Sparse Recovery and Fourier Sampling

by

Eric Price

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2013

Author .....
Department of Electrical Engineering and Computer Science
August 26, 2013

Certified by .....
Piotr Indyk
Professor
Thesis Supervisor

Accepted by .....
Leslie A. Kolodziejski
Chair, Department Committee on Graduate Students

Sparse Recovery and Fourier Sampling

by

Eric Price

Submitted to the Department of Electrical Engineering and Computer Science
on August 26, 2013, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computer Science

Abstract

In the last decade a broad literature has arisen studying *sparse recovery*, the estimation of sparse vectors from low dimensional linear projections. Sparse recovery has a wide variety of applications such as streaming algorithms, image acquisition, and disease testing. A particularly important subclass of sparse recovery is the sparse Fourier transform, which considers the computation of a discrete Fourier transform when the output is sparse. Applications of the sparse Fourier transform include medical imaging, spectrum sensing, and purely computation tasks involving convolution.

This thesis describes a coherent set of techniques that achieve optimal or near-optimal upper and lower bounds for a variety of sparse recovery problems. We give the following state-of-the-art algorithms for recovery of an approximately k-sparse vector in n dimensions:

- Two sparse Fourier transform algorithms, respectively taking $O(k \log n \log(n/k))$ time and $O(k \log n \log^c \log n)$ samples. The latter is within $\log^c \log n$ of the optimal sample complexity when $k < n^{1-\epsilon}$.
- An algorithm for *adaptive* sparse recovery using $O(k \log \log(n/k))$ measurements, showing that adaptivity can give substantial improvements when k is small.
- An algorithm for C-approximate sparse recovery with $O(k \log_C(n/k) \log^* k)$ measurements, which matches our lower bound up to the $\log^* k$ factor and gives the first improvement for $1 \ll C \ll n^\epsilon$.

In the second part of this thesis, we give lower bounds for the above problems and more.

Thesis Supervisor: Piotr Indyk
Title: Professor

Acknowledgments

This thesis is the culmination of several great years of grad school at MIT. I would therefore like to thank everyone who made it a happy and productive time.

My primary gratitude lies of course with my advisor, Piotr Indyk. It's hard to express how thankful I am for his unwavering support over the last five years. Piotr's combination of wisdom, good humor, and dedication to his students is incredible. Thanks, Piotr.

I also want to thank the rest of my coauthors: Khanh Do Ba, Badih Ghazi, Rishi Gupta, Haitham Hassanieh, Mikhail Kapralov, Dina Katabi, Gregory Minton, Jelani Nelson, Yaron Rachlin, Lixin Shi, David Woodruff, and Mary Wootters; this thesis is really a joint effort, and I've enjoyed the collaboration. I'd particularly like to thank David, with whom I've collaborated multiple times and learned much from the experience.

Lastly, I would like to thank my parents for a lifetime of support.

Contents

List of Figures

List of Algorithms

Chapter 1

Introduction

This thesis gives algorithms and lower bounds for computing sparse Fourier transforms and for other problems involving the estimation of sparse vectors from linear sketches.

Sparse Fourier Transforms. The discrete Fourier transform (DFT) is a fundamental tool used in a wide variety of domains, including signal processing, medical imaging, compression, and multiplication. The fastest known algorithm for computing the n-dimensional DFT is the $O(n \log n)$ time fast Fourier transform (FFT) [CT65]. Because of its breadth of applications, the FFT is one of the most heavily optimized algorithms in existence [FJ98].

A natural question is: under what circumstances can we compute the DFT in less than $n \log n$ time? And we might ask for more: when can we compute the DFT in *much* less than $n \log n$ time, in particular in *sublinear* time? Sublinear time algorithms are in some ways quite restricted, since they cannot even read the entire input or output all n values. So a necessary condition for a sublinear time DFT is that the output be *sparse*, i.e., have $k \ll n$ nonzero (or "important") values. If the other components are all zero we say that the DFT is *exactly k-sparse*; otherwise we say that it is *approximately k-sparse*.

Fortunately, many signals have sparse Fourier transforms. For example, in compression (including standard image, audio and video formats such as JPEG, MP3, and MPEG[1]) the whole point of using the DFT is that the resulting vector is sparse. Lossy compression schemes will zero out the "negligible" components, storing only $k \ll n$ terms of the DFT. Because we know empirically that these compression schemes work well for many signals, this means that many signals have sparse Fourier transforms.

In this thesis, we will give an algorithm to compute the DFT faster than the FFT whenever the DFT is k-sparse for $k = o(n)$. When the DFT is only approximately k-sparse then the output must necessarily involve some error; we will achieve $1 + \epsilon$ times the best possible error of any k-sparse output. This *sparse Fourier transform* problem had been studied before [Man92, GGI+02a, AGS03, GMS05, Iwe10, Aka10] but with slower running times. Ours is the first result that improves over the FFT in every sublinear sparsity regime.

A broader perspective. The sparse Fourier transform problem is one instance of a general class of problems involving the estimation of a sparse vector from a low dimensional linear sketch. This more general class—known variously as sparse recovery, compressed sensing, or compressive sampling—has applications such as image acquisition [DDT+08b], genetic testing [ECG+09], and streaming algorithms [CCF02, CM06].

[1] Some of these use not the DFT but the essentially equivalent discrete cosine transform (DCT).

Because problems in this class share many of the same properties, techniques developed for one problem can often be refined and applied other problems. This thesis will demonstrate this, building techniques in simpler sparse recovery problems before applying them to sparse Fourier transforms.

1.1 Sparse recovery overview

More formally, the problem of *sparse recovery* involves the observation of a linear sketch $Ax \in \mathbb{R}^m$ of an vector $x \in \mathbb{R}^n$, where $A \in \mathbb{R}^{m \times n}$ is the *measurement matrix*[2]. We would like to use Ax to recover an estimate x' of x such that, if x is "close" to k-sparse, then x' is similarly "close" to x. In particular, we would like that

$$\|x' - x\| \leq C \min_{k\text{-sparse } x_{(k)}} \|x - x_{(k)}\| \tag{1.1}$$

for some (possibly different) norms $\|\cdot\|$ and approximation factor C. We will also allow randomized algorithms, where A is drawn from a random distribution and (1.1) may fail with some "small" probability δ.

This thesis gives both upper and lower bounds for several problems in this general class. There are two types of variation among sparse recovery problems. The first type is variation of *sensing modality*: what kinds of matrices A may be chosen in the given application. The second type is variation of *recovery guarantee*: the choice of the norms, the approximation factor C, and the failure probability δ. In every situation, we would like to minimize two objectives: primarily the *number of measurements* m, but also the *running time* of the recovery algorithm.

A foundational result in sparse recovery theory is: if A is a uniformly random projection matrix, $C = \Theta(1) > 2$, and the norms are ℓ_2, then $m = \Theta(k \log(n/k))$ is necessary and sufficient to achieve (1.1) with $\delta = e^{-\Omega(m)}$ failure probability [CRT06b, BDDW08, Wai09]. Furthermore, it is possible to perform the recovery in polynomial time using ℓ_1 minimization. We will consider more general C and sensing modalities, getting faster running times and fewer measurements; the main tradeoff is that our techniques will also yield higher failure probabilities.

1.1.1 Sensing modality

The main differentiator among sparse recovery applications is the sensing modality. Different applications have different "hardware" to do the observation, which imposes different constraints on the measurement matrix A. For example:

The single pixel camera [DDT$^+$08b] flips the architecture of a camera from having an array of a million photosensors to having a single photosensor and an array of a million adjustable mirrors that can apply many different masks to the pixels. If x is sparse in the wavelet basis W, so Wx denotes the image in the pixel basis, then one can observe MWx for a binary matrix M. Hence the sensing modality is that $A = MW$ for a binary matrix M. A more careful model would impose additional constraints; for example, we want M to be dense so that a good fraction of the light reaches the photosensor.

The streaming heavy hitters problem [CCF02, GGI$^+$02b] is to estimate the most common elements of a multiset under a stream of insertions and deletions. If the number of items is very large (say, the set of all URLs on the web) then one cannot store the histogram $x \in \mathbb{Z}^n$ of counts directly. Instead one can store a low-dimensional sketch Ax, and use that $A(x + \Delta) = Ax + A\Delta$

[2]One can also use $\mathbb{C}$ instead of $\mathbb{R}$, which we will do in the Fourier setting.

to maintain it under insertions and deletions. In this case, we can choose A arbitrarily, but would prefer it to be sparse (so we can compute the update $A\Delta$ quickly) and possible to store implicitly with sublinear space (for example, using hash functions).

In genetic testing, we would like to determine which k of n people is a carrier for a rare recessive disease. Rather than test all n people, we can mix together samples and estimate the fraction of the DNA in the mixture that has the recessive gene [ECG+09]. Because each person's blood sample can only be split so finely, the measurement matrix must again be sparse. Additionally, the pipetting machine that mixes samples may impose constraints; for example, a machine with four heads works best when placing four adjacent inputs into four adjacent outputs, so the matrix should decompose into length-four diagonal sequences.

Magnetic resonance imaging (MRI) machines essentially sample from the 2D Fourier transform of the desired image [LDSP08]. Decreasing the sample complexity directly corresponds to decreasing the time patients must spend lying still in the machine. If the image is sparse in the pixel domain (e.g. in angiograms), then the sensing modality is that A is a subset of the 2D Fourier matrix. This is the same problem as the 2D sparse Fourier transform, except with an emphasis on measurement complexity rather than time complexity. If instead the image is sparse in another basis like the wavelet basis W, then A must be a subset of FW.

As we see, there are many different sensing modalities. For this thesis, we consider three simple ones that represent the breadth of power of the modalities. From most powerful to least powerful, these are:

- *Adaptive*: the measurements $\langle A_i, x\rangle$ are made in series and subsequent rows A_j may be chosen arbitrarily and dependent on $\langle A_i, x\rangle$ for $i < j$.
- *Standard* or *nonadaptive*: A may be chosen from an arbitrary distribution independent of x.
- *Fourier*: each row of A must be one of the n rows of the $n \times n$ discrete Fourier matrix F.[3]

Our algorithms in the adaptive and nonadaptive settings will use sparse matrices.

1.1.2 Recovery guarantee

Outside Chapter 7, our thesis uses the ℓ_2 norm for (1.1). This norm is appealing because (1) it is basis independent and (2) vectors are often sparse in the ℓ_2 norm but not in the ℓ_1 norm. In Chapter 7 we study the ℓ_1 norm, which has been studied in other work. We go into more detail on the relationship between various norms in Section 1.5.1.

C corresponds to the "noise tolerance" of the algorithm, and our value of C will vary throughout the thesis. The "default" setting is $C = O(1)$, but we will consider both $C = 1 + \epsilon$ for $\epsilon = o(1)$ and $C = \omega(1)$ at times.

Unfortunately, no deterministic algorithm with $m = o(n)$ satisfies the ℓ_2 guarantee [CDD09], so we must use a randomized algorithm with some chance δ of failure. We will generally allow δ to be a constant (say, 1/4). This failure probability can be decreased to an arbitrary $\delta > 0$ via repetition, with an $O(\log(1/\delta))$ loss in time and number of measurements.

Thesis overview. This thesis is divided into two parts. Part I gives algorithms for sparse recovery in the different sensing modalities and with some variation of C. Part II gives lower bounds on

[3] In the Fourier modality, our algorithms are nonadaptive while our lower bound applies to the more general adaptive setting.

Norm	Sensing modality	m (upper bound)	m (lower bound)	Running time
ℓ_2	Adaptive	$\frac{1}{\epsilon}k \log\log(n/k)$	$\frac{1}{\epsilon}k + \log\log n$	$n \log^c n$
	Nonadaptive	$k \log_C(n/k) \log^* k$	$k \log_C(n/k)$	$n \log^c n$
	Fourier	$\frac{1}{\epsilon}k \log n \log(n/k)$	$\frac{1}{\epsilon}k \log(n/k)/\log\log n$	$k \log n \log(n/k)$
	Fourier	$\frac{\log(1/\epsilon)}{\epsilon}k \log n \log^c \log n$	$\frac{1}{\epsilon}k \log(n/k)/\log\log n$	$k \log^2 n \log^c \log n$
ℓ_1	Adaptive		$\frac{1}{\sqrt{\epsilon}\log(k/\epsilon)}k$	
	Nonadaptive	$\frac{\log^c(1/\epsilon)}{\sqrt{\epsilon}}k \log n$	$\frac{1}{\sqrt{\epsilon}\log(k/\epsilon)}k$	$n \log^c n$

Figure 1-1: Table of main results for $C = (1+\epsilon)$-approximate recovery. Results written in terms of ϵ require $\epsilon \leq O(1)$. Results ignore constant factors and $c = \Theta(1)$ is a constant. The $\frac{1}{\epsilon}k$ lower bound for adaptive sensing is due to [ACD11]; the other results appear in this thesis.

the measurement complexity of these algorithms, as well as some upper and lower bounds for ℓ_1 recovery.

The rest of the introduction proceeds as follows. Section 1.2 gives a summary of our main upper and lower bounds. Section 1.3 gives notation used throughout the thesis. Section 1.4 describes the key techniques used in our Part I upper bounds. Finally, Section 1.5 surveys the techniques used in Part II.

1.2 Results and previous work

Figure 1-1 describes the main results of this thesis.

1.2.1 In the ℓ_2 guarantee

Adaptive. In the adaptive modality, we give an algorithm using $O(\frac{1}{\epsilon}k \log\log(n/k))$ measurements for $1+\epsilon$-approximate ℓ_2 recovery. This is smaller than the $\Omega(\frac{1}{\epsilon}k \log(n/k))$ measurements necessary in the nonadaptive modality, and exponentially smaller when k and ϵ are constants. Moreover, this exponential improvement is optimal: we also show that $\Omega(\log\log n)$ measurements are necessary in the adaptive setting.

The adaptive measurement model has received a fair amount of attention [JXC08, CHNR08, HCN09, HBCN09, MSW08, AWZ08], see also [Def10]. In particular [HBCN09] showed that adaptivity can achieve subconstant ϵ given $\Theta(k \log(n/k))$ measurements. Our result was the first to improve on the $\Theta(k \log(n/k))$ bound for general ϵ. In terms of lower bounds, we also know that $\Omega(\frac{1}{\epsilon}k)$ measurements are necessary [ACD11].

Nonadaptive. We give an $\Omega(k \log_C(n/k))$ lower bound on the number of measurements required, and an algorithm that uses a nearly matching $O(k \log_C(n/k) \log^* k)$ measurements for $C \gg 1$.

The nonadaptive setting is well studied. For the lower bound, multiple authors [Wai09, IT10, ASZ10, CD11] have bounds similar to our $\Omega(k \log_C(n/k))$ but in slightly different settings. The results of [Wai09, IT10, ASZ10] assume a Gaussian distribution on the measurement matrix, while the result of [CD11] is essentially comparable to ours (and appeared independently). Moreover, all these proofs—including ours—have a broadly similar structure.

For the upper bound, before our work, optimal $O(k \log_C(n/k))$ algorithms were known for $\epsilon = \Theta(1)$ [CRT06b] and then for all $\epsilon = O(1)$ [GLPS10]. At the other extreme, when $C = n^{\Omega(1)}$, the situation is close to the "noiseless" setting and $O(k)$ results were essentially known [EG07, BJCC12]. Ours is the first result to perform well in the intermediate regime of C.

Fourier. We devote two chapters to upper bounds in the Fourier modality, which optimize running time and sample complexity, respectively.

Chapter 4 gives an algorithm using $O(k \log n \log(n/k))$ time and measurements. Chapter 5 improves the sample complexity to $O(k \log n \log^c \log n)$ and takes $O(k \log^2 n \log^c \log n)$ time. In terms of lower bounds, the $\Omega(k \log(n/k))$ lower bound for nonadaptive measurements applies if the Fourier samples are chosen nonadaptively (as is the case in our algorithms). Even if the samples are chosen adaptively, we show that $\Omega(k \log(n/k)/\log\log n)$ Fourier samples are necessary: adaptivity cannot help dramatically like it does with arbitrary measurements, because one can only choose amongst n possibilities.

Our second result is thus within $(\log\log n)^c$ of optimal sample complexity for $k < n^{.99}$. Our first result takes one log factor more time than the optimal nonadaptive sample complexity; this seems like a natural barrier, given that the FFT has the same property.

Our algorithms in the Fourier modality assume that n is a power of two (or more generally, smooth) and gives an output up to $n^{-O(1)}$ relative precision.

Reference	Time	Samples	Approximation
[CT06, RV08, CGV12]	$n \log^c n$	$k \log n \log^3 k$	$C > 2$
[CP10]	$n \log^c n$	$k \log n$	$C > \log^2 n$
[GMS05]	$k \log^{O(1)} n$	$k \log^3 n$	any $C > 1$
Chapter 4	$k \log n \log(n/k)$	$k \log n \log(n/k)$	any $C > 1$
Chapter 5	$k \log^2 n (\log\log n)^c$	$k \log n (\log\log n)^c$	any $C > 1$

Figure 1-2: Sparse Fourier transform algorithms.

The goal of designing efficient DFT algorithms for (approximately) sparse signals has been a subject of a large body of research [Man92, GGI+02a, AGS03, GMS05, Iwe10, Aka10, LWC12, BCG+12, HAKI12, PR13, HKPV13]. These works show that, for a wide range of signals, both the time complexity and the number of signal samples taken can be significantly sub-linear in n. From a different perspective, minimizing the sampling complexity for signals that are approximate sparse in the Fourier domain was also a focus of an extensive research in the area of compressive sensing [CT06, RV08, CGV12, CP10]. The best known results are summarized in Figure 1-2. Our first result is the fastest known, and our second result uses the fewest samples to achieve a constant approximation factor.

1.2.2 In the ℓ_1 guarantee

To achieve the ℓ_1 guarantee for constant ϵ, it is known that $\Theta(k \log(n/k))$ measurements are necessary and sufficient [CRT06b, GLPS10, DIPW10, FPRU10]. Remaining open, however, was the dependence on ϵ.

A number of applications would like $\epsilon \ll 1$. For example, a radio wave signal can be modeled as $x = x^* + w$ where x^* is k-sparse (corresponding to a signal over a narrow band) and the noise w is i.i.d. Gaussian with $\|w\|_p \approx D\|x^*\|_p$ [TDB09]. Then sparse recovery with $C = 1 + \alpha/D$ allows

the recovery of a $(1-\alpha)$ fraction of the true signal x^*. Since x^* is concentrated in a small band while w is located over a large region, it is often the case that $\alpha/D \ll 1$.

The previous best upper bounds achieved the same $O(\frac{1}{\epsilon}k\log n)$ as in the ℓ_2 setting [CM04, GLPS10]. We show that this dependence is not optimal. We give upper and lower bounds that show that, up to $\log^c n$ factors, $k/\sqrt{\epsilon}$ is the correct dependence in the ℓ_1 norm, in both the nonadaptive and adaptive sensing modalities.

1.3 Notation and definitions

In this section we define notation used throughout the thesis.

- $[n]$ denotes the set $\{1,\dots,n\}$.
- For $S \subset [n]$, $\overline{S} := [n] \setminus S$.
- x_S: for $S \subset [n]$ and $x \in \mathbb{R}^n$, x_S denotes the vector $x' \in \mathbb{R}^n$ given by $x'_i = x_i$ if $i \in S$, and $x'_i = 0$ otherwise.
- $\mathrm{supp}(x)$ denotes the support of x.
- x_{-i}: for $i \in [n]$, $x_{-i} = x_{\overline{\{i\}}}$.
- $f \lesssim g$ if there exists a universal constant C such that $f \le Cg$, i.e. $f = O(g)$.[4]
- $f \ll g$ if $f = o(g)$.
- $g = \widetilde{O}(f)$ denotes that $g \le f\log^c f$ for some $c \lesssim 1$.

