7}$$

Additionally, for the random variable $(X_i \mid E_{j_1^{i-1}})$ over $[n]$, we define the number of bits that the distribution knows about the location of X_i as:

$$\begin{aligned} b_i(j_1, \ldots, j_{i-1}) &= H(\mathcal{U}([n])) - H(X_i \mid E_{j_1^{i-1}}) \\ &= \log(n) - H(X_i \mid E_{j_1^{i-1}}). \end{aligned}$$

We show for every i and j_1^{i-1} that $M_{j_1^{i-1},J_i}$ is small on average over J_i:

Lemma 2.1. *Consider random variables $X_1, \ldots, X_k \in [n]$ with joint probability distribution $p(l_1, \ldots l_k) = \Pr[X_1 = l_1, \ldots, X_k = l_k]$ and suppose we know that $X_1 \in T_{j_1}, \ldots X_{i-1} \in T_{j_1^{i-1}}$. Suppose that J_i is the discrete random variable that denotes the j_i such that $X_i \in T_{j_1^i}$ conditioned on $X_1 \in T_{j_1^1}, \ldots X_{i-1} \in T_{j_1^{i-1}}$. Then,*

$$\mathop{\mathbb{E}}_{J_i}[\log\left(1 + M_{j_1^{i-1},J_i}\right)] \leq O(b_i(j_1, \ldots, j_{i-1}) + 1).$$

Proof. Using (2.7) we get the bound:

$$\begin{aligned} \mathop{\mathbb{E}}_{J_i}[\log\left(1 + M_{j_1^{i-1},J_i}\right)] &\leq \mathop{\mathbb{E}}_{J_i}\left[\log\left(1 + \frac{2^{J_i+1}}{q_{j_1^{i-1},J_i}}\right)\right] \\ &= \sum_{j_i=0}^{\infty} q_{j_1^i} \log\left(1 + \frac{2^{j_i+1}}{q_{j_1^i}}\right) \\ &\leq \sum_{j_i=0}^{\infty} q_{j_1^i} \log(1 + 2^{j_i+1}) + \sum_{j=0}^{\infty} q_{j_1^i} \log\left(1 + \frac{1}{q_{j_1^i}}\right) \\ &\leq \sum_{j_i=0}^{\infty} j_i q_{j_1^i} + \sum_{j_i=0}^{\infty} 2q_{j_1^i} + \sum_{j=0}^{\infty} q_{j_1^i} \log\left(1 + \frac{1}{q_{j_1^i}}\right). \end{aligned}$$

Since $\sum_{j_i=0}^{\infty} q_{j_1^i} = 1$, Lemma A.2 implies:

$$\mathop{\mathbb{E}}_{J_i}[\log\left(1 + M_{i,j_1^{i-1},J_i}\right)] \leq O(b_i(j_1, \ldots, j_{i-1}) + 1).$$

□

For every i and collection of measurement vectors $v_1, \ldots, v_m$, we now show that the amount of "signal energy" for X_i is bounded even conditioned on the partition J_1^k.

Lemma 2.2. *Let $X_1, \ldots, X_k$ be random variables over $[n]$ with joint probability distribution $p(l_1, \ldots l_k) = Pr[X_1 = l_1, \ldots, X_k = l_k]$. For all $i \in [k]$, define $b_i = \log(n) - H(X_i \mid X_1, \ldots, X_{i-1})$. Let $v_1, \ldots, v_m \in \mathbb{R}^{nk}$ be a fixed set of vectors. Define random variable $Z_{i,j_1^k} \stackrel{\text{def}}{=} \mathbb{E}_{X_i | E_{j_1^k}}[\sum_{s=1}^m (v_s)^2_{n\cdot(i-1)+X_i}]$ and random variable $M_{i,j_1^k} \stackrel{\text{def}}{=} n \cdot \max_{l \in T_{j_1^i}}(p_i(l \mid E_{j_1^k}))$. Then, for any $i \in [k]$,*

1. $\log(1 + Z_{i,J_1^k}) \leq \log\left(1 + (\frac{\sum_{s=1}^m \left\|v_{s|i}\right\|_2^2}{n})\right) + \log(1 + M_{i,J_1^k})$
2. $\mathbb{E}_{J_1,\ldots,J_k}\left[\log(1 + M_{i,J_1^k})\right] \leq O(b_i + 1)$

where $v_{s|i}$ denotes the restriction of v_s to the the index set $[n(i-1)+1, ni]$.

Proof. Using the definition of Z_{i,j_1^k} and M_{i,j_1^k}, we can write:

$$Z_{j_1^i} = \sum_{s=1}^m \sum_{t \in [n]} (v_s)^2_{n\cdot(i-1)+t} \cdot \Pr[X_i = t \mid E_{j_1^i}] \leq \Big(\frac{\sum_{s=1}^m \left\|v_{s|i}\right\|_2^2}{n}\Big) M_{j_1^i}.$$

Let J_i be the discrete random variable that denotes the j_i such that $X_i \in T_{j_1^i}$ conditioned on $E_{j_1^{i-1}}$. Then, using Lemma 2.1,

$$\mathop{\mathbb{E}}_{J_i}[\log(1 + M_{j_1^{i-1},J_i})] \leq O(b_i(j_1, \ldots, j_{i-1}) + 1).$$

We wish to bound $\mathbb{E}_{J_1\ldots,J_k}[\log(1+M_{i,J_1^k})]$. Using the concavity of log,

$$\mathop{\mathbb{E}}_{J_1\ldots,J_k}[\log(1+Z_{i,J_1^k})] \leq \mathop{\mathbb{E}}_{J_1\ldots,J_i}[\log\left(1+\mathop{\mathbb{E}}_{J_{i+1},\ldots,J_k}[M_{i,J_1^k}]\right)].$$

From the definitions of M_{i,J_1^k} and $M_{J_1^i}$, we know that:

$$\begin{aligned}
\mathop{\mathbb{E}}_{J_{i+1},\ldots,J_k}[M_{i,J_1^k}] &= \mathop{\mathbb{E}}_{J_{i+1},\ldots,J_k}\left[\mathop{\mathbb{E}}_{X_i|E_{J_1^k}}\left[n\cdot\max_{l\in T_{j_1^i}} p_i(l \mid E_{j_1^k})\right]\right] \\
&= \mathop{\mathbb{E}}_{X_i|E_{J_1^i}}\left[n\cdot\max_{l\in T_{J_1^i}} p_i(l \mid E_{J_1^i})\right] \\
&= M_{J_1^i}.
\end{aligned}$$

So,

$$\begin{aligned}
\mathop{\mathbb{E}}_{J_1\ldots,J_k}[\log(1+M_{i,J_1^k})] &\leq \mathop{\mathbb{E}}_{J_1\ldots,J_i}[\log(1+M_{J_1^i})] \\
&\leq O(\mathop{\mathbb{E}}_{J_1\ldots,J_{i-1}}[b_i(J_1,\ldots,J_{i-1})+1]).
\end{aligned}$$

Since conditioning decreases entropy, we also know:

$$\begin{aligned}
\mathop{\mathbb{E}}_{J_1,\ldots,J_{i-1}}[b_i(J_1,\ldots,J_{i-1})] &= H(\mathcal{U}([n])) - H(X_i \mid E_{J_1^{i-1}}) \\
&\leq H(\mathcal{U}([n])) - H(X_i \mid X_1\ldots X_{i-1}) \\
&= b_i
\end{aligned}$$

and hence,

$$\mathop{\mathbb{E}}_{J_1\ldots,J_k}[\log(1+M_{i,J_1^k})] \leq O(b_i+1).$$

□

We can now show the key lemma, that if b bits of information are known from the previous rounds, the next round will only reveal roughly $m(\frac{b}{k}+\log k)$ more bits of information.

Lemma 2.3. *Suppose $X_1, \ldots, X_k$ are random variables over $[n]$ and $W = \{l_1, l_2, \ldots, l_{|W|}\} \subseteq [k]$ be a subset such that $|W| = ck$ where $c \leq 1$ is a constant. We define the number of bits of information revealed about the subset W, conditioned on the variables $\{X\}_{[n]\setminus W}$ as*

$$b = |W| \log(n) - H((X)_W \mid (X)_{[n]\setminus W}).$$

Define $\tilde{X} = \sum_{i=1}^{k} e_{(i-1)\cdot n + X_i} + N(0, I_N \sigma^2/N)$ where $\sigma^2 = \Theta(k)$. Consider a fixed set of measurement vectors $v_1, \ldots, v_m \in \mathbb{R}^N$ independent of $X_1, \ldots, X_k$ with $\|v_j\|_2^2 = N$ for all $j \in [m]$, and define $Y_j = \langle v_j, \tilde{X} \rangle$. Then, for all $0 < \alpha < \gamma < 1$, with probability $1 - \gamma$, there exists a subset $W' \subseteq W$, $|W'| \geq (1 - \frac{\alpha}{\gamma})\,|W|$ such that

$$I((X)_{W'}; Y_1^m \mid (X)_{[n]\setminus W'}, W') \leq c_3 \frac{m}{\alpha} \frac{b}{k} + m \log(k) + \frac{c_4 m}{\alpha} + c_2(b + k)$$

for some constants c_2, c_3, c_4.

Proof. Since we wish to condition out the indices not in W, we may perform the analysis on a fixed set of values for $(X)_{[n]\setminus W}$ and then use the fact that $I(A; B|C) = \mathbb{E}_c[I(A; B \mid C = c)]$ to arrive at the theorem statement.

Suppose that for all $i \in [n] \setminus W$, $X_i = x_i$. Then, the number of bits of information known about $(X)_W$ may be denoted $\tilde{b} = b((x)_{[n]\setminus W}) = |W| \log(n) - H((X)_W \mid (x)_{[n]\setminus W})$. Now, we may construct sequentially conditioned partitions only on the domains of $(X)_W$ and in the order $l_1, l_2, \ldots, l_{|W|}$. We will denote by J_W the conditioning over the partitions of the $(X)_W$ in the chosen order.

Let $W' \subseteq W$ be a set of indices which we shall choose later. Consider the mutual information between a set of random variables $(X)_{W'}$ and the measurements conditioned on the variables not in W'. Using the chain rule of mutual information:

$$I((X)_{W'}; Y_1^m \mid (X)_{W\setminus W'}, (x)_{[n]\setminus W}, W') \\ \leq I((X)_{W'}; Y_1^m \mid E_{J_W}, (X)_{W\setminus W'}, (x)_{[n]\setminus W}, W') + H(J_W \mid (x_{[n]\setminus W})).$$

Using Lemma A.2, there exists a constant c_2 such that for all $i \in [|W|]$, $H(J_{l_i} \mid J_{l_1}^{l_{i-1}}, (x)_{[n]\setminus W}) \leq c_2(\log(n) - H(X_{l_1} \mid J_{l_1}^{l_{i-1}}, (x)_{[n]\setminus W}) + 1)$. Since conditioning only reduces entropy, we know that $H(J_{l_i} \mid J_{l_1}^{l_{i-1}}, (x)_{[n]\setminus W}) \leq c_2(\log(n) - H(X_{l_1} \mid X_{l_1}, \ldots, X_{l_{i-1}}, (x)_{[n]\setminus W}) + 1)$. So, $H(J_W \mid (x_{[n]\setminus W})) = \sum_{i\in[|W|]} H(J_{l_i} \mid J_{l_1}^{l_{i-1}}, (x)_{[n]\setminus W}) \leq c_2(\tilde{b} + k)$. Using the definition of conditional mutual information, and the fact that measurements are chosen independently,

$$I((X)_{W'}; Y_1^m \mid (X)_{W\setminus W'}, (x)_{[n]\setminus W}, W') \\ \leq \mathop{\mathbb{E}}_{(x)_{W\setminus W'}} \Big(\sum_{s=1}^{m} I((X)_{W'}; Y_s \mid E_{J_W}, (x)_{[n]\setminus W'}, W') \Big) + c_2(\tilde{b} + k).$$

Applying the Data Processing Inequality to the first term, we get:

$$I((X)_{W'}; Y_1^m \mid (X)_{W\setminus W'}, (x)_{[n]\setminus W}, W') \\ \leq \mathop{\mathbb{E}}_{(x)_{W\setminus W'}} \Big(\sum_{s=1}^{m} I\big(\sum_{i\in W'} (v_s)_{(i-1)n+X_i}; Y_s \mid E_{J_W}, (x)_{[n]\setminus W'}, W' \big) \Big) \\ + c_2(\tilde{b} + k).$$

Observe that $Y_s = \sum_{i\in W'} (v_s)_{(i-1)n+X_i} + \sum_{i\in[n]\setminus W'} (v_s)_{(i-1)n+x_i} + N(0, \sigma^2)$. Since

$(x)_{[n]\setminus W'}$ are conditioned out, we may subtract their contribution and we get:

$$I\big(\sum_{i\in W'}(v_s)_{(i-1)n+X_i}; Y_s \mid E_{J_W}, (x)_{[n]\setminus W'}, W'\big)$$
$$= I\big(\sum_{i\in W'}(v_s)_{(i-1)n+X_i}; \sum_{i\in W'}(v_s)_{(i-1)n+X_i} + \eta \mid E_{J_W}, (x)_{[n]\setminus W'}, W'\big)$$

where $\eta \sim N(0,\sigma^2)$ is additive white gaussian noise. We may now use the Shannon-Hartley Theorem (Theorem A.1) on this quantity to bound the mutual information in terms of a variance term:

$$I((X)_{W'}; Y_1^m \mid (X)_{W\setminus W'}, (x)_{[n]\setminus W}, W')$$
$$\leq \mathop{\mathbb{E}}_{(x)_{W\setminus W'}} \sum_{s=1}^{m} \mathop{\mathbb{E}}_{j_W} \big[\log\big(1 + \frac{\mathbb{E}_{(X)_{W'}|E_{j_W},(x)_{[n]\setminus W'}}(\sum_{i\in W'}[(v_s)_{(i-1)n+X_i}])^2}{\sigma^2}\big)\big]$$
$$+ c_2(\tilde{b}+k).$$

Using Cauchy-Schwartz, then applying Jensen's inequality, and then using the convexity of log and the definition of Z_{i,J_W}:

$$I((X)_{W'}; Y_1^m \mid (X)_{W\setminus W'}, (x)_{[n]\setminus W}, W')$$
$$\leq \mathop{\mathbb{E}}_{j_W}\Big(\sum_{s=1}^{m}\log\big(1+|W'|\frac{\sum_{i\in W'}\mathbb{E}_{X_i|E_{j_W}}[(v_s)^2_{(i-1)n+X_i}]}{\sigma^2}\big)\Big) + c_2(\tilde{b}+k)$$
$$\leq m\mathop{\mathbb{E}}_{j_W}\Big(\log\big(1+\frac{|W'|\sum_{i\in W'}\mathbb{E}_{X_i|E_{j_W}}[\sum_{s=1}^{m}(v_s)^2_{(i-1)n+X_i}]}{\sigma^2\cdot m}\big)\Big) + c_2(\tilde{b}+k)$$
$$= m\mathop{\mathbb{E}}_{j_W}\Big(\log\big(1+\frac{|W'|\sum_{i\in W'}Z_{i,j_W}}{\sigma^2\cdot m}\big)\Big) + c_2(\tilde{b}+k). \tag{2.8}$$

We need to set W' to be the set that contains indices in W with low values of Z_{i,j_W}. More precisely, for a fixed partition sequence j_w, we set

$$W' = \Big\{ i\in W \;\Big|\; \log(1+Z_{i,j_W}) < \log(1+\frac{\sum_{s=1}^{m}\left\|v_{s|i}\right\|_2^2}{n}) + (\frac{c_3}{\alpha})\cdot(\frac{\tilde{b}}{k}+1)\Big\}$$

where c_3 is a constant which will be set later. Suppose that $M_{i_l,j_W} = n \cdot \max_{l \in T_{j_1^i}}(\Pr[X_{i_l} = x_{i_l} \mid E_{j_W}])$. We may use Lemma 2.2 on the indices in W since the indices in $[n] \setminus W$ has been fixed. So, there is a constant c_1 such that for all $i_l \in W$,

$$\mathop{\mathbb{E}}_{J_{i_1},\ldots,J_{i_{|W|}}} \Big[\log\big(1 + M_{i_l,J_W}\big) \Big] \leq c_1(\tilde{b}_{i_l} + 1)$$

where $\tilde{b}_{i_l} = \log(n) - H(X_{i_l} \mid X_{i_1}, \ldots, X_{i_{l-1}}, (x)_{[n]\setminus W})$. Observe that $\sum \tilde{b}_{i_l} = |W| \log(n) - \sum H(X_{i_l} | X_{i_1}, \ldots, X_{i_{l-1}}, (x)_{[n]\setminus W}) = |W| \log(n) - H(X_{i_1}, \ldots, X_{i_{|W|}} \mid (x)_{[n]\setminus W}) = \tilde{b}$. Suppose I is distributed uniformly over W. Then using Jensen's inequality,

$$\begin{aligned}
\mathop{\mathbb{E}}_{I} \Big[\mathop{\mathbb{E}}_{J_W} \big[\log\big(1 + M_{I,J_W}\big) \big] \Big] &\leq c_1 \mathop{\mathbb{E}}_{I}[\tilde{b}_I + 1] \\
&\leq \frac{c_1(\tilde{b} + k)}{|W|} \\
&\leq \frac{c_3(\tilde{b} + k)}{k}
\end{aligned}$$

where the third inequality follows because we are only considering W such that $|W| = ck$ for a constant fraction c and $c_3 = (c_1/c)$.

Now, since each $M_{I,J_W} \geq 0$, we may use Markov's inequality to show that:

$$\Pr_{(I,J_W)} \Big[\log(1 + M_{I,J_W}) \geq \frac{c_3(\tilde{b} + k)}{\alpha k} \Big] \leq \alpha.$$

Define $U = \{(i, j_W) \mid i \in W, \log(1 + M_{i,j_W}) < c_3(\tilde{b} + k)/\alpha k\}$ and for all $i \in W$, we may define $p_i^U = \Pr_{J_W}[(i, J_W) \notin U]$. Note that $E[|W \setminus W'|] \leq \sum_{i \in W} p_i^U \leq \alpha |W|$ and using Markov's inequality, we say that $\Pr[|W \setminus W'| \geq \alpha |W| / \gamma] \leq$

γ. Plugging the definition of W' and $\sigma^2 = \Theta(k) = c'k$, into (2.8) gives

$$\begin{aligned}
&I((X)_{W'}; Y_1^m \mid (X)_{W\setminus W'}, (x)_{[n]\setminus W}, W') \\
&\quad \le m \log\left(1 + \frac{|W'| \cdot \sum_{i\in W'} 2^{\log(1+\frac{1}{n}\sum_{s=1}^m \left\|v_{s|i}\right\|_2^2) + \frac{1}{\alpha}\left(c_3(\tilde{b}/k)+1\right)}}{c'mk}\right) + c_2(\tilde{b}+k) \\
&\quad \le m \log\left(1 + \frac{|W'| \cdot 2^{\frac{1}{\alpha}\left(c_3(\frac{\tilde{b}}{k})+1\right)} \sum_{i\in W'} \left(1 + \frac{1}{n}\sum_{s=1}^m \left\|v_{s|i}\right\|_2^2\right)}{c'mk}\right) + c_2(\tilde{b}+k).
\end{aligned} \tag{2.9}$$

Since $\sum_{i\in[n]} \left\|v_{s|i}\right\|_2^2 = N$, we know that $\sum_{i\in W'} \left(1 + \frac{1}{n}\sum_{s=1}^m \left\|v_{s|i}\right\|_2^2\right) \le |W'| + \frac{Nm}{n} = |W'| + km$. Plugging this into (2.9), we get:

$$\begin{aligned}
&I((X)_{W'}; Y_1^m \mid (X)_{W\setminus W'}, (x)_{[n]\setminus W}, W') \\
&\quad \le m \log\left(1 + \frac{|W'| \cdot 2^{\frac{c_3}{\alpha}(\tilde{b}/k+1)} \cdot (W' + km)}{c'mk}\right) + c_2(\tilde{b}+k) \\
&\quad \le m \log(1 + \frac{|W'|}{c'}) + m \log(1 + 2^{\frac{c_3}{\alpha}(\frac{\tilde{b}}{k}+1)}) + c_2(\tilde{b}+k) \\
&\quad \le m \log(1+k) + m \log(1 + \frac{c}{c'}) + m \log(1 + 2^{\frac{c_3}{\alpha}(\frac{\tilde{b}}{k}+1)}) + c_2(\tilde{b}+k) \\
&\quad \le 2m + m \log(k) + m \log(1 + \frac{c}{c'}) + 2m + m \log(2^{\frac{c_3}{\alpha}(\frac{\tilde{b}}{k}+1)}) + c_2(\tilde{b}+k) \\
&\quad \le m \log(k) + \frac{c_3 m\tilde{b}}{\alpha k} + \frac{c_4 m}{\alpha} + c_2(\tilde{b}+k)
\end{aligned}$$

where $c_4 = 4 + \log(1 + c/c')$ is a constant. So, with probability $1 - \gamma$ there exists a set $W' \subseteq W$ such that $|W'| \ge (1 - \alpha/\gamma)\,|W|$ and

$$I((X)_{W'}; Y_1^m \mid (X)_{W\setminus W'}, (x)_{[n]\setminus W}, W') \le c_3 \frac{m\tilde{b}}{\alpha k} + m \log(k) + \frac{c_4 m}{\alpha} + c_2(\tilde{b}+k).$$

Now, taking the expectation of this term over $(x)_{[n]\setminus W}$, with probability $1 - \gamma$ there exists a set $W' \subseteq W$ such that $|W'| \ge (1 - \alpha/\gamma)\,|W|$ and

$$I((X)_{W'}; Y_1^m \mid (X)_{[n]\setminus W'}, W') \le c_3 \frac{mb}{\alpha k} + m \log(k) + \frac{c_4 m}{\alpha} + c_2(b+k).$$

□

By applying Lemma 2.3 every round, we get the desired lower bound on m.