Definition 1.3.1. *Define*

$$H_k(x) := \operatorname*{arg\,max}_{\substack{S\subset[n]\\|S|=k}} \|x_S\|_2$$

to be the largest k coefficients in x.

Definition 1.3.2. *For any vector x, we define the "heavy hitters" to be those elements that are both (i) in the top k and (ii) large relative to the mass outside the top k. In particular, we define*

$$H_{k,\epsilon}(x) := \{j \in H_k(x) \mid x_j^2 \ge \frac{\epsilon}{k}\|x_{\overline{H_k(x)}}\|_2^2\}$$

Definition 1.3.3. *Define the "noise" or "error" as*

$$\mathrm{Err}_k^2(x) := \|x_{\overline{H_k(x)}}\|_2^2 = \min_{k\text{-sparse } y} \|x-y\|_2^2$$

[4] We remark in passing that big-O notation is often ill-defined with respect to multiple parameters [How08]. We will define $f = O(g)$ if $f \le Cg$ for some universal constant C, i.e. we drop the "for sufficiently large inputs" clause in the one-variable definition of big-O notation that causes difficulty with multiple variables. The clause is irrelevant when $f, g \ge 1$ everywhere, and is mainly useful to allow us to claim that $1 = O(\log n)$ despite $\log 1 = 0$. We instead redefine the log function inside big-O notation to mean $\log_a b := \max(1, \log_a b)$. This means, for example, $k = O(k\log_C(n/k))$ for all C. It does have the unintuitive result that $\log_a b \log a = \Theta(\log(ab))$ for $b \ge 1$.

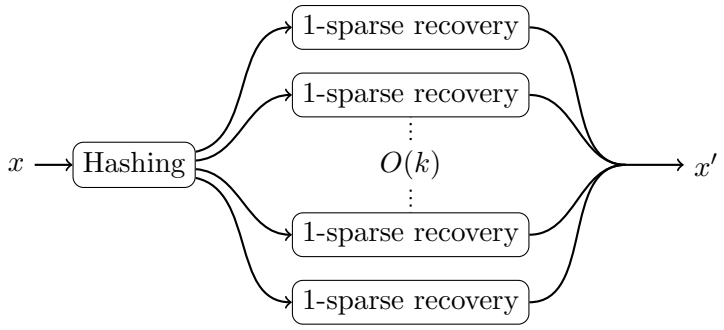


Figure 1-3: Partial sparse recovery finds 90% of the k heavy hitters

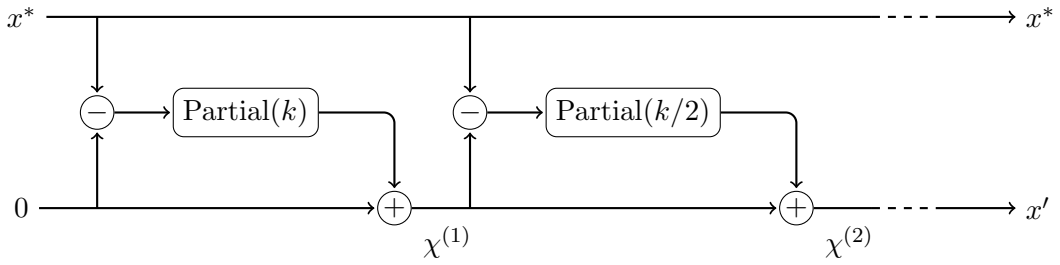


Figure 1-4: Full sparse recovery recovers all k heavy hitters. We iteratively refine an estimate χ of x^* by running partial sparse recovery on $x = x^* - \chi$.

Signal-to-noise ratio. A core concept in the intuition behind our ℓ_2 results is that of *signal-to-noise ratio*, which we generally define as

$$SNR := \|x_{H_k(x)}\|_2^2 / \|x_{\overline{H_k(x)}}\|_2^2$$

although we will vary the definition slightly at times. Essentially, one expects to get at most $O(\log(1+SNR))$ bits of information per measurement (which we show formally in our nonadaptive lower bound). Sparse recovery therefore becomes harder as the SNR decreases, except that $C = 1+\epsilon$ approximate recovery is only meaningful for $SNR > \epsilon$. (At $SNR < \epsilon$, the zero vector is a valid output.) Thus, in the "hard" cases, we will have $SNR \approx \epsilon$.

1.4 Upper bound techniques

Our Part I sparse recovery algorithms all use the same underlying architecture, which was previously used e.g. in [GGI+02b, GLPS10] and is represented in Figures 1-3 and 1-4. We first develop efficient methods of *1-sparse recovery* in the given modality. These methods are different in each different modality, but because $k = 1$ the problem is fairly simple. We then "hash" the k-sparse problem on n coordinates into $\Theta(k)$ "buckets" of $\Theta(n/k)$ coordinates. Most of the k "heavy hitters" (i.e. large coordinates) will be the only large coordinate in their bucket, and thus found with the 1-sparse recovery algorithm. This gives *partial k-sparse recovery*, where we recover "most" of the heavy hitters. We then iterate partial sparse recovery on the residual until we recover the heavy hitters. By making k decrease in each iteration, we can turn a partial sparse recovery algorithm into a full sparse recovery algorithm with only a constant factor overhead in the time and sample complexity.

Our Fourier-modality algorithm that optimizes measurements (Chapter 5) adds another stage

to the architecture, one of *SNR reduction*. We discuss this after describing the other results.

1.4.1 1-sparse recovery

Definition 1.4.1 (1-sparse recovery)**.** *Suppose $x \in \mathbb{C}^n$ has a single coordinate i^* with*

$$|x_{i^*}| \geq C\|x_{-i^*}\|_2$$

for some value C. The 1-sparse recovery problem Locate1Sparse*(x, C, δ) is to find i^* using as few measurements of x as possible (of the desired measurement modality), with success probability $1 - \delta$.*

We will only consider 1-sparse recovery for C larger than an arbitrarily large fixed constant. This makes the algorithms somewhat simpler, and smaller C are dealt with by the same method that converts from 1-sparse to general k-sparsity. Ignoring constant factors, our results are as follows:

Sensing modality	m	Running time
Nonadaptive	$\log_C(n/\delta)$	$\widetilde{O}(n)$
Adaptive	$\log\log_C n \log_C(1/\delta)$	$\widetilde{O}(n)$
Fourier	$\log n \log(1/\delta)$	$\log^{1.1} n$
Fourier	$\log_C n \log_C(\frac{1}{\delta}\log n)$	$\log_C n \log_C(\frac{1}{\delta}\log n)$

In the Fourier case, we have two algorithms: one which has optimal sample complexity for constant C and δ, and one which has better dependence on C at the cost of a $\log\log n$ factor. The two techniques could probably be merged into one $\log_C n \log_C(1/\delta)$ measurement algorithm, but we do not do so.

Note that in the nonadaptive and adaptive case, we are chiefly concerned with measurement complexity while in the Fourier case we also care about running time.

We now describe the techniques involved in each modality.

Nonadaptive measurements. Consider the two linear measurements

$$u(x) = \sum_{i\in[n]} s(i)x_i \qquad \text{and} \qquad u'(x) = \sum_{i\in[n]} i \cdot s(i)x_i$$

for random signs $s(i) \in \{\pm 1\}$, and consider the distribution of

$$\tilde{i} := u'(x)/u(x).$$

If x is zero outside i^* (i.e., $C = \infty$), then $\tilde{i} = i$. More generally, we can show that

$$|\tilde{i} - i^*| \lesssim n/\sqrt{C}$$

with $1 - 1/C$ probability over the signs s. We then consider randomly permuting x before applying the linear measurements. If we "vote" for every coordinate whose permutation lies within $O(n/\sqrt{C})$ of the observed u'/u, then i^* will get a vote with all but C^{-1} probability and $i' \neq i^*$ will get a vote with $O(C^{-1/2})$ probability. Hence, if we perform this process $r = O(\log_C(n/\delta))$ times, then with

probability $1-\delta$, by a Chernoff bound i^* will get more than $r/2$ votes and every other coordinate will get fewer than $r/2$ votes. This lets us recover i^*.

Adaptive measurements. With adaptive measurements, our goal is to "rule out" coordinates early, to make later stages have a higher "effective" value of C. We keep track of a set $S \subseteq [n]$ of "candidate" indices at each stage, and observe the same measurements as in the adaptive case but restricted to S:

$$u(x) = \sum_{i \in S} s(i) x_i \qquad \text{and} \qquad u'(x) = \sum_{i \in S} \pi(i) \cdot s(i) x_i$$

for random signs s and a random permutation π. Then as in the previous cases, for $\tilde{i} = u'(x)/u(x)$ we have

$$|\tilde{i} - \pi(i^*)| \lesssim n/\sqrt{D}$$

with probability $1 - 1/D$, where $D = |x_{i^*}|/\|x_{S \setminus \{i^*\}}\|_2$ is the "effective" value of C. We then remove every i from S that has $|\tilde{i} - \pi(i)| > cn/\sqrt{D}$ for an appropriate constant c. This means each $i \neq i^*$ has a $1 - 1/\sqrt{D}$ chance of being ruled out, and i^* remains in S with $1 - 1/D$ probability.

Therefore in the next iteration, we expect $\|x_{S \setminus \{i^*\}}\|_2^2$ to decrease by a $\sqrt{D}$ factor, so the new value of D is $D^{5/4}$. Then in $O(\log \log_C n)$ iterations we will have D grow from C to n^2, at which point S will be exactly $\{i^*\}$.

Fourier measurements. In the Fourier setting, there are two problems with the above nonadaptive method to overcome: first, the measurements were not Fourier measurements, and second in this setting we would like a polylogarithmic in n—rather than superlinear in n—running time.

To use Fourier measurements, we consider the two observations

$$\widehat{x}_a = \sum_i \omega^{ai} x_i \qquad \text{and} \qquad \widehat{x}_{a+t} = \sum_i \omega^{(a+t)i} x_i.$$

Then we have that

$$\widehat{x}_{a+t}/\widehat{x}_a \approx \omega^{i^* t}.$$

This is known as the "OFDM trick," after orthogonal frequency division multiplexing; it is also a special case of the Prony method. In particular, one can show that for each t as a distribution over a,

$$|\widehat{x}_{a+t}/\widehat{x}_a - \omega^{i^* t}| \lesssim 1/\sqrt{C} \tag{1.2}$$

with probability $1 - 1/C$. Because there are only $O(n/\sqrt{C})$ different i with ω^{it} inside this circle, we can now use the same voting technique as in the general nonadaptive method to get an equivalent result. Essentially, a is taking the place of the random signs s and t is taking the place of the random permutation. For random t, the true i^* gets a vote with $1 - 1/C$ probability and other i get votes with $1/\sqrt{C}$ probability, so $O(\log_C(n/\delta))$ random samples would suffice to identify i^* with $1 - \delta$ probability.

However, the running time for this voting procedure would be at least $\Theta(n/\sqrt{C})$ in each iteration, which is unacceptable for the sparse Fourier transform motivation. We will improve the running time at some cost to the dependence on δ. To describe this, we will first describe an alternate method that loses a $\log_C(\frac{1}{\delta} \log_C n)$ factor in measurements but is fast; we will then see how to merge these two methods into a fast sample-optimal method.

Suppose that the property (1.2) held with probability 1 (and, for simplicity, with a sufficiently small constant). Then consider using $O(\log_C n)$ samples to evaluate $\widehat{x}_{a+t}/\widehat{x}_a$ at $t = 1, C^{1/2}, C^{2/2}, C^{3/2}, \ldots, \Theta(n)$. In the first round, we will learn i^* to a region R_1 containing $n/\sqrt{C}$ indices, all consecutive. In the second round, the ω^{ti} for indices $i \in R_1$ will be uniformly spread on the unit circle, so we will learn i^* to a consecutive region R_2 of n/C indices, then R_3 of $n/C^{3/2}$ consecutive indices, and so on. In the end, we will identify i^* exactly. Because our regions contain consecutive indices, we can compute all this in time linear in the $O(\log_C n)$ sample complexity.

Of course, (1.2) does not hold with probability 1 for each t. However, if we evaluate $\widehat{x}_{a+t}/\widehat{x}_a$ for $O(\log_C(\frac{1}{\delta}\log_C n))$ random values of a, then for each t with $1 - \delta/\log_C n$ probability the median estimate will satisfy (1.2). Then a union bound over t gives a $1-\delta$ failure probability overall, using $O(\log_C n \log_C(\frac{1}{\delta}\log_C n))$ samples and time. This technique appears in, e.g., [GMS05], and we use it in Chapter 5 where we are willing to lose $\log\log n$ factors.

We merge the two techniques in Chapter 4 in the constant C, δ case. One can think of our first method as a random code that expands $\log n$ bits into $O(\log n)$ bits—having time and failure probability exponential in $\log n$. Our second method learns $\log C = O(1)$ bits at a time; we can think of this as partitions the goal into $\log n$ blocks of $O(1)$ bits, each of which is randomly coded into $O(\log\log n)$ bits, giving failure probability exponential in $\log\log n$ and time exponential in 1. To merge the two techniques, we instead use $\log n/\log\log n$ blocks of $\log\log n$ bits. Then decoding each block has time and failure probability exponential in $\log\log n$. A union bound then gets an $O(\log^{1.1} n)$ time, $O(\log n)$ sample 1-sparse recovery method with $1/\log^c n$ failure probability. One could probably extend this technique to general C and δ to get $O(\log_C n \log_C(1/\delta))$ measurements and $O(\log_C^{1.1} n)$ time.

This method is related to tree codes [Sch93, GMS11] and spinal codes [PIF+12]. With adaptive sampling, one could use noisy binary search [KK07] to achieve a better dependence on the failure probability; however, adaptive sampling would not work in our architecture for building a k-sparse recovery algorithm.

1.4.2 Partial k-sparse recovery

Our first step in building a k-sparse recovery algorithm from a 1-sparse recovery algorithm is "partial k-sparse recovery," which recovers *most* of the k heavy hitters.

Definition 1.4.2. *Let $x \in \mathbb{C}^n$. The* partial sparse recovery problem PartialSparse*$(x, k, f, \epsilon, \delta)$ is to find x' with*

$$\operatorname{Err}_{fk}^2(x' - x) \le (1+\epsilon)\operatorname{Err}_k^2(x)$$

with probability $1-\delta$ using (the given modality of) measurements on x.

For this intuition, we will describe the case of f and δ being small constants. We describe the modalities from simplest to most complicated; each analysis builds upon the previous one.

Adaptive. In the adaptive setting, once we locate a heavy coordinate i we can observe x_i directly to make $(x' - x)_i = 0$.

The idea is to partition the coordinates randomly into $\Theta(k/\epsilon)$ "buckets", then perform 1-sparse recovery on each bucket. Each coordinate will most likely (i.e., with probability $1 - f$ for constant f) land in a bucket that both (1) does not include any of the other k heavy hitters and (2) has roughly the average level of "noise" from non-heavy hitters, i.e. $\Theta((\epsilon/k)\operatorname{Err}_k^2(x))$. Therefore each i with $|x_i|^2 > (\epsilon/k)\operatorname{Err}_k^2(x)$ will with $1-f$ probability contain a large constant fraction of the mass in the bucket that it lies in, in which case the 1-sparse recovery guarantee says that i will probably

be found and removed from the residual. Therefore of the top k elements, we expect at most $O(fk)$ elements larger than $(\epsilon/k)\operatorname{Err}_k^2(x)$ to remain in the residual. This gives the partial sparse recovery guarantee.

Since the buckets have size $n\epsilon/k$, 1-sparse recovery takes $O(\log\log(n\epsilon/k))$ measurements per bucket or $O(k\log\log(n\epsilon/k))$ measurements overall.

Nonadaptive. In the nonadaptive setting, we cannot simply observe x_i after locating i and make the residual $x_i' - x_i$ zero. Therefore we need to introduce an *estimation* stage that lets us estimate coordinate values using nonadaptive measurements. For any $\epsilon \lesssim 1$, we can randomly partition the coordinates into $O(k/\epsilon)$ buckets and observe $u(x) = \sum_i s(i)x_i$ within each bucket for random signs $s(i) \in \{\pm 1\}$. Then $s(i)u(x) \approx x_i$, in particular $|s(i)u(x) - x_i|^2 \le (\epsilon/k)\operatorname{Err}_k^2(x)$ with large constant (say, 3/4) probability[5]. To estimate x_i, we take the median of $\log(1/f\delta)$ such estimates. With $1 - f\delta$ probability, most estimates will have small error and hence so will the median.

For partial sparse recovery with large C, as in the adaptive setting we partition the coordinates randomly into $O(k)$ buckets and perform C-approximate 1-sparse recovery on each bucket. With probability $1 - \delta$, this will return a set L of locations such that

$$\operatorname{Err}_{fk}^2(x_L - x) \le C\operatorname{Err}_k^2(x)$$

We then use the estimation stage (run with $O(k)$ buckets) to get $\tilde{x}_L \approx x_L$, such that

$$\operatorname{Err}_{2fk}^2(\tilde{x}_L - x) \le \underbrace{\operatorname{Err}_{fk}^2(\tilde{x}_L - x_L)}_{\text{Estimation error}} + \underbrace{\operatorname{Err}_{fk}^2(x_L - x)}_{\text{Location error}} \le (1 + C)\operatorname{Err}_k^2(x)$$

with probability $1 - 2\delta$. This gives partial sparse recovery with $O(k\log_C n + k) = O(k\log_C n)$ measurements for constant f and δ.

Fourier. Our algorithm for the Fourier modality is modeled after the previous ones, but somewhat more tricky to implement. For this discussion, we will assume $C = \Theta(1)$ for simplicity. The difficulty lies in partitioning the coordinates randomly into $O(k)$ buckets and performing 1-sparse recovery within each bucket. With arbitrary linear measurements we choose matrix rows containing $\Theta(n/k)$ nonzeros; Fourier measurements, by contrast, have n nonzeros.

The trick is to post-process the measurements into ones that are "close" to the desired 1-sparse measurements of buckets. To do this, we introduce good "filters" $G \in \mathbb{C}^n$ that are localized in both time and frequency domains. More specifically, let $I \in \mathbb{R}^n$ be the rectangular function of width n/k, so $I_i = 1$ for $|i| \le n/(2k)$ and 0 otherwise, and for intuition suppose that the k large coordinates of x were in random positions. Then $x \cdot I$, where $\cdot$ denotes entrywise multiplication, can be viewed as x restricted to a "bucket" of n/k coordinates. Since $x \cdot I$ is likely to be 1-sparse, we would ideally like to sample from the Fourier transform of $x \cdot I$ and perform 1-sparse recovery. This is hard to do exactly, but for some tolerance R, we construct $G \in \mathbb{C}^n$ such that roughly[6]

$$\begin{aligned}\|G - I\|_2 &\lesssim \|I\|_2/R\\ |\operatorname{supp}(\widehat{G})| &\lesssim k\log R\end{aligned}$$

That is, G is an approximation of I with sparse Fourier transform.

[5]Since the k heavy hitters will probably miss the bucket, and $(\epsilon/k)\operatorname{Err}_k^2(x)$ is the expected contribution of the other coordinates.

[6]In particular, if we ignore coordinates near the boundary where I changes from 1 to 0.

Figure 1-5 shows how to use $G, \widehat{G}$ to estimate something about x. We can compute $\widehat{G} \cdot \widehat{x}$ in order $|\operatorname{supp}(\widehat{G})| \lesssim k \log R$ time. If we then took its n-dimensional DFT, we would get the convolution $G * x$. To be more efficient in time complexity, we can instead "alias" the observation $\widehat{G} \cdot \widehat{x}$ into k terms (i.e., sum up the first k, second k, ... terms) and take the k-dimensional DFT. This gives us a subsampling of $G * x$: a vector $u \in \mathbb{C}^k$ with $u_i = (G * x)_{in/k}$, using $O(k \log R)$ samples and $O(k \log(Rk))$ time.

We can think of u in another way, as $u_i = \sum_j ((G * e_{in/k}) \cdot x)_j$, since our G is symmetric. In this view, $(G * e_{in/k}) \cdot x \approx (I * e_{in/k}) \cdot x$ approximates x restricted to n/k coordinates; it represents a "bucketing" of x. Then u_i, as the sum of elements in this bucket, represents the 0th Fourier coefficient of the bucketed signal. More generally, if (as in the figure) we shift $\widehat{G}$ by an offset a before multiplying with $\widehat{x}$, then u_i will represent the ath Fourier coefficient of the bucketed signal, for each bucket $i \in [k]$.

The resulting u is therefore the desired measurements of buckets achieved by post-processing. Hence we can run this procedure to compute u for multiple a as desired by our 1-sparse recovery algorithm, and if the bucketed signal $(G * e_{in/k}) \cdot x$ is 1-sparse then we will recover the large coordinate with $O(k \log R \log(n/k))$ samples.

The above description uses a deterministic bucketing, which will always fail on inputs with many nearby large coordinates. To randomize it, we imagine sampling from $\widehat{x}''_i = \widehat{x}_{\sigma i} \omega^{-\sigma b i}$ for random $\sigma, b \in [n]$ with σ invertible mod n. As in [GGI+02a, GMS05], the corresponding inverse Fourier transform x'' is an approximately pairwise independent permutation of x. Since we can easily simulate samples from $\widehat{x}''$ using samples of $\widehat{x}$, this gives us a randomized bucketing.

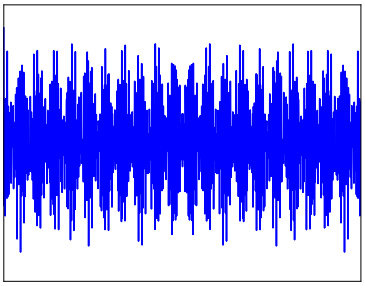
To get partial sparse recovery with this setup, we just need R to be large enough that the "leakage" between buckets is negligible. In our Chapter 4 result optimizing time complexity, we set $R = n^{O(1)}$. This is simple and easily suffices when our goal is $n^{O(1)}$ precision, but incurs a $\log R = \log n$ loss in sample complexity. In Chapter 5, we show that we only need that R be at least the "signal-to-noise ratio" $\|x\|_2^2 / \operatorname{Err}_k^2(x)$. The signal-to-noise ratio may be quite large; but in those cases C-approximate recovery for $C = R^{\Omega(1)}$ gives useful information using $O(k \log R \log_R(n/k) \log \log n) = O(k \log(Rn/k) \log \log n)$ measurements. In Section 1.4.4 we describe the intuition of this procedure.

1.4.3 General k-sparse recovery

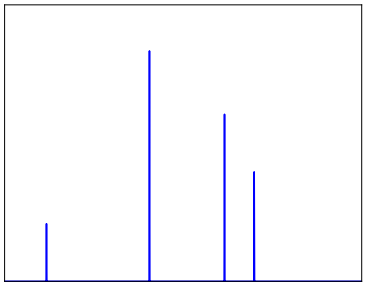
Suppose our original signal is x^*. Our first call to PartialSparse recovers a $\chi^{(1)}$ where $x^* - \chi^{(1)}$ is (say) $k/2$-sparse. We would then like to recursively call PartialSparse on $x = x^* - \chi^{(1)}$ with different parameters to get χ' and set $\chi^{(2)} = \chi^{(1)} + \chi'$, then on $x = x^* - \chi^{(2)}$ to get $\chi^{(3)}$, and so on. Two questions arise: first, how can we implement observations of $x = x^* - \chi$ using observations of x^* and knowledge of χ? Second, what *schedule* of of parameters $(k_r, f_r, \epsilon_r, \delta_r)$ should we use?

Implementing observations. For any row v of A, we can find $\langle v, x \rangle$ by observing $\langle v, x^* \rangle$, directly computing $\langle v, \chi \rangle$, and subtracting the two. In the adaptive and nonadaptive modalities this works great because (a) we are not concerned with running time and (b) the running time is fast in any case because v is sparse. In the Fourier setting neither of these hold. We will implement observations in different ways in the two Fourier chapters.

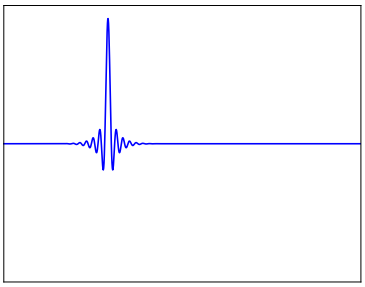
In Chapter 4, where $R = n^{\Omega(1)}$, we use that the "buckets" are very well isolated and each coordinate of χ only contributes non-negligibly to a single bucket. That is, the post-processed measurements u are essentially sparse linear transformations, so we can compute the influence of χ efficiently.



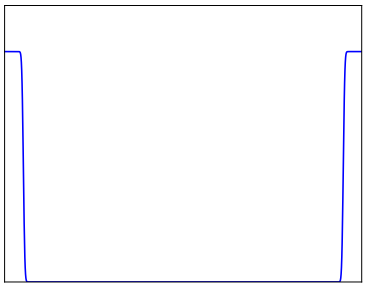
(a) Accessible signal $\widehat{x}$



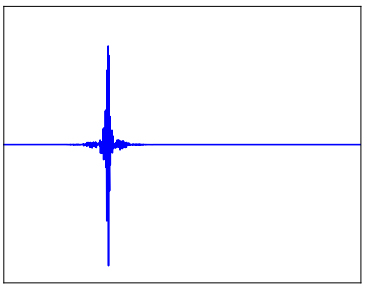
(b) Desired result x



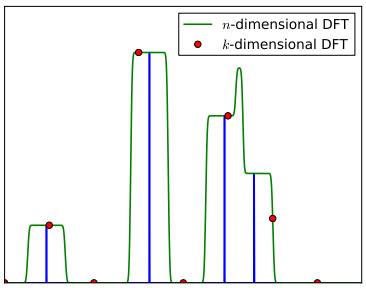
(c) Filter $\widehat{G}$ at some offset a.



(d) Absolute value of G, which is independent of a.



(e) Observation $\widehat{G} \cdot \widehat{x}$



(f) Computed k-dimensional Fourier transform of $\widehat{G} \cdot \widehat{x}$ (red dots) is a subsampling of $G * x$ (green line).

Figure 1-5: Computation using filters. For clarity, the signal x is exactly k-sparse. The red points in (f) can be computed in $O(|\operatorname{supp}(\widehat{G})| + k \log k)$ time.

In Chapter 5, because R may be smaller this is no longer the case. Instead, we use that the sampling pattern on $\widehat{x}$ consists of arithmetic sequences of length larger than k. By the semi-equispaced Fourier transform, we can compute $\widehat{\chi}$ at those locations with only a logarithmic loss in time ([DR93, PST01], see also Section 5.12). We then subtract $\widehat{\chi}$ from $\widehat{x}^*$ to get $\widehat{x}$ before post-processing the measurements.

Parameter schedule. We will vary the schedule, but the basic idea is to choose the f_i somehow and then:

$$\begin{aligned} k_i &= k \prod_{j<i} f_i \\ \epsilon_i &= \frac{1}{10}\epsilon/i^2 \\ \delta_i &= \frac{1}{10}\delta/i^2 \end{aligned}$$

for $i \le r$ where r is such that $\prod_{i\le r} f_i < 1/k$. Then with probability $1 - \sum_i \delta_i > 1 - \delta$, all the calls to PartialSparse succeed. In this case, by telescoping the guarantee of PartialSparse we have that the final result has

$$\|x^* - \chi^{(r)}\|_2^2 = \operatorname{Err}^2_{f_r k_r}(x^* - \chi^{(r)}) \le \operatorname{Err}^2_k(x^*) \prod_{i\le r}(1+\epsilon_i) \tag{1.3}$$

If $\epsilon < 1$, one can show $\prod_{i\le r}(1+\epsilon_i) \le 1+\epsilon$ which gives the desired guarantee (1.1).

The choice of f_i may affect the number of rounds r, the number of measurements, and the running time, but not correctness. Because of the logarithmic dependence of PartialSparse on f and linear dependence on k, we will be able to set $f_i = 2^{-\Omega(1/f_i)}$ and still have the sample complexity dominated by the first call to PartialSparse. Then f decays as a power tower, so only $r \lesssim \log^* k$ rounds are necessary.

When $C \gg 1$, (1.3) does not give the guarantee (1.1) but instead

$$\|x^* - \chi^{(r)}\|_2^2 \le C^{O(\log^* k)} \operatorname{Err}^2_k(x^*).$$

Hence in the nonadaptive modality this gives a $C^{O(\log^* k)}$-approximate sparse recovery algorithm using $O(k \log_C(n/k))$ measurements. Redefining C, this gives a C-approximate algorithm with $O(k \log_C(n/k) \log^* k)$ measurements.

1.4.4 SNR reduction

Our Fourier-modality algorithm optimizing measurements (Chapter 5) adds an *SNR reduction* stage to the architecture. Let $O^*(\cdot)$ hide $\log\log n$ factors. The preceding sections describe how, with arbitrary measurements, we can achieve C-approximate recovery with $O^*(k \log_C(n/k))$ measurements. Furthermore with Fourier measurements, if the SNR $\|x\|_2^2/\operatorname{Err}^2_k(x) \lesssim R$, we can perform C-approximate recovery with $O^*(k \log R \log_C(n/k))$ measurements.

By definition, the C-approximate recovery guarantee states that the residual $x - x'$ has SNR C^2. Therefore, starting with an upper bound on the SNR of $R = n^{O(1)}$, we can perform $R^{0.1}$-approximate recovery using $O^*(k \log(Rn/k))$ measurements and recurse on the residual with $R \to R^{0.2}$. After $\log\log n$ iterations, the SNR will be constant and we can solve the problem directly for the desired approximation factor $1+\epsilon$.

1.5 Lower bound techniques

1.5.1 ℓ_p/ℓ_q guarantees

Generally the two norms in (1.1) can be different; an "ℓ_p/ℓ_q guarantee" uses ℓ_p on the left and ℓ_q on the right, and scales C as $k^{1/p-1/q}$; this scaling compensates for the behavior if the vectors on both sides were dominated by k comparable large values. The common (p, q) in the literature are $(\infty, 2)$, $(2, 2)$, $(2, 1)$ and $(1, 1)$, in order of decreasing strength of the guarantee. In every chapter of this thesis except Chapter 7, we choose $p = q = 2$.

Choosing $q = 2$ is important because the guarantee (1.1) is trivial on many signals when $q = 1$. In particular, vectors are often sparse because their coordinates decay as a power law. Empirically, power laws $x_i \approx i^{-\alpha}$ typically have decay exponent $\alpha \in [1/2, 2/3]$ [CSN09, Mit04][7], which is precisely the range where the right hand side of (1.1) converges when $q = 2$ but not when $q = 1$. For such signals, the $q = 1$ guarantee is trivially satisfied by the zero vector. The downside of this stronger guarantee is that it is harder to achieve. In fact, no deterministic algorithm with $m = o(n)$ can achieve $q = 2$ [CDD09], so we must allow randomized algorithms with a δ chance of failure.

For the left hand side, having $p = 2$ is nice because it is basis independent. So if we sparsify in one basis (e.g., the wavelet basis for images) we can still get an approximation guarantee in another basis (e.g. the pixel basis).

A significant fraction of the work on sparse recovery considers *post-measurement noise*. In this model, one observes $Ax + w$ for an exactly k-sparse x and wants to recover x' with

$$\|x' - x\|_2 \le C\|w\|_2.$$

To be nontrivial, one adds some restriction on size of entries in A, such as that all rows have unit norm. This model behaves quite similarly to the ℓ_2/ℓ_2 pre-measurement noise model, and algorithms that work in one of the two models usually also apply to the other one. We will focus on pre-measurement noise, but most of our algorithms and lower bounds can be modified to apply in a post-measurement noise setting.

1.5.2 ℓ_2: Gaussian channel capacity

In the ℓ_2 setting, we can often get tight lower bounds the number of measurements required for sparse recovery using the information capacity of an AWGN Gaussian channel. Similar techniques have appeared in [Wai09, IT10, ASZ10, CD11].

Basic technique. To elucidate the technique, we start with an $\Omega(\log_C n)$ lower bound for arbitrary nonadaptive measurements with $k = 1$ and $C \ge 2$. We draw x from the distribution

$$x = e_{i^*} + w$$

for $i^* \in [n]$ uniformly at random and $w \sim N(0, \Theta(1/(C^2 n))I_n)$ normally distributed. We will show that

$$\log n \lesssim I(i^*; Ax) \lesssim m \log C \tag{1.4}$$

[7] Corresponding to Pareto distributions with parameter $\alpha \in [2, 3]$.

to get the result, where $I(a;b)$ denotes the Shannon mutual information between a and b. For the left inequality, we know that sparse recovery must find x' with

$$\|x' - e_{i^*}\|_2 \leq \|x' - x\|_2 + \|w\|_2 \leq (C+1)\|w\|_2 \tag{1.5}$$

with constant probability. But $\|w\|_2$ is strongly concentrated about its mean $\Theta(1/C)$, so with appropriate constants $\|x' - e_{i^*}\|_2 < 1/4$ with constant probability. But then rounding x' to the nearest e_i will recover the $\log n$-bit i^* with constant probability. By Fano's inequality, this means $I(i^*; x') \gtrsim \log n$. Since $i^* \to Ax \to x'$ is a Markov chain, $I(i^*; Ax) \geq I(i^*; x') \gtrsim \log n$.

For the upper bound on $I(i^*, Ax)$, we note that each individual measurement is of the form

$$\langle v, x\rangle = v_{i^*} + \langle v, w\rangle$$

for some row v of A. Since w is isotropic Gaussian, $\langle v, w\rangle$ is simply a Gaussian random variable independent of v_{i^*}. This channel $i^* \to \langle v, x\rangle$ is an *additive white Gaussian noise (AWGN) channel*, which is well studied in information theory. In his paper introducing information theory [Sha48], Shannon proved that it has information capacity

$$I(i^*, \langle v, x\rangle) \leq \frac{1}{2}\log(1 + SNR) \tag{1.6}$$

where SNR denotes the "signal to noise ratio" $\frac{\mathbb{E}[v_{i^*}^2]}{\mathbb{E}[\langle v,w\rangle^2]}$. In our setting, we have $\mathbb{E}[v_{i^*}^2] = \|v\|_2^2/n$ and $\mathbb{E}[\langle v, w\rangle^2] = \|v\|_2^2/(C^2 n)$, giving

$$I(i^*, \langle v, x\rangle) \lesssim \log C.$$

Using the chain rule of information and properties of linearity, we then show that $I(i^*, Ax) \lesssim m \log C$ to get (1.4), and hence $m = \Omega(\log_C n)$.

General nonadaptive. To support $k > 1$, we instead set

$$x = \tilde{x} + w$$

where $\tilde{x} \in \{0, \pm 1\}^n$ is k-sparse and $w \sim N(0, k/(C^2 n)I)$. In particular, we draw $\mathrm{supp}(\tilde{x})$ from a *code* $\mathcal{F}$ of $2^{\Omega(k\log(n/k))}$ different supports, where every pair of supports in the code differ in $\Omega(k)$ positions. Then in a very similar fashion to the above, we get the result from

$$k\log(n/k) \lesssim I(\mathrm{supp}(\tilde{x}); Ax) \lesssim m \log C.$$

The lower bound follows from the distance property, and the upper bound just needs that

$$\mathop{\mathbb{E}}_{x}[\langle v, \tilde{x}\rangle^2] = k/n \tag{1.7}$$

for every unit vector v. If $\tilde{x}$ is drawn uniformly conditioned on its support S and the code is sufficiently "symmetric", we have

$$\mathop{\mathbb{E}}_{x}[\langle v, \tilde{x}\rangle^2] = \mathop{\mathbb{E}}_{S}[\|v_S\|_2^2] = k/n.$$

The same framework also supports $C = 1 + \epsilon$ for $\epsilon = o(1)$. First, to make the upper bound of (1.4) hold, we increase w to $N(0, k/(\epsilon n)I)$. The lower bound of (1.4) then requires more care,

but still holds.

Adaptive Fourier. In the Fourier setting, we would like to lower bound *adaptively chosen* measurements sampled from the Fourier matrix.

The above technique is mostly independent of adaptivity, with the exception of (1.7). In the adaptive setting, v is not independent of x, which makes (1.7) not hold. To work around this, we observe that for any fixed row v of the Fourier matrix and $S = \text{supp}(\tilde{x})$, if the signs of $\tilde{x}$ are chosen uniformly at random, then

$$\Pr[\langle v, \tilde{x}\rangle^2 > kt/n] < e^{-\Omega(t)}$$

by subgaussian concentration inequalities. Therefore with high probability over the signs,

$$\langle v, \tilde{x}\rangle^2 < k(\log n)/n$$

for all v in the Fourier matrix. Then regardless of the choice of v, the SNR is bounded by $C^2 \log n$, giving $I(\text{supp}(\tilde{x}); Ax) \lesssim m \log(C \log n)$. Thus with Fourier measurements, adaptivity can only give a $\log\log n$ factor improvement.

Arbitrary adaptive. When the measurement vector v can be chosen arbitrarily, the above lower bound (which relies on $\|v\|_\infty = 1/\sqrt{n}$) does not apply. In fact, in this case we know that $O(k \log\log(n/k))$ measurements suffice. Using a more intricate Gaussian channel capacity argument, we show that $\Omega(\log\log n)$ measurements are necessary. Hence our upper bound is tight for $k = 1$.

The core idea of the proof is as follows. At any stage of the adaptive algorithm, we have some posterior distribution p on i^*. This represents some amount of information $b = H(i^*) - H(p)$. Intuitively, with b bits of information we can restrict i^* to a subset of size $n/2^b$, which increases the SNR by a 2^b factor giving channel capacity $\frac{1}{2}\log(1 + SNR) \lesssim 1 + b$. For general distributions p this naive analysis based solely on the SNR does not work; nevertheless, with some algebra we show for all distributions p that $I(i^*; \langle v, x\rangle \mid p) \lesssim 1 + b$. This means it takes $\Omega(\log\log n)$ measurements to reach $\log n$ bits.

1.5.3 ℓ_1: Communication complexity

In Chapter 7, we consider recovery guarantees other than the standard ℓ_2 guarantee. The Gaussian channel technique in the previous section does not extend well to the ℓ_1 setting. It relies on the Gaussian being both 2-stable and the maximum entropy distribution under an ℓ_2 constraint. The corresponding distributions in ℓ_1 are not the same as each other, so we need a different technique.

We use communication complexity. Alice has an input $x \in \mathbb{R}^n$ and Bob has an input $y \in \mathbb{R}^n$, and they want to compute some function $f(x, y)$ of their inputs. Communication complexity studies how many bits they must transmit between themselves to compute the function; for some functions, it is known that many bits must be transmitted. We show that a variant of a known hard problem, $\mathsf{Gap}\ell_\infty$, is hard on a particular distribution.