Theorem 2.4. *Any scheme using R adaptive rounds with $m_1, \ldots, m_R$ measurements in each round and m total measurements has a set $W \subseteq [k], |W| \geq \Omega(k)$ such that with probability $\geq 3/4$*

$$I((X_i)_{i\in W}; Y_1, \ldots, Y_m \mid (X_i)_{i\notin W}, W) \leq \left(\prod_{j=2}^{R} \left(2c_5 + \frac{32c_6R^2m_j}{k}\right)\right) \max\{k\log(k), m_1\}$$

where c_5 and c_6 are constants. Consequently, for (k, C)-sparse recovery with $C = O(1)$, it must hold that

$$m \geq \frac{k}{C'R} \min\left\{\left(\log(N/k)\right)^{1/R}, \left(\frac{\log(N/k)}{\log(k)}\right)^{1/(R-1)}\right\}$$

for some constant C'.

Proof. Let A^r be the measurement matrix in round r (which we may assume is deterministically chosen as a function of all the previous rounds). Since the first round is non-adaptive, we may use the Shannon-Hartley Theorem (as per [PW12]) to show that for $W_2 = [k]$,

$$I((X_i)_{i\in W_2}; Y_{1,1}, \ldots, Y_{1,m_1} \mid (X_i)_{i\notin W_2}, W_2) \leq m_1.$$

For each round r, by p_r we denote Bob's prior distribution at the beginning of that round. We also denote by $b^{(r)} = |W_r|\log(n) - H(X_{W_r} \mid X_{[n]\setminus W_r})$ the number of bits of information in the prior $(X_i)_{i\in W_r}$ conditioned on $(X_i)_{i\notin W_r}$.

Since the rows of A^r are deterministic given the observations in previous rounds, we may apply Lemma 2.3 with $\alpha = 1/(16R^2)$, $\gamma = 1/4R$, and with probability $(1 - (1/4R))$ obtain a set $W_{r+1} \subseteq W_r$ such that $|W_{r+1}| \geq (1 - \frac{\alpha}{\gamma})\,|W_r|$ and:

$$\begin{aligned} &I((X_i)_{i\in W_{r+1}}; Y^{r+1} \mid y^1, \ldots, y^r, (X_i)_{i\notin W_{r+1}}, W_{r+1}) \\ &\qquad \leq c_3 \frac{m_{r+1} b_r}{\alpha k} + m_{r+1}\log(k) + \frac{c_4 m_{r+1}}{\alpha} + c_2(b_r + k). \end{aligned}$$

Let us define $B_{r+1} = I((X_i)_{i\in W_{r+1}}; Y^{r+1}, \ldots, Y^1 \mid (X_i)_{i\notin W_{r+1}}, W_{r+1})$. Using the chain rule of mutual information for $r > 1$

$$\begin{aligned} B_{r+1} &= I((X_i)_{i\in W_{r+1}}; Y^r, \ldots, Y^1 \mid (X_i)_{i\notin W_{r+1}}) \\ &\qquad + I((X_i)_{i\in W_{r+1}}; Y^{r+1} \mid Y^r, \ldots, Y^1, (X_i)_{i\notin W_{r+1}}, W_{r+1}) \\ &\leq B_r + \mathop{\mathbb{E}}_{y^1,\ldots,y^r}[I((X_i)_{i\in W_{r+1}}; Y^{r+1} \mid y^1, \ldots, y^r, (X_i)_{i\notin W_{r+1}}, W_{r+1})]. \end{aligned}$$

So,

$$\begin{aligned} B_{r+1} &\leq B_r + c_3 \frac{m_{r+1} B_r}{\alpha k} + m_{r+1}\log(k) + \frac{c_4 m_{r+1}}{\alpha} + c_2(B_r + k) \\ &\leq \big(c_5 + \frac{c_3 m_{r+1}}{\alpha k}\big) B_r + m_{r+1}\log(k) + \frac{c_4 m_{r+1}}{\alpha} + c_2 k \end{aligned} \tag{2.10}$$

where $c_5 = c_2 + 1$. We know using the Shannon-Hartley Theorem that $B_1 \leq m_1$. Further, we assume that $B_1 \geq k\log(k)$. While this weakens our lower bound, it allows us to make a cleaner inductive argument into Claim A.3. Plugging $\alpha = 1/16R^2$ in Claim A.3, we get:

$$B_R \leq \Big(\prod_{j=2}^{R} \big(2c_5 + \frac{32 c_6 R^2 m_j}{k}\big)\Big) \max\{k\log(k), m_1\}.$$

It follows using the AM-GM inequality that:

$$B_R \leq \max\left\{k \cdot \left(2c_5 + \frac{32c_6R \cdot m}{k}\right)^R, k\log(k) \cdot \left(2c_5 + \frac{32c_6R \cdot m}{k}\right)^{(R-1)}\right\}.$$

So, after R rounds with probability $\geq 3/4$, we have a set W_R such that $|W_R| \geq (1 - \frac{\alpha}{\gamma})^R k \geq e^{-4}k$ with $I((X_i)_{i\in W_R}; Y^R, \ldots, Y^1 \mid (X_i)_{i\notin W_R}, W_R)$ bounded as above. We may scale the variance of w (gaussian noise) by appropriate constants, so that for sparse recovery to succeed $k(1 - \frac{1}{2e^4})$ indices must be fully recovered with probability $\geq 3/4$. So, for the set W_R it must hold that $I((X_i)_{i\in W_R}; Y^R, \ldots, Y^1 \mid (X_i)_{i\notin W_R}, W_R) \geq \frac{k}{2e^4}\log(N/k)$ and as a consequence, it must hold that:

$$\max\left\{\left(2c_5 + \frac{32c_6R \cdot m}{k}\right)^R, \left(2c_5 + \frac{32c_6R \cdot m}{k}\right)^{(R-1)} k\log(k)\right\} \geq \frac{k}{2e^4}\log(N/k)$$

If we simplify this and set $C' = 32c_6$, we get

$$m \geq \min\left\{\frac{k}{C'R}\left(\log(N/k)\right)^{1/R}, \frac{k}{C'R}\left(\frac{\log(N/k)}{\log(k)}\right)^{1/(R-1)}\right\}$$

□

If we restrict our sparsity parameter k to be $O(2^{(\log(N))^{1/R}})$ we observe that this lower bound is tight.

Corollary 2.5. *Let $C > 1$. Any (k, C)-sparse recovery scheme for vectors in $\mathbb{R}^N$ that uses R adaptive rounds and m total measurements with $k = O(2^{\log^{1/R} N})$ must satisfy*

$$m \geq \frac{k}{C'R}\left(\log(N/k)\right)^{1/R}$$

for some constant C'.

2.4 Upper Bound for Adaptive Sparse Recovery with Limited Adaptivity

In this section we present our algorithm for (k, C)-sparse recovery in R rounds. The main goal is to prove Theorem 2.10 which shows that Algorithm 2.4.2 achieves (k, C) sparse recovery using $O(k \log_C(n/k)^{1/R} \log^*(k) \cdot 2^R)$ measurements. Lemma 2.7 shows that in each round we lose a small amount of mass from the vector. Lemma 2.8 and Lemma 2.9 show that with a constant increase in the sparsity parameter from one round to the next, we can ensure that the "noise" carried over to the next round decreases by a factor.

2.4.1 Preliminaries

We start with a few definitions. Let x be an n-dimensional vector.

Definition 2.4.1. Define

$$H_k(x) = \arg\max_{\substack{S \in [n] \\ |S|=k}} \|x_S\|_2$$

to be the largest k coefficients in x.

Definition 2.4.2. Define the "noise" or "error"

$$\mathrm{Err}^2(x, k) = \left\| x_{\overline{H_k(x)}} \right\|_2^2.$$

Definition 2.4.3. Given a vector x, a recovered vector x^* satisfies (k, C)-sparse recovery under the ℓ_2/ℓ_2 guarantee if:

$$\|x - x^*\|_2^2 \leq C\,\mathrm{Err}^2(x, k).$$

Definition 2.4.4. Given a hash function $h : [n] \to [D]$, a (D, h)-**gaussian hash projection** of a vector $x \in \mathbb{R}^n$ into $\mathbb{R}^D$ is given by $y \in \mathbb{R}^D$ such that $y_j = \sum_{i:h(i)=j} x_i \cdot g_i$ where $g_i \sim \mathcal{N}(0, 1)$ is i.i.d normal with variance 1 and mean 0.

We denote by HighSNR-Recover(x, k, C, δ) a black-box algorithm which makes linear measurements on the input x and whose output achieves (k, C) sparse recovery with probability $1 - \delta$. The best known algorithm for achieving (k, C)-sparse recovery when $C \geq 1$ is the algorithm from [KP20]:

Theorem 2.6. *There exists an algorithm that takes $O\big(k \log_C(\frac{n}{k}) \log(\frac{1}{\delta})\big)$ linear measurements and outputs a k-sparse vector that achieves (k, C)-sparse recovery under the ℓ_2/ℓ_2 guarantee with success probability $1 - \delta$.*

We prove this theorem in Chapter 3.

2.4.2 Algorithm

procedure 1-RoundSNRBoost(x, n, D, C, k, δ) ▷ Recover most of the mass of heavy hitters while reducing noise by factor D/k
For $i \in [n], h(i) \leftarrow [D]$
For $i \in [n], t \in \{1, 2, 3\}$ $g_i^{(t)} \leftarrow \mathcal{N}(0, 1)$
For $j \in [D], t \in \{1, 2, 3\}$ define $y_j^{(t)} = \sum_{i \in h^{-1}(j)} g_i^{(t)} x_i$
For $t \in \{1, 2, 3\}$, $U^{(t)} \leftarrow \text{supp}(\text{HighSNR-Recover}(y^{(t)}, k, C, \delta/3))$
return $\cup_{j \in U^{(1)} \cup U^{(2)} \cup U^{(3)}} h^{-1}(j)$
end procedure

Algorithm 2.4.1: 1 round SNR-Boost

procedure R-ROUND-K-SPARSEREC(x, k, C, R)
  $S_0 = [n]$
  $C_0 = C/8$
  for $r \leftarrow 1, \ldots, R-1$ **do**
    $k_r \leftarrow k5^{r-1}$
    $D_r \leftarrow k_r C_0^{5(\log_{C_0}(n))^{r/R}}$
    $C_r \leftarrow C_0^{(\log_{C_0}(n))^{(r-1)/R}}$
    $\delta_r \leftarrow 2^{-(r+3)}$
    $S_r \leftarrow$ 1-ROUNDSNRBOOST$(x_{S_{r-1}}, |S_{r-1}|, D_r, C_r, k_r, \delta_r)$
  end for
  return $\hat{x} \leftarrow$ HIGHSNR-RECOVER$(x_{S_{R-1}}, 5k_{R-1}, C_0^{(\log_{C_0} n)^{\frac{(R-1)}{R}}}, 2^{-(R+3)})$
end procedure

Algorithm 2.4.2: R-Round-k-Sparse Recovery

Lemma 2.7. *Let $x \in \mathbb{R}^n$, $D \geq k$, $C \geq 1$. Suppose $h : [n] \to [D]$ is drawn from a fully independent family of hash functions and $y^{(1)}$, $y^{(2)}$ and $y^{(3)}$ are independent (D, h)-gaussian hash projections of x. Then, if $\mathcal{A}$ is an algorithm that achieves (k, C) sparse recovery with probability $\geq 8/9$, and $U^{(t)} = \operatorname{supp}(\mathcal{A}(y^{(t)}))$ for $t \in \{1, 2, 3\}$,*

$$\mathbb{E}\left[\sum_{\substack{j \in [D] \\ j \notin U^{(1)} \cup U^{(2)} \cup U^{(3)}}} \left\| x_{h^{-1}(j)} \right\|_2^2 \,\middle|\, \mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3 \right] \leq 9C \operatorname{Err}^2(x, k)$$

where $\mathcal{E}^{(t)}$ represents the event that $\mathcal{A}(y^{(t)})$ successfully performs (k, C)-sparse recovery.

Proof. Let y be a (D, h)-gaussian hash projection of x. From the definition of $H_k(y)$, we know that for all S such that $|S| \leq k$, $\operatorname{Err}^2(y, k) = \sum_{j \in \overline{H_k(y)}} y_j^2 \leq \sum_{j \in \overline{S}} y_j^2$. If we choose $S = h(H_k(x))$, we get $\operatorname{Err}^2(y, k) \leq \sum_{j \in \overline{h(H_k(x))}} y_j^2$.

Furthermore,

$$\begin{aligned}
\mathop{\mathbb{E}}_{g}[\mathrm{Err}^2(y,k)] &\le \mathop{\mathbb{E}}_{g}[\sum_{j\in\overline{h(H_k(x))}} y_j^2] \\
&= \mathop{\mathbb{E}}_{g}[\sum_{j\in\overline{h(H_k(x))}} \Big(\sum_{i\in h^{-1}(j)} x_i\cdot g_i\Big)^2] \\
&= \sum_{j\in\overline{h(H_k(x))}} \sum_{i\in h^{-1}(j)} x_i^2 \\
&\le \sum_{i\in\overline{H_k(x)}} x_i^2 = \mathrm{Err}^2(x,k)
\end{aligned}$$

where the second equality follows because $g_i \sim \mathcal{N}(0,1)$ for all $i \in [n]$.

Let E_j be the indicator random variable for the event that $j \notin U$ where $U = \mathrm{supp}(\mathcal{A}(y))$. For a successful run of $\mathcal{A}$, the ℓ_2 mass of the unrecovered indices is bounded by:

$$\sum_{j\in[D]} E_j y_j^2 \le C\,\mathrm{Err}^2(y,k).$$

Let $\mathcal{E}$ be the event that $\mathcal{A}(y)$ satisfies the (k,C)-sparse recovery guarantee for y. Then, if $\mathbb{I}(\mathcal{E})$ is the indicator random variable for the event $\mathcal{E}$,

$$\begin{aligned}
\mathop{\mathbb{E}}_{g,\mathcal{A}}[\sum_{j\in[D]} E_j y_j^2 \mid \mathcal{E}] &\le \mathop{\mathbb{E}}_{g,\mathcal{A}}[(\sum_{j\in[D]} E_j y_j^2)\mathbb{I}(\mathcal{E})]/\Pr_{g,\mathcal{A}}[\mathcal{E}] \\
&\le \frac{9C}{8}\mathop{\mathbb{E}}_{g}[\mathrm{Err}^2(y,k)] \\
&\le \frac{9C}{8}\mathrm{Err}^2(x,k). \qquad (2.11)
\end{aligned}$$

Let $q_j = \mathbb{E}_{g,\mathcal{A}}\left[E_j \;\middle|\; \mathcal{E}\right]$ denote the probability $\big($over (D,h) projections and $\mathcal{A}\big)$ that $j \notin \mathrm{supp}(\mathcal{A}(y))$. Then for $j \in [D]$ and any $\theta > 0$,

$$\mathop{\mathbb{E}}_{g,\mathcal{A}}[E_j y_j^2 | \mathcal{E}] \ge \Pr\left[\Big(j \notin U\Big) \wedge \Big(\,|y_j| \ge (q_j/2)\theta\Big) \;\middle|\; \mathcal{E}\right]\cdot\theta^2.$$

Observe that:

$$\Pr\left[\left(j \notin U\right) \wedge \left(|y_j| \geq \theta\right) \;\middle|\; \mathcal{E}\right] \geq 1 - \Pr\left[j \in U \;\middle|\; \mathcal{E}\right] - \Pr\left[|y_j| < \theta \;\middle|\; \mathcal{E}\right].$$

Since $y_j \sim \mathcal{N}(0, \theta^2)$ we may use the gaussian anti-concentration inequality i.e. $\Pr[|X| \leq \delta\theta] \leq \delta$ to get:

$$\Pr\left[\left(j \notin U\right) \wedge \left(|y_j| \geq \theta\right) \;\middle|\; \mathcal{E}\right] \geq 1 - (1 - q_j) - \frac{\theta}{\left\|x_{h^{-1}(j)}\right\|_2}.$$

Setting $\theta = \frac{q_j}{2}\left\|x_{h^{-1}(j)}\right\|_2$:

$$\Pr\left[\left(j \notin U\right) \wedge \left(|y_j| \geq \frac{q_j}{2}\left\|x_{h^{-1}(j)}\right\|_2\right) \;\middle|\; \mathcal{E}\right] \geq q_j/2$$

and for all $j \in [D]$,

$$\mathop{\mathbb{E}}_{g,A}[E_j y_j^2 \mid \mathcal{E}] \geq \frac{q_j^3}{8}\left\|x_{h^{(-1)}(j)}\right\|_2^2. \tag{2.12}$$

Now, consider the $U^{(t)} = \mathrm{supp}(\mathcal{A}(y^{(t)}, k, C))$ for $t = 1, 2, 3$ where $y^{(1)}, y^{(2)}, y^{(3)}$ are independent (D, h) gaussian projections of x. Then,

$$\begin{aligned}
&\mathbb{E}\Big[\sum_{\substack{j\in[D]:\\ j\notin U^{(1)}\cup U^{(2)}\cup U^{(3)}}} \left\|x_{h^{-1}(j)}\right\|_2^2 \;\Big|\; \mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3\Big] \\
&= \sum_{j\in[D]} \left\|x_{h^{-1}(j)}\right\|_2^2 \cdot \mathbb{E}[E_j^{(1)} \cdot E_j^{(2)} \cdot E_j^{(3)} \mid \mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3] \\
&= \sum_{j\in[D]} \left\|x_{h^{-1}(j)}\right\|_2^2 \cdot \mathbb{E}[E_j^{(1)} \mid \mathcal{E}_1] \cdot \mathbb{E}[E_j^{(2)} \mid \mathcal{E}_2] \cdot \mathbb{E}[E_j^{(3)} \mid \mathcal{E}_3] \\
&= \sum_{j\in[D]} \left\|x_{h^{-1}(j)}\right\|_2^2 \cdot q_j^3
\end{aligned}$$

where the expectation is taken over $g^{(1)}, g^{(2)}, g^{(3)}, \mathcal{A}(y^{(1)}), \mathcal{A}(y^{(2)}), \mathcal{A}(y^{(3)})$. The second equality follows from the independence of $y^{(1)}, y^{(2)}, y^{(3)}$. So, using (2.11) and (2.12),

$$\begin{aligned}\mathbb{E}\Big[\sum_{\substack{j\in[D]:\\ j\notin U^{(1)}\cup U^{(2)}\cup U^{(3)}}} \left\|x_{h^{-1}(j)}\right\|_2^2 \mid \mathcal{E}_1,\mathcal{E}_2,\mathcal{E}_3\Big] &\le \sum_{j\in[D]} \left\|x_{h^{-1}(j)}\right\|_2^2 \cdot q_j^3 \\ &\le 8\sum_{j\in[D]} \mathop{\mathbb{E}}_{g,\mathcal{A}}[E_j y_j^2 \mathbb{I}(\mathcal{E})] \\ &\le 9C\,\mathrm{Err}^2(x,k).\end{aligned}$$

□

Lemma 2.8. *Let $z \in \mathbb{R}^n$ and $h : [n] \to [D]$ be randomly chosen from a fully independent family of hash functions where $D \le n$. Then, with probability $1 - 2\delta$,*

$$\max_{l\in[D]}\left[\sum_{i\in h^{-1}(l)} z_i^2\right] \le 4\left(\frac{\|z\|_2^2}{D} + 5\,\|z\|_\infty^2 \log\big(\frac{D\cdot\log(n/\delta)}{\delta}\big)\right).$$

Proof. Let $\beta_j = \|z\|_\infty^2 \cdot 2^{-j}$ for all $j \in \mathbb{Z}$ and let $t = O(\log(n/\delta))$. Partition $[n]$ into $t+2$ sets: $R_j = \{i \in [n] \mid \beta_{j+1} \le z_i^2 \le \beta_j\}$ for all $0 \le j \le t$ and $R_{t+1} = \{i \in [n] \mid z_i^2 \le \beta_{t+1}\}$. Then, for a fixed R_j and $l \in [D]$ we may apply the Bernstein bounds (Theorem A.4) to get:

$$\Pr\Big[\big|R_j \cap h^{-1}(l)\big| \ge \frac{|R_j|}{D} + 4\log(1/\delta) + 4\sqrt{\frac{\log(1/\delta)\,|R_j|}{D}}\Big] \le \delta.$$

Taking a union bound over all $R_0, \ldots, R_t$ and all $l \in [D]$:

$$\Pr\Big[\exists j \in [t], l \in [D] \;\Big|\; \big|R_j \cap h^{-1}(l)\big| \ge \frac{|R_j|}{D} + 4\log\frac{D\cdot t}{\delta} + 4\sqrt{\frac{\log(\frac{D\cdot t}{\delta})\,|R_j|}{D}}\Big] \le \delta.$$

The ℓ_2 mass from $R_0, \ldots, R_t$ falling into any $j \in [D]$ is bounded by:

$$\begin{aligned}\sum_{j=0}^{t} \beta_j \Big(\frac{|R_j|}{D} + 4\log(\frac{D\cdot t}{\delta}) + 4\sqrt{\frac{\log(\frac{D\cdot t}{\delta})\,|R_j|}{D}} \Big) &\leq 2\sum_{j=0}^{t} \beta_j \big(\frac{|R_j|}{D} + 4\log(\frac{D\cdot t}{\delta}) \big) \\ &\leq 4\big(\frac{\|z\|_2^2}{D} + 4\log(\frac{D\cdot t}{\delta})\beta_0 \big) \\ &= 4\big(\frac{\|z\|_2^2}{D} + 4\log(\frac{D\cdot t}{\delta})\,\|z\|_\infty^2 \big)\end{aligned}$$

where the second inequality follows because $\sum_{j=0}^{t} |R_j|\,\beta_j \leq 2\sum_{i\in[n]} z_i^2 \leq 2\,\|z\|_2^2$.