We then show that sparse recovery solves our $\mathsf{Gap}\ell_\infty$ variant. In the nonadaptive setting, the idea is that Alice sends Ax to Bob, who subtracts Ay to compute $A(x - y)$. Bob then runs sparse recovery on $x - y$, and the result lets us solve our $\mathsf{Gap}\ell_\infty$ variant. This shows that Ax must have $\Omega(k/\sqrt{\epsilon})$ bits. In the adaptive setting, the same idea applies except the communication is two-way—so Alice and Bob can both compute subsequent rows of the matrix—and we get the same bound.

Of course, the entries of A are real numbers so Ax may have arbitrarily many bits. We show that only $O(\log n)$ bits per entry are "important": rounding A to $O(\log n)$ bits per term gives a negligible perturbation to the result of sparse recovery. Because our lower bound instance has $n \approx k/\epsilon$, this gives a lower bound on the number of measurements of $\Omega(k/(\sqrt{\epsilon}\log(k/\epsilon)))$.

Part I

Algorithms

Chapter 2

Adaptive Sparse Recovery

(Based on parts of [IPW11])

2.1 Introduction

In this chapter we give an algorithm for $1+\epsilon$ approximate ℓ_2/ℓ_2 sparse recovery using only $O(\frac{1}{\epsilon}k \log\log n)$ measurements if we allow the measurement process to be *adaptive*. In the adaptive case, the measurements are chosen in rounds, and the choice of the measurements in each round depends on the outcome of the measurements in the previous rounds.

Implications. Our new bounds lead to potentially significant improvements to efficiency of sparse recovery schemes in a number of application domains. Naturally, not *all* applications support adaptive measurements. For example, network monitoring requires the measurements to be performed simultaneously, since we cannot ask the network to "re-run" the packets all over again. However, a surprising number of applications are capable of supporting adaptivity. For example:

- Streaming algorithms for data analysis: since each measurement round can be implemented by one pass over the data, adaptive schemes simply correspond to multiple-pass streaming algorithms (see [McG09] for some examples of such algorithms).
- Compressed sensing of signals: several architectures for compressive sensing, e.g., the single-pixel camera of [DDT+08a], already perform the measurements in a sequential manner. In such cases the measurements can be made adaptive[1]. Other architectures supporting adaptivity are under development [Def10].
- Genetic data analysis and acquisition: as above.

Running time of the recovery algorithm. In the nonadaptive model, the running time of the recovery algorithm is well-defined: it is the number of operations performed by a procedure that takes Ax as its input and produces an approximation x' to x. The time needed to generate the measurement vectors A, or to encode the vector x using A, is not included. In the adaptive case, the distinction between the matrix generation, encoding and recovery procedures does not exist, since

[1] We note that, in any realistic sensing system, minimizing the number of measurements is only one of several considerations. Other factors include: minimizing the computation time, minimizing the amount of communication needed to transfer the measurement matrices to the sensor, satisfying constraints on the measurement matrix imposed by the hardware etc. A detailed cost analysis covering all of these factors is architecture-specific, and beyond the scope of this thesis.

new measurements are generated based on the values of the prior ones. Moreover, the running time of the measurement generation procedure heavily depends on the representation of the matrix. If we suppose that we may output the matrix in sparse form and receive encodings in time bounded by the number of nonzero entries in the matrix, our algorithms run in $n \log^{O(1)} n$ time. Moreover, if we may *implicitly* output the matrix (e.g. output a circuit that computes each entry of the matrix in each round) then one could probably implement our algorithm with $k \log^{O(1)} n$ time.

2.2 1-sparse recovery

This section discusses recovery of 1-sparse vectors with $O(\log \log n)$ adaptive measurements. First, in Lemma 2.2.1 we show that if the heavy hitter x_{j^*} is $B \gg 1$ times larger than the ℓ_2 norm of everything else (the "signal-to-noise ratio" is n^2), then with two nonadaptive measurements we can find an estimate $\tilde{j}$ of j^* with $|\tilde{j} - j^*| \lesssim n/B$. The idea is quite simple: we observe

$$u(x) = \sum_i s(i)x_i \qquad \text{and} \qquad u'(x) = \sum_i i \cdot s(i)x_i$$

for random signs $s(i) \in \{\pm 1\}$, and round $u'(x)/u(x)$ to get our estimate of j^*.

Lemma 2.2.1. *Consider the measurements*

$$u(x) = \sum_i s(i)x_i \qquad \textit{and} \qquad u'(x) = \sum_i i \cdot s(i)x_i$$

for pairwise independent random signs $s(i) \in \{\pm 1\}$. Then with probability $1 - \delta$,

$$\left| \frac{u'(x)}{u(x)} - j^* \right| \leq 2\sqrt{2} \frac{n}{\sqrt{\delta}} \frac{\|x_{-j^*}\|_2}{|x_{j^*}|}$$

for any j^* for which this bound is less than n.

Proof. By assumption, we have that j^* satisfies $|x_{j^*}|^2 \geq (8/\delta)\|x_{-j^*}\|_2^2$. We have that

$$\frac{u'(x)}{u(x)} = j^* + \frac{\sum_i (i - j^*)x_i s(i)}{\sum_i x_i s(i)}$$

and will bound the numerator and denominator of this error term. We have by pairwise independence of s that

$$\mathbb{E}\left[\left(\sum_i (i - j^*)x_i s(i) \right)^2 \right] = \sum_i (i - j^*)^2 x_i^2 \leq n^2 \|x_{-j^*}\|_2^2.$$

so by Markov's inequality, with probability $1 - \delta/2$ we can bound the numerator by

$$|\sum_i (i - j^*)x_i s(i)| \leq \sqrt{2/\delta} n \|x_{-j^*}\|_2$$

For the denominator, we have

$$|\sum_i x_i s(i)| \geq |x_{j^*}| - |\sum_{i \neq j^*} x_i s(i)|.$$

Because $\mathbb{E}[|\sum_{i\neq j^*} x_i s(i)|^2] = \|x_{-j^*}\|_2^2 \leq \delta |x_{j^*}|^2/8$ by assumption, by Markov's inequality we have with probability $1-\delta/2$ that

$$|\sum_{i\neq j^*} x_i s(i)| \leq |x_{j^*}|/2$$

and so the denominator is at least $|x_{j^*}|/2$. Combining, we have

$$\left|\frac{u'(x)}{u(x)} - j^*\right| = \left|\frac{\sum_i (i-j^*)x_i s(i)}{\sum_i i \cdot x_i s(i)}\right| \leq \frac{\sqrt{2/\delta} n \|x_{-j^*}\|_2}{|x_{j^*}|/2} = 2\sqrt{2}\frac{n}{\sqrt{\delta}}\frac{\|x_{-j^*}\|_2}{|x_{j^*}|}$$

as desired. □

This lemma is great if $|x_{j^*}| \gg n\|x_{-j^*}\|_2$, in which case $u(x)/u'(x)$ rounds to j^*. At lower signal-to-noise ratios, it still identifies j^* among a relatively small set of possibilities. However, because those possibilities (i.e. the nearby indices) are deterministic this lemma does not allow us to show that the *energy* of the extraneous possibilities goes down. To fix this, we extend the result to allow an arbitrary hash function from $[n]$ to $[D]$:

Lemma 2.2.2. *Let $h : [n] \to [D]$ be fixed and consider the measurements*

$$u(x) = \sum_i s(i)x_i \qquad \text{and} \qquad u'(x) = \sum_i h(i) \cdot s(i)x_i$$

for pairwise independent random signs $s(i) \in \{\pm 1\}$. Then with probability $1-\delta$,

$$\left|\frac{u'(x)}{u(x)} - h(j^*)\right| \leq 2\sqrt{2}\frac{D}{\sqrt{\delta}}\frac{\|x_{-j^*}\|_2}{|x_{j^*}|}$$

for any j^* for which this bound is less than D.

Proof. The proof is identical to Lemma 2.2.1 but replacing $(i-j^*)^2 \leq n^2$ with $(h(i)-h(j^*))^2 \leq D^2$. □

Lemma 2.2.3. *Suppose there exists a j^* with $|x_{j^*}| \geq C\frac{B}{\delta^{3/2}}\|x_{-j^*}\|_2$ for some constant C and parameters B and δ. Then with two nonadaptive measurements,* NonAdaptiveShrink*$(x, B/\delta)$ returns a set $S \subset [n]$ such that $j^* \in S$, $\|x_{S\setminus\{j^*\}}\|_2 \leq \|x_{-j^*}\|_2/\sqrt{B}$, and $|S| \leq 1 + n/B$ with probability $1-3\delta$.*

Proof. Let $D = B/\delta$ and $h : [n] \to [D]$ be a pairwise independent hash function. Define $C_{j^*} = \{i \neq j^* : h(i) = h(j^*)\}$ to be the set of indices that "collide" with j^*. We have by pairwise independence that each $i \neq j^*$ has $\Pr[i \in C_{j^*}] = 1/D$, so

$$\mathbb{E}[|C_{j^*}|] = (n-1)/D < \delta n/B$$
$$\mathbb{E}[\|x_{C_{j^*}}\|_2^2] = \|x_{-j^*}\|_2^2/D = \delta\|x_{-j^*}\|_2^2/B$$

and so with probability $1-2\delta$ we have $|C_{j^*}| \leq n/B$ and $\|x_{C_{j^*}}\|_2 \leq \|x_{-j^*}\|_2/\sqrt{B}$, so

$$S = \{i : h(i) = h(j^*)\} = C_{j^*} \cup \{j^*\}$$

would be a satisfactory output. By Lemma 2.2.2, we have with probability $1-\delta$ that

$$|\frac{u'(x)}{u(x)} - h(j^*)| \leq 2\sqrt{2}\frac{D}{\sqrt{\delta}}\frac{\|x_{-j^*}\|_2}{x_{j^*}} \leq 2\sqrt{2}\frac{B}{\delta^{3/2}}\frac{\delta^{3/2}}{CB} = \frac{2\sqrt{2}}{C} < 1/2$$

procedure NonAdaptiveShrink(x, D) ▷ Find smaller set S containing heavy coordinate x_{j^*}
  $s : [n] \to \{\pm 1\}$ and $h : [n] \to [D]$ pairwise independent.
  $u \leftarrow \sum s(i)x_i$ ▷ Observation
  $u' \leftarrow \sum h(i)s(i)x_i$ ▷ Observation
  $j \leftarrow \text{Round}(u'/u)$.
  return $\{i \mid h(i) = j\}$.
end procedure
procedure AdaptiveOneSparseRecovery(x) ▷ Recover heavy coordinate x_{j^*}
  $S \leftarrow [n]$
  $B \leftarrow 2$, $\delta \leftarrow 1/4$
  while $|S| > 1$ **do**
    $S \leftarrow \text{NonAdaptiveShrink}(x_S, 4B/\delta)$
    $B \leftarrow B^{3/2}$, $\delta \leftarrow \delta/2$.
  end while
  return The single index in S
end procedure

Algorithm 2.2.1: Adaptive 1-sparse recovery

for sufficiently large constant C. But then $\frac{u'(x)}{u(x)}$ rounds to $h(j^*)$, so the output is satisfactory with $1 - 3\delta$ probability. □

Lemma 2.2.4 (1-sparse recovery)**.** *Suppose there exists a j^* with $|x_{j^*}| \geq C\|x_{[n]\setminus\{j^*\}}\|_2$ for some sufficiently large constant C. Then $O(\log\log n)$ adaptive measurements suffice to recover j^* with probability $1/2$.*

Proof. Let C' be the constant from Lemma 2.2.3. Define $B_0 = 8$ and $B_{r+1} = B_r^{3/2}/(2\sqrt{2})$ for $r \geq 0$. Define $\delta_r = 2^{-r}/12$ for $r \geq 0$. Suppose $C \geq C'B_0/\delta_0^{3/2}$.

Define $R = O(\log\log n)$ so $B_R \geq n$. Starting with $S_0 = [n]$, our algorithm iteratively applies Lemma 2.2.3 with parameters $B = B_r$ and $\delta = \delta_r$ to x_{S_r} to identify a set $S_{r+1} \subset S_r$ with $j^* \in S_{r+1}$.

We prove by induction that Lemma 2.2.3 applies at the ith iteration. We chose C to match the base case. For the inductive step, suppose $\|x_{S_r\setminus\{j\}}\|_2 \leq |x_{j^*}|/(C'\frac{B_r}{\delta_r^{3/2}})$. Then by Lemma 2.2.3,

$$\|x_{S_{r+1}\setminus\{j^*\}}\|_2 \leq \|x_{S_r\setminus\{j^*\}}\|_2/\sqrt{B_r} \leq |x_{j^*}|/(C'\frac{B_r^{3/2}}{\delta_r^{3/2}}) = |x_{j^*}|/(C'\frac{B_{r+1}}{\delta_{r+1}^{3/2}})$$

so the lemma applies in the next iteration as well, as desired.

After r iterations, we have $S_r \leq 1 + n/B_r^2 < 2$, so we have uniquely identified $j^* \in S_r$. The probability that any iteration fails is at most $\sum_{r\geq 0} 3\delta_r < 6\delta_0 = 1/2$. □

2.3 k-sparse recovery

Given a 1-sparse recovery algorithm using m measurements, one can use subsampling to build a k-sparse recovery algorithm using $O(km)$ measurements and achieving constant success probability. Our method for doing so is quite similar to one used in [GLPS10]. The main difference is that, in order to identify one large coefficient among a subset of coordinates, we use the adaptive algorithm from the previous section as opposed to error-correcting codes.

For intuition, straightforward subsampling at rate $1/k$ will, with constant probability, recover (say) 90% of the heavy hitters using $O(km)$ measurements. This reduces the problem to $k/10$-sparse recovery: we can subsample at rate $10/k$ and recover 90% of the remainder with $O(km/10)$ measurements, and repeat $\log k$ times. The number of measurements decreases geometrically, for $O(km)$ total measurements. Naively doing this would multiply the failure probability and the approximation error by $\log k$; however, we can make the number of measurements decay less quickly than the sparsity. This allows the failure probability and approximation ratios to also decay exponentially so their total remains constant.

To determine the number of rounds, note that the initial set of $O(km)$ measurements can be done in parallel for each subsampling, so only $O(m)$ rounds are necessary to get the first 90% of heavy hitters. Repeating $\log k$ times would require $O(m \log k)$ rounds. However, we can actually make the sparsity in subsequent iterations decay super-exponentially, in particular as a power tower. This give $O(m \log^* k)$ rounds.

Theorem 2.3.1. *There exists an adaptive $(1+\epsilon)$-approximate k-sparse recovery scheme with $O(\frac{1}{\epsilon}k \log \frac{1}{\delta} \log\log(n\epsilon/k))$ measurements and success probability $1-\delta$. It uses $O(\log^* k \log\log(n\epsilon))$ rounds.*

To prove this, we start from the following lemma:

Lemma 2.3.2. *We can perform $O(\log\log(n/k))$ adaptive measurements and recover an $\widehat{\imath}$ such that, for any $j \in H_{k,1}(x)$ we have $\Pr[\widehat{\imath} = j] = \Omega(1/k)$.*

Proof. Let $S = H_k(x)$. Let $T \subset [n]$ contain each element independently with probability $p = 1/(4C^2k)$, where C is the constant in Lemma 2.2.4. Let $j \in H_{k,1}(x)$. Then we have

$$\mathbb{E}[\|x_{T\setminus S}\|_2^2] = p\|x_{\overline{S}}\|_2^2$$

so with probability at least 3/4,

$$\|x_{T\setminus S}\|_2 \le \sqrt{4p}\|x_{\overline{S}}\|_2 = \frac{1}{C\sqrt{k}}\|x_{\overline{S}}\|_2 \le |x_j|/C$$

where the last step uses that $j \in H_{k,1}(x)$. Furthermore we have $\mathbb{E}[|T \setminus S|] < pn$ so $|T \setminus S| < n/k$ with probability at least $1 - 1/(4C^2) > 3/4$. By the union bound, both these events occur with probability at least 1/2.

Independently of this, we have

$$\Pr[T \cap S = \{j\}] = p(1-p)^{k-1} > p/e$$

so all these events hold with probability at least $p/(2e)$. Assuming this,

$$\|x_{T\setminus\{j\}}\|_2 \le |x_j|/C$$

and $|T| \le 1 + n/k$. But then Lemma 2.2.4 applies, and $O(\log\log|T|) = O(\log\log(n/k))$ measurements of x_T can recover j with probability 1/2. This is independent of the previous probability, for a total success chance of $p/(4e) = \Omega(1/k)$. □

Lemma 2.3.3 (Partial k-sparse recovery)**.** *With $O(\frac{1}{\epsilon}k \log \frac{1}{f\delta} \log\log(n\epsilon/k))$ adaptive measurements, we can recover T with $|T| \le k$ and*

$$\mathrm{Err}_{fk}^2(x_{\overline{T}}) \le (1+\epsilon)\,\mathrm{Err}_k^2(x)$$

with probability at least $1-\delta$*. The number of rounds required is* $O(\log\log(n\epsilon/k))$*.*

Proof. Repeat Lemma 2.3.2 $m = O(\frac{1}{\epsilon}k\log\frac{1}{f\delta})$ times in parallel with parameters n and k/ϵ to get coordinates $T' = \{t_1, t_2, \ldots, t_m\}$. For each $j \in H_{k,\epsilon}(x) \subseteq H_{k/\epsilon,\epsilon}(x)$ and $i \in [m]$, the lemma implies $\Pr[j = t_i] \geq \epsilon/(Ck)$ for some constant C. Then $\Pr[j \notin T'] \leq (1-\epsilon/(Ck))^m \leq e^{-\epsilon m/(Ck)} \leq f\delta$ for appropriate m. Thus

$$\mathbb{E}[|H_{k,\epsilon}(x) \setminus T'|] \leq f\delta|H_{k,\epsilon}(x)| \leq f\delta k$$
$$\Pr\left[|H_{k,\epsilon}(x) \setminus T'| \geq fk\right] \leq \delta.$$

Now, observe $x_{T'}$ directly and set $T \subseteq T'$ to be the locations of the largest k values. Then, since $H_{k,\epsilon}(x) \subseteq H_k(x)$, $|H_{k,\epsilon}(x) \setminus T| = |H_{k,\epsilon}(x) \setminus T'| \leq fk$ with probability at least $1-\delta$.

Suppose this occurs, and let $y = x_{\overline{T}}$. Then

$$\begin{aligned}
\operatorname{Err}^2_{fk}(y) &= \min_{|S|\leq fk} \|y_{\overline{S}}\|_2^2 \\
&\leq \|y_{\overline{H_{k,\epsilon}(x)\setminus T}}\|_2^2 \\
&\leq \|x_{\overline{H_{k,\epsilon}(x)}}\|_2^2 \\
&= \|x_{\overline{H_k(x)}}\|_2^2 + \|x_{H_k(x)\setminus H_{k,\epsilon}(x)}\|_2^2 \\
&\leq \|x_{\overline{H_k(x)}}\|_2^2 + k\|x_{H_k(x)\setminus H_{k,\epsilon}(x)}\|_\infty^2 \\
&\leq (1+\epsilon)\|x_{\overline{H_k(x)}}\|_2^2 \\
&= (1+\epsilon)\operatorname{Err}^2_k(x)
\end{aligned}$$

as desired. □

procedure AdaptiveKSparseRecovery(x, k, ϵ, δ) ▷ Recover approximation $\widehat{x}$ of x
  $R_0 \leftarrow [n]$
  $\delta_0 \leftarrow \delta/2$, $\epsilon_0 \leftarrow \epsilon/e$, $f_0 \leftarrow 1/32$, $k_0 \leftarrow k$.
  $J \leftarrow \{\}$
  for $i \leftarrow 0, \ldots, O(\log^* k)$ **do** ▷ While $k_i \geq 1$
    for $t \leftarrow 0, \ldots, \Theta(\frac{1}{\epsilon_i}k_i\log\frac{1}{\delta_i})$ **do**
      $S_t \leftarrow$ Subsample($R_i, \Theta(\epsilon_i/k_i)$)
      J.add(AdaptiveOneSparseRecovery(x_{S_t}))
    end for
    $R_{i+1} \leftarrow [n] \setminus J$
    $\delta_{i+1} \leftarrow \delta_i/8$
    $\epsilon_{i+1} \leftarrow \epsilon_i/2$
    $f_{i+1} \leftarrow 1/2^{1/(4^{i+1}f_i)}$
    $k_{i+1} \leftarrow k_i f_i$
  end for
  $\widehat{x} \leftarrow x_J$ ▷ Direct observation
  return $\widehat{x}$
end procedure

Algorithm 2.3.1: Adaptive k-sparse recovery

Theorem 2.3.4. *We can perform $O(\frac{1}{\epsilon}k\log\frac{1}{\delta}\log\log(n\epsilon/k))$ adaptive measurements and recover a set T of size at most $2k$ with*

$$\|x_{\overline{T}}\|_2 \le (1+\epsilon)\|x_{\overline{H_k(x)}}\|_2.$$

with probability $1-\delta$. The number of rounds required is $O(\log^* k \log\log(n\epsilon))$.

Proof. Define $\delta_i = \frac{\delta}{2\cdot 2^i}$ and $\epsilon_i = \frac{\epsilon}{e\cdot 2^i}$. Let $f_0 = 1/32$ and $f_i = 2^{-1/(4^i f_{i-1})}$ for $i > 0$, and define $k_i = k\prod_{j<i} f_j$. Let $R_0 = [n]$.

Let $r = O(\log^* k)$ such that $f_{r-1} < 1/k$. This is possible since $\alpha_i = 1/(4^{i+1}f_i)$ satisfies the recurrence $\alpha_0 = 8$ and $\alpha_i = 2^{\alpha_{i-1}-2i-2} > 2^{\alpha_{i-1}/2}$. Thus $\alpha_{r-1} > k$ for $r = O(\log^* k)$ and then $f_{r-1} < 1/\alpha_{r-1} < 1/k$.

For each round $i = 0,\dots,r-1$, the algorithm runs Lemma 2.3.3 on x_{R_i} with parameters ϵ_i, k_i, f_i, and δ_i to get T_i. It sets $R_{i+1} = R_i \setminus T_i$ and repeats. At the end, it outputs $T = \cup T_i$.

The total number of measurements is of order

$$\begin{aligned}\sum \frac{1}{\epsilon_i}k_i\log\frac{1}{f_i\delta_i}\log\log(n\epsilon_i/k_i) &\lesssim \sum \frac{2^i(k_i/k)\log(1/f_i)}{\epsilon}k(i+\log\frac{1}{\delta})\log(\log(k/k_i)+\log(n\epsilon/k))\\ &\lesssim \frac{1}{\epsilon}k\log\frac{1}{\delta}\log\log(n\epsilon/k)\sum 2^i(k_i/k)\log(1/f_i)(i+1)\log\log(k/k_i)\end{aligned}$$

using the very crude bounds $i+\log(1/\delta) \le (i+1)\log(1/\delta)$ and $\log(a+b) \le 2\log a\log b$ for $a,b \ge e$. But then

$$\begin{aligned}\sum 2^i(k_i/k)\log(1/f_i)(i+1)\log\log(k/k_i) &\le \sum 2^i(i+1)f_i\log(1/f_i)\log\log(1/f_i)\\ &\lesssim \sum 2^i(i+1)\sqrt{f_i}\\ &\lesssim 1\end{aligned}$$

since $f_i \lesssim /16^i$, giving $O(\frac{1}{\epsilon}k\log\frac{1}{\delta}\log\log(n\epsilon/k))$ total measurements. The probability that any of the iterations fail is at most $\sum\delta_i < \delta$. The result has size $|T| \le \sum k_i \le 2k$. All that remains is the approximation ratio $\|x_{\overline{T}}\|_2 = \|x_{R_r}\|_2$.

For each i, we have

$$\mathrm{Err}^2_{k_{i+1}}(x_{R_{i+1}}) = \mathrm{Err}^2_{f_ik_i}(x_{R_i\setminus T_i}) \le (1+\epsilon_i)\,\mathrm{Err}^2_{k_i}(x_{R_i}).$$

Furthermore, $k_r < kf_{r-1} < 1$. Hence

$$\|x_{R_r}\|_2^2 = \mathrm{Err}^2_{k_r}(x_{R_r}) \le \left(\prod_{i=0}^{r-1}(1+\epsilon_i)\right)\mathrm{Err}^2_{k_0}(x_{R_0}) = \left(\prod_{i=0}^{r-1}(1+\epsilon_i)\right)\mathrm{Err}^2_k(x)$$

But $\prod_{i=0}^{r-1}(1+\epsilon_i) < e^{\sum\epsilon_i} < e$, so

$$\prod_{i=0}^{r-1}(1+\epsilon_i) < 1+\sum e\epsilon_i \le 1+2\epsilon$$

and hence

$$\|x_{\overline{T}}\|_2 = \|x_{R_r}\|_2 \le (1+\epsilon)\|x_{\overline{H_k(x)}}\|_2$$

as desired. □

Once we find the support T, we can observe x_T directly with $O(k)$ measurements to get a

$(1+\epsilon)$-approximate k-sparse recovery scheme, proving Theorem 2.3.1.

Chapter 3

Nonadaptive Upper Bound for High SNR

(Based on parts of [PW12])

This chapter gives an upper bound for C-approximate sparse recovery that uses $O(k \log^* k \log_C(n/k))$ measurements for $C \gg 1$. This matches, up to a $\log^* k$ factor, the lower bound in Chapter 6. Previous results were optimal when $C = O(1)$ or $C = n^{\Omega(1)}$ (see [CRT06b] for $C = 1 + \Theta(1)$, [GLPS10] for $C = 1 + o(1)$, and in a slightly different setting [EG07, BJCC12] for $C = n^{\Omega(1)}$), but this is the first result we are aware of that performs well in the intermediate regime.

We first focus on recovery of a single heavy coordinate. We then study "partial" sparse recovery, i.e. recovery of 90% of the heavy hitters for general k. We conclude with recovery of all the heavy hitters.

3.1 1-sparse recovery

We observe $2r$ measurements, for some $r = O(\log_C n)$. Let $D = C/16$. For $i \in [r]$, we choose pairwise independent hash functions $h\colon [n] \to [D]$ and $s\colon [n] \to \{\pm 1\}$. We then observe

$$u = \sum_j s(j)x_j \qquad\qquad u' = \sum_j h(j)s(j)x_j.$$

Lemma 3.1.1 (1-sparse recovery)**.** *Suppose there exists a $j^* \in [n]$ such that $|x_{j^*}| \geq C\|x_{-j^*}\|_2$. Then if C is larger than a sufficiently large constant, we can choose $r = O(\log_C(n/\delta))$ and $D = \Theta(\sqrt{C})$ so that* Locate1Sparse *returns j^* with probability $1 - \delta$.*

Proof. These measurements u_i and u'_i are identical to those used and analyzed for 1-sparse recovery in Chapter 2. By Lemma 2.2.2, for each $i \in [r]$ with probability $1 - 1/C$ over s_i we have

$$\left|\frac{u'_i}{u_i} - h_i(j^*)\right| \leq 2\sqrt{2}\frac{D}{\sqrt{1/C}}\frac{\|x_{-j^*}\|_2}{|x_{j^*}|} < 1/2$$

for appropriate constant in $D = \Theta(\sqrt{C})$. Hence for $\alpha_i = \text{ROUND}(\frac{u'_i}{u_i})$, we have

$$\Pr[\alpha_i \neq h_i(j^*)] \leq 1/C. \tag{3.1}$$

```
procedure LOCATE1SPARSE(x, C)
    for i ∈ [r] do                                          ▷ r = O(log_C(n/δ))
        Choose h_i : [n] → [D] and s_i : [n] → {±1} pairwise independent.   ▷ D = √C/16
        Sample u_i = Σ_j s_i(j)x_j and u'_i = Σ_j h_i(j)s_i(j)x_j.
        α_i ← ROUND(u'/u)
    end for
    c_j ← |{i ∈ [r] | h_i(j) = α_i}| for j ∈ [n].
    S ← {j ∈ [n] | c_j > r/2}.
    if |S| = 1 then
        return the single element j ∈ S
    else
        return ⊥
    end if
end procedure
```

Algorithm 3.1.1: Nonadaptive 1-sparse location

This is independent for each $i \in [r]$, so the probability this happens $r/2$ times is

$$\Pr[j^* \notin S] \le \binom{r}{r/2}(1/C)^{r/2} \le (4/C)^{r/2} = C^{-\Omega(r)} \le \delta/(2n).$$

for $r = \Theta(\log_C(n/\delta))$. Similarly, we have for $j \neq j^*$ that

$$\Pr[\alpha_i = h_i(j)] \le \Pr[h_i(j) = h_i(j^*)] + \Pr[\alpha_i \neq h_i(j^*)] \le 1/D + 1/C \le 2/D.$$

Hence

$$\Pr[j \in S] \le \binom{r}{r/2}(2/D)^{r/2} \le D^{-\Omega(r)} \le \delta/(2n).$$

Therefore a union bound gives that $S = \{j^*\}$ with probability $1 - \delta$. □

3.2 Partial k-sparse recovery

3.2.1 Location

For general k, we locate a set L of $O(k)$ coordinates by partitioning the coordinates into $B = O(k)$ sets of size n/B and applying LOCATE1SPARSE. To be specific, we use pairwise independent hash functions $h\colon [n] \to [B]$ to partition into B sets $\{i : h(i) = u\}$ for each $u \in [B]$.

The algorithm LOCATEMOST performs this for $R = O(\log(1/f\delta))$ different hash functions h and outputs the set of indices $j \in [n]$ that are located in most of the hashing h.

First we analyze the probability that a "heavy hitter" is located in a single hashing:

Lemma 3.2.1. *Each round of* LOCATEMOST *uses $O(k \log_C(n/k))$ measurements and returns a set L of size $O(k)$ such that each $j \in H_{k,C}(x)$ has $j \in L$ with probability at least $3/4$.*

Proof. Let $h : [n] \to [B]$ be the pairwise independent hash function used in this round. Let $S = H_k(x)$ denote the indices of the largest k coefficients of x, and let $j \in H_{k,C}(x) \subset S$ so $|x_j|^2 \ge \frac{C}{k}\operatorname{Err}_k^2(x)$. Define $C_j = \{i \neq j : h(i) = h(j)\}$ to be the set of elements that "collide" with j.

procedure LocateMost(x, k, f, δ, C)
  for $r \leftarrow [R]$ **do** ▷ $R = O(\log(1/f\delta))$
    Choose $h : [n] \to [B]$ pairwise independent. ▷ $B = O(k)$
    $L_r \leftarrow \{\text{Locate1Sparse}(x \text{ restricted to } \{j : h(j) = i\}, \sqrt{C}) \mid i \in [k]\}$
  end for
  $c_j \leftarrow |\{r \mid j \in L_r\}|$ for $j \in [n]$.
  $L \leftarrow \{j \mid c_j > R/2\}$
  return $\tilde{x}_L$
end procedure
procedure EstimateMost(x, k, f, δ, L)
  for $r \leftarrow [R]$ **do** ▷ $R = O(\log(1/f\delta))$
    Choose $h : [n] \to [B]$ and $s : [n] \to \{\pm 1\}$ pairwise independent. ▷ $B = O(k)$
    Observe $y_j = \sum_{h(i)=j} s(i)x_i$ for each $j \in [B]$.
    $\tilde{x}_i^{(r)} \leftarrow s(i)y_{h(i)}$ for each $i \in L$.
  end for
  $\tilde{x}_i \leftarrow \text{median}_r\, \tilde{x}_i^{(r)}$
  return $\tilde{x}_L$
end procedure

Algorithm 3.2.1: Nonadaptive partial k-sparse recovery

By pairwise independence of h we have that

$$\mathbb{E}[|C_j|] \le n/B$$
$$\mathbb{E}[\|x_{C_j \setminus S}\|_2^2] = \frac{1}{B}\operatorname{Err}_k^2(x)$$
$$\mathbb{E}[|C_j \cap S|] < k/B.$$

Hence if $B \ge 24k$, with 7/8 probability we will have that

$$|C_j| \le n/k$$
$$\|x_{C_j \setminus S}\|_2^2 = \frac{1}{k}\operatorname{Err}_k^2(x)$$
$$|C_j \cap S| < 1.$$

In this case, $T = \{i : h(i) = h(j)\} = C_j \cup \{j\}$ has

$$\|x_{T \setminus \{j\}}\|_2 \le |x_j|/\sqrt{C}$$

and $|T| \le 1 + n/k$. Therefore x restricted to T is a size $1 + n/k$ vector, and Lemma 3.1.1 shows that with 7/8 probability Locate1Sparse(x restricted to T, $\sqrt{C}$) returns j using $\log_{\sqrt{C}}(8|T|) \lesssim \log_C(n/k)$ measurements. □

Corollary 3.2.2. *With $O(k \log_C(n/k) \log(1/f\delta))$ measurements,* LocateMost *returns a set L of size $O(k)$ such that each $j \in H_{k,C}(x)$ has $j \in L$ with probability at least $1 - f\delta$.*

Proof. We repeat the method of Lemma 3.2.1 $O(\log(1/f\delta))$ times, and take all coordinates that are listed in more than half the sets L_i. This at most doubles the output size, and by a Chernoff bound each $j \in H_{k,C}(x)$ lies in the output with probability at least $1 - f\delta$. □

Corollary 3.2.2 gives a good method for finding the heavy hitters, but we also need to estimate them.

3.2.2 Estimation

We estimate using Count-Sketch [CCF02], with $R = O(\log(1/f\delta))$ hash tables of size $O(k/\epsilon)$.

Lemma 3.2.3. *Suppose $|L| \lesssim k$. With $O(k \log(\frac{1}{f\delta}))$ measurements,* EstimateMost *returns $\tilde{x}_L$ so that with probability $1-\delta$ we have*

$$\mathrm{Err}^2_{fk}(x_L - \tilde{x}_L) < \mathrm{Err}^2_k(x)$$

Proof. Consider any $j \in [n]$ and round $r \in [R]$ with hash functions h, s. Let $C_j = \{i \neq j : h(i) = h(j)\}$. We have

$$\tilde{x}^{(r)} - x_j = s(j) \sum_{i \in C_j} s(i) x_i$$

and so

$$\mathop{\mathbb{E}}_{s}[(\tilde{x}^{(r)} - x_j)^2] = \|x_{C_j}\|_2^2.$$

Hence for each j and r, with $7/8$ probability we have

$$(\tilde{x}^{(r)} - x_j)^2 \leq 8\|x_{C_j}\|_2^2.$$

Now, by pairwise independence of h we have that

$$\mathop{\mathbb{E}}_{h}[|C_j \cap H_k(x)|] \leq k/B$$
$$\mathop{\mathbb{E}}_{h}[\|x_{C_j \setminus H_k(x)}\|_2^2] = \frac{1}{B}\mathrm{Err}^2_k(x)$$

so if $B > 100k$ then we have that $|C_j \cap H_k(x)| = 0$ with $99/100$ probability and with $9/10$ probability $8\|x_{C_j \setminus H_k(x)}\|_2^2 \leq \frac{8 \cdot 10}{100}\frac{1}{k}\mathrm{Err}^2_k(x) < \frac{1}{k}\mathrm{Err}^2_k(x)$.

Hence by a union bound, with $1 - 1/100 - 1/8 - 1/10 > 3/4$ probability we have

$$(\tilde{x}^{(r)} - x_j)^2 < \frac{1}{k}\mathrm{Err}^2_k(x) \tag{3.2}$$

for each coordinate j and round r. By a Chernoff bound, using $R = O(\log 1/(f\delta))$ rounds, with at least $1 - f\delta$ probability we will have that more than $R/2$ of the $r \in [R]$ satisfy (3.2). In this case, the median $\tilde{x}_j$ must satisfy it as well.

Therefore with $1-\delta$ probability, at most fk of the $|L| \leq k$ elements $j \in L$ have

$$(\tilde{x}_j - x_j)^2 \geq \frac{1}{k}\mathrm{Err}^2_k(x),$$

giving the result. □

3.2.3 Combining the two

Lemma 3.2.4 (Partial sparse recovery)**.** *The result $\tilde{x}_L$ of* LocateMost *followed by* EstimateMost *satisfies*

$$\operatorname{Err}^2_{fk}(x - \tilde{x}_L) \le C \operatorname{Err}^2_k(x)$$

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

Proof. Let T contain the largest k coordinates of x. By Corollary 3.2.2, each $j \in H_{k,C}(x)$ has $j \in L$ with probability $1 - f\delta$, so with probability $1 - \delta$ we have $|H_{k,C}(x) \setminus L| \le fk$. Therefore

$$\begin{aligned}
\operatorname{Err}^2_{fk}(x_{\overline{L}}) &\le \|x_{\overline{H_{k,C}(x) \cup L}}\|_2^2 \\
&\le \|x_{\overline{H_{k,C}(x)}}\|_2^2 \\
&= \|x_{\overline{H_k(x)}}\|_2^2 + \|x_{H_k(x) \setminus H_{k,C}(x)}\|_2^2 \\
&\le \operatorname{Err}^2_k(x) + k\|x_{H_{k,C}(x)}\|_\infty^2 \\
&\le (1 + C) \operatorname{Err}^2_k(x)
\end{aligned}$$

and so

$$\begin{aligned}
\operatorname{Err}^2_{2fk}(x - \tilde{x}_L) &\le \operatorname{Err}^2_{fk}(x_L - \tilde{x}_L) + \operatorname{Err}^2_{fk}(x_{\overline{L}}) \\
&\le (2 + C) \operatorname{Err}^2_k(x) \\
&\le 3C \operatorname{Err}^2_k(x)
\end{aligned}$$

with probability $1 - \delta$ by Lemma 3.2.3. Rescale f, δ, and C to get the result. □

3.3 General k-sparse recovery

procedure RecoverAll(x, k, C)
  Choose k_r, f_r, δ_r, D per proof of Theorem 3.3.1
  $\chi^{(0)} \leftarrow 0$
  for $r \leftarrow 0, 1, \ldots, r-1$ **do**
    $L^{(r)} \leftarrow$ LocateMost$(x - \chi^{(r)}, k_r, f_r, \delta_r, D)$
    $\tilde{x}^{(r)} \leftarrow$ EstimateMost$(x - \chi^{(r)}, k_r, f_r, \delta_r, L)$
    $\chi^{(r+1)} \leftarrow \chi^{(r)} + \tilde{x}^{(r)}$
  end for
  return $\chi^{(R)}$
end procedure

Algorithm 3.3.1: Nonadaptive general k-sparse recovery

Theorem 3.3.1. RecoverAll *achieves C-approximate ℓ_2/ℓ_2 sparse recovery with $O(k \log_C(n/k) \log^* k)$ measurements and $3/4$ success probability.*

Proof. We will achieve $D^{O(\log^* k)}$-approximate recovery using $O(k \log_D(n/k))$ measurements. Substituting $\log C = \log D \log^* k$ gives the result.