Next, we bound contribution of R_{t+1} to the ℓ_2 mass hashed to each location. The total ℓ_2 mass in R_{t+1} is $\big\|z_{R_{t+1}}\big\|_2^2 \leq \beta_{t+1}\cdot n$. So, the expected amount of ℓ_2 mass in a given location $l \in [D]$ is $\leq n\beta_{t+1}/D$. Using Markov's inequality, with probability $1-\delta$, we know that the ℓ_2 mass from R_{t+1} hashed to each location in $[D]$ is $\leq n\cdot\|z\|_\infty^2 \cdot 2^{-(t+1)}/\delta \leq \|z\|_\infty^2$. So,

$$\max_{l\in[D]} \Big[\sum_{i\in h^{-1}(l)} z_i^2 \Big] \leq 4\Big(\frac{\|z\|_2^2}{D} + 5\,\|z\|_\infty^2 \log\big(\frac{D\cdot\log(n/\delta)}{\delta}\big) \Big).$$

□

Lemma 2.9. *Let $z \in \mathbb{R}^n$, $k \leq D \leq n$ and $h : [n] \to [D]$ be randomly chosen from a fully independent family of hash functions. Then, with probability $1-\delta$, for all $U \subseteq [D]$:*

$$\mathrm{Err}^2(z_{h^{-1}(U)}, |U|+k) \leq \|z\|_2^2\,\frac{|U|\,O(\log(n/\delta))}{\sqrt{kD\delta}}.$$

Proof. Consider all indices in the set $J = \{i \in [n] \mid z_i^2 \geq \|z\|_2^2/L\}$ where $L = \sqrt{kD\delta}$. Observe that the expected number of collisions among these elements

under the hash function h is $\leq \binom{L}{2}/D \leq k\delta/2$. By Markov's inequality, the number of collisions is at most k with probability $1-(\delta/2)$. So, with probability $1-\delta/2$:

$$\forall U \subset [D], \left|J \cap h^{-1}(U)\right| \leq |U| + k \tag{2.13}$$

Suppose, we restricted ourselves only to the indices in the set $\overline{J}$. Observe that $\|z_{\overline{J}}\|_2 \leq \|z\|_2$ and $\|z_{\overline{J}}\|_\infty^2 \leq \|z\|_2^2/L$. Applying Lemma 2.8, with probability $1-\delta/2$:

$$\begin{aligned}\max_{l\in[D]}\left[\sum_{i\in\overline{J}:h(i)=l} z_i^2\right] &\leq 4\left(\frac{\|z\|_2^2}{D} + \frac{5\|z\|_2^2}{L}\log\left(\frac{4D\log(4n/\delta)}{\delta}\right)\right)\\ &= O\left(\|z\|_2^2\,\frac{O(\log(n/\delta))}{L}\right)\end{aligned} \tag{2.14}$$

So, with probability $1-\delta$, both (2.13) and (2.14) hold. Hence,

$$\begin{aligned}\mathrm{Err}^2(z_{h^{-1}(U)}, |U|+k) &\leq |U|\cdot\left(\|z\|_2^2\,\frac{O(\log(n/\delta))}{L}\right)\\ &\leq \|z\|_2^2\,\frac{|U|\,O(\log(n/\delta))}{\sqrt{kD\delta}}\end{aligned}$$

□

Theorem 2.10. *Suppose an algorithm that takes $O(k\log_C(\frac{n}{k})\log(\frac{1}{\delta})\cdot g(k))$ linear measurements of its input where $g(k)$ is a non-decreasing function in k and outputs a k sparse vector that achieves (k, C) sparse recovery with probability $(1-\delta)$. Then, for $R \leq \log\log(n/k)/2\log\log\log(n)$ and $C > 16$, Algorithm 2.4.2 takes $O(k5^R(\log_C(n/k))^{1/R}\cdot g(5^R k))$ linear measurements of $x \in \mathbb{R}^n$ over R adaptive rounds and outputs a vector that achieves (k, C) sparse recovery of x with probability $\geq \frac{3}{4}$.*

Proof. In this proof, we will achieve $(k, 16C)$ sparse recovery for all $C > 1$. We may rescale C to get the theorem statement. We define

$$\begin{aligned}
\delta_r &= 2^{-(r+3)} \\
k_r &= k5^{r-1} \\
D_r &= k_r C^{5(\log_C(n/k))^{r/R}} \\
C_r &= C^{(\log_C(n/k))^{(r-1)/R}}
\end{aligned}$$

for $r > 0$ and $S_0 = [n]$

In each round $r \in \{1, \ldots, R-1\}$, we use Algorithm 2.4.1 with these parameters to get a subset $S_r \subseteq S_{r-1}$. We sample a random hash function $h : S_{r-1} \to [D_r]$ and generate 3 independent (D_r, h)-gaussian hash projections $y^{(1)}, y^{(2)}, y^{(3)}$ of $x_{S_{r-1}}$ and perform HighSNR-Recover on each of them with parameters $(k_r, C_r, \delta_r/3)$. Let $U^{(1)}, U^{(2)}, U^{(3)}$ be supports of the recovered vectors. Since HighSNR-Recover generates k_r sparse output, $\left|U^{(1)}\right|, \left|U^{(2)}\right|, \left|U^{(3)}\right| \leq k_r$. Let $U_r = U^{(1)} \cup U^{(2)} \cup U^{(3)}$, and set $S_r = h^{-1}(U_r) \subseteq S_{r-1}$ to be the set of indices carried into the next round. So, if we set $z = x_{S_{r-1} \cap \overline{H_{k_{r-1}}(x_{S_{r-1}})}}$ and let $U = U_r$ in Lemma 2.9:

$$\begin{aligned}
\mathrm{Err}^2(z_{h^{-1}(U_r)}, |U_r| + k_{r-1}) &\leq \frac{\|z\|_2^2}{\sqrt{D_r \delta_r / k_r O(\log(n/\delta_r))}} \\
&\leq \frac{\|z\|_2^2}{2^{2(\log(n))^{r/R}}}
\end{aligned}$$

where the second inequality follows because $\log(n) = o(C^{2(\log_C(n))^{1/R}})$ when $2^r \leq C^{2(\log_C(n))^{r/R}}$ and $R \leq \frac{\log\log(n)}{2\log\log\log(n)}$. Since $z = x_{S_{r-1} \cap \overline{H_{k_{r-1}}(x_{S_{r-1}})}}$, we have

both $\|z\|_2^2 = \left\|x_{S_{r-1}\cap\overline{H_{k_{r-1}}(x_{S_{r-1}})}}\right\|_2^2 = \mathrm{Err}^2(x_{S_{r-1},k_{r-1}})$ and $\mathrm{Err}^2(z_{h^{-1}(U)}, |U| + k_r) \geq \mathrm{Err}^2(x_{h^{-1}(U)}, |U| + k_{r-1} + k_{r-1})$. Since $|U| \leq 3k_{r-1}$ and $5k_{r-1} = k_r$, we conclude:

$$\mathrm{Err}^2(x_{S_r}, k_r) \leq \frac{\mathrm{Err}^2(x_{S_{r-1}}, k_{r-1})}{C^{2(\log_C(n))^{r/R}}}$$

If we successively apply Theorem 2.7 under the above parameters for rounds $1, \ldots, R-1$, then for any $r \in \{1, \ldots, R-1\}$

$$\begin{aligned}\mathbb{E}[\left\|x_{S_r} - x_{S_{r+1}}\right\|_2^2] &\leq C_r \,\mathrm{Err}^2(x_{S_r}, k_r) \\ &\leq \frac{C_r}{C^{2(\log_C(n))^{r/R}}} \,\mathrm{Err}^2(x_{S_{r-1}}, k_{r-1})\end{aligned}$$

Since $\mathrm{Err}^2(x_{S_{r-1}}, k_{r-1}) \leq \mathrm{Err}^2(x, k)$ and we have set $C_r = C^{(\log_C(n/k))^{(r-1)/R}}$,

$$\mathbb{E}[\left\|x_{S_r} - x_{S_{r+1}}\right\|_2^2] \leq \frac{1}{C^{(\log_C(n))^{r/R}}} \,\mathrm{Err}^2(x, k)$$

In the final round, we run HighSNR-Recover$(x_{S_{R-1}}, k_R, C_R)$ and find $\hat{x}$ such that $\left\|x_{S_{R-1}} - \hat{x}\right\|_2^2 \leq C_R \,\mathrm{Err}^2(x_{S_{R-1}}, k_R)$. So,

$$\begin{aligned}\mathbb{E}\left[\,\|x - \hat{x}\|_2^2\,\right] &\leq \sum_{r=1}^{R-1} \mathbb{E}\left[\,\left\|x_{S_{r-1}} - x_{S_r}\right\|_2^2\,\right] + \mathbb{E}\left[\,\left\|x_{S_{R-1}} - \hat{x}\right\|_2^2\,\right] \\ &\leq \sum_{r=1}^{R} C_r \,\mathrm{Err}^2(x_{S_r}, k_r) \\ &\leq C \,\mathrm{Err}^2(x_{S_1}, k_1) + \sum_{r=2}^{R} \frac{1}{C^{(\log_C(n))^{(r-1)/R}}} \,\mathrm{Err}^2(x, k) \\ &\leq 2C \,\mathrm{Err}^2(x, k)\end{aligned}$$

So, with probability $\geq 7/8$, after R rounds $\|x - \hat{x}\|_2^2 \leq 16C \,\mathrm{Err}^2(x, k)$. In each round, we use independently call HighSNR-Recover$(x_{S_{r-1}}, k_r, C_r)$

thrice with failure probability $\delta_r/3 = 2^{-(r+3)}/3$ and condition on them being successful. So, over R rounds all calls to HighSNR-Recover are successful with probability $\geq 1 - \sum_{r=1}^{R} \delta_r = 1 - \sum_{r=1}^{R} 2^{-(r+3)} = 7/8$.

The total number of measurements over R rounds is bounded by:

$$\begin{aligned}
&\sum_{r=1}^{R} 3k_r \log(3/\delta_r) \cdot g(5^r k) \cdot (\log_{C_{r-1}}(D_r/k)) \\
&= \sum_{r=1}^{R} 3k_r \log(3/\delta_r) \cdot g(5^r k) \cdot (\log_C(n/k))^{1/R} \\
&\leq \sum_{r=1}^{R} 3k \cdot 5^r \cdot 2r \cdot g(5^r k)(\log_C(n/k))^{1/R} \\
&= O(5^R k(\log_C(n/k))^{1/R} \cdot g(5^R k))
\end{aligned}$$

So, the output of Algorithm 2.4.2 achieves $(k, 16C)$ sparse recovery in R rounds with probability $\geq 3/4$ and uses $O(5^R k(\log_C(n/k))^{1/R} \cdot g(5^R k))$ measurements. If we rescale C by a factor of 16, we get the desired guarantee. □

As a consequence of Theorem 2.10 and Theorem 2.6, we get the following guarantee on our algorithm:

Corollary 2.11. *For $R \leq \frac{\log\log(n/k)}{\log\log\log(n)}$ and $C > 16$, Algorithm 2.4.2 takes $O(k5^R(\log_C(n/k))^{1/R} \cdot \log^*(5^R k))$ linear measurements of $x \in \mathbb{R}^n$ over R adaptive rounds and outputs a vector that achieves (k, C) sparse recovery of x with probability $\geq \frac{3}{4}$.*

Chapter 3

Sparse Recovery under High SNR

Most sparse recovery literature has focused on the case where $C = (1+\epsilon)$ for a small $\epsilon > 0$ and in the non-adaptive case tight bounds on measurement complexity are know. However, the exact measurement complexity for larger approximation ratios i.e. $C \gg 1$ is an open question.

In this chapter[1], we give upper bounds on the measurement complexity for ℓ_2/ℓ_2-sparse recovery when $C \gg 1$. We prove bounds which match the lower bound of [PW11, PW12]. Formally, we show that:

Theorem 3.1. *Suppose $C > 16$. Then, there exists an algorithm that achieves (k, C)-approximate ℓ_2/ℓ_2 sparse recovery with $O(k \log_C(\frac{n}{k}) \log(\frac{1}{\delta}))$ measurements and with probability $1 - \delta$. The running time of the recovery algorithm is $O(k \operatorname{polylog} n)$.*

The techniques we use to prove this theorem allow us to achieve a somewhat stronger guarantee which we refer to as (k, C)-approximate ℓ_∞/ℓ_2 sparse recovery guarantee. This is also known as "heavy hitters". Formally,

$$\|x - x^*\|_\infty^2 \leq \frac{C^2}{k} \min_{k\text{-sparse } x'} \|x - x'\|_2^2 . \tag{3.1}$$

[1]The results presented in this chapter appear in [KP20].

We may compare this to the ℓ_2/ℓ_2 guarantee:

$$\|x - x^*\|_2^2 \leq C^2 \min_{k\text{-sparse } x'} \|x - x'\|_2^2 \tag{3.2}$$

and notice that a vector that satisfies the ℓ_∞/ℓ_2 guarantee accurately recovers *every* coordinate of x, whereas a vector that satisfies the ℓ_2/ℓ_2 guarantee only recovers a vector such that the sum of errors is bounded.

It is known that any algorithm achieving (3.1) can achieve (3.2) (with $C \to \sqrt{C^2+1}$) by thresholding the result to $2k$ coordinates. This guarantee is also achievable with $O(k \log n)$ measurements using, for example, CountSketch [CCF02]. We show that the ℓ_∞/ℓ_2 guarantee can be achieved with $O(k \log_C n)$ measurements.

Theorem 3.2. *Suppose $C > 16$. Then, there exists an algorithm that achieves (k, C)-approximate ℓ_∞/ℓ_2 sparse recovery with $O(k \log_C(n) \log(\frac{1}{\delta}))$ measurements and with probability $1 - \delta$. The running time of the recovery algorithm is $O(k \operatorname{polylog} n)$.*

Our ℓ_∞/ℓ_2 algorithm is almost identical to our ℓ_2/ℓ_2 algorithm, only differing in the last step. Whether this sample complexity is optimal—or if $O(k \log_C(\frac{n}{k}))$ is possible—is an open question even for constant C.

In Table 3.1, we list the various upper bounds and lower bounds for high-SNR sparse recovery under both the ℓ_2/ℓ_2 and ℓ_∞/ℓ_2 guarantee.

	Measurement Complexity Bound	Paper	Comment
ℓ_2/ℓ_2	$\Omega(k \log_C(\frac{n}{k}))$	[PW12]	
	$O(k \log_C(\frac{n}{k}) \log^*(k) \log(\frac{1}{\delta}))$	[PW12]	
	$O(k \log_C(\frac{n}{k}) \log(\frac{1}{\delta}))$	Theorem 3.1	
	$O(k \log_C(\frac{n}{k}) + \log(\frac{1}{\delta}))$	[KP20]	$O(n^k)$ recovery time
ℓ_∞/ℓ_2	$\Omega(k \log_C(\frac{n}{k}))$	[PW12]	Implied by ℓ_2/ℓ_2 lower bound
	$\Omega(k \log(\frac{1}{\delta}))$	[LNW18]	when $C = O(1)$
	$O(k \log(n) \log(\frac{1}{\delta}))$	[PW11]	when $C = O(1)$
	$O(k \log_C(n) \log(\frac{1}{\delta}))$	Theorem 3.2	

Table 3.1: Results for (k, C)-sparse recovery under the ℓ_2/ℓ_2 guarantee and the ℓ_∞/ℓ_2 guarantee.

3.1 A Discussion of Previous Results

We denote the error of $x \in \mathbb{R}^n$ with respect to k-sparse vectors in $\mathbb{R}^n$ as $\mathrm{Err}^2(x, k) = \min_{k\text{-sparse } x'} \|x - x'\|_2^2$. We use S to denote the set of heavy hitter indices in a vector $x \in \mathbb{R}^n$ i.e. $S = \{i \in [n] \mid |x_i|^2 > \frac{C^2}{k} \mathrm{Err}^2(x, k)\}$.

Our algorithm is similar to [PW12], which built on [GLPS10]. In [GLPS10], the goal is to perform $(k, 1+\epsilon)$-approximate ℓ_2/ℓ_2 sparse recovery. They run $O(\log(k))$ iterations such that in each iteration, they identify and peel off $\frac{3}{4}$ fraction of the remaining heavy hitters.

In the next round, they perform the same process with parameters $\frac{k}{4}$ and ϵ. With these parameters, heavy hitters are indices i such that $x_i^2 \geq 4\frac{\epsilon}{k} \mathrm{Err}^2(x, k)$. So, some indices that were originally heavy hitters (e.g. i such that $x_i^2 \approx 2\frac{\epsilon}{k} \mathrm{Err}^2(x, k)$) may be ignored in this iteration. However, the total weight of heavy hitters "dropped" in this manner over all the iterations is $\leq \epsilon \, \mathrm{Err}^2(x, k)$. Since they focus on achieving an ℓ_2/ℓ_2 guarantee, these heavy hitters may be ignored. [PW12] uses similar ideas with slightly more compli-

cated parameters. Their algorithm and analysis also allow for heavy hitters to be dropped.

3.2 An Overview of Our Results

We will iteratively identify heavy hitters, estimate them and peel them off similar to [PW12], [GLPS10] while ensuring that we never drop them. However, we stop after pruning the number of heavy hitters down to $O(\frac{k}{\log C})$ elements. Thereafter, performing $(\frac{k}{\log C}, O(1))$-approximate ℓ_∞/ℓ_2 sparse recovery using an algorithm like [LNNT16] will allow us to identify all the remaining heavy hitters using $O(\frac{k}{\log C} \cdot \log(n)) = O(k \cdot \log_C(n))$ measurements in total.