Define $\delta_i = \frac{1}{8 \cdot 2^i}$. Let $f_0 = 1/16$ and $f_{i+1} = 2^{-1/(4^i f_i)}$. Let $k_i = k \prod_{j<i} f_j$. Then for $R = O(\log^* k)$, $k_R < 1$.

We set $\tilde{x}^{(0)} = 0$, and iterate LocateMost and EstimateMost on $x - \chi^{(r)}$ in each round r with δ_r, f_r, k_r, D as parameters, getting update $\tilde{v}^{(r)}$ and setting $\chi^{(r+1)} = \chi^{(r)} + \tilde{v}^{(r)}$.

The probability that Lemma 3.2.4 fails at any stage is at most $\sum \delta_i < 1/4$. Assuming it does not fail, the error guarantee telescopes, giving

$$\operatorname{Err}^2_{k_r}(x - \chi^{(r)}) \leq D^r \operatorname{Err}^2_k(x)$$

so $\|x - \chi^{(R)}\|_2^2 \leq D^R \operatorname{Err}^2_k(x)$, which is $D^{O(\log^* k)}$-approximate recovery.

The total number of measurements is

$$\begin{aligned}
&\sum_{i=0}^{R} k_i \log_D(n/k_i) \log(\frac{1}{\delta_i f_i}) \\
=&\sum_{i=0}^{R} k(\prod_{j<i} f_j) \log_D(\frac{n}{k} \prod_{j<i}(1/f_j))(3 + i + \frac{1}{4^i f_{i-1}}) \\
\leq& 2\sum_{i=0}^{R} k\frac{1}{4^i}(\prod_{j<i-1} f_j) \log_D(\frac{n}{k} \prod_{j<i}(1/f_j)) \\
=&O(k \log_D \frac{n}{k}) + \frac{k}{\log D} \sum_{i=0}^{R} \frac{2}{4^i}(\prod_{j<i-1} f_j) \sum_{j<i} \frac{1}{4^j f_{j-1}} \\
=&O(k \log_D \frac{n}{k}).
\end{aligned}$$

□

Chapter 4

Sparse Fourier Transform: Optimizing Time

(Based on parts of [HIKP12a])

4.1 Introduction

The discrete Fourier transform (DFT) is one of the most important and widely used computational tasks. Its applications are broad and include signal processing, communications, and audio/image/video compression. Hence, fast algorithms for DFT are highly valuable. Currently, the fastest such algorithm is the Fast Fourier Transform (FFT), which computes the DFT of an n-dimensional signal in $O(n \log n)$ time. The existence of DFT algorithms faster than FFT is one of the central questions in the theory of algorithms.

A general algorithm for computing the exact DFT must take time at least proportional to its output size, i.e., $\Omega(n)$. In many applications, however, most of the Fourier coefficients of a signal are small or equal to zero, i.e., the output of the DFT is (approximately) *sparse*. This is the case for video signals, where a typical 8x8 block in a video frame has on average 7 non-negligible frequency coefficients (i.e., 89% of the coefficients are negligible) [CGX96]. Images and audio data are equally sparse. This sparsity provides the rationale underlying compression schemes such as MPEG and JPEG. Other sparse signals appear in computational learning theory [KM91, LMN93], analysis of Boolean functions [KKL88, O'D08], compressed sensing [Don06, CRT06a], multi-scale analysis [DRZ07], similarity search in databases [AFS93], spectrum sensing for wideband channels [LVS11], and datacenter monitoring [MNL10].

For sparse signals, the $\Omega(n)$ lower bound for the complexity of DFT no longer applies. If a signal has a small number k of nonzero Fourier coefficients – the *exactly k-sparse* case – the output of the Fourier transform can be represented succinctly using only k coefficients. Hence, for such signals, one may hope for a DFT algorithm whose runtime is sublinear in the signal size n. Even for a general n-dimensional signal x – the *general case* – one can find an algorithm that computes the best *k-sparse approximation* of its Fourier transform $\widehat{x}$ in sublinear time. The goal of such an algorithm is to compute an approximation vector $\widehat{x}'$ that satisfies the following ℓ_2/ℓ_2 *guarantee*:

$$\|\widehat{x} - \widehat{x}'\|_2 \leq C \min_{k\text{-sparse } y} \|\widehat{x} - y\|_2, \tag{4.1}$$

where C is some approximation factor and the minimization is over k-sparse signals. We allow the algorithm to be *randomized*, and only succeed with constant (say, 2/3) probability.

The past two decades have witnessed significant advances in sublinear sparse Fourier algorithms.

The first such algorithm (for the Hadamard transform) appeared in [KM91] (building on [GL89]). Since then, several sublinear sparse Fourier algorithms for complex inputs have been discovered [Man92, GGI+02a, AGS03, GMS05, Iwe10, Aka10, HIKP12c]. These algorithms provide[1] the guarantee in Equation (4.1).[2]

The main value of these algorithms is that they outperform FFT's runtime for sparse signals. For very sparse signals, the fastest algorithm is due to [GMS05] and has $O(k \log^c(n) \log(n/k))$ runtime, for some[3] $c > 2$. This algorithm outperforms FFT for any k smaller than $\Theta(n/\log^a n)$ for some $a > 1$. For less sparse signals, the fastest algorithm is due to [HIKP12c], and has $O(\sqrt{nk} \log^{3/2} n)$ runtime. This algorithm outperforms FFT for any k smaller than $\Theta(n/\log n)$.

Despite impressive progress on sparse DFT, the state of the art suffers from two main limitations:

1. None of the existing algorithms improves over FFT's runtime for the whole range of sparse signals, i.e., $k = o(n)$.

2. Most of the aforementioned algorithms are quite complex, and suffer from large "big-Oh" constants (the algorithm of [HIKP12c] is an exception, but has a running time that is polynomial in n).

Results. In this chapter, we address these limitations by presenting two new algorithms for the sparse Fourier transform. We require that the length n of the input signal is a power of 2. We show:

- An $O(k \log n)$-time algorithm for the exactly k-sparse case, and
- An $O(k \log n \log(n/k))$-time algorithm for the general case.

The key property of both algorithms is their ability to achieve $o(n \log n)$ time, and thus improve over the FFT, for *any* $k = o(n)$. These algorithms are the first known algorithms that satisfy this property. Moreover, if one assume that FFT is optimal and hence the DFT cannot be computed in less than $O(n \log n)$ time, the algorithm for the exactly k-sparse case is *optimal*[4] as long as $k = n^{\Omega(1)}$. Under the same assumption, the result for the general case is at most one $\log \log n$ factor away from the optimal runtime for the case of "large" sparsity $k = n/\log^{O(1)} n$.

Furthermore, our algorithm for the exactly sparse case (depicted as Algorithm 4.3.1 on page 5) is quite simple and has low big-Oh constants. In particular, our preliminary implementation of a variant of this algorithm is faster than FFTW, a highly efficient implementation of the FFT, for $n = 2^{22}$ and $k \leq 2^{17}$ [HIKP12b]. In contrast, for the same signal size, prior algorithms were faster than FFTW only for $k \leq 2000$ [HIKP12c].[5]

In Chapter 6 we complement our algorithmic results by showing that any algorithm that works for the general case must use at least $\Omega(k \log(n/k)/\log \log n)$ samples of x. This bound holds even for *adaptive* sampling, where the algorithm selects the samples based on the values of the previously sampled coordinates. Note that our algorithms are *nonadaptive*, and thus limited by the more stringent $\Omega(k \log(n/k))$ lower bound.

[1] The algorithm of [Man92], as stated in the paper, addresses only the exactly k-sparse case. However, it can be extended to the general case using relatively standard techniques.

[2] All of the above algorithms, as well as the algorithms in this chapter, need to make some assumption about the precision of the input; otherwise, the right-hand-side of the expression in Equation (4.1) contains an additional additive term. See Preliminaries for more details.

[3] The paper does not estimate the exact value of c. We estimate that $c = 3$.

[4] One also needs to assume that k divides n. See Section 4.5 for more details.

[5] Note that both numbers ($k \leq 2^{17}$ and $k \leq 2000$) are for the exactly k-sparse case. The algorithm in [HIKP12c] can deal with the general case, but the empirical runtimes are higher.

Techniques – overview. We start with an overview of the techniques used in prior works. At a high level, sparse Fourier algorithms work by binning the Fourier coefficients into a small number of bins. Since the signal is sparse in the frequency domain, each bin is likely[6] to have only one large coefficient, which can then be located (to find its position) and estimated (to find its value). The binning has to be done in sublinear time, and thus these algorithms bin the Fourier coefficients using an n-dimensional filter vector G that is concentrated both in time and frequency. That is, G is zero except at a small *number* of time coordinates, and its Fourier transform $\widehat{G}$ is negligible except at a small *fraction* (about $1/k$) of the frequency coordinates, representing the filter's "pass" region. Each bin essentially receives only the frequencies in a narrow range corresponding to the pass region of the (shifted) filter $\widehat{G}$, and the pass regions corresponding to different bins are disjoint. In this chapter, we use filters introduced in [HIKP12c]. Those filters (defined in more detail in Preliminaries) have the property that the value of $\widehat{G}$ is "large" over a constant fraction of the pass region, referred to as the "super-pass" region. We say that a coefficient is "isolated" if it falls into a filter's super-pass region and no other coefficient falls into filter's pass region. Since the super-pass region of our filters is a constant fraction of the pass region, the probability of isolating a coefficient is constant.

To achieve the stated running times, we need a fast method for locating and estimating isolated coefficients. Further, our algorithm is iterative, so we also need a fast method for updating the signal so that identified coefficients are not considered in future iterations. Below, we describe these methods in more detail.

New techniques – location and estimation. Our location and estimation methods depends on whether we handle the exactly sparse case or the general case. In the exactly sparse case, we show how to estimate the position of an isolated Fourier coefficient using only two samples of the filtered signal. Specifically, we show that the phase difference between the two samples is linear in the index of the coefficient, and hence we can recover the index by estimating the phases. This approach is inspired by the frequency offset estimation in orthogonal frequency division multiplexing (OFDM), which is the modulation method used in modern wireless technologies (see [HT01], Chapter 2).

In order to design an algorithm[7] for the general case, we employ a different approach. Specifically, we can use two samples to estimate (with constant probability) individual bits of the index of an isolated coefficient. Similar approaches have been employed in prior work. However, in those papers, the index was recovered bit by bit, and one needed $\Omega(\log \log n)$ samples per bit to recover *all* bits correctly with constant probability. In contrast, in this chapter we recover the index one *block of bits* at a time, where each block consists of $O(\log \log n)$ bits. This approach is inspired by the fast sparse recovery algorithm of [GLPS10]. Applying this idea in our context, however, requires new techniques. The reason is that, unlike in [GLPS10], we do not have the freedom of using arbitrary "linear measurements" of the vector $\widehat{x}$, and we can only use the measurements induced by the inverse Fourier transform.[8] As a result, the extension from "bit recovery" to "block recovery" is the most technically involved part of the algorithm. Section 4.4.1 contains further intuition on this part.

[6] One can randomize the positions of the frequencies by sampling the signal in time domain appropriately [GGI+02a, GMS05]. See Preliminaries for the description.

[7] We note that although the two-sample approach employed in our algorithm works in theory only for the exactly k-sparse case, our preliminary experiments show that using a few more samples to estimate the phase works surprisingly well even for general signals.

[8] In particular, the method of [GLPS10] uses measurements corresponding to a random error correcting code.

New techniques – updating the signal. The aforementioned techniques recover the position and the value of any isolated coefficient. However, during each filtering step, each coefficient becomes isolated only with constant probability. Therefore, the filtering process needs to be repeated to ensure that each coefficient is correctly identified. In [HIKP12c], the algorithm simply performs the filtering $O(\log n)$ times and uses the median estimator to identify each coefficient with high probability. This, however, would lead to a running time of $O(k \log^2 n)$ in the k-sparse case, since each filtering step takes $k \log n$ time.

One could reduce the filtering time by subtracting the identified coefficients from the signal. In this way, the number of nonzero coefficients would be reduced by a constant factor after each iteration, so the cost of the first iteration would dominate the total running time. Unfortunately, subtracting the recovered coefficients from the signal is a computationally costly operation, corresponding to a so-called *non-uniform* DFT (see [GST08] for details). Its cost would override any potential savings.

In this chapter, we introduce a different approach: instead of subtracting the identified coefficients from the *signal*, we subtract them directly from the *bins* obtained by filtering the signal. The latter operation can be done in time linear in the number of subtracted coefficients, since each of them "falls" into only one bin. Hence, the computational costs of each iteration can be decomposed into two terms, corresponding to filtering the original signal and subtracting the coefficients. For the exactly sparse case these terms are as follows:

- The cost of filtering the original signal is $O(B \log n)$, where B is the number of bins. B is set to $O(k')$, where k' is the the number of yet-unidentified coefficients. Thus, initially B is equal to $O(k)$, but its value decreases by a constant factor after each iteration.
- The cost of subtracting the identified coefficients from the bins is $O(k)$.

Since the number of iterations is $O(\log k)$, and the cost of filtering is dominated by the first iteration, the total running time is $O(k \log n)$ for the exactly sparse case.

For the general case, we need to set k' and B more carefully to obtain the desired running time. The cost of each iterative step is multiplied by the number of filtering steps needed to compute the location of the coefficients, which is $\Theta(\log(n/B))$. If we set $B = \Theta(k')$, this would be $\Theta(\log n)$ in most iterations, giving a $\Theta(k \log^2 n)$ running time. This is too slow when k is close to n. We avoid this by decreasing B more slowly and k' more quickly. In the r-th iteration, we set $B = k/\mathrm{poly}(r)$. This allows the total number of bins to remain $O(k)$ while keeping $\log(n/B)$ small—at most $O(\log \log k)$ more than $\log(n/k)$. Then, by having k' decrease according to $k' = k/r^{\Theta(r)}$ rather than $k/2^{\Theta(r)}$, we decrease the number of rounds to $O(\log k / \log \log k)$. Some careful analysis shows that this counteracts the $\log \log k$ loss in the $\log(n/B)$ term, achieving the desired $O(k \log n \log(n/k))$ running time.

Organization of the chapter. In Section 4.2, we give notation and definitions used throughout the chapter. Sections 4.3 and 4.4 give our algorithm in the exactly k-sparse and the general case, respectively. Section 4.5 gives the reduction to the exactly k-sparse case from a k-dimensional DFT. Section 4.6 describes how to efficiently implement our filters.

4.2 Preliminaries

This section introduces the notation, assumptions, and definitions used in the rest of this chapter.

Note that the Fourier transform and the inverse Fourier transform are equivalent problems. Therefore, to simplify notation and make it consistent with the rest of this thesis, we will consider

the *inverse* Fourier transform problem: approximating a sparse vector x from samples of its Fourier transform $\widehat{x}$.

Notation. We use $[n]$ to denote the set $\{1, \ldots, n\}$, and define $\omega = e^{-2\pi \mathbf{i}/n}$ to be an nth root of unity. For any complex number a, we use $\phi(a) \in [0, 2\pi]$ to denote the *phase* of a. For a complex number a and a real positive number b, the expression $a \pm b$ denotes a complex number a' such that $|a - a'| \leq b$. For a vector $x \in \mathbb{C}^n$, its support is denoted by $\mathrm{supp}(x) \subset [n]$. We use $\|x\|_0$ to denote $|\mathrm{supp}(x)|$, the number of nonzero coordinates of x.

The Fourier transform of x is denoted by $\widehat{x}$, with

$$\widehat{x}_i = \frac{1}{\sqrt{n}} \sum_{j \in [n]} \omega^{ij} x_j.$$

The inverse transform is then

$$x_i = \frac{1}{\sqrt{n}} \sum_{j \in [n]} \omega^{-ij} \widehat{x}_j.$$

For a vector of length n, indices should be interpreted modulo n, so $x_{-i} = x_{n-i}$. This allows us to define *convolution*

$$(x * y)_i = \sum_{j \in [n]} x_j y_{i-j}$$

and the *coordinate-wise product* $(x \cdot y)_i = x_i y_i$, so $\widehat{x \cdot y} = \widehat{x} * \widehat{y}$.

When $i \in \mathbb{Z}$ is an index into an n-dimensional vector, sometimes we use $|i|$ to denote $\min_{j \equiv i \pmod n} |j|$.

Definitions. We use two tools introduced in previous papers: (pseudorandom) spectrum permutation [GGI+02a, GMS05, GST08] and flat filtering windows [HIKP12c].

Definition 4.2.1. *Suppose σ^{-1} exists mod n. We define the* permutation $P_{\sigma,a,b}$ *by*

$$(P_{\sigma,a,b}\widehat{x})_i = \widehat{x}_{\sigma(i+a)} \omega^{-\sigma b i}.$$

We also define $\pi_{\sigma,b}(i) = \sigma(i - b) \bmod n$.

Claim 4.2.2. *Let $\mathcal{F}^{-1}(x)$ denote the inverse Fourier transform of x. Then*

$$(\mathcal{F}^{-1}(P_{\sigma,a,b}\widehat{x}))_{\pi_{\sigma,b}(i)} = x_i \omega^{a\sigma i}.$$

Proof.

$$\begin{aligned}
\mathcal{F}^{-1}(P_{\sigma,a,b}\widehat{x})_{\sigma(i-b)} &= \frac{1}{\sqrt{n}} \sum_{j \in [n]} \omega^{-\sigma(i-b)j} (P_{\sigma,a,b}\widehat{x})_j \\
&= \frac{1}{\sqrt{n}} \sum_{j \in [n]} \omega^{-\sigma(i-b)j} \widehat{x}_{\sigma(j+a)} \omega^{-\sigma b j} \\
&= \omega^{a\sigma i} \frac{1}{\sqrt{n}} \sum_{j \in [n]} \omega^{-i\sigma(j+a)} \widehat{x}_{\sigma(j+a)} \\
&= x_i \omega^{a\sigma i}.
\end{aligned}$$

□

Definition 4.2.3. *We say that* $(\widehat{G}, G') = (\widehat{G}_{B,\delta,\alpha}, G'_{B,\delta,\alpha}) \in \mathbb{R}^n \times \mathbb{R}^n$ *is a* flat window function *with parameters* $B \geq 1$, $\delta > 0$, *and* $\alpha > 0$ *if* $|\operatorname{supp}(\widehat{G})| = O(\frac{B}{\alpha} \log(n/\delta))$ *and* G' *satisfies*

- $G'_i = 1$ *for* $|i| \leq (1-\alpha)n/(2B)$
- $G'_i = 0$ *for* $|i| \geq n/(2B)$
- $G'_i \in [0,1]$ *for all* i
- $\|G' - G\|_\infty < \delta$.

The above notion corresponds to the $(1/(2B), (1-\alpha)/(2B), \delta, O(B/\alpha \log(n/\delta))$-flat window function in [HIKP12c]. In Section 4.6 we give efficient constructions of such window functions, where $\widehat{G}$ can be computed in $O(\frac{B}{\alpha} \log(n/\delta))$ time and for each i, G'_i can be computed in $O(\log(n/\delta))$ time. Of course, for $i \notin [(1-\alpha)n/(2B), n/(2B)]$, $G'_i \in \{0,1\}$ can be computed in $O(1)$ time.

The fact that G'_i takes $\omega(1)$ time to compute for $i \in [(1-\alpha)n/(2B), n/(2B)]$ will add some complexity to our algorithm and analysis. We will need to ensure that we rarely need to compute such values. A practical implementation might find it more convenient to precompute the window functions in a preprocessing stage, rather than compute them on the fly.

We use the following lemma from [HIKP12c]:

Lemma 4.2.4. (Lemma 3.6 of [HIKP12c]) *If* $j \neq 0$, n *is a power of two, and* σ *is a uniformly random odd number in* $[n]$, *then* $\Pr[\sigma j \in [-C, C] \pmod n] \leq 4C/n$.

Assumptions. Through the chapter, we require that n, the dimension of all vectors, is an integer power of 2. We also make the following assumptions about the precision of the vectors x:

- For the exactly k-sparse case, we assume that $x_i \in \{-L, \ldots, L\}$ for some precision parameter L. To simplify the bounds, we assume that $L = n^{O(1)}$; otherwise the $\log n$ term in the running time bound is replaced by $\log L$.
- For the general case, we only achieve Equation (4.1) if $\|x\|_2 \leq n^{O(1)} \cdot \min_{k\text{-sparse } y} \|x - y\|_2$. In general, for any parameter $\delta > 0$ we can add $\delta \|x\|_2$ to the right hand side of Equation (4.1) and run in time $O(k \log(n/k) \log(n/\delta))$.

4.3 Algorithm for the exactly sparse case

In this section we assume $x_i \in \{-L, \ldots, L\}$, where $L \leq n^c$ for some constant $c > 0$, and x is k-sparse. We choose $\delta = 1/(4n^2 L)$. The algorithm (NoiselessSparseFFT) is described as Algorithm 4.3.1. The algorithm has three functions:

- HashToBins. This permutes the spectrum of $x - \chi$ with $P_{\sigma,a,b}$, then "hashes" to B bins. The guarantee will be described in Lemma 4.3.3.
- NoiselessSparseFFTInner. Given time-domain access to $\widehat{x}$ and a sparse vector χ such that $x - \chi$ is k'-sparse, this function finds "most" of $x - \chi$.
- NoiselessSparseFFT. This iterates NoiselessSparseFFTInner until it finds x exactly.

We analyze the algorithm "bottom-up", starting from the lower-level procedures.

procedure HashToBins($\widehat{x}$, χ, $P_{\sigma,a,b}$, B, δ, α)
  Compute $y_{jn/B}$ for $j \in [B]$, where $y = G_{B,\alpha,\delta} \cdot (P_{\sigma,a,b}x)$
  Compute $y'_{jn/B} = y_{jn/B} - (G'_{B,\alpha,\delta} * P_{\sigma,a,b}\chi)_{jn/B}$ for $j \in [B]$
  return u given by $u_j = y'_{jn/B}$.
end procedure
procedure NoiselessSparseFFTInner($\widehat{x}$, k', χ, α)
  Let $B = k'/\beta$, for sufficiently small constant β.
  Let $\delta = 1/(4n^2L)$.
  Choose σ uniformly at random from the set of odd numbers in $[n]$.
  Choose b uniformly at random from $[n]$.
  $u \leftarrow \text{HashToBins}(\widehat{x}, \chi, P_{\sigma,0,b}, B, \delta, \alpha)$.
  $u' \leftarrow \text{HashToBins}(\widehat{x}, \chi, P_{\sigma,1,b}, B, \delta, \alpha)$.
  $w \leftarrow 0$.
  Compute $J = \{j : |u_j| > 1/2\}$.
  for $j \in J$ **do**
    $a \leftarrow u_j/u'_j$.
    $i \leftarrow \sigma^{-1}(\text{round}(\phi(a)\frac{n}{2\pi})) \bmod n$. ▷ $\phi(a)$ denotes the phase of a.
    $v \leftarrow \text{round}(u_j)$.
    $w_i \leftarrow v$.
  end for
  return w
end procedure
procedure NoiselessSparseFFT($\widehat{x}$, k)
  $\chi \leftarrow 0$
  for $t \in 0, 1, \ldots, \log k$ **do**
    $k_t \leftarrow k/2^t$, $\alpha_t \leftarrow \Theta(2^{-t})$.
    $\chi \leftarrow \chi + \text{NoiselessSparseFFTInner}(\widehat{x}, k_t, \chi, \alpha_t)$.
  end for
  return χ
end procedure

Algorithm 4.3.1: Exact k-sparse recovery

Analysis of NoiselessSparseFFTInner and HashToBins.. For any execution of NoiselessSparseFFTInner, define the support $S = \text{supp}(x - \chi)$. Recall that $\pi_{\sigma,b}(i) = \sigma(i - b) \bmod n$. Define $h_{\sigma,b}(i) = \text{round}(\pi_{\sigma,b}(i)B/n)$ and $o_{\sigma,b}(i) = \pi_{\sigma,b}(i) - h_{\sigma,b}(i)n/B$. Note that therefore $|o_{\sigma,b}(i)| \leq n/(2B)$. We will refer to $h_{\sigma,b}(i)$ as the "bin" that the frequency i is mapped into, and $o_{\sigma,b}(i)$ as the "offset". For any $i \in S$ define two types of events associated with i and S and defined over the probability space induced by σ and b:

- "Collision" event $E_{coll}(i)$: holds iff $h_{\sigma,b}(i) \in h_{\sigma,b}(S \setminus \{i\})$, and
- "Large offset" event $E_{off}(i)$: holds iff $|o_{\sigma,b}(i)| \geq (1 - \alpha)n/(2B)$.

Claim 4.3.1. *For any $i \in S$, the event $E_{coll}(i)$ holds with probability at most $4|S|/B$.*

Proof. Consider distinct $i, j \in S$. By Lemma 4.2.4,

$$\begin{aligned}\Pr[h_{\sigma,b}(i) = h_{\sigma,b}(j)] &\leq \Pr[\pi_{\sigma,b}(i) - \pi_{\sigma,b}(j) \bmod n \in [-n/B, n/B]] \\ &= \Pr[\sigma(i-j) \bmod n \in [-n/B, n/B]] \\ &\leq 4/B.\end{aligned}$$

By a union bound over $j \in S$, $\Pr[E_{coll}(i)] \leq 4|S|/B$. □

Claim 4.3.2. *For any $i \in S$, the event $E_{off}(i)$ holds with probability at most α.*

Proof. Note that $o_{\sigma,b}(i) \equiv \pi_{\sigma,b}(i) \equiv \sigma(i-b) \pmod{n/B}$. For any odd σ and any $l \in [n/B]$, we have that $\Pr_b[\sigma(i-b) \equiv l \pmod{n/B}] = B/n$. Since only $\alpha n/B$ offsets $o_{\sigma,b}(i)$ cause $E_{off}(i)$, the claim follows. □

Lemma 4.3.3. *Suppose B divides n. The output u of* HashToBins *satisfies*

$$u_j = \sum_{h_{\sigma,b}(i)=j} (x-\chi)_i (G'_{B,\delta,\alpha})_{-o_{\sigma,b}(i)} \omega^{a\sigma i} \pm \delta\|x\|_1.$$

Let $\zeta = |\{i \in \mathrm{supp}(\chi) \mid E_{off}(i)\}|$. The running time of HashToBins *is $O(\frac{B}{\alpha}\log(n/\delta) + \|\chi\|_0 + \zeta\log(n/\delta))$.*

Proof. Define the flat window functions $G = G_{B,\delta,\alpha}$ and $G' = {G'}_{B,\delta,\alpha}$. We have

$$\begin{aligned}y &= G \cdot P_{\sigma,a,b} x = G * P_{\sigma,a,b} x \\ y' &= G' * P_{\sigma,a,b}(x-\chi) + (G - G') * P_{\sigma,a,b} x\end{aligned}$$

By Claim 4.2.2, the coordinates of $P_{\sigma,a,b}x$ and x have the same magnitudes, just different ordering and phase. Therefore

$$\|(G-G') * P_{\sigma,a,b}x\|_\infty \leq \|G - G'\|_\infty \|P_{\sigma,a,b}x\|_1 \leq \delta\|x\|_1$$

and hence

$$\begin{aligned}u_j = {y'}_{jn/B} &= \sum_{|l|<n/(2B)} {G'}_{-l} (P_{\sigma,a,b}(x-\chi))_{jn/B+l} \pm \delta\|x\|_1 \\ &= \sum_{|\pi_{\sigma,b}(i) - jn/B| < n/(2B)} {G'}_{jn/B - \pi_{\sigma,b}(i)} (P_{\sigma,a,b}(x-\chi))_{\pi_{\sigma,b}(i)} \pm \delta\|x\|_1 \\ &= \sum_{h_{\sigma,b}(i)=j} {G'}_{-o_{\sigma,b}(i)} (x-\chi)_i \omega^{a\sigma i} \pm \delta\|x\|_1\end{aligned}$$

as desired.

We can compute HashToBins via the following method:

1. Compute y with $\|y\|_0 = O(\frac{B}{\alpha}\log(n/\delta))$ in $O(\frac{B}{\alpha}\log(n/\delta))$ time.
2. Compute $v \in \mathbb{C}^B$ given by $v_i = \sum_j y_{i+jB}$.
3. Because B divides n, by the definition of the Fourier transform (see also Claim 3.7 of [HIKP12c]) we have $y_{jn/B} = v_j$ for all j. Hence we can compute it with a B-dimensional FFT in $O(B\log B)$ time.

4. For each coordinate $i \in \mathrm{supp}(\chi)$, decrease $y_{\frac{n}{B}h_{\sigma,b}(i)}$ by $G'_{-o_{\sigma,b}(i)}\chi_i\omega^{a\sigma i}$. This takes $O(\|\chi\|_0 + \zeta\log(n/\delta))$ time, since computing $G'_{-o_{\sigma,b}(i)}$ takes $O(\log(n/\delta))$ time if $E_{off}(i)$ holds and $O(1)$ otherwise.

□

Lemma 4.3.4. *Consider any $i \in S$ such that neither $E_{coll}(i)$ nor $E_{off}(i)$ holds. Let $j = h_{\sigma,b}(i)$. Then*

$$round(\phi(u_j/u'_j))\frac{n}{2\pi}) = \sigma i \pmod n,$$
$$round(u_j) = x_i - \chi_i,$$

and $j \in J$.

Proof. We know that $\|x\|_1 \le k\|x\|_\infty \le kL < nL$. Then by Lemma 4.3.3 and $E_{coll}(i)$ not holding,

$$u_j = (x-\chi)_i G'_{-o_{\sigma,b}(i)} \pm \delta nL.$$

Because $E_{off}(i)$ does not hold, $G'_{-o_{\sigma,b}(i)} = 1$, so

$$u_j = (x-\chi)_i \pm \delta nL. \tag{4.2}$$

Similarly,

$$u'_j = (x-\chi)_i\omega^{\sigma i} \pm \delta nL$$

Then because $\delta nL < 1 \le |(x-\chi)_i|$, the phase is

$$\phi(u_j) = 0 \pm \sin^{-1}(\delta nL) = 0 \pm 2\delta nL$$

and $\phi(u'_j) = -\sigma i\frac{2\pi}{n} \pm 2\delta nL$. Thus $\phi(u_j/u'_j) = \sigma i\frac{2\pi}{n} \pm 4\delta nL = \sigma i\frac{2\pi}{n} \pm 1/n$ by the choice of δ. Therefore

$$\mathrm{round}(\phi(u_j/u'_j)\frac{n}{2\pi}) = \sigma i \pmod n.$$

Also, by Equation (4.2), $\mathrm{round}(u_j) = x_i - \chi_i$. Finally, $|\mathrm{round}(u_j)| = |x_i - \chi_i| \ge 1$, so $|u_j| \ge 1/2$. Thus $j \in J$. □

For each invocation of NoiselessSparseFFTInner, let P be the the set of all pairs (i, v) for which the command $w_i \leftarrow v$ was executed. Claims 4.3.1 and 4.3.2 and Lemma 4.3.4 together guarantee that for each $i \in S$ the probability that P does not contain the pair $(i, (x-\chi)_i)$ is at most $4|S|/B + \alpha$. We complement this observation with the following claim.

Claim 4.3.5. *For any $j \in J$ we have $j \in h_{\sigma,b}(S)$. Therefore, $|J| = |P| \le |S|$.*

Proof. Consider any $j \notin h_{\sigma,b}(S)$. From Equation (4.2) in the proof of Lemma 4.3.4 it follows that $|u_j| \le \delta nL < 1/2$. □

Lemma 4.3.6. *Consider an execution of* NoiselessSparseFFTInner*, and let $S = \mathrm{supp}(x-\chi)$. If $|S| \le k'$, then*

$$E[\|x - \chi - w\|_0] \le 8(\beta+\alpha)|S|.$$

Proof. Let e denote the number of coordinates $i \in S$ for which either $E_{coll}(i)$ or $E_{off}(i)$ holds. Each such coordinate might not appear in P with the correct value, leading to an incorrect value of w_i. In fact, it might result in an arbitrary pair (i', v') being added to P, which in turn could

lead to an incorrect value of $w_{i'}$. By Claim 4.3.5 these are the only ways that w can be assigned an incorrect value. Thus we have

$$\|x - \chi - w\|_0 \leq 2e.$$

Since $E[e] \leq (4|S|/B + \alpha)|S| \leq (4\beta + \alpha)|S|$, the lemma follows. □

Analysis of NoiselessSparseFFT.. Consider the tth iteration of the procedure, and define $S_t = \mathrm{supp}(x - \chi)$ where χ denotes the value of the variable at the beginning of loop. Note that $|S_0| = |\mathrm{supp}(x)| \leq k$.

We also define an indicator variable I_t which is equal to 0 iff $|S_t|/|S_{t-1}| \leq 1/8$. If $I_t = 1$ we say the the tth iteration was not *successful*. Let $\gamma = 8 \cdot 8(\beta + \alpha)$. From Lemma 4.3.6 it follows that $\Pr[I_t = 1 \mid |S_{t-1}| \leq k/2^{t-1}] \leq \gamma$. From Claim 4.3.5 it follows that even if the tth iteration is not successful, then $|S_t|/|S_{t-1}| \leq 2$.

For any $t \geq 1$, define an event $E(t)$ that occurs iff $\sum_{i=1}^{t} I_i \geq t/2$. Observe that if none of the events $E(1) \ldots E(t)$ holds then $|S_t| \leq k/2^t$.

Lemma 4.3.7. *Let $E = E(1) \cup \ldots \cup E(\lambda)$ for $\lambda = 1 + \log k$. Assume that $(4\gamma)^{1/2} < 1/4$. Then $\Pr[E] \leq 1/3$.*

Proof. Let $t' = \lceil t/2 \rceil$. We have

$$\Pr[E(t)] \leq \binom{t}{t'} \gamma^{t'} \leq 2^t \gamma^{t'} \leq (4\gamma)^{t/2}$$

Therefore

$$\Pr[E] \leq \sum_t \Pr[E(t)] \leq \frac{(4\gamma)^{1/2}}{1 - (4\gamma)^{1/2}} \leq 1/4 \cdot 4/3 = 1/3.$$

□

Theorem 4.3.8. *Suppose x is k-sparse with entries from $\{-L, \ldots, L\}$ for some known $L = n^{O(1)}$. Then the algorithm* NoiselessSparseFFT *runs in expected $O(k \log n)$ time and returns the correct vector x with probability at least $2/3$.*

Proof. The correctness follows from Lemma 4.3.7. The running time is dominated by $O(\log k)$ executions of HashToBins.

Assuming a correct run, in every round t we have

$$\|\chi\|_0 \leq \|x\|_0 + |S_t| \leq k + k/2^t \leq 2k.$$

Therefore

$$\mathbb{E}[|\{i \in \mathrm{supp}(\chi) \mid E_{off}(i)\}|] \leq \alpha \|\chi\|_0 \leq 2\alpha k,$$

so the expected running time of each execution of HashToBins is $O(\frac{B}{\alpha} \log(n/\delta) + k + \alpha k \log(n/\delta)) = O(\frac{B}{\alpha} \log n + k + \alpha k \log n)$. Setting $\alpha = \Theta(2^{-t/2})$ and $\beta = \Theta(1)$, the expected running time in round t is $O(2^{-t/2} k \log n + k + 2^{-t/2} k \log n)$. Therefore the total expected running time is $O(k \log n)$. □

4.4 Algorithm for the general case

This section shows how to achieve Equation (4.1) for $C = 1 + \epsilon$. Pseudocode is in Algorithm 4.4.1 and 4.4.2.

Define $\widehat{x}^*$ to be the initial input to the algorithm. Our algorithm will repeatedly construct refined estimates χ of x^*, and recursively apply itself to $x = x^* - \chi$.

4.4.1 Intuition

Let S denote the "heavy" $O(k/\epsilon)$ coordinates of x. The overarching algorithm SparseFFT works by first "locating" a set L containing most of S, then "estimating" x_L to get χ. It then repeats on $x - \chi$. We will show that each heavy coordinate has a large constant probability of both being in L and being estimated well. As a result, $x - \chi$ is probably nearly $k/4$-sparse, so we can run the next iteration with $k \to k/4$. The later iterations then run faster and achieve a higher success probability, so the total running time is dominated by the time in the first iteration and the total error probability is bounded by a constant.

In the rest of this intuition, we will discuss the first iteration of SparseFFT with simplified constants. In this iteration, hashes are to $B = O(k/\epsilon)$ bins and, with $3/4$ probability, we get χ so $x - \chi$ is nearly $k/4$-sparse. The actual algorithm will involve a parameter α in each iteration, roughly guaranteeing that with $1 - \sqrt{\alpha}$ probability, we get χ so $x - \chi$ is nearly $\sqrt{\alpha}k$-sparse; the formal guarantee will be given by Lemma 4.4.8. For this intuition we only consider the first iteration where α is a constant.

Location. As in the noiseless case, to locate the "heavy" coordinates we consider the "bins" computed by HashToBins with $P_{\sigma,a,b}$. This roughly corresponds to first permuting the coordinates according to the (almost) pairwise independent permutation $P_{\sigma,a,b}$, partitioning the coordinates into $B = O(k/\epsilon)$ "bins" of n/B consecutive indices, and observing the sum of values in each bin. We get that each heavy coordinate i has a large constant probability that the following two events occur: no other heavy coordinate lies in the same bin, and only a small (i.e., $O(\epsilon/k)$) fraction of the mass from non-heavy coordinates lies in the same bin. For such a "well-hashed" coordinate i, we would like to find its location $\tau = \pi_{\sigma,b}(i) = \sigma(i - b)$ among the $\epsilon n/k < n/k$ consecutive values that hash to the same bin. Let

$$\theta_j^* = \frac{2\pi}{n}(j + \sigma b) \pmod{2\pi}. \tag{4.3}$$

so $\theta_\tau^* = \frac{2\pi}{n}\sigma i$. In the noiseless case, we showed that the difference in phase in the bin using $P_{\sigma,0,b}$ and using $P_{\sigma,1,b}$ is θ_τ^* plus a negligible $O(\delta)$ term. With noise this may not be true; however, we can say for any $\beta \in [n]$ that the difference in phase between using $P_{\sigma,a,b}$ and $P_{\sigma,a+\beta,b}$, as a distribution over uniformly random $a \in [n]$, is $\beta\theta_\tau^* + \nu$ with (for example) $\mathbb{E}[\nu^2] = 1/100$ (all operations on phases modulo 2π). We can only hope to get a constant number of bits from such a "measurement". So our task is to find τ within a region Q of size n/k using $O(\log(n/k))$ "measurements" of this form.