Identify most heavy hitters in a round: In one round, given sparsity parameter k and approximation parameter $C > 2$, we hash the indices $[n]$ down to $[16k]$ buckets. Since there are at most $k + \frac{k}{C}$ heavy hitters, a heavy hitter does not collide with any other heavy hitters with probability $\frac{7}{8}$. If the weight from the tail (the non-heavy hitters) that lands in that bucket is $\approx \frac{1}{k}\operatorname{Err}^2(x, k)$, we can perform $(1, C)$-approximate recovery within a bucket and recover that heavy hitter. So, we recover a set L of at most $16k$ elements(some of these are heavy hitters and some are non-heavy hitters). The identification procedure uses $O(k \log_C(\frac{n}{k}))$ measurements.

Probability Amplification The locations of $\frac{7}{8}$ fraction of the heavy hitters are recovered in L with constant probability. We can amplify the success

probability by repeating this procedure $\log(\frac{1}{\delta})$ times and constructing a set containing only those elements that were recovered in more than half of the attempts. Using Markov's inequality, we get a set L of cardinality $\leq 32k$ that contains more than $\frac{3}{4}$ fraction of the heavy hitters with probability $1 - \delta$.

Estimate elements in a round: We can then perform Count-Sketch with $O(\log(\frac{128}{\delta}))$ tables of size $O(16k)$. This sketch gives us an estimate for all the elements identified in L. Let $v_L \in \mathbb{R}^n$ be a vector of these estimates. After pruning these estimates off, we have a residual vector $x' \in \mathbb{R}^n$ given by

$$x' \leftarrow x - v_L.$$

The Count-Sketch with $O(\log(\frac{128}{\delta}))$ tables mis-estimates $\frac{1}{128}$ fraction of the identified elements with probability $1 - \delta$. So, the number of mis-estimated elements is at most $32k \cdot \frac{1}{128} = \frac{k}{4}$. The mis-estimation by the Count-Sketch might estimate a non-heavy hitter as being heavy and when we peel it off, that index might become heavy in x'. So, the total number of heavy hitters in the residual for the following round is at most $\frac{k}{2}$. So, in the following round the top $k' = \frac{k}{2}$ heaviest indices, of x' contains the unrecovered heavy hitters in x'.

Keeping track of the heavy hitters: In the next round, we perform a similar procedure to prune out the heavy hitters that we haven't recovered. If we perform the aforementioned identification and estimation procedures on x' with parameters $\frac{k}{2}$ and the same value of C, we risk dropping out elements that are heavy hitters in x (similar to [GLPS10] and [PW12]). We would only

be recovering indices i such that $\|x_i\|_2^2 \geq \frac{2C^2}{k} \operatorname{Err}^2(x,k)$ and might not recover indices i' whose weight is $\|x_{i'}\|_2^2 \geq \frac{C^2}{k} \operatorname{Err}^2(x,k)$.

In order to get around this, we use a different SNR parameter $C' = \sqrt{C}$ and sparsity parameter $\frac{k}{2}$ in the next round. As a result, in the second round we will find almost all coordinates larger than

$$\begin{aligned} \frac{2(C')^2}{k} \operatorname{Err}^2(x', \frac{k}{2}) &= \frac{2C'}{k} \operatorname{Err}^2(x', \frac{k}{2}) \\ &\leq \frac{C^2}{k} \operatorname{Err}^2(x,k), \end{aligned} \tag{3.3}$$

which includes the original heavy hitters.

More formally, in round r we choose our SNR parameter C_r and the sparsity parameter k_r such that $\frac{C^2}{k} \operatorname{Err}^2(x,k) \geq \frac{C_r}{k_r} \operatorname{Err}^2(x_r, k_r)$ where x_r is the pruned vector in round r. By doing this we ensure that the set of (k,C)-heavy hitters in x that have yet to be pruned out are also (k_r, C_r)-heavy hitters in x_r.

Total Number of Measurements and Total Error: We carefully set parameters δ_r, k_r and C_r such that the number of measurements performed in round r is geometrically decreasing in r. So, the total number of measurements in the first phase is $O(k \log_C(\frac{n}{k}) \log(\frac{1}{\delta}))$. We also ensure that failure probability can decreases such that $\sum_r \delta_r \leq \delta$.

At the end of round $\log \log C$, we will have peeled off all but $\frac{k}{\log C}$ heavy hitters. We perform the $(\frac{k}{\log C}, O(1))$ approximate ℓ_∞/ℓ_2 sparse recovery algo-

rithm of [LNNT16] to peel off remaining heavy hitters with $O(k \log_C(n) \log(\frac{1}{\delta}))$ measurements.

Comparison to previous work. The above approach is very similar to [GLPS10] and [PW12], but with different settings of parameters. The two differences are (I) because $C \gg 1$, we can iterate $C \to \sqrt{C}$ as in (3.3) and still find all heavy hitters while $\prod_r C_r$ remains bounded; and (II) once the residual is only $\frac{k}{\log C}$–sparse, we clean up with a difference $O(1)$-approximate algorithm. These differences also allow us to improve the analysis to get the ℓ_∞/ℓ_2 guarantee.

3.3 Proof of Our Upper Bounds

Recall, that we define the error of $x \in \mathbb{R}^n$ with respect to k-sparse vectors in $\mathbb{R}^n$ as

$$\mathrm{Err}^2(x,k) = \min_{k\text{-sparse } x'} \|x - x'\|_2^2$$

and the heavy hitter indices in a vector $x \in \mathbb{R}^n$ as:

$$S = \Big\{ i \in [n] \mid |x_i|^2 \geq \frac{C^2}{k} \mathrm{Err}^2(x,k) \Big\}.$$

The core of our algorithm is the following $(1, C)$ approximate sparse recovery algorithm. Given $x \in \mathbb{R}^n$, the indices $[n]$ are hashed using a pairwise independent hash function, $h : [n] \to [C]$, into C buckets. If x has a heavy hitter, due to the high SNR, we will be able to recover the hash location of the heavy hitter. This gives us $\log C$ bits of information about the index of the heavy

hitter. We need to learn $\log n$ bits of information about the heavy hitter to learn the index exactly. So, we can learn the exact index of the heavy hitter using $O(\log_C(\frac{n}{k}))$ linear measurements.

procedure IDENTIFYSINGLE(x)
  $r \leftarrow \log_C(n) + \log(\frac{1}{\delta})$
  for $i \in [r]$ **do**
    Pick a pairwise independent hash functions $h_i : [n] \to [C]$ and $s_i : [n] \to \{\pm 1\}$
    Measurement 1: $y_{2i} \leftarrow \sum_{j \in [n]} x_j \cdot s_i(j) \cdot h_i(j)$
    Measurement 2 :$y_{2i+1} \leftarrow \sum_{j \in [n]} x_j \cdot s_i(j)$
    $\alpha_i \leftarrow \mathsf{Round}(\frac{y_{2i}}{y_{2i+1}})$ for $i \in [r]$
  end for
  $c_j \leftarrow |\{i \in [r] \mid h_i(j) = \alpha_i\}|$ for $j \in [n]$
  $S \leftarrow \{j \in [n] \mid c_j \geq \frac{5r}{8}\}$
  if $|S| = 1$ **then**
    return $j \in S$
  else
    return $\perp$
  end if
end procedure

Algorithm 3.3.1: 1-sparse identification: In round i, hash items down to C buckets and recover the identity of the buckets using measurements y_{2i} and y_{2i+1}. Select the element that whose hash value has been the most over r rounds

We then use an algorithm that recovers *most* coordinates, using the desired number of measurements.

procedure IDENTIFYMOST(x, k, δ)
  $r \leftarrow \log(\frac{1}{\delta})$
  for $r \leftarrow [R]$ **do**
    Pick pairwise independent hash function $h : [n] \to [16k]$
    $L_r \leftarrow \{\mathsf{IdentifySingle}(x_{h^{-1}(i)}) \mid i \in [16k]\}$
  end for
  $c_j \leftarrow |\{r \mid j \in L_r\}|$ for $j \in [n]$
  $L \leftarrow \{j \in [n] \mid c_j \geq \frac{R}{2}\}$
  return L
end procedure

Algorithm 3.3.2: Identify most heavy coordinates. In each round r, hash indices $[n]$ into $[16k]$ buckets. Identify a single coordinate if it is heavy within that bucket. Output a list of indices that have been identified in more than $\frac{5r}{8}$ rounds.

The estimation algorithm EstimateMost runs Count-Sketch with $\log(\frac{1}{\delta})$ hash-tables of size $O(\frac{k}{\epsilon})$ where $\epsilon = \frac{1}{2}$.

Lemma 3.3. *With $O(k \log_C(\frac{n}{k}) \log(\frac{1}{\delta}))$ measurements, IdentifyMost returns a set L of size $32k$ such that each $j \in S$ has $j \in L$ with probability $1 - \delta$.*

Lemma 3.4. *(Lemma 10 from [PW12]) The result $\hat{x}_L$ of IdentifyMost followed by EstimateMost satisfies*

$$\mathrm{Err}^2(x - \hat{x}_L, fk) \leq C^2 \mathrm{Err}^2(x, k)$$

with probability $1 - \delta$ and uses $O(k \log_C(\frac{n}{k}) \log(\frac{1}{f\delta}) + k \log(\frac{1}{f\delta}))$ measurements.

These lemmas were proven in [PW12]. With $O(k \log_C(\frac{n}{k}) \log(\frac{64}{f\delta}))$ measurements, IdentifyMost recovers a set of elements L of cardinality $32k$ such that all but $\frac{fk}{2}$ elements of S are contained in L. Furthermore, performing

Count-Sketch with $O(k \log(\frac{128}{f\delta}))$ measurements yields estimates for all but $\frac{fk}{4}$ elements to within $\epsilon \operatorname{Err}^2(x, k)$. Lemma 3.4 follows by this bound on the total weight of these elements.

Observe that the total number of elements that are either mis-estimated and become heavy hitters or were un-recovered is bounded by fk. We use this crucial observation in our proof of Lemma 3.6.

We show that a more careful analysis and choice of parameters can yield two improvements over [PW12]: getting the optimal sample complexity, and getting an ℓ_∞/ℓ_2 bound. We carefully tune the relevant parameters — the sparsity(k_r), failure probability(δ_r), and approximation ratio(C_r) — so that the total failure probability, final approximation ratio, and total number of measurements are bounded as desired.

procedure RECOVERALL(y, δ)
  $k_0 \leftarrow k, \delta_0 \leftarrow \frac{\delta}{16}, \hat{x}^{(1)} \leftarrow 0, C_0 = C^{2^{-1}}, R = \log\log C$
  for $r \leftarrow [R]$ **do**
    $y' \leftarrow y^{(r)} - A^{(r)}\hat{x}^{(r)}$
    $L^{(r)} \leftarrow \mathsf{IdentifyMost}(y', k_r, \delta_r)$
    $\hat{v}^{(r)} \leftarrow \mathsf{EstimateMost}(y', k_r, \delta_r, L^{(r)})$
    $\hat{x}^{(r+1)} \leftarrow \hat{x}^{(r)} + \hat{v}^{(r)}$
    $\delta_{r+1} \leftarrow \delta_0 \cdot 2^{-r+1}$, $f_{r+1} \leftarrow \frac{1}{16 \cdot 8^r}$, $C_{r+1} \leftarrow C^{2^{-(r+1)}}$, $k_{r+1} \leftarrow f_{r+1}k$.
  end for
  $y' \leftarrow y^{(R+1)} - A^{(R+1)}\hat{x}^{(R+1)}$
  $\hat{v}^{(R+1)} \leftarrow \mathsf{ExpanderSketch}(y', k_{R+1}, \delta_0)$
  $\hat{x}_{out} \leftarrow \hat{x}^{(R+1)} + \hat{v}^{(R+1)}$
  return $\hat{x}_{out}$
end procedure

Algorithm 3.3.3: Identify all heavy coordinates. Each of the first $\log\log C$ rounds identifies a large number of the heavy hitters and peels them off until there are only $\frac{k}{\log C}$ heavy hitters remaining. ExpanderSketch[LNNT16] is used to peel off the rest.

Matrices $A^{(r)}$ for $r = 0 \ldots, R$, are the measurement matrices chosen by IdentifyMost and EstimateMost and the matrix $A^{(R+1)}$ is the measurement matrix of ExpanderSketch[2]. The algorithm ExpanderSketch is the recovery algorithm of [LNNT16] which achieves $(k, O(1))$ ℓ_∞/ℓ_2 with probability $1 - \delta$ using $O(k \log(n) \log(\frac{1}{\delta}))$ linear measurements. We will see later that this cleanup round requires $O(k \log_C(n) \log(\frac{1}{\delta}))$ measurements and does not match the lower bound of [PW12]. However, for an ℓ_2/ℓ_2 sparse recovery guarantee, the final cleanup round need not achieve an ℓ_∞/ℓ_2 guarantee and we may use the recovery algorithm of [GLPS10] instead of ExpanderSketch to achieve a

[2]Note: While we describe the algorithm in an iterative fashion over rounds, the actual measurement matrices are chosen non-adaptively

tight upper bound of $O(k \log_C(\frac{n}{k}) \log(\frac{1}{\delta}))$.

First, we show that the number of measurements made by our algorithm is $O(k \log_C(n))$.

Lemma 3.5. *The RecoverAll algorithm uses $O(k \log_C(n) \log(\frac{1}{\delta}))$ linear measurements.*

Proof. The number of measurements made in round r is $k_r \log_{C_r}(\frac{n}{k_r}) \log(\frac{16}{\delta_r})$. So, the total number of measurements made in rounds $0, \ldots, R$ is:

$$\begin{aligned}
&\sum_{r=0}^{R} k_r \log_{C_r}(\frac{n}{k_r}) \log(\frac{16}{\delta_r}) \\
&= \sum_{r=0}^{R} \frac{k}{16 \cdot 8^r} \cdot \log_C(\frac{n}{k_r}) 2^r \cdot \log(\frac{1}{\delta}) + \sum_{r=0}^{R} \frac{k}{16 \cdot 4^r} \cdot \log_C(\frac{n}{k}) 2^r \cdot (r+5) \\
&\leq k \log_C(\frac{n}{k}) \log(\frac{1}{\delta}) \sum_{r=0}^{R} \frac{r+5}{16 \cdot 2^r} \\
&= O(k \log_C(\frac{n}{k}) \log(\frac{1}{\delta})).
\end{aligned}$$

The number of measurements in the final round is:

$$\begin{aligned}
k_{R+1} \log(n) \log(\frac{1}{\delta_0}) &= \frac{k}{\log C} \log(n) \log(\frac{1}{\delta}) \\
&= O(k \log_C(n) \log(\frac{1}{\delta})).
\end{aligned}$$

So, the total number of measurements is $O(k \log_C(n) \log(\frac{1}{\delta}))$. □

In order to get an ℓ_∞/ℓ_2 guarantee, we keep track of the heavy hitters and show that all of them will have been peeled off by the final round.

Lemma 3.6. *RecoverAll achieves the C-approximate ℓ_∞/ℓ_2 guarantee with probability $1-\delta$.*

Proof. Our proof keeps track of the (k, C)-heavy hitters in round r i.e. :

$$\hat{S}_r = \Big\{ j \in [n] \mid (x - x^{(r)})_j^2 \geq \frac{C^2}{k} \operatorname{Err}^2(x, k) \Big\}.$$

In each round $r \in [R]$, we recover all but $\frac{1}{8}$ fraction of the (k_r, C_r) 'local' heavy hitters in that round. The set of (k_r, C_r) heavy hitters in round r is defined as:

$$S_r = \Big\{ j \in [n] \mid (x - x^{(r)})_j^2 \geq \frac{C_r^2}{k_r} \operatorname{Err}^2(x - x^{(r)}, k) \Big\}.$$

So, it suffices to show that $\hat{S}_r \subseteq S_r$ for each $r \in [R+1]$. Thereafter, all elements in $\hat{S}_{R+1}$ will be recovered by ExpanderSketch. First, observe that in particular round r, Lemma 3.4 telescopes and gives us

$$\operatorname{Err}^2(x - \hat{x}^{(r)}, k_r) \leq C^{\sum_{j=1}^{r+1} 2^{-j}} \operatorname{Err}^2(x, k).$$

Using Corollary 3.3, we know that the number of (k_r, C_r) heavy hitters that are not recovered is at most $\frac{f_r}{16} k_r$ with probability $1 - \frac{\delta_r}{2}$. The number of elements that are mis-estimated by EstimateMost is at most $\frac{f_r}{16} k_r$ with probability $1 - \frac{\delta_r}{2}$. All other elements have ℓ_2 weight $\leq \frac{C_r^2}{k_r} \operatorname{Err}^2(x - \hat{x}^{(r)}, k_r)$. So, any element in $\hat{S}_{r+1}$ will be in the top $k_{r+1} = \frac{1}{8} k_r$ coordinates of $(x - \hat{x}^{(r)})$. These elements

are also in S_{r+1} because:

$$\begin{aligned}\frac{C_{r+1}}{k_{r+1}}\operatorname{Err}^2(x-\hat{x}^{(r)},k_{r+1}) &\leq \frac{C_{r+1}}{k_{r+1}}\operatorname{Err}^2(x,k)\prod_{i=1}^{r}C_i \\ &\leq \frac{1}{kf_{r+1}}\operatorname{Err}^2(x,k)\prod_{i=1}^{r+1}C^{-2^{i+1}} \\ &\leq \frac{C^{\frac{1}{2}}(\log C)^3}{k}\cdot\operatorname{Err}^2(x,k) \\ &\leq \frac{C^2}{k}\operatorname{Err}^2(x,k)\end{aligned}$$

where the second inequality follows from the definitions of the quantities C_r, k_r, f_r and the third inequality follows by observing that $R \leq \log\log C$ implies that $\frac{1}{f_r} < (\log C)^3$. Further, $\prod_{r=1}^{i} C^{-2^{r+1}} \leq C^{\frac{1}{2}}$.

So, in each round $r \in [R+1]$, we have $\hat{S}_r \subseteq S_r$. Consequently, $\hat{S}_{R+1}$ has at most k_{R+1} elements and all of them are $(k_{R+1}, O(1))$ heavy. So, they will be identified and recovered in the final clean-up round with probability $1-\delta'$. The failure probability of the entire procedure can now be bounded by the probability that elements of $\hat{S}_r$ are not (k_r, C_r) heavy for some round $r \in [R]$ or ExpanderSketch fails:

$$\begin{aligned}&\Pr[\textsf{RecoverAll fails}] \\ &\leq \Pr[\exists i \in [R], \textsf{RecoverAll fails in round } i] + \Pr[\textsf{ExpanderSketch fails}] \\ &\leq \sum_{i=0}^{R} 2\delta_i + \delta_0 \\ &\leq 2\delta_0 \leq \delta.\end{aligned}$$

So, RecoverAll achieves (k, C)-approximate ℓ_∞/ℓ_2 with probability $1-\delta$. $\square$

The proof of Theorem 3.2 follows from Lemma 3.6 and Lemma 3.5. Also, observe that for the first R rounds, the computation per measurement is $O(1)$ and hence the time complexity for those rounds is $O(k \log_C(\frac{n}{k}) \log(\frac{1}{\delta}))$ and the time complexity for the final round is $O(k\text{polylog}(n, \frac{1}{\delta}))$. So, the time complexity of RecoverAll is $O(k\text{polylog}(n, \frac{1}{\delta}))$.

We also claimed in Theorem 3.1 and in the preceding section that the same algorithm using [GLPS10] in the cleanup round achieves ℓ_2/ℓ_2 guarantee using $O(k \log(\frac{n}{k}) \log(\frac{1}{\delta}))$ linear measurements.

Proof of Theorem 3.1. Observe that in Lemma 3.6 after the first R rounds fail to recover at most $k_{R+1} = \frac{k}{\log C}$ elements which are $(k_{R+1}, O(1))$ heavy in $x - \hat{x}^{(r)}$. At this stage we can replace the final round that applies ExpanderSketch in Algorithm 1 with the ℓ_2/ℓ_2 recovery algorithm of [GLPS10] to obtain an $(\frac{k}{\log C}, O(1))$ approximate ℓ_2/ℓ_2-recovery of $x - \hat{x}_{R+1}$. So, this implies a (k, C) approximate ℓ_2/ℓ_2-recovery of x. The measurement complexity in the cleanup round will be

$$O(k_{R+1} \log(\frac{n}{k_{R+1}}) \log(\frac{1}{\delta})) = O(k \log_C(\frac{n}{k}) \log(\frac{1}{\delta}))$$

and from the proof of Lemma 3.5, the first R rounds use $O(k \log_C(\frac{n}{k}) \log(\frac{1}{\delta}))$ measurements. So, the total measurement complexity is $O(k \log_C(\frac{n}{k}) \log(\frac{1}{\delta}))$.

□

Chapter 4

Deterministic L2 Heavy Hitters in the Insertion-Only Model

In this chapter, we study the ℓ_2-heavy hitters problem[1]. This problem is closely related to the sparse recovery problem and we use communication complexity techniques here to prove lower bounds.

4.1 Introduction to Streaming Algorithms and Heavy Hitters

A data stream is a sequence of data that is too large to be stored in memory. Some examples of this include message or event logs on the internet, sensor data from autonomous vehicles and scientific streams (in genomics and astronomy). In such scenarios, the goal is to compute a function of the data without having to store the data on disk. This has led to the study of the streaming model of computation.

In the streaming model, we parse a sequence of elements $a_1, \ldots, a_m$ where each element is drawn from a universe $\mathcal{U} = [n]$. The goal is to compute a function of this data by parsing this sequence a few times.

[1]The results presented in this chapter appear in [KPW20].

Clearly, one could parse the sequence and store it on disk and compute the function. This, however, is not practically feasible. For example, when parsing logs of internet traffic, we might need to parse petabytes of data on a machine that has a few gigabytes of memory. The main goal in the streaming model is to minimize the space complexity of our algorithms.

One of the most fundamental problems in data streams is the heavy hitters problem (also referred to as frequent elements or top-k elements). The goal here is to find a list of size at most k that describes the elements that occurred with high frequency in the stream. In this chapter, we will restrict ourselves to the problem of ℓ_2-heavy hitters which is defined as follows:

Definition 4.1.1. Given a stream of data $a_1, \ldots, a_m \in [n]$, let $f \in \mathbb{R}^n$ be the frequency vector where f_i denotes the number of occurrences of i in the stream. Then, $i \in [n]$ is an ϵ-ℓ_2-heavy hitter if:

$$|f_i| \geq \epsilon \|f\|_2 .$$

Note the similarity between this guarantee and the ℓ_∞/ℓ_2 guarantee. The sparse recovery algorithms discussed in Chapters 2 and 3 may be used in streaming algorithms as well. For algorithms that are allowed only 1 pass over the stream, we may use non-adaptive sparse recovery algorithms and for multi-pass algorithms, adaptive sparse recovery algorithms may be used.

A common approach to solving streaming problems is to use a linear sketch. In the case of the heavy hitters problem, the sketching algorithm stores

a linear sketch $A \cdot f$ of the frequency vector f. There is a rich literature on sketching algorithms to solve streaming problems.

It is known that any deterministic compressed sensing algorithm that achieves an ℓ_2/ℓ_2 guarantee must use $\Omega(n)$ linear measurements [CDD09]. In [KPW20], we showed that this lower bound holds even for the ℓ_2-heavy hitters problem in the insertion-only model.

The best known algorithm for the ℓ_2-heavy hitters problem in the insertion only model is the Misra-Gries algorithms[MG82]. This deterministic algorithm finds all the ϵ-ℓ_1-heavy hitters using $\frac{1}{\epsilon}$ counters. Since all ϵ-ℓ_2-heavy hitters are $\frac{\epsilon}{\sqrt{n}}$-$\ell_1$-heavy hitters, all ϵ-ℓ_2-heavy hitters can be recovered using $O(\frac{\sqrt{n}}{\epsilon})$ counters (or $O(\frac{\sqrt{n}}{\epsilon}\log m)$ bits).

The Misra-Gries algorithms[MG82] is a non-linear algorithm and hence the lower bound of [KPW20] does not apply. We show that a lower bound of $\Omega(\frac{\sqrt{n}}{\epsilon})$ bits applies for any algorithm that solves the ℓ_2-heavy hitters problem in the insertion only model.

Our Results: We show that this lower bound holds using a reduction from a communication complexity problem which we call *Mostly Set Disjointness* (or MostlyDISJ). This problem is a generalization of the multi-party Set Disjointness problem. Set Disjointness is a well-studied problem both in the two-party model [SK87] and more recently in the context of multi-party communication models[BEO$^+$13, BO15]. We prove lower bounds on the communication

complexity of Mostly Set Disjointness by using techniques which were first developed in [BJKS04].

Thereafter we describe a reduction from MostlyDISJ to the ℓ_2-heavy hitters problem and this reduction allows us to infer a lower bound on the space complexity of streaming algorithms for ℓ_2-heavy hitters.

The results proven in this chapter resemble the results in [KPW20] where we prove lower bounds on the communication complexity of Mostly Set Disjointness problem with δ-error. In this chapter we prove a lower bound when $\delta = 0$ using simpler techniques.

4.2 Communication Complexity Lower Bound

In this section, we prove lower bounds on the deterministic multi-party communication complexity of the Mostly Set Disjointness problem. We use techniques from communication complexity in order to prove these lower bounds.

4.2.1 Preliminaries

Information Theoretic Measures We use the following measures of distance between distributions in our proofs.

Definition 4.2.1. Let P and Q be probability distributions over the same countable universe $\mathcal{U}$. The total variation distance between P and Q is defined

as:

$$d_{\mathrm{TV}}(P,Q) = \frac{1}{2}\left\|P-Q\right\|_1$$

and the squared Hellinger distance between P and Q is defined as:

$$h^2(P,Q) = 1 - \sum_{x\in\mathcal{U}} \sqrt{P(x)\cdot Q(x)} = \frac{1}{2}\cdot\sum_{x\in\mathcal{U}}\left(\sqrt{P(x)}-\sqrt{Q(x)}\right).$$

In this chapter, we will sometimes abuse notation and consider distances between random variables instead of the underlying distributions.

Lemma 4.1. *For any two probability distributions P and Q, the Hellinger and total variation distances are related in the following manner:*

$$h^2(P,Q) \le d_{TV}(P,Q) \le h(P,Q)\cdot\sqrt{2-h^2(P,Q)} \le 1.$$

Multi-party Communication Model This model is a generalization of the more well-known notion of two-party communication. We consider t-ary functions $F : \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_t \to \mathcal{Z}$. There are t parties (or players) who receive inputs $X_1, \ldots, X_t$ which are jointly distributed according to some distribution η. In any protocol π, players speak in a particular order. The message of player i is a function of the messages of the previous players, his input and randomness i.e. $m_i = M_i(X_i, m_1, \ldots, m_{i-1}, R_i)$. The final player's message is the output of the protocol.

In this model of communication, every player's message is visible to every other player. This is more commonly known as the *blackboard model* of communication.

The communication cost of a multi-party protocol π is the sum of the lengths of the individual messages $\|\pi\| = \sum |M_j|$.

The deterministic communication complexity of the function f is the cost of the deterministic protocol of smallest communication cost that computes the function and is denoted by $D(f)$.

A protocol π is a δ-error protocol for the function f if for every input $x \in \mathcal{L}$, the output of the protocol equals $f(x)$ with probability $1 - \delta$.

The randomized communication complexity of f, denoted $R_\delta(f)$, is the cost of the cheapest randomized protocol that computes f correctly on every input with error at most δ over the randomness of the protocol.

The distributional communication complexity of the function f for error parameter δ is denoted as $D^\delta_\mu(f)$. This is the communication cost of the cheapest deterministic protocol which computes the function f with error at most δ under the input distribution μ.

By Yao's minimax theorem, $R_\delta(f) = \max_\mu D^\delta_\mu(f)$ and hence it suffices to prove a lower bound on the distributional communication complexity for a hard distribution μ.

In this chapter, instead of bounding the deterministic communication complexity, we bound the randomized communication complexity of protocols that do not err. Note that since every deterministic protocol for f is also a 0-error randomized protocol, $D(f) \geq R_0(f)$.

Conditional Information Complexity and Direct Sum Theorem Our lower bound on the randomized communication complexity will use the notion of conditional information complexity and the direct sum theorem of [BJKS04]. We define some of these terms here:

Definition 4.2.2. Let π be a randomized protocol whose inputs belong to $\mathcal{K} \subseteq \mathcal{X}_1 \times \mathcal{X}_2 \ldots \times \mathcal{X}_t$. Suppose $((X_1, X_2, \ldots, X_t), D) \sim \eta$ where η is a distribution over $\mathcal{K} \times \mathcal{D}$ for some set $\mathcal{D}$. The **conditional information cost** of π with respect to η is defined as:

$$cCost_\eta(\pi) = I(X_1, \ldots, X_t; \pi(X_1, \ldots, X_t) \mid D).$$

Definition 4.2.3. The δ-error **conditional information complexity** of f with respect to η, denoted $CIC_{n,\delta}(f)$ is defined as the minimum information cost of a δ-error protocol for f with respect to η.

Under these definitions the conditional information complexity of a function with respect to any valid distribution lower bounds the randomized communication complexity of that function. We may prove lower bounds on conditional information complexity with respect to a hard distribution in order to prove randomized communication complexity lower bounds.

Proposition 4.2 (Corollary 4.7 of [BJKS04])**.** *Let $f : \mathcal{K} \to \{0, 1\}$, and let η be a distribution over $\mathcal{K} \times \mathcal{D}$ for some set $\mathcal{D}$. Then, $R_\delta(f) \geq CIC_{\eta,\delta}(f)$.*

The Direct Sum Theorem allows us to reduce a t-player communication problem with n-dimensional input (to each player) to a t-player communication

problem with a 1-dimensional input. This theorem applies only when the function is decomposable and the input distribution is collapsing. We define both these notions here.

Definition 4.2.4. Suppose $\mathcal{K} \subseteq \mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_t$ and $\mathcal{K}_n \subseteq \mathcal{K}^n$. A function $f : \mathcal{K}_n \to \{0,1\}$ is **g-decomposable** with primitive $h : \mathcal{K} \to \{0,1\}$ if it can be written as:

$$f(X_1, \ldots, X_t) = g(h(X_{1,1}, \ldots, X_{1,t}), \ldots, h(X_{n,1}, \ldots, X_{n,t})).$$

for $g : \{0,1\}^n \to \{0,1\}$.

Definition 4.2.5. Suppose $\mathcal{K} \subseteq \mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_t$ and $\mathcal{K}_n \subseteq \mathcal{K}^n$. A distribution η over $\mathcal{K}_n$ is a **collapsing distribution** for $f : \mathcal{K}_n \to \{0,1\}$ with respect to $h : \mathcal{K} \to \{0,1\}$ if for all $Y_1, \ldots, Y_n$ in the support of η, for all $y \in \mathcal{K}$ and for all $i \in [n]$,

$$f(Y_1, \ldots, Y_{i-1}, y, Y_{i+1}, \ldots, Y_n) = h(y).$$

We state the Direct Sum Theorem for conditional information complexity below. The proof of this theorem in [BJKS04] applies to the blackboard model of multi-party communication. We state this in the most general form here.

Theorem 4.3 (Multi-party version of Theorem 5.6 of [BJKS04])**.** *Let $\mathcal{K} \subseteq \mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \mathcal{X}_t$ and let $\mathcal{K}_n \subseteq \mathcal{K}^n$. Suppose that the following conditions hold:*

(i) $f : \mathcal{K}_n \to \{0,1\}$ is a decomposable function with primitive $h : \mathcal{K} \to \{0,1\}$,

(ii) ζ *is a distribution over* $\mathcal{K} \times \mathcal{D}$*, such that for any* $d \in \mathcal{D}$ *the distribution* $(\zeta \mid D = d)$ *is a product distribution,*

(iii) $\eta = \zeta^n$ *is a product distribution over* $\mathcal{K}_n \times \mathcal{D}^n$*, and*

(iv) the marginal probability distribution of η *over* $\mathcal{K}_n$ *is a collapsing distribution for* f *with respect to* h*.*

Then,

$$CIC_{\eta,\delta}(f) \geq n \cdot CIC_{\zeta,\delta}(h).$$

The Mostly Set Disjointness Problem:

Definition 4.2.6. Denote by $\mathsf{MostlyDISJ}_{n,t}$, the multi-party Mostly Set Disjointness problem in which every player $j \in [t]$ receives an n-dimensional input vector $X_j = (X_{j,1}, \ldots, X_{j,n})$ where $X_{j,i} \in \{0,1\}$ and the input is guaranteed to fall into one of the following two cases:

- **NO:** For all $i \in [n]$, $\sum_{j \in [t]} X_{j,i} \leq 1$.
- **YES:** There exists a unique $i \in [n]$ such that $\sum_{j \in [t]} X_{j,i} = \frac{t}{2}$ and for all other $i' \neq i, \sum_j X_{j,i'} \leq 1$.

At the end of the protocol, the final player must output 1 if the input is in the YES case and 0 in the NO case.

Let $\mathcal{L} \subset \{0,1\}^t$ be the set of valid inputs along one index in $[n]$ for $\mathsf{MostlyDISJ}_{n,t}$ i.e. the set of elements in $x \in \{0,1\}^t$ with $\sum_{j \in [t]} x_j \leq 1$ or

$\sum_{j\in[t]} x_j = \frac{t}{2}$. Let $\mathcal{L}_n \subset \mathcal{L}^n$ denote the set of valid inputs to the $\mathsf{MostlyDISJ}_{n,t}$ function.

Observe that $\mathsf{MostlyDISJ}_{n,t} : \mathcal{L}_n \to \{0,1\}$ can be written as:

$$\mathsf{MostlyDISJ}_{n,t}(X_1, \ldots, X_t) = \bigvee_{i\in[n]} F_t(X_{1,i}, \ldots, X_{t,i})$$

for the function $F_t : \mathcal{L} \to \{0,1\}$ defined as:

$$F_t(x_1, \ldots, x_t) = \bigvee_{\substack{S\subseteq[t]\\|S|=\frac{t}{2}}} \bigwedge_{j\in S} x_j.$$

In particular, $\mathsf{MostlyDISJ}_{n,t}$ is OR-decomposable into n copies of F_t.

In order to prove a lower bound on the conditional information complexity, we need to define a "hard" distribution over the inputs to $\mathsf{MostlyDISJ}_{n,t}$. We define the distribution η over $\mathcal{L}_n \times \mathcal{D}^n$ where $\mathcal{D} = [t]$ as follows:

- For each $i \in [n]$ pick $D_i \in [t]$ uniformly at random and sample $X_{D_i,i}$ uniformly from $\{0,1\}$ and for all $j' \neq D_i$ set $X_{j',i} = 0$.
- Pick $I \in [n]$ uniformly at random and $Z \in \{0,1\}$.
- If $Z = 1$, pick a set $S \subseteq [t]$ such that $|S| = \frac{t}{2}$ uniformly at random and for all $j \in S$ set $X_{j,I} = 1$ and for all $j \notin S$, set $X_{j,I} = 0$.

Let μ_0 denote the distribution for each $i \in [n]$ conditioned on $Z = 0$. For any $d \in [t]$, when $D = d$, the conditional distribution $(\mu_0 \mid D = d)$ over $\mathcal{L}$ is the uniform distribution over $\{0, e_d\}$ and hence a product distribution. Let η_0 be the distribution η conditioned on the event that $Z = 0$. Clearly, $\eta_0 = \mu_0^n$.

This definition of $\mathsf{MostlyDISJ}_{n,t}$ and the hard distribution η_0 allows us to apply the Direct Sum theorem (Corollary 4.3). This will enable us to prove a lower bound on the conditional information complexity of the simpler single coordinate multi-party problem, F_t, and as a consequence obtain a lower bound on the conditional information complexity of $\mathsf{MostlyDISJ}_{n,t}$.

Corollary 4.4 (of Theorem 4.3)**.** *Consider* $\mathsf{MostlyDISJ}_{n,t}$ *with input distribution* η_0 *over* $\mathcal{L}_n \times \mathcal{D}^n$ *and* F_t *with input distribution* μ_0 *over* $\mathcal{L} \times \mathcal{D}$*. We have the direct sum relation between the respect conditional information complexities:*

$$CIC_{\eta_0,\delta}(\mathsf{MostlyDISJ}_{n,t}) \geq n \cdot CIC_{\mu_0,\delta}(F_t)$$

Proof. Observe that

(i) $\mathsf{MostlyDISJ}_{n,t}$ is OR-decomposable by F_t,

(ii) μ_0 is a distribution over $\mathcal{L} \times [t]$ such that the marginal distribution $(\mu_0 \mid D = d)$ over the $\mathcal{L}$ is uniform over $\{0, e_d\}$ (and hence a product distribution),

(iii) $\eta_0 = \mu_0^n$, and

(iv) since $\mathsf{MostlyDISJ}_{n,t}$ is OR-decomposable and η_0 has support only on inputs in the NO case, η_0 is a collapsing distribution for $\mathsf{MostlyDISJ}_{n,t}$ with respect to F_t.

We may apply Theorem 4.3 and conclude:

$$CIC_{\eta_0,\delta}(\mathsf{MostlyDISJ}_{n,t}) \geq n \cdot CIC_{\mu_0,\delta}(F_t)$$

□

4.2.2 Proof of Our Lower Bound

We prove a lower bound on the deterministic communication complexity of this problem:

Theorem 4.5.

$$D(\mathsf{MostlyDISJ}_{n,t}) = \Omega(n).$$

As we stated earlier, we prove a lower bound on the randomized communication complexity of computing $\mathsf{MostlyDISJ}_{n,t}$ with 0-error. Since $R_0(f) \leq D(f)$, this implies a lower bound on the deterministic communication complexity of $\mathsf{MostlyDISJ}_{n,t}$.

Using Proposition 4.2, we know that to prove $R_0(\mathsf{MostlyDISJ}_{n,t}) \geq \Omega(n)$ it is sufficient to prove that $CIC_{\eta_0,0}(\mathsf{MostlyDISJ}_{n,t}) \geq \Omega(n)$. Instead we prove that $CIC_{\mu_0,0}(F_t)$ and Theorem 4.5 follows by an application of the Direct Sum Theorem (Corollary 4.4).

In order to lower bound the $CIC_{\mu_0,0}(f)$, we need show a lower bound on $cCost_{\mu_0}(\pi)$ for every π that does not err. Using the connection between conditional mutual information and Hellinger distance established in [BJKS04] (Lemma 6.2) we know that:

$$cCost_{\mu_0}(\pi) = I(\pi(X_1, \ldots, X_t); X_1, \ldots, X_t \mid D) \geq \mathop{\mathbb{E}}_{i}[h^2(\pi_{e_i}, \pi_0)]$$

where π_x denotes the distribution of the transcript of the protocol π on input x.

Instead of bounding this expectation, we bound the sum of total-variation distance (denoted d_{TV}) of the same distributions.

Lemma 4.6. *Consider any n-player communication protocol π where each player i has input $X_i \in \{0,1\}$ and l messages are sent in the protocol, such that for any set S with $|S| = k$, $d_{TV}(\pi_{e_S}, \pi_0) = 1$, then:*

$$\sum_{i=1}^{n} d_{TV}(\pi_{e_i}, \pi_0) \geq n - k + 1.$$

Proof. We prove the theorem using induction on n, l and k.

Base Case: When $k = 1$ for any n, observe that $d_{\mathrm{TV}}(\pi_{e_i}, \pi_0) = 1$ for all $i \in [n]$ by supposition. So, $\sum_{i=1}^{n} d_{\mathrm{TV}}(\pi_{e_i}, \pi_0) = n - k + 1$.

Induction: Let $n = n'$, $k = k'$ and $l = l'$ and suppose n' players speak in the order $i_1, i_2, \cdots, i_{l'}$ and for any set S such that $|S| = k$, $d_{\mathrm{TV}}(\pi_{e_S}, \pi_0) = 1$. We say that a message m_i sent by player i is *ambiguous* if it could be sent when $X_i = 0$ or $X_i = 1$. Let 'E' denote the event that m_{i_1} is ambiguous and $p := \Pr[E \mid X_{i_1} = 0]$. We prove the claim under two cases:

Case 1: Player i_1 does not speak again.

Using the definition of p note that $d_{\mathrm{TV}}(\pi_{e_{i_1}}, \pi_0) \geq 1 - p$.