One method for doing so would be to simply do measurements with random $\beta \in [n]$. Then each measurement lies within $\pi/4$ of $\beta\theta_\tau^*$ with at least $1 - \frac{\mathbb{E}[\nu^2]}{\pi^2/16} > 3/4$ probability. On the other hand, for $j \neq \tau$ and as a distribution over β, $\beta(\theta_\tau^* - \theta_j^*)$ is roughly uniformly distributed around the circle. As a result, each measurement is probably more than $\pi/4$ away from $\beta\theta_j^*$. Hence $O(\log(n/k))$ repetitions suffice to distinguish among the n/k possibilities for τ. However, while the number of measurements is small, it is not clear how to decode in polylog rather than $\Omega(n/k)$ time.

To solve this, we instead do a t-ary search on the location for $t = \Theta(\log n)$. At each of $O(\log_t(n/k))$ levels, we split our current candidate region Q into t consecutive subregions $Q_1, \ldots, Q_t$, each of size w. Now, rather than choosing $\beta \in [n]$, we choose $\beta \in [\frac{n}{16w}, \frac{n}{8w}]$. By the upper bound on β, for each $q \in [t]$ the values $\{\beta\theta_j^* \mid j \in Q_q\}$ all lie within $\beta w\frac{2\pi}{n} \leq \pi/4$ of each other on the circle.

On the other hand, if $|j-\tau| > 16w$, then $\beta(\theta^*_\tau - \theta^*_j)$ will still be roughly uniformly distributed about the circle. As a result, we can check a single candidate element e_q from each subregion: if e_q is in the same subregion as τ, each measurement usually agrees in phase; but if e_q is more than 16 subregions away, each measurement usually disagrees in phase. Hence with $O(\log t)$ measurements, we can locate τ to within $O(1)$ consecutive subregions with failure probability $1/t^{\Theta(1)}$. The decoding time is $O(t \log t)$.

This primitive LocateInner lets us narrow down the candidate region for τ to a subregion that is a $t' = \Omega(t)$ factor smaller. By repeating LocateInner $\log_{t'}(n/k)$ times, LocateSignal can find τ precisely. The number of measurements is then $O(\log t \log_t(n/k)) = O(\log(n/k))$ and the decoding time is $O(t \log t \log_t(n/k)) = O(\log(n/k) \log n)$. Furthermore, the "measurements" (which are actually calls to HashToBins) are nonadaptive, so we can perform them in parallel for all $O(k/\epsilon)$ bins, with $O(\log(n/\delta))$ average time per measurement. This gives $O(k \log(n/k) \log(n/\delta))$ total time for LocateSignal.

This lets us locate every heavy and "well-hashed" coordinate with $1/t^{\Theta(1)} = o(1)$ failure probability, so every heavy coordinate is located with arbitrarily high constant probability.

Estimation. By contrast, estimation is fairly simple. As in Algorithm 4.3.1, we can estimate x_i as $u_{h_{\sigma,b}(i)}$, where u is the output of HashToBins. Unlike in Algorithm 4.3.1, we now have noise that may cause a single such estimate to be poor even if i is "well-hashed". However, we can show that for a random permutation $P_{\sigma,a,b}$ the estimate is "good" with constant probability. EstimateValues takes the median of $R_{est} = O(\log \frac{1}{\epsilon})$ such samples, getting a good estimate with $1 - \epsilon/64$ probability. Given a candidate set L of size k/ϵ, with $7/8$ probability at most $k/8$ of the coordinates are badly estimated. On the other hand, with $7/8$ probability, at least $7k/8$ of the heavy coordinates are both located and well estimated. This suffices to show that, with $3/4$ probability, the largest k elements J of our estimate w contains good estimates of $3k/4$ large coordinates, so $x - w_J$ is close to $k/4$-sparse.

4.4.2 Formal definitions

As in the noiseless case, we define $\pi_{\sigma,b}(i) = \sigma(i-b) \bmod n$, $h_{\sigma,b}(i) = \text{round}(\pi_{\sigma,b}(i)B/n)$ and $o_{\sigma,b}(i) = \pi_{\sigma,b}(i) - h_{\sigma,b}(i)n/B$. We say $h_{\sigma,b}(i)$ is the "bin" that frequency i is mapped into, and $o_{\sigma,b}(i)$ is the "offset". We define $h^{-1}_{\sigma,b}(j) = \{i \in [n] \mid h_{\sigma,b}(i) = j\}$.

In each iteration of SparseFFT, define $x = x^* - \chi$, and let

$$\begin{aligned}\rho^2 &= \text{Err}^2_k(x) + \delta^2 n \|x^*\|_1^2 \\ \mu^2 &= \epsilon\rho^2/k \\ S &= \{i \in [n] \mid |x_i|^2 \geq \mu^2\}\end{aligned}$$

Then $|S| \leq (1+1/\epsilon)k = O(k/\epsilon)$ and $\|x - x_S\|_2^2 \leq (1+\epsilon)\rho^2$. We will show that each $i \in S$ is found by LocateSignal with probability $1 - O(\alpha)$, when $B = \Omega(\frac{k}{\alpha\epsilon})$.

For any $i \in S$ define three types of events associated with i and S and defined over the probability space induced by σ and b:

- "Collision" event $E_{coll}(i)$: holds iff $h_{\sigma,b}(i) \in h_{\sigma,b}(S \setminus \{i\})$;
- "Large offset" event $E_{off}(i)$: holds iff $|o_{\sigma,b}(i)| \geq (1-\alpha)n/(2B)$; and
- "Large noise" event $E_{noise}(i)$: holds iff $\|x_{h^{-1}_{\sigma,b}(h_{\sigma,b}(i)) \setminus S}\|_2^2 \geq \text{Err}^2_k(x)/(\alpha B)$.

procedure SPARSEFFT($\widehat{x}^*$, k, ϵ, δ)
  $R \leftarrow O(\log k / \log\log k)$ as in Theorem 4.4.9.
  $\chi^{(1)} \leftarrow 0$
  for $r \in [R]$ **do**
    Choose B_r, k_r, α_r as in Theorem 4.4.9.
    $R_{est} \leftarrow O(\log(\frac{B_r}{\alpha_r k_r}))$ as in Lemma 4.4.8.
    $L_r \leftarrow$ LOCATESIGNAL($\widehat{x}^*, \chi^{(r)}, B_r, \alpha_r, \delta$)
    $\chi^{(r+1)} \leftarrow \chi^{(r)} +$ ESTIMATEVALUES($\widehat{x}^*, \chi^{(r)}, 3k_r, L_r, B_r, \delta, R_{est}$).
  end for
  return $\chi^{(R+1)}$
end procedure
procedure ESTIMATEVALUES($\widehat{x}^*$, χ, k', L, B, δ, R_{est})
  for $r \in [R_{est}]$ **do**
    Choose $a_r, b_r \in [n]$ uniformly at random.
    Choose σ_r uniformly at random from the set of odd numbers in $[n]$.
    $u^{(r)} \leftarrow$ HASHTOBINS($\widehat{x}^*, \chi, P_{\sigma,a_r,b}, B, \delta$).
  end for
  $w \leftarrow 0$
  for $i \in L$ **do**
    $w_i \leftarrow \text{median}_r\, u^{(r)}_{h_{\sigma,b}(i)} \omega^{-a_r \sigma i}$. ▷ Separate median in real and imaginary axes.
  end for
  $J \leftarrow \arg\max_{|J|=k'} \|w_J\|_2$.
  return w_J
end procedure

Algorithm 4.4.1: k-sparse recovery for general signals, part 1/2.

procedure LocateSignal($\widehat{x}^*$, χ, B, α, δ)
  Choose uniformly at random $\sigma, b \in [n]$ with σ odd.
  Initialize $l_i^{(1)} = (i-1)n/B$ for $i \in [B]$.
  Let $w_0 = n/B, t = \log n, t' = t/4, D_{max} = \log_{t'}(w_0 + 1)$.
  Let $R_{loc} = \Theta(\log_{1/\alpha}(t/\alpha))$ per Lemma 4.4.5.
  for $D \in [D_{max}]$ **do**
    $l^{(D+1)} \leftarrow$ LocateInner($\widehat{x}^*, \chi, B, \delta, \alpha, \sigma, \beta, l^{(D)}, w_0/(t')^{D-1}, t, R_{loc}$)
  end for
  $L \leftarrow \{\pi_{\sigma,b}^{-1}(l_j^{(D_{max}+1)}) \mid j \in [B]\}$
  return L
end procedure

▷ δ, α parameters for G, G'
▷ $(l_1, l_1 + w), \ldots, (l_B, l_B + w)$ the plausible regions.
▷ $B \approx k/\epsilon$ the number of bins
▷ $t \approx \log n$ the number of regions to split into.
▷ $R_{loc} \approx \log t = \log\log n$ the number of rounds to run

procedure LocateInner($\widehat{x}^*$, χ, B, δ, α, σ, b, l, w, t, R_{loc})
  Let $s = \Theta(\alpha^{1/3})$.
  Let $v_{j,q} = 0$ for $(j, q) \in [B] \times [t]$.
  for $r \in [R_{loc}]$ **do**
    Choose $a \in [n]$ uniformly at random.
    Choose $\beta \in \{\frac{snt}{4w}, \ldots, \frac{snt}{2w}\}$ uniformly at random.
    $u \leftarrow$ HashToBins($\widehat{x}^*, \chi, P_{\sigma,a,b}, B, \delta, \alpha$).
    $u' \leftarrow$ HashToBins($\widehat{x}^*, \chi, P_{\sigma,a+\beta,b}, B, \delta, \alpha$).
    for $j \in [B]$ **do**
      $c_j \leftarrow \phi(u_j/u'_j)$
      for $q \in [t]$ **do**
        $m_{j,q} \leftarrow l_j + \frac{q-1/2}{t} w$
        $\theta_{j,q} \leftarrow \frac{2\pi(m_{j,q}+\sigma b)}{n} \bmod 2\pi$
        if $\min(|\beta\theta_{j,q} - c_j|, 2\pi - |\beta\theta_{j,q} - c_j|) < s\pi$ **then**
          $v_{j,q} \leftarrow v_{j,q} + 1$
        end if
      end for
    end for
  end for
  for $j \in [B]$ **do**
    $Q^* \leftarrow \{q \in [t] \mid v_{j,q} > R_{loc}/2\}$
    if $Q^* \neq \emptyset$ **then**
      $l'_j \leftarrow \min_{q \in Q^*} l_j + \frac{q-1}{t} w$
    else
      $l'_j \leftarrow \perp$
    end if
  end for
  return l'
end procedure

Algorithm 4.4.2: k-sparse recovery for general signals, part 2/2.

By Claims 4.3.1 and 4.3.2, $\Pr[E_{coll}(i)] \le 4|S|/B = O(\alpha)$ and $\Pr[E_{off}(i)] \le 2\alpha$ for any $i \in S$.

Claim 4.4.1. *For any $i \in S$, $\Pr[E_{noise}(i)] \le 4\alpha$.*

Proof. For each $j \neq i$, $\Pr[h_{\sigma,b}(j) = h_{\sigma,b}(i)] \le \Pr[|\sigma j - \sigma i| < n/B] \le 4/B$ by Lemma 4.2.4. Then

$$\mathbb{E}[\|x_{h^{-1}_{\sigma,b}(h_{\sigma,b}(i))\setminus S}\|_2^2] \le 4\|x_{[n]\setminus S}\|_2^2/B$$

The result follows by Markov's inequality. □

We will show for $i \in S$ that if none of $E_{coll}(i), E_{off}(i)$, and $E_{noise}(i)$ hold then SparseFFTInner recovers x_i with $1 - O(\alpha)$ probability.

Lemma 4.4.2. *Let $a \in [n]$ uniformly at random, B divide n, and the other parameters be arbitrary in*

$$u = \text{HashToBins}(\widehat{x}^*, \chi, P_{\sigma,a,b}, B, \delta, \alpha).$$

Then for any $i \in [n]$ with $j = h_{\sigma,b}(i)$ and none of $E_{coll}(i)$, $E_{off}(i)$, or $E_{noise}(i)$ holding,

$$\mathbb{E}[|u_j - x_i\omega^{a\sigma i}|^2] \le 2\frac{\rho^2}{\alpha B}$$

Proof. Let $G' = {G'}_{B,\delta,\alpha}$. Let $T = h^{-1}_{\sigma,b}(j) \setminus \{i\}$. We have that $T \cap S = \{\}$ and ${G'}_{-o_{\sigma,b}(i)} = 1$. By Lemma 4.3.3,

$$u_j - x_i\omega^{a\sigma i} = \sum_{i' \in T} {G'}_{-o_\sigma(i')} x_{i'}\omega^{a\sigma i'} \pm \delta\|x^*\|_1.$$

Because the $\sigma i'$ are distinct for $i' \in T$, we have by Parseval's theorem

$$\mathop{\mathbb{E}}_a \left|\sum_{i' \in T} {G'}_{-o_\sigma(i')} x_{i'}\omega^{a\sigma i'}\right|^2 = \sum_{i' \in T} ({G'}_{-o_\sigma(i')} x_{i'})^2 \le \|x_T\|_2^2$$

Since $|X + Y|^2 \le 2|X|^2 + 2|Y|^2$ for any X, Y, we get

$$\begin{aligned}\mathop{\mathbb{E}}_a[|u_j - x_i\omega^{a\sigma i}|^2] &\le 2\|x_T\|_2^2 + 2\delta^2\|x^*\|_1^2 \\ &\le 2\operatorname{Err}_k^2(x)/(\alpha B) + 2\delta^2\|x^*\|_1^2 \\ &\le 2\rho^2/(\alpha B).\end{aligned}$$

□

4.4.3 Properties of LocateSignal

In our intuition, we made a claim that if $\beta \in [n/(16w), n/(8w)]$ uniformly at random, and $i > 16w$, then $\frac{2\pi}{n}\beta i$ is "roughly uniformly distributed about the circle" and hence not concentrated in any small region. This is clear if β is chosen as a random real number; it is less clear in our setting where β is a random integer in this range. We now prove a lemma that formalizes this claim.

Lemma 4.4.3. *Let $T \subset [m]$ consist of t consecutive integers, and suppose $\beta \in T$ uniformly at random. Then for any $i \in [n]$ and set $S \subset [n]$ of l consecutive integers,*

$$\Pr[\beta i \bmod n \in S] \le \lceil im/n \rceil (1 + \lfloor l/i \rfloor)/t \le \frac{1}{t} + \frac{im}{nt} + \frac{lm}{nt} + \frac{l}{it}$$

Proof. Note that any interval of length l can cover at most $1 + \lfloor l/i \rfloor$ elements of any arithmetic sequence of common difference i. Then $\{\beta i \mid \beta \in T\} \subset [im]$ is such a sequence, and there are at most $\lceil im/n \rceil$ intervals $an + S$ overlapping this sequence. Hence at most $\lceil im/n \rceil(1 + \lfloor l/i \rfloor)$ of the $\beta \in [m]$ have $\beta i \bmod n \in S$. Hence

$$\Pr[\beta i \bmod n \in S] \leq \lceil im/n \rceil(1 + \lfloor l/i \rfloor)/t.$$

□

Lemma 4.4.4. *Let $i \in S$. Suppose none of $E_{coll}(i), E_{off}(i)$, and $E_{noise}(i)$ hold, and let $j = h_{\sigma,b}(i)$. Consider any run of* LocateInner *with $\pi_{\sigma,b}(i) \in [l_j, l_j + w]$. Let $f > 0$ be a parameter such that*

$$B = \frac{Ck}{\alpha f \epsilon}.$$

for C larger than some fixed constant. Then $\pi_{\sigma,b}(i) \in [l'_j, l'_j + 4w/t]$ with probability at least $1 - tf^{\Omega(R_{loc})}$,

Proof. Let $\tau = \pi_{\sigma,b}(i) \equiv \sigma(i - b) \pmod n$, and for any $j \in [n]$ define

$$\theta^*_j = \frac{2\pi}{n}(j + \sigma b) \pmod{2\pi}$$

so $\theta^*_\tau = \frac{2\pi}{n}\sigma i$. Let $g = \Theta(f^{1/3})$, and $C' = \frac{B\alpha\epsilon}{k} = \Theta(1/g^3)$.

To get the result, we divide $[l_j, l_j + w]$ into t "regions", $Q_q = [l_j + \frac{q-1}{t}w, l_j + \frac{q}{t}w]$ for $q \in [t]$. We will first show that in each round r, c_j is close to $\beta\theta^*_\tau$ with $1 - g$ probability. This will imply that Q_q gets a "vote," meaning $v_{j,q}$ increases, with $1 - g$ probability for the q' with $\tau \in Q_{q'}$. It will also imply that $v_{j,q}$ increases with only g probability when $|q - q'| > 3$. Then R_{loc} rounds will suffice to separate the two with $1 - f^{-\Omega(R_{loc})}$ probability. We get that with $1 - tf^{-\Omega(R_{loc})}$ probability, the recovered Q^* has $|q - q'| \leq 3$ for all $q \in Q^*$. If we take the minimum $q \in Q^*$ and the next three subregions, we find τ to within 4 regions, or $4w/t$ locations, as desired.

In any round r, define $u = u^{(r)}$ and $a = a_r$. We have by Lemma 4.4.2 and that $i \in S$ that

$$\begin{aligned}\mathbb{E}[|u_j - \omega^{a\sigma i}x_i|^2] &\leq 2\frac{\rho^2}{\alpha B} = \frac{2k}{B\alpha\epsilon}\mu^2 \\ &= \frac{2}{C'}\mu^2 \leq \frac{2}{C'}|x_i|^2.\end{aligned}$$

Note that $\phi(\omega^{a\sigma i}) = -a\theta^*_\tau$. Thus for any $p > 0$, with probability $1 - p$ we have

$$|u_j - \omega^{a\sigma i}x_i| \leq \sqrt{\frac{2}{C'p}}|x_i|$$

$$\|\phi(u_j) - (\phi(x_i) - a\theta^*_\tau)\|_\bigcirc \leq \sin^{-1}(\sqrt{\frac{2}{C'p}})$$

where $\|x - y\|_\bigcirc = \min_{\gamma \in \mathbb{Z}} |x - y + 2\pi\gamma|$ denotes the "circular distance" between x and y. The

analogous fact holds for $\phi(u'_j)$ relative to $\phi(x_i)-(a+\beta)\theta^*_\tau$. Therefore with at least $1-2p$ probability,

$$\begin{aligned}\|c_j - \beta\theta^*_\tau\|_\bigcirc &= \|\phi(u_j) - \phi(u'_j) - \beta\theta^*_\tau\|_\bigcirc \\ &= \left\| (\phi(u_j) - (\phi(x_i) - a\theta^*_\tau)) - \left(\phi(u'_j) - (\phi(x_i) - (a+\beta)\theta^*_\tau)\right) \right\|_\bigcirc \\ &\leq \|\phi(u_j) - (\phi(x_i) - a\theta^*_\tau)\|_\bigcirc + \|\phi(u'_j) - (\phi(x_i) - (a+\beta)\theta^*_\tau)\|_\bigcirc \\ &\leq 2\sin^{-1}(\sqrt{\frac{2}{C'p}})\end{aligned}$$

by the triangle inequality. Thus for any $s = \Theta(g)$ and $p = \Theta(g)$, we can set $C' = \frac{2}{p\sin^2(s\pi/4)} = \Theta(1/g^3)$ so that

$$\|c_j - \beta\theta^*_\tau\|_\bigcirc < s\pi/2 \tag{4.4}$$

with probability at least $1 - 2p$.

Equation (4.4) shows that c_j is a good estimate for i with good probability. We