Suppose $X_{i_1} = 0$ and m_{i_1} is ambiguous. Note that $\pi' = \pi \mid m_{i_1}$ is a protocol in which for all $S' \subseteq [n] \setminus i_1$ such that $|S'| = k - 1$, we must have $d_{\mathrm{TV}}(\pi'_{e_{S'}}, \pi_0) = 1$ (else $d_{\mathrm{TV}}(\pi_{e_{S' \cup i_1}}, \pi_0) < 1$ and the lemma condition

is contradicted). We may now apply the induction hypothesis to π' with $n = n' - 1$, $k = k' - 1$ and $l = l' - 1$ and we get:

$$\sum_{\substack{i \in [n] \\ i \neq i_1}} d_{\mathrm{TV}}(\pi_{e_i}, \pi_0 \mid E) \geq (n' - 1) - (k' - 1) + 1 = n' - k' + 1.$$

When $X_{i_1} = 0$ and m_{i_1} is unambiguous, $\pi' = \pi \mid m_{i_1}$ is a protocol with $n = n' - 1$ players, $k = k'$ and $l = l' - 1$ where the lemma conditions hold. We may apply the induction hypothesis to obtain:

$$\sum_{\substack{i \in [n] \\ i \neq i_1}} d_{\mathrm{TV}}(\pi_{e_i}, \pi_0 \mid \overline{E}) \geq (n' - 1) - (k') + 1 = n' - k'.$$

Now we have

$$\begin{aligned}
\sum_{\substack{i \in [n] \\ i \neq i_1}} d_{\mathrm{TV}}(\pi_{e_i}, \pi_0) &= \sum_{\substack{i \in [n] \\ i \neq i_1}} \mathop{\mathbb{E}}_{m_{i_1}} d_{\mathrm{TV}}(\pi_{e_i}, \pi_0 \mid m_{i_1}) \\
&= \sum_{\substack{i \in [n] \\ i \neq i_1}} d_{\mathrm{TV}}(\pi_{e_i}, \pi_0 \mid E) \times \Pr[E] + \sum_{\substack{i \in [n] \\ i \neq i_1}} d_{\mathrm{TV}}(\pi_{e_i}, \pi_0 \mid \overline{E}) \times \Pr[\overline{E}] \\
&\geq p(n - k + 1) + (1 - p)(n - k) \\
&= n - k + p.
\end{aligned}$$

As desired the sum of the total variation distances is:

$$\begin{aligned}
\sum_{i \in [n]} d_{\mathrm{TV}}(\pi_{e_i}, \pi_0) &= d_{\mathrm{TV}}(\pi_{\{i_1\}}, \pi_0) + \sum_{\substack{i \in [n] \\ i \neq i_1}} d_{\mathrm{TV}}(\pi_{e_i}, \pi_0) \\
&\geq (1 - p) + (n - k + p) \\
&= n - k + 1.
\end{aligned}$$

Case 2: Player i_1 speaks again in the protocol.

If $X_{i_1} = 0$ and m_{i_1} is ambiguous, the induction hypothesis still holds for $\pi \mid m_{i_1}$ with $n = n'$, $k = k'$ and $l = l' - 1$. So, $\sum_{i=1}^{n} d_{\mathrm{TV}}(\pi_{e_i}, \pi_0 \mid E) \geq n' - k' + 1$. Suppose $X_{i_1} = 0$ and m_{i_1} is unambiguous. We define π' to be a protocol in which player i_2 simulates i_1 in the protocol $\pi \mid m_{i_1}$. Now, the induction hypothesis holds for π' with $n = n' - 1$, $k = k'$ and $l \leq l' - 1$. Hence

$$\begin{aligned}\sum_{i=1}^{n} d_{\mathrm{TV}}(\pi_{e_i}, \pi_0 \mid \overline{E}) &= d_{\mathrm{TV}}(\pi_{e_{i_1}}, \pi_0 \mid \overline{E}) + \sum_{\substack{i \in [n] \\ i \neq i_1}} d_{\mathrm{TV}}(\pi'_{e_i}, \pi'_0) \\ &\geq 1 + (n' - k').\end{aligned}$$

So we have $\sum_{i=1}^{n} d_{\mathrm{TV}}(\pi_{e_i}, \pi_0) \geq n' - k' + 1$ and we have proven the claim for all n, k, l using induction. □

We know that any protocol which computes F_t with 0 error must have $d_{\mathrm{TV}}(\pi_{e_S}, \pi_0) = 1$ for all $S \subseteq [t]$ such that $|S| = \frac{t}{2}$. Using Lemma 4.6:

$$\begin{aligned}\sum_{i=1}^{t} d_{\mathrm{TV}}(\pi_{e_i}, \pi_0) &\geq t - \frac{t}{2} + 1 \\ &= \frac{t}{2} + 1.\end{aligned}$$

Since $d_{\mathrm{TV}}(P, Q) \leq 1$, we know that:

$$|\{i \in [t] \mid d_{\mathrm{TV}}(\pi_{e_i}, \pi_0) \leq 1/4\}| \leq 2t/3.$$

From Lemma 4.1 we know that for all P, Q, $h(P, Q) \geq \frac{d_{\mathrm{TV}}(P,Q)}{\sqrt{2}}$. Hence

$$\left|\{i \in [t] \mid h(\pi_{e_i}, \pi_0) \geq \frac{d_{\mathrm{TV}}(\pi_{e_i}, \pi_0)}{\sqrt{2}} > \frac{1}{4\sqrt{2}}\}\right| > t/3.$$

Hence,

$$\sum_{i=1}^{t} h^2(\pi_{e_i}, \pi_0) > \frac{t}{3} \cdot \frac{1}{32} = \frac{t}{96}.$$

So, we get a lower bound on the conditional information cost of any protocol for F_t:

Corollary 4.7. *Suppose π is a t-player 0-error randomized protocol for F_t. Then,*

$$cCost_{\mu_0}(\pi) \geq \mathop{\mathbb{E}}_{i} \left[h^2(\pi_{e_i}, \pi_0)\right] = \Omega(1).$$

This corollary together with the Direct Sum theorem implies a lower bound on the deterministic communication complexity of MostlyDISJ.

Proof. (of Theorem 4.5)

$$\begin{aligned} D(\mathsf{MostlyDISJ}_{n,t}) &\geq R_0(\mathsf{MostlyDISJ}_{n,t}) \\ &\geq CIC_{\eta_0,0}(\mathsf{MostlyDISJ}_{n,t}) \\ &\geq n \cdot CIC_{\mu_0,0}(F_t) \\ &\geq \Omega(n) \end{aligned}$$

where the first equality uses the fact that all deterministic algorithms are also randomized, the second inequality uses Proposition 4.2, the third inequality uses Corollary 4.4 and the fourth inequality uses Corollary 4.7 and the definition of conditional information complexity. □

4.3 Reduction to L2 Heavy Hitters

Now we show that a lower bound for the ℓ_2-heavy hitters problem follows using reductions from the Mostly Disjointness problem and the communication lower bound.

Definition 4.3.1. In the ϵ-ℓ_2-heavy hitters problem, we are given $\epsilon \in (0, 1)$ and a stream of items $a_1, \ldots, a_m$ where $a_i \in [n]$. If f_i denotes the frequency of item i in the stream, the algorithm should output a list of all elements $j \subseteq [n]$ such that

$$f_j \geq \epsilon \|f\|_2 .$$

Theorem 4.8. *Given $\epsilon \in (\frac{1}{\sqrt{n}}, \frac{1}{2})$, any deterministic r-pass insertion-only streaming algorithm for ϵ-ℓ_2-heavy hitters must have space complexity of $\Omega(\frac{\sqrt{n}}{r\epsilon})$ bits.*

Proof. Let $\mathcal{A}$ be a deterministic r-pass streaming algorithm for ϵ-ℓ_2-heavy hitters in the insertion-only model. We describe a multi-party protocol to deterministically solve the Mostly Set Disjointness problem i.e. $\mathsf{MostlyDISJ}_{n,4\epsilon\sqrt{n}}$ that uses the $\mathcal{A}$. The players simulate a stream which updates a vector $x \in \mathbb{R}^{2n}$. Instead of starting with 0^{2n} (as is the case with most streaming algorithms), the protocol starts off with a frequency vector defined as follows.

$$f_0 = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} 0 \\ \vdots \\ 0 \end{matrix}}\right\} n \\ \left.\vphantom{\begin{matrix} 1 \\ \vdots \\ 1 \end{matrix}}\right\} n \end{matrix}$$

Each player performs an update $f \leftarrow f + \delta_i$ to the vector and passes the state of $\mathcal{A}$ to the next player. The update vector δ_i that is processed by player i is just their input x_i padded to length $2n$.

$$\delta = \begin{pmatrix} x_i \\ \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Observe that if the input to the players is a NO-instance of $\mathsf{MostlyDISJ}_{n,4\epsilon\sqrt{n}}$, then the final vector f' in the turnstile stream consists of 0-1 entries with at least n 1-s. Since $\|f'\|_2^2 \geq n$ and $\epsilon \geq \frac{1}{\sqrt{n}}$, no element is a ϵ-ℓ_2 heavy hitter.

If the input is a YES-instance, then the final vector, f', consists of $\leq 2n-1$ entries that are 1 and one entry at which is $2\epsilon\sqrt{n}$. Since $4\epsilon^2 n \geq \epsilon^2(2n+4\epsilon^2 n)$, that entry is a ϵ-heavy hitter. Using the lower bound of Theorem 4.5, we know that the total communication in the protocol is $\Omega(n)$. Since the number of messages sent over r rounds in the protocol is $r \cdot 4\epsilon\sqrt{n}$, there exists at least one player whose message in a given round is $\Omega(\frac{\sqrt{n}}{r\epsilon})$ bits and this is a lower bound on the space complexity of $\mathcal{A}$. $\square$

Chapter 5

Compressed Sensing with Generative Models

In this chapter, we study compressed sensing with a new notion of structure[1]. In compressed sensing, one would like to learn a structured signal $x \in \mathbb{R}^n$ from a limited number of linear measurements $y \approx Ax$. The unknown signals x being observed are structured or "compressible": although x lies in $\mathbb{R}^n$, it would take far fewer than n floating point numbers to describe x. In such a situation, one can hope to estimate x well from a number of linear measurements that is closer to the size of the *compressed representation* of x than to its ambient dimension n.

In order to do compressed sensing, you need a formal notion of how signals are expected to be structured. As we noticed in Chapters 2 and 3, the classic answer is to use *sparsity*. In sparse recovery, given linear measurements $y = Ax$ of an arbitrary vector $x \in \mathbb{R}^n$, one can hope to recover an estimate $\hat{x}$ of x satisfying

$$\|x - \hat{x}\| \leq C \min_{k\text{-sparse } x'} \|x - x'\| \tag{5.1}$$

for some constant C and norm $\|\cdot\|$. In this chapter, we focus on achieving a similar guarantee with $3/4$ probability. Thus, if x is well-approximated by

[1]The results presented in this chapter appeared in [KKP20].

a k-sparse vector x', it should be accurately recovered. Classic results such as [CRT06b] show that (5.1) is achievable when A consists of $m = O(k \log \frac{n}{k})$ independent Gaussian linear measurements. This bound is tight, and in fact no distribution of matrices with fewer rows can achieve this guarantee in either ℓ_1 or ℓ_2 [DIPW10].

Although compressed sensing has had success, sparsity is a limited notion of structure. Can we learn a richer model of signal structure from data, and use this to perform recovery? Generative models are one such form of structure that model the manifold of "natural images". Over the last decade neural networks based models like generative adversarial networks (GANs) [GPAM$^+$14] and variational autoencoders (VAEs) [KW14] been used successfully to produce generative models $G : \mathbb{R}^k \to \mathbb{R}^n$ that allow approximate sampling from the distribution of "natural images". So one obvious question is: can these models can be used as a form of structure for compressed sensing.

In [BJPD17] it was shown how to use generative models to achieve a guarantee analogous to (5.1): for any L-Lipschitz $G : \mathbb{R}^k \to \mathbb{R}^n$, one can achieve

$$\|x - \hat{x}\|_2 \leq C \min_{z' \in B_k^2(r)} \|x - G(z')\|_2 + \delta, \tag{5.2}$$

where $r, \delta > 0$ are parameters, $B_k^2(r)$ denotes the radius-r ℓ_2 ball in $\mathbb{R}^k$ and Lipschitzness is defined with respect to the ℓ_2-norms, using only $m = O(k + k \log \frac{Lr}{\delta})$ measurements. Thus, the recovered vector is almost as good as the nearest point in the *range of the generative model*, rather than in the set of

k-sparse vectors. We will refer to the problem of achieving the guarantee (5.2) as "generative-model recovery".

Our first theorem is that the [BJPD17] result is tight: for any setting of parameters n, k, L, r, δ, there exists an L-Lipschitz function $G : \mathbb{R}^k \to \mathbb{R}^n$ such that the [BJPD17] measurement bound is optimal for achieving (5.2):

Theorem 5.1. *Consider any n, k, L, r, δ. There exists an L-Lipschitz function $G^* : \mathbb{R}^k \to \mathbb{R}^n$ such that, if $\mathcal{A}$ is an algorithm which picks a matrix $A \in \mathbb{R}^{m \times n}$, and given Ax returns an $\hat{x}$ satisfying (5.2) with probability at least $3/4$, then $m = \Omega(\min(k + k \log(Lr/\delta), n))$.*

The same result holds if the ℓ_2 norms in (5.2) are replaced with ℓ_1 norms.

That our lower bound caps out at $m = \Theta(n)$ is of course necessary, since the problem is trivial for $m = n$; thus our bound is tight for the whole range of possible parameters. Notably, and in contrast to sparse recovery, the additive error δ is necessary for Lipschitz generative model recovery. One cannot achieve (5.2) with $\delta = 0$ and $m = o(n)$.

Our second result is to directly relate the two notions of structure: sparsity and generative models. We produce a simple ReLU-based neural network $G_{sp} : \mathbb{R}^{2k} \to \mathbb{R}^n$ whose range is precisely the set of all k-sparse vectors.

Theorem 5.2. *There exists a 2-hidden-layer ReLU-based neural network $G_{sp} : \mathbb{R}^{2k} \to \mathbb{R}^n$ with width $O(nk)$ such that $\mathrm{Im}(G) = \{x \mid \|x\|_0 \leq k\}$.*

This matches a second result of [BJPD17], which shows that for ReLU-based neural networks, one can avoid the additive δ term and achieve a different result from (5.2):

$$\|x - \hat{x}\|_2 \leq C \min_{z' \in \mathbb{R}^k} \|x - G(z')\|_2 \tag{5.3}$$

using $O(kd \log W)$ measurements, if d is the depth and W is the maximum number of activations per layer. Applying this result to our sparsity-producing network G_{sp} implies, with $O(k \log n)$ measurements, recovery achieving the standard sparsity guarantee (5.1). So the generative-model representation of structure really is more powerful than sparsity.

Connecting the results. Theorem 5.2 directly implies a weaker form of Theorem 5.1. The network G_{sp} produces all k-sparse binary vectors from seeds of radius $r = n\sqrt{k}$ and with $L = 2$. The standard sparse recovery lower bound shows that recovering these vectors for $\delta = \sqrt{k}$ requires $\Omega(k \log(\frac{n}{k}))$ measurements, which is $\Omega(k \log n)$ for $n > k^{1.1}$. Therefore we immediately see an $\Omega(k \log \frac{Lr}{\delta})$ bound for Lipschitz recovery for these parameters. The advantage of Theorem 5.1 over such an approach is that it applies to *all* values of L, r, and δ, rather than these polynomially-bounded ones; and indeed, such an approach would not show that the additive δ is necessary in (5.2).

Concurrent work. This chapter presents the results of [KKP20]. A concurrent paper [LS20] proves a very similar lower bound to our Theorem 5.1. However, the [LS20] result is weaker in an important way, analogous to the

implication from Theorem 5.2: it requires n to equal $\frac{Lr}{\delta}$, so the lower bound is equal to $\Theta(k \log n)$. As a result, it neither applies to superpolynomial L, nor does it imply that any dependence on δ is necessary.

Our result is also stronger than [LS20] in a couple other ways. Our bound applies to *non-uniform* algorithms where each matrix A only works for 3/4 of possible inputs x, rather than requiring A to work for all x, and our bound applies to the ℓ_1 as well as the ℓ_2 guarantee. The [LS20] approach likely can be extended to non-uniform algorithms, but extending their techniques to ℓ_1 seems quite challenging. Even in the standard sparse-recovery setting, our communication-complexity–based techniques extend to the ℓ_1 guarantee, while (to our knowledge) the information-theory techniques used in [LS20] do not.

5.1 Overview of Our Results

As described above, this section contains two results: a tight lower bound for compressed sensing relative to a Lipschitz generative model, and an $O(1)$-layer generative model whose range contains all sparse vectors. The techniques are independent, and are outlined below.

5.1.1 Lower Bound for Compressed Sensing with Generative Models

Over the last decade, lower bounds for sparse recovery have been studied extensively. The techniques in this chapter are most closely related to the techniques used in [DIPW10].

Similar to [DIPW10], our proof is based on communication complexity. We will exhibit an L-Lipschitz function G and a large finite set $Z \subset \mathrm{Im}(G) \subset B_n^p(R)$ of points that are well-separated. Then, given a point x that is picked uniformly at random from Z, we show how to identify it from Ax using the generative model recovery algorithm. This implies Ax also contains a lot of information, so m must be fairly large.

Formally, we produce a generative model whose range includes a large, well-separated set:

Theorem 5.3. *Given $R > 0$ satisfying $R > 2Lr$, $p \in \{1, 2\}$, there exists an $O(L)-$Lipschitz function $G : \mathbb{R}^k \to \mathbb{R}^n$, and $X \subseteq B_k^p(r)$ such that*

(1) for all $x \in X$, $G(x) \in \{\pm \frac{R}{n^{\frac{1}{p}}}\}^n$

(2) hence for all $x \in X$, $\|G(x)\|_p = R$

(3) for all $x, y \in X$, $\|G(x) - G(y)\|_p \geq \frac{R}{6^{\frac{1}{p}}}$

(4) $\log(|X|) = \Omega\left(\min(k \log(\frac{Lr}{R})), n\right)$

Now, suppose we have an algorithm that can perform generative model recovery with respect to G from Theorem 5.3, with approximation factor C, and error $\delta < R/24$ within the radius r ball in k-dimensions. Set $t = \Theta(\log n)$, and for any $z_1, z_2, \ldots, z_t \in Z = G(X)$ take

$$z = \epsilon^t z_1 + \epsilon^{t-1} z_2 + \epsilon^{t-1} z_3 + \ldots + z_t$$

for $\epsilon = \frac{1}{48(C+1)}$ a small constant. The idea of the proof is the following: given $y = Az$, we can recover $\widehat{z}$ such that

$$\begin{aligned}\|\widehat{z} - z_t\| &\leq \|z - z_t\| + \|\widehat{z} - z\| + \delta \\ &\leq (C+1)\|z - z_t\| + \delta \\ &\leq (C+1)\frac{\epsilon R}{1-\epsilon} + \delta \\ &< R/24 + R/24 = R/12,\end{aligned}$$

where the first inequality comes from the generative model recovery guarantee for z_t when treating $z - z_t$ as noise. Now, because Z has minimum distance $\frac{R}{6^{1/p}}$, we can exactly recover z_t by rounding $\widehat{z}$ to the nearest element of Z. But then we can repeat the process on $(Az - Az_t)$ to find z_{t-1}, then z_{t-2}, up to z_1, and learn $t \lg |Z| = \Omega(tk \log(\frac{Lr}{R}))$ bits total. Thus Az must contain this many bits of information; but if the entries of A are rational numbers with poly(n) bounded numerators and (the same) poly(n) bounded denominator, then each entry of Az can be described in $O(t + \log n)$ bits, so

$$m \cdot O(t + \log n) \geq \Omega(tk \log(\frac{Lr}{R}))$$

or $m \geq \Omega(k \log(\frac{Lr}{R}))$.

There are two issues that make the above outline not totally satisfactory, which we only briefly address how to resolve here. First, the theorem statement makes no supposition on the entries of A being polynomially bounded. To resolve this, we perturb z with a tiny (polynomially small) amount of additive Gaussian noise, after which discretizing Az at an even tinier

(but still polynomial) precision has negligible effect on the failure probability. The second issue is that the above outline requires the algorithm to recover all t vectors, so it only applies if the algorithm succeeds with $1 - \frac{1}{t}$ probability rather than constant probability. This is resolved by using a reduction from the *augmented indexing* problem, which is a one-way communication problem where Alice has $z_1, z_2, \ldots, z_t \in Z$, Bob has $i \in [t]$ and $z_{i+1}, \cdots, z_n$, and Alice must send Bob a message so that Bob can output z_i with $2/3$ probability. This still requires $\Omega(t \log |Z|)$ bits of communication, and can be solved in $O(m(t + \log n))$ bits of communication by sending Az as above.

Constructing the set. The above lower bound approach, relies on finding a large, well-separated set $Z = G(X)$ as in Theorem 5.3.

We construct this set in two stages. First, we consider the $k = 1$ case, producing a Lipschitz map from $\mathbb{R}$ to $\mathbb{R}^n$ with $\frac{Lr}{R}$ points of appropriate distance. We do this by linearly interpolating between elements of a high-distance code over $\{\pm\frac{R}{n^{1/p}}\}^n$; because codewords are $\Theta(R)$ apart, an L-Lipschitz function from $[-r, r]$ can reach $\frac{Lr}{R}$ such elements (as long as this is less than the $2^{\Omega(n)}$ total number of codewords).

To extend this construction to a mapping from $\mathbb{R}^k$ to $\mathbb{R}^n$, we take the product distribution of k such functions, each run with $n' = n/k$. This results in a Lipschitz generative model with the desired radius and number of elements; unfortunately, the minimum distance would be too small. We fix this by concatenating the code: we use an error correcting code over $[n/k]^k$ to

choose a subset of these points that is still large enough but has the desired distance.

5.1.2 A Sparsity-Producing Generative Model

For our second result, to produce a generative model whose range consists of all k-sparse vectors, we start by mapping $\mathbb{R}^2$ to the set of positive 1-sparse vectors. For any pair of angles θ_1, θ_2, we can use a constant number of unbiased ReLUs to produce a neuron that is only active at points whose representation (r, θ) in polar coordinates has $\theta \in (\theta_1, \theta_2)$. Moreover, because unbiased ReLUs behave linearly, the activation can be made an arbitrary positive real by scaling r appropriately. By applying this n times in parallel, we can produce n neurons with disjoint activation ranges, making a network $\mathbb{R}^2 \to \mathbb{R}^n$ whose range contains all 1-sparse vectors with nonnegative coordinates.

By doing this k times and adding up the results, we produce a network $\mathbb{R}^{2k} \to \mathbb{R}^n$ whose range contains all k-sparse vectors with nonnegative coordinates. To support negative coordinates, we just extend the $k = 1$ solution to have two ranges within which it is non-zero: for one range of θ the output is positive, and for another the output is negative. This results in Theorem 5.2.

5.2 Proof of Our Lower Bound

In this section, we prove a lower bound for the sample complexity of generative model recovery by a reduction from a communication game. We show that the communication game can be won by sending a vector Ax and

then performing generative model recovery. A lower bound on the communication complexity of the game implies a lower bound on the number of bits used to represent Ax if Ax is discretized. We can then use this to lower bound the number of measurements in A.

Since we are dealing in bits in the communication game and the entries of a sparse recovery matrix can be arbitrary reals, we will need to discretize each measurement. We show first that discretizing the measurement matrix by rounding does not change the resulting measurement too much and will allow for our reduction to proceed.

Notation. We use $B_k^p(r) = \{x \in \mathbb{R}^k \mid \|x\|_p \leq r\}$ to denote the k-dimensional ℓ_p ball of radius r. Given a function $g : \mathbb{R}^a \to \mathbb{R}^b$, $g^{\otimes k} : \mathbb{R}^{ak} \to \mathbb{R}^{bk}$ denotes a function that the maps a point $(x_1, \ldots, x_{ak})$ to $(g(x_1, \ldots, x_a), g(x_{a+1}, \ldots, x_{2a}),$ $\ldots, g(x_{a(k-1)+1}, \ldots, x_{ak}))$. For any function $G : A \to B$, we use $\mathrm{Im}(G)$ to denote $\{G(x) \mid x \in A\}$.

Matrix conditioning. We first show that, without loss of generality, we may assume that the measurement matrix A is well-conditioned. In particular, we may assume that the rows of A are orthonormal.

We can multiply A on the left by any invertible matrix to get another measurement matrix with the same recovery characteristics. If we consider the singular value decomposition $A = U\Sigma V^*$, where U and V are orthonormal and Σ is 0 off the diagonal, this means that we can eliminate U and make the

entries of Σ be either 0 or 1. The result is a matrix consisting of m orthonormal rows.

Discretization. For well-conditioned matrices, we use the following lemma (similar to one from [DIPW10]) to show that we can discretize the entries without changing the behavior by much:

Lemma 5.4. *Let $A \in \mathbb{R}^{m \times n}$ be a matrix with orthonormal rows. Let A' be the result of rounding A to b bits per entry. Then for any $v \in \mathbb{R}^n$ there exists an $s \in \mathbb{R}^n$ with $A'v = A(v - s)$ and $\|s\|_p < n^2 2^{-b} \|v\|_p$ for $p \in \{1, 2\}$.*

Proof. Let $A'' = A - A'$ be the error when discretizing A to b bits, so each entry of A'' is less than 2^{-b}. Then for any v and $s = A^T A'' v$, we have $As = A''v$. For $p = 2$, we have:

$$\begin{aligned} \|s\|_2 = \left\|A^T A'' v\right\|_2 &\leq \|A'' v\|_2 \\ &\leq m 2^{-b} \|v\|_2 \leq n 2^{-b} \|v\|_2 \end{aligned}$$

and for $p = 1$,

$$\begin{aligned} \|s\|_1 = \left\|A^T A'' v\right\|_1 &\leq \sqrt{n} \|A'' v\|_1 \\ &\leq m\sqrt{n} 2^{-b} \|v\|_1 \leq n^2 2^{-b} \|v\|_1 . \end{aligned}$$

□

The Augmented Indexing problem. As in [DIPW10], we use the Augmented Indexing communication game which is defined as follows: There are two parties, Alice and Bob. Alice is given a string $y \in \{0,1\}^d$. Bob is given an index $i \in [d]$, together with $y_{i+1}, y_{i+2}, \ldots, y_d$. The parties also share an arbitrarily long common random string r. Alice sends a single message $M(y,r)$ to Bob, who must output y_i with probability at least $2/3$, where the probability is taken over r. We refer to this problem as Augmented Indexing. The communication cost of Augmented Indexing is the minimum, over all correct protocols, of length $|M(y,r)|$ on the worst-case choice of r and y.

The following theorem is well-known and follows from Lemma 13 of [MNSW98] (see, for example, an explicit proof in [DIPW10])

Theorem 5.5. *The communication cost of* Augmented Indexing *is* $\Omega(d)$.

A well-separated set of points. We would like to prove Theorem 5.3, getting a large set of well-separated points in the image of a Lipschitz generative model. Before we do this, though, we prove a $k = 1$ analog:

Lemma 5.6. *Given $p \in \{1,2\}$, there is a set of points P in $B_n^p(1) \subset \mathbb{R}^n$ of size $2^{\Omega(n)}$ such that for each pair of points $x, y \in P$*

$$\|x - y\| \in \left[\left(\frac{1}{3}\right)^{1/p}, \left(\frac{2}{3}\right)^{1/p} \right].$$

Proof. Consider a τ-balanced linear code over the alphabet $\{\pm\frac{1}{n^{1/p}}\}$ with message length M. It is known that such codes exist with block length $O(M/\tau^2)$

[BATS09]. Setting the block length to be n and $\tau = 1/6$, we get that there is a set of $2^{\Omega(n)}$ points in $\mathbb{R}^n$ such that the pairwise hamming distance is between $\left[\frac{n}{3}, \frac{2n}{3}\right]$ i.e. the pairwise ℓ_p distance is between $\left[\left(\frac{1}{3}\right)^{1/p}, \left(\frac{2}{3}\right)^{1/p}\right]$. □

Now we wish to extend this result to arbitrary k while achieving the parameters in Theorem 5.3.

Proof of Theorem 5.3. We first define an $O(L)$-Lipschitz map $g : \mathbb{R} \to \mathbb{R}^{n/k}$ that goes through a set of points that are pairwise $\Theta\left(\frac{R}{k^{1/p}}\right)$ apart. Consider the set of points P from Lemma 5.6 scaled to $B^p_{n/k}(\frac{R}{k^{1/p}})$.

Observe that $|P| \geq \exp\left(\Omega\left(n/k\right)\right) \geq \min\left(\exp\left(\Omega\left(n/k\right)\right), \frac{Lr}{R}\right)$. Choose subset P' such that it contains exactly $\min\left(\frac{Lr}{R}, \exp(\Omega(n/k))\right)$ points and let $g_1 : [0, \frac{r}{k^{1/p}}] \to P'$ be a piecewise linear function that goes through all the points in P' in any order. Then, we define $g : \mathbb{R} \to \mathbb{R}^{n/k}$ as:

$$g(x) = \begin{cases} g_1(0) & \text{if } x < 0 \\ g_1(x) & \text{if } 0 \leq x \leq \frac{r}{k^{1/p}} \\ g_1(\frac{R}{k^{1/p}}) & \text{if } x \geq \frac{r}{k^{1/p}} \end{cases}$$

Let $I = \{\frac{r}{k^{1/p}|P'|}, \ldots, \frac{r}{k^{1/p}}\}$ be the points that are pre-images of elements of P'. Observe that g is $O(L)$-Lipschitz since within the interval $[0, \frac{r}{k^{1/p}}]$, since it maps each interval of length $\frac{r}{k^{1/p}|P'|} \geq \frac{rR}{k^{1/p}Lr} = \frac{R}{Lk^{1/p}}$ to an interval of length at most $O(\frac{R}{k^{1/p}})$.

Now, consider the function $G := g^{\otimes k} : \mathbb{R}^k \to \mathbb{R}^n$. Observe that G is

also $O(L)$ Lipschitz,

$$\begin{aligned}
&\|G(x_1,\ldots,x_k) - G(y_1,\ldots,y_k)\|_p^p \\
&= \sum_{i\in[k]} \|g(x_i) - g(y_i)\|_p^p \\
&\le \sum_{i\in[k]} O(L^p)\,\|x_i - y_i\|_p^p \\
&= O(L^p)\,\|x - y\|_p^p .
\end{aligned}$$

Also, for every point $(x_1,\ldots,x_k) \in I^k$, we know that $\|G(x_1,\ldots,x_k)\|_p = (\sum_{i\in[k]} \|g(x_i)\|_p^p)^{1/p} \le R$. However, there still exist distinct points $x, y \in I^k$(e.g. points that differ at one coordinate) such that $\|G(x) - G(y)\|_p \le O(\frac{R}{k^{1/p}})$.

We construct a large subset of the points in I^k such that any two points in this subset are far apart using error correcting codes. Consider the $A \subset P'$ s.t. $|A| > |P'|/2$ is a prime. For any integer $z > 0$, there is a prime between z and $2z$, so such a set A exists. Consider a Reed-Solomon code of block length k, message length $k/2$, distance $k/2$ and alphabet A. The existence of such a code implies that there is a subset X' of $(P')^k$ of size at least $(\frac{|P'|}{2})^{k/2}$ such that every pair of distinct elements from this set disagree in $k/2$ coordinates.

This translates into a distance of $\frac{R}{6^{1/p}}$ in p-norm. So, if we set $G = g^{\otimes k}$ and $X \subset I^k$ to $G^{-1}(X')$, we get a set of points of cardinality $(\frac{|P'|}{2})^{k/2} \ge (\min(\exp(\Omega(n/k)), \frac{Lr}{R}))^{k/2}$ with minimum distance $\frac{R}{6^{1/p}}$ in the p-norm that lie within the ℓ_p ball of radius R. □

Lower bound. We now prove the lower bound for generative model recovery.

Proof of Theorem 5.1. An application of Theorem 5.3 with $R = \sqrt{Lr\delta}$ gives us a set of points Z and G such that $Z = G(X) \subseteq \mathbb{R}^n$ such that $\log(|Z|) = \Omega(\min(k \log(\frac{Lr}{\delta}), n))$, and for all $x \in Z$, $\|x\| \leq \sqrt{Lr\delta}$ and for all $x, x' \in Z$, $\|x - x'\| \geq \sqrt{Lr\delta}/6$. Let $d = \lfloor \log |X| \rfloor \log n$, and let $D = 48(C + 1)$.

We will show how to solve the Augmented Indexing problem on instances of size $d = \log(|Z|) \cdot \log(n) = \Omega(k \log(Lr) \log n)$ with communication cost $O(m \log n)$. The theorem will then follow by Theorem 5.5.

Alice is given a string $y \in \{0, 1\}^d$, and Bob is given $i \in [d]$ together with $y_{i+1}, y_{i+2}, \ldots, y_d$, as in the setup for Augmented Indexing.

Alice splits her string y into $\log n$ contiguous chunks $y^1, y^2, \ldots, y^{\log n}$:

$$\underbrace{y_1, \ldots, y_{\log |X|}}_{y^1}, \underbrace{y_{\log |X|+1}, \ldots, y_{2 \log |X|}}_{y^2}, \ldots, \underbrace{y_{d - \log |X|}, \ldots, y_d}_{y^{\log n}}$$

where each chunk contains $\lfloor \log |X| \rfloor$ bits and represents an index into X.

She uses y^j as an index into the set X to choose x_j. Alice defines

$$x = D^1 x_1 + D^2 x_2 + \cdots + D^{\log n} x_{\log n}.$$

Alice and Bob use the common randomness $\mathcal{R}$ to agree on a recovery matrix A with orthonormal rows. Both Alice and Bob round A to form A' with $b = \Theta(\log(n))$ bits per entry. Alice computes $A'x$ and transmits it to Bob. Note that, since $x \in \left\{\pm \frac{1}{n^{1/p}}\right\}$ the x's need not be discretized.

From Bob's input i, he can compute the chunk $j = j(i)$ for which the bit y_i occurs in y^j. Bob's input also contains $y_{i+1}, \ldots, y_n$, from which he can

reconstruct $x_{j+1}, \ldots, x_{\log n}$, and in particular can compute

$$z = D^{j+1}x_{j+1} + D^{j+2}x_{j+2} + \cdots + D^{\log n}x_{\log n}.$$

Set $w = \frac{1}{D^j}(x - z) = \frac{1}{D^j}\sum_{i=1}^{j} D^i x_i$. Bob then computes $A'z$, and using $A'x$ and linearity, he can compute $\frac{1}{D^j} \cdot A'(x - z) = A'w$. Then

$$\|w\| \leq \frac{1}{D^j}\sum_{i=1}^{j} R \cdot D^i < R.$$

So from Lemma 5.4, there exists some s with $A'w = A(w - s)$ and

$$\|s\| < n^2 2^{-b} \|w\| < \frac{R}{D^j n^2}.$$

Ideally, Bob would perform recovery on the vector $A(w - s)$ and show that the correct point x_j is recovered. However, since s is correlated with A and w, Bob needs to use a slightly more complicated technique.

Bob first chooses another vector u uniformly from $B_n^p(\frac{R}{D^j})$ and computes $A(w - s - u) = A'w - Au$. He then runs the estimation algorithm $\mathcal{A}$ on A and $A(w - s - u)$, obtaining $\hat{w}$. We have that u is independent of w and s, and that $\|u\| \leq \frac{R}{D^j}(1 - 1/n^2) \leq \frac{R}{D^j} - \|s\|$ with probability $\frac{\mathrm{Vol}(B_n^p(\frac{R}{D^j}(1-1/n^2)))}{\mathrm{Vol}(B_n^p(\frac{R}{D^j}))} = (1-1/n^2)^n > 1-1/n$. But $\{w-u \mid \|u\| \leq \frac{R}{D^j} - \|s\|\} \subseteq \{w-s-u \mid \|u\| \leq \frac{R}{D^j}\}$, so as a distribution over u, the ranges of the random variables $w - s - u$ and $w-u$ overlap in at least a $1-1/n$ fraction of their volumes. Therefore $w-s-u$ and $w - u$ have statistical distance at most $1/n$. The distribution of $w - u$ is independent of A, so running the recovery algorithm on $A(w-u)$ would work with probability at least $3/4$. Hence with probability at least $\frac{3}{4} - \frac{1}{n} \geq \frac{2}{3}$ (for

n large enough), $\hat{w}$ satisfies the recovery criterion for $w - u$, meaning

$$\|w - u - \hat{w}\| \leq C \min_{w' \in \mathrm{Im}(G)} \|w - u - w'\| + \delta.$$

Now,

$$\begin{aligned}
\|x_j - \hat{w}\| &\leq \|w - u - x_j\| + \|w - u - \hat{w}\| \\
&\leq (1 + C) \|w - u - x_j\| + \delta \\
&\leq (1 + C) \left(\|u\| + \frac{1}{D^j} \cdot \sum_{i=1}^{j-1} \left\| D^i x_i \right\| \right) + \delta \\
&\leq 2(1 + C)R/D + \delta \\
&< R \cdot \frac{2(1 + C)}{D} + \delta \\
&= \frac{1}{24} \cdot R + \delta.
\end{aligned}$$

Since $\delta < Lr/24$, this distance is strictly bounded by $R/12$. Since the minimum distance in X is $R/6$, this means $\|D^j x_j - \hat{w}\| < \|D^j x' - \hat{w}\|$ for all $x' \in X, x' \neq x_j$. So Bob can correctly identify x_j with probability at least $2/3$. From x_j he can recover y^j, and hence the bit y_i that occurs in y^j.

Hence, Bob solves Augmented Indexing with probability at least $2/3$ given the message $A'x$. Each entry of $A'x$ takes $O(\log n)$ bits to describe because A' is discretized to up to $\log(n)$ bits and $x \in \{\pm \frac{1}{n^{1/p}}\}^n$. Hence, the communication cost of this protocol is $O(m \cdot \log n)$. By Theorem 5.5, $m \log n = \Omega(\min(k \log(Lr/\delta), n) \cdot \log n)$, or $m = \Omega(\min(k \log(Lr/\delta), n))$. $\square$

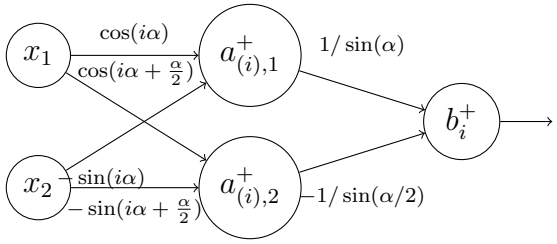
5.3 Construction of a Sparsity Producing Generative Model

We show that the set of all k-sparse vectors in $\mathbb{R}^n$ is contained in the image of a 2 layer neural network. This shows that generative model recovery is a generalization of sparse recovery.

Lemma 5.7. *There exists a 2 layer neural network $G : \mathbb{R}^2 \to \mathbb{R}^n$ with width $O(n)$ such that $\{x \mid \|x\|_0 = 1\} \subseteq \mathrm{Im}(G)$*

Our construction is intuitively very simple. We define two gadgets G_i^+ and G_i^-. $G_i^+ \geq 0$ and $G_i^+(x_1, x_2) \neq 0$ iff $\arctan(x_2/x_1) \in [i \cdot \frac{2\pi}{n}, (i+1) \cdot \frac{2\pi}{n}]$. Similarly $G_i^-(x_1, x_2) \leq 0$ and $G_i^-(x_1, x_2) \neq 0$ iff $\arctan(x_2/x_1) \in [\pi + i \cdot \frac{2\pi}{n}, \pi + (i+1) \cdot \frac{2\pi}{n}]$. Then, we set the i^{th} output node $(G(x_1, x_2))_i = G_i^+(x_1, x_2) + G_i^-(x_1, x_2)$. Varying the distance of (x_1, x_2) from the origin will allow us to get the desired value at the output node i.

Proof. Let $\alpha = \frac{\pi}{n+1}$. Let $[x]_+ = x \cdot \mathbb{I}(x \geq 0)$ denote the unbiased ReLU function that preserves positive values and $[x]_- = x \cdot \mathbb{I}(x \leq 0)$ denote the unbiased ReLU function that preserves negative values. We define $G_i^+ : \mathbb{R}^2 \to \mathbb{R}$ as follows:



G_i^+ is a 2 layer neural network gadget that produces positive values at output node i of G. We define each of the hidden nodes of the neural network G_i^+ as follows:

$$\begin{aligned} a^+_{(i),1} &= \Big[\cos(i\alpha)x_1 - \sin(i\alpha)x_2\Big]_+ \\ a^+_{(i),2} &= \Big[\cos\big(i\alpha + \frac{\alpha}{2}\big)x_1 - \sin\big(i\alpha + \frac{\alpha}{2}\big)x_2\Big]_+ \\ b^+_{(i)} &= \Big[\frac{a^+_{(i),1}}{\sin(\alpha)} - \frac{a^+_{(i),2}}{\sin(\alpha/2)}\Big]_+. \end{aligned}$$

In a similar manner, G_i^- which produces negative values at output node i of G with the internal nodes defined as:

$$\begin{aligned} a^-_{(i),1} &= \Big[\cos(\pi + i\alpha)x_1 - \sin(\pi + i\alpha)x_2\Big]_+ \\ a^-_{(i),2} &= \Big[\cos\big(\pi + i\alpha + \frac{\alpha}{2}\big)x_1 - \sin\big(\pi + i\alpha + \frac{\alpha}{2}\big)x_2\Big]_+ \\ b^-_{(i)} &= \Big[\frac{a^-_{(i),2}}{\sin(\alpha/2)} - \frac{a^-_{(i),1}}{\sin(\alpha)}\Big]_- \end{aligned}$$

The last ReLU activation preserves only negative values. Since G_i^+ and G_i^- are identical up to signs in the second hidden layer, we only analyze G_i^+'s. Consider $i \in [n]$. Let $\beta = i\alpha$ and $(x_1, x_2) = (t\sin(\theta), t\cos(\theta))$. Then using the identity $\sin(A)\cos(B) - \cos(A)\sin(B) = \sin(A - B)$,

$$\begin{aligned} \cos(\beta)x_1 - \sin(\beta)x_2 &= t\big(\cos(\beta)\sin(\theta) - \sin(\beta)\cos(\theta)\big) \\ &= t\sin(\theta - \beta). \end{aligned}$$

This is positive only when $\theta \in (\beta, \pi + \beta)$. Similarly, $\cos(\beta + \alpha/2)x_1 - \sin(\beta + \alpha/2)x_2 = t\sin(\theta - (\beta + \alpha/2))$ and is positive only when $\theta \in (\beta + \alpha/2, \pi + \beta + \alpha/2)$.

So, $a^+_{(i),1}$ and $a^+_{(i),2}$ are both non-zero when $\theta \in (\beta + \alpha/2, \pi + \beta)$. Using some elementary trigonometry, we may see that:

$$\begin{aligned}
&\frac{a_1^{(i)}}{\sin(\alpha)} - \frac{a_2^{(i)}}{\sin(\alpha/2)} \\
&= t\Big(\frac{\sin(\theta - \beta)}{\sin(\alpha)} - \frac{\sin(\theta - (\beta + \frac{\alpha}{2}))}{\sin(\alpha/2)}\Big) \\
&= \frac{t\sin(\beta - \theta + \alpha)}{\sin(\alpha/2)}.
\end{aligned}$$

In Fact B.1, we show a proof of the above identity. Observe that when $\theta > \beta + \alpha$, this term is negative and hence $b^i = 0$. So, we may conclude that $G_i^+((x_1, x_2)) \neq 0$ if and only if $(x_1, x_2) = (t\sin(\theta), t\cos(\theta))$ with $\theta \in ((i-1)\alpha, i\alpha)$. Also, observe that $G_i^+(t\sin(\beta + \alpha/2), t\cos(\beta + \alpha/2)) = t$. Similarly G_i^- is non-zero only if and only if $\theta \in [\pi + i\alpha, \pi + (i+1)\alpha]$ and $G_i^-(t\sin(\pi + i\alpha + \alpha/2), t\cos(\pi + i\alpha + \alpha/2)) = -t$. Since $\alpha = \frac{\pi}{n+1}$, the intervals within which each of $G_1^+, \ldots, G_n^+$,$G_1^-, \ldots, G_n^-$ are non-zero do not intersect.

So, given a vector z' such that $\|z\|_0 = 1$ with $z_{i'} \neq 0$, if $z_{i'} > 0$, set

$$x_1 = |z_{i'}|\sin(i'\alpha + \alpha/2)$$

$$x_2 = |z_{i'}|\cos(i'\alpha + \alpha/2)$$

and if $z_{i'} < 0$, set

$$x_1 = |z_{i'}|\sin(\pi + i'\alpha + \alpha/2)$$

$$x_2 = |z_{i'}|\cos(\pi + i'\alpha + \alpha/2).$$

Observe that:

$$G_{i'}^+((x_1, x_2)) + G_{i'}^-((x_1, x_2)) = z_{i'}$$

and for all $j \neq i'$

$$G_j^+((x_1, x_2)) + G_j^-((x_1, x_2)) = 0.$$

So, if $G(x) = (G_1^+(x) + G_1^-(x), \ldots, G_n^+(x) + G_n^-(x))$, G is a 2-layer neural network with width $O(n)$ such that $\mathrm{Im}(G) = \{x \mid \|x\|_0 \leq 1\}$. □

Now, we extend this gadget to a construction whose image is the set of all k-sparse vectors.

Proof of Theorem 5.2. Given a vector z that is non-zero at k coordinates, let $i_1 < i_2 < \cdots < i_k$ be the indices at which z is non-zero. We may use copies of G from Lemma 5.7 to generate 1-sparse vectors $v_1, \ldots, v_k$ such that $(v_j)_{i_j} = z_{i_j}$. Then, we add these vectors to obtain z. It is clear that we only used k copies of G to create G_{sp}. So, G_{sp} can be represented by a neural network with 2 layers. □

Theorem 5.1 provides a reduction which uses only 2 layers. Then, using the algorithm from Theorem 5.3, we can recover the correct k-sparse vector using $O(kd \log(nk))$ measurements. Since $d = 4$ and $\leq n$, this requires only $O(k \log n)$ linear measurements to perform ℓ_2/ℓ_2 (k, C)-sparse recovery.

Appendices

Appendix A

Theorems for Chapter 2

Theorem A.1 (Shannon-Hartley)**.** *Let S be a random variable such that $\mathbb{E}[S^2] = \tau^2$. Consider the random variable $S + T$, where $T \sim \mathcal{N}(0, \sigma^2)$. Then*

$$I(S; S+T) \leq \frac{1}{2} \lg \left(1 + \frac{\tau^2}{\sigma^2}\right).$$

Lemma A.2. *Consider a random variable $X \in [n]$ with probability distribution $p(l) = \Pr[X = l]$. Suppose $b = \lg(n) - H(X)$. Let $T_i = \{j \mid 2^i \leq np(j) \leq 2^{i+1}\}$ and $T_0 = \{j \mid np(j) \leq 2\}$ and let $q_i = \sum_{j \in T_i} p(j)$. Then,*

(a) $\sum_{i=0}^{\infty} i q_i \leq b + 1$

(b) $\sum_{i=0}^{\infty} q_i \lg(1 + \frac{1}{q_i}) \leq O(b+1)$

(c) if J is the random variable that denotes the index of the partition containing X, then $H(J) < O(b+1)$.

Proof.

$$\begin{aligned}
\sum_{i=0}^{\infty} iq_i &= \sum_{i>0} \sum_{j \in T_i} \Pr[X = j] \cdot i \\
&\leq \sum_{i>0} \sum_{j \in T_i} \Pr[X = j] \lg(n \Pr[X = j]) \\
&= b - \sum_{j \in T_0} \Pr[X = j] \lg(n \Pr[X = j]) \\
&= b - q_0 \lg(nq_0 / |T_0|) \\
&\leq b + |T_0| / ne
\end{aligned}$$

using convexity and minimizing $x \lg(ax)$ at $x = 1/ae$. Hence,

$$\sum_{i=0}^{\infty} iq_i \leq b + 1 \tag{A.1}$$

Next, consider $\sum_{i=0}^{\infty} q_i \lg(1 + \frac{1}{q_i})$. When $q_i \leq 1/2$, we have $\lg(1 + \frac{1}{q_i}) \leq 2 \lg(\frac{1}{q_i})$. So,

$$\sum_{i=0}^{\infty} q_i \lg(1 + \frac{1}{q_i}) \leq 2\Big(\sum_{i|q_i \leq 1/2} t_i \lg(1/t_i) + \sum_{i|q_i > 1/2} 1 \Big) \leq 2\Big(H(J) + 1 \Big) \tag{A.2}$$

Now, in order to bound the entropy term, consider the partition $T_+ = \{i \mid q_i > 1/2^i\}$ and $T_- = \{i \mid q_i \leq 1/2^i\}$. Then

$$\begin{aligned}
H(J) &= \sum_i q_i \lg(\frac{1}{q_i}) \\
&\leq \sum_{i \in T_+} iq_i + \sum_{i \in T_-} q_i \lg(\frac{1}{q_i}) \\
&\leq b + 1 + \sum_{i \in T_-} q_i \lg(\frac{1}{q_i})
\end{aligned}$$

Observe that $x\log(1/x)$ increases on $[0, 1/e]$, so

$$\sum_{i\in T_-} q_i \lg(\frac{1}{q_i}) \leq q_0 \log(\frac{1}{q_0}) + q_1 \lg(\frac{1}{q_1}) + \sum_{i\geq 2} \frac{1}{2^i} \lg(1/2^i) \leq 2/e + 3/2 < 3$$

Hence $H(J) < b + 4$. So, in (A.2),

$$\sum_{i=0}^{\infty} q_i \lg(1 + \frac{1}{q_i}) \leq 2(b+5) \tag{A.3}$$

□

Claim A.3. *Let the sequence $B_1 \leq B_2 \leq B_3 \ldots,$ satisfy $B_1 \geq k\log(k)$,$B_1 \leq \max\{m_1, k\log(k)\}$ and for all $r \geq 1$,*

$$B_{r+1} \leq \big(c_5 + \frac{c_3 m_{r+1}}{\alpha k}\big) B_r + m_{r+1}\log(k) + \frac{c_4 m_{r+1}}{\alpha} + c_2 k$$

for constants $c_2, c_3, c_4, c_5 > 1$. Then, for all $r \geq 1$,

$$B_r \leq \left(\prod_{j=2}^{r+1} \big(2c_5 + \frac{2c_6 m_j}{k\alpha}\big)\right) \max\{k\log(k), m_1\}$$

where c_6 is a constant.

Proof. The base case holds because :

$$B_1 = \max\{m_1, k\log(k)\}$$

Now, assume that the claim holds for r, then:

$$\begin{aligned}
B_{r+1} &\leq B_r\left(c_5 + \frac{c_3 m_{r+1}}{\alpha k}\right) + m_{r+1}\log(k) + \frac{c_4 m_{r+1}}{\alpha} + c_2 k \\
&= B_r\left(c_5 + \frac{c_3 \cdot m_{r+1}}{\alpha k}\right) + \frac{m_{r+1}}{k}(k\log(k)) + \frac{c_4 m_{r+1}}{\alpha} + c_2 k \\
&\leq 2B_r\left(c_5 + \frac{c_6 \cdot m_{r+1}}{\alpha k}\right) \\
&\leq \left(\prod_{j=2}^{r+1} \big(2c_5 + \frac{2c_6 m_j}{k\alpha}\big)\right) \max\{k\log(k), m_1\}
\end{aligned}$$

where the third line follows because $B_r \geq B_1 \geq k\log(k)$ and $B_r \geq B_1 \geq m_1$ and $c_6 = \max(c_3, c_4 + 1)$ is a constant. □

The following form of Bernstein's inequality is well known:

Theorem A.4 (Bernstein)**.** *Let $X_1, \ldots, X_n$ be i.i.d Bernoulli random variables with parameter p and $X = \sum_{i=1}^{n} X_i$. Then,*

$$\Pr[X \geq np + 4\log(1/\delta) + 4\sqrt{np\log(1/\delta)}] \leq \delta.$$

Appendix B

Theorems for Chapter 5

Fact B.1.

$$\frac{\sin(\beta+\frac{\alpha}{2}-\theta)}{\sin(\alpha/2)}-\frac{\sin(\beta-\theta)}{\sin(\alpha)}=\frac{\sin(\beta-\theta+\alpha)}{\sin(\alpha/2)}$$

Proof.

$$\begin{aligned}
&\frac{\sin(\beta+\frac{\alpha}{2}-\theta)}{\sin(\alpha/2)}-\frac{\sin(\beta-\theta)}{\sin(\alpha)}\\
&=\frac{\sin(\beta+\frac{\alpha}{2}-\theta)\sin(\alpha)-\sin(\beta-\theta)\sin(\alpha/2)}{\sin(\alpha)\sin(\alpha/2)}\\
&=\frac{1}{2\sin(\alpha)\sin(\frac{\alpha}{2})}\Big(\cos(\beta-\theta-\frac{\alpha}{2})-\cos(\beta-\theta+\frac{3\alpha}{2})\\
&\qquad\qquad-\cos(\beta-\theta-\frac{\alpha}{2})+\cos(\beta-\theta+\frac{\alpha}{2})\Big)\\
&=\frac{\cos(\beta-\theta+\frac{\alpha}{2})-\cos(\beta-\theta+\frac{3\alpha}{2})}{2\sin(\alpha)\sin(\alpha/2)}\\
&=\frac{\sin(\beta-\theta+\alpha)\sin(\alpha)}{\sin(\alpha)\sin(\alpha/2)}\\
&=\frac{\sin(\beta-\theta+\alpha)}{\sin(\alpha/2)}
\end{aligned}$$

where we use the identity that $\sin(A)\sin(B)=\frac{1}{2}[\cos(A-B)-\cos(A+B)]$ □

Bibliography

[ACD13] Ery Arias-Castro, Emmanuel J. Candès, and Mark A. Davenport. On the Fundamental Limits of Adaptive Sensing. *IEEE Transactions on Information Theory*, 59(1), 2013.

[BATS09] Avraham Ben-Aroya and Amnon Ta-Shma. Constructing Small-Bias Sets from Algebraic-Geometric Codes. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2009.

[BEO+13] Mark Braverman, Faith Ellen, Rotem Oshman, Toniann Pitassi, and Vinod Vaikuntanathan. A Tight Bound for Set Disjointness in the Message-Passing Model. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2013.

[BJKK04] Ziv Bar-Yossef, T. S. Jayram, Robert Krauthgamer, and Ravi Kumar. The Sketching Complexity of Pattern Matching. In *Approximation, Randomization, and Combinatorial Optimization, Algorithms and Techniques, 7th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX-RANDOM)*, 2004.

[BJKS04] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *Journal of Computer and System Sciences*, 68(4), 2004.

[BJPD17] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G. Dimakis. Compressed Sensing Using Generative Models. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

[BO15] Mark Braverman and Rotem Oshman. The Communication Complexity of Number-In-Hand Set Disjointness with No Promise. *Electronic Colloquium on Computational Complexity, (ECCC)*, 22, 2015.

[CCF02] M. Charikar, K. Chen, and M. Farach-Colton. Finding Frequent Items in Data Streams. In *Proceedings of the 29th International Colloquium on Automata, Languages and Programming (ICALP)*, 2002.

[CDD09] Albert Cohen, Wolfgang Dahmen, and Ronald DeVore. Compressed sensing and best k-term approximation. *American Mathematical Society*, 22, 2009.

[CHNR08] Rui M. Castro, Jarvis D. Haupt, Robert D. Nowak, and Gil M. Raz. Finding Needles in Noisy Haystacks. In *Proceedings of the*

IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2008.

[CM04] G. Cormode and S. Muthukrishnan. Improved Data Stream Summaries: The Count-Min Sketch and its Applications. In *LATIN 2004: Theoretical Informatics, 6th Latin American Symposium*, 2004.

[CM06] G. Cormode and S. Muthukrishnan. Combinatorial Algorithms for Compressed Sensing. In *Proceedings of Structural Information and Communication Complexity, 13th International Colloquium, (SIROCCO)*, 2006.

[CRT06a] E. Candes, J. Romberg, and T. Tao. Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information. *IEEE Transactions on Information Theory*, 52(2), 2006.

[CRT06b] E. J. Candès, J. Romberg, and T. Tao. Stable Signal Recovery from Incomplete and Inaccurate Measurements. *Communications on Pure and Applied Mathematics*, 59(8), 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. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2010.

[ECG+09] Yaniv Erlich, Kenneth Chang, Assaf Gordon, Roy Ronen, Oron Navon, Michelle Rooks, and Gregory J Hannon. DNA Sudoku-harnessing high-throughput sequencing for multiplexed specimen analysis. *Genome Research*, 19(7), 2009.

[GGI+02] Anna C. Gilbert, Sudipto Guha, Piotr Indyk, Yannis Kotidis, S. Muthukrishnan, and Martin Strauss. Fast, Small-Space Algorithms for Approximate Histogram Maintenance. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC)*, 2002.

[GLPS10] Anna C. Gilbert, Yi Li, Ely Porat, and Martin J. Strauss. Approximate Sparse Recovery: Optimizing Time and Measurements. In *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC)*, 2010.

[GPAM+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS)*, 2014.

[GSB16] Raja Giryes, Guillermo Sapiro, and Alexander M. Bronstein. Deep Neural Networks with Random Gaussian Weights: A Universal Classification Strategy? *IEEE Transactions on Signal Processing*, 64(13), 2016.

[HBCN12] Jarvis D. Haupt, Richard G. Baraniuk, Rui M. Castro, and Robert D. Nowak. Sequentially designed compressed sensing. In *Proceedings of the IEEE Statistical Signal Processing Workshop (SSP)*, 2012.

[HCN11] Jarvis D. Haupt, Rui M. Castro, and Robert D. Nowak. Distilled Sensing: Adaptive Sampling for Sparse Detection and Estimation. *IEEE Transactions on Information Theory*, 57(9), 2011.

[IPW11] Piotr Indyk, Eric Price, and David P. Woodruff. On the Power of Adaptivity in Sparse Recovery. In *Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2011.

[JXC08] Shihao Ji, Ya Xue, and Lawrence Carin. Bayesian Compressive Sensing. *IEEE Transactions on Signal Processing*, 56(6), 2008.

[KKP20] Akshay Kamath, Sushrut Karmalkar, and Eric Price. On the Power of Compressed Sensing with Generative Models. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

[KP19] Akshay Kamath and Eric Price. Adaptive Sparse Recovery with Limited Adaptivity. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms, (SODA)*, 2019.

[KP20] Akshay Kamath and Eric Price. Optimal Algorithms for Sparse Recovery under High SNR. *Manuscript*, 2020.

[KPW20] Akshay Kamath, Eric Price, and David Woodruff. Lower Bounds for Insertion-only Deterministic L2 Heavy Hitters. *Manuscript*, 2020.

[KW14] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, 2014.

[LDSP08] M. Lustig, D.L. Donoho, J.M. Santos, and J.M. Pauly. Compressed Sensing MRI. *IEEE Signal Processing Magazine*, 25(2), 2008.

[LNNT16] Kasper Green Larsen, Jelani Nelson, Huy L. Nguyen, and Mikkel Thorup. Heavy Hitters via Cluster-Preserving Clustering. In *Proceedings of the IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, 2016.

[LNW18] Yi Li, Vasileios Nakos, and David P. Woodruff. On Low-Risk Heavy Hitters and Sparse Recovery Schemes. In *Approxima-*

tion, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM), 2018.

[LS20] Z. Liu and J. Scarlett. Information-Theoretic Lower Bounds for Compressive Sensing With Generative Models. *IEEE Journal on Selected Areas in Information Theory*, 1(1), 2020.

[MG82] Jayadev Misra and David Gries. Finding Repeated Elements. *Science of Computer Programming*, 2(2), 1982.

[MNSW98] Peter Bro Miltersen, Noam Nisan, Shmuel Safra, and Avi Wigderson. On Data Structures and Asymmetric Communication Complexity. *Journal of Computer and System Sciences*, 57(1), 1998.

[MSW08] Dmitry M. Malioutov, Sujay Sanghavi, and Alan S. Willsky. Compressed Sensing with Sequential Observations. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2008.

[NSWZ18] Vasileios Nakos, Xiaofei Shi, David P. Woodruff, and Hongyang Zhang. Improved Algorithms for Adaptive Compressed Sensing. In *Proceedings of the 45th International Colloquium on Automata, Languages, and Programming (ICALP)*, 2018.

[Pol90] David Pollard. *Section 4: Packing and covering in Euclidean spaces*, volume Volume 2 of *Regional Conference Series in Prob-*

ability and Statistics, pages 14–20. Institute of Mathematical Statistics and American Statistical Association, 1990.

[PW11] Eric Price and David P. Woodruff. (1 + eps)-Approximate Sparse Recovery. In *Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2011.

[PW12] Eric Price and David P. Woodruff. Applications of the Shannon-Hartley Theorem to Data Streams and Sparse Recovery. In *Proceedings of the 2012 IEEE International Symposium on Information Theory (ISIT)*, 2012.

[PW13] Eric Price and David P. Woodruff. Lower Bounds for Adaptive Sparse Recovery. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2013.

[Sha11] Ohad Shamir. A Variant of Azuma's Inequality for Martingales with Subgaussian Tails. *CoRR*, abs/1110.2392, 2011.

[SK87] Georg Schnitger and Bala Kalyanasundaram. The probabilistic communication complexity of set intersection. In *Proceedings of the Second Annual Conference on Structure in Complexity Theory*, 1987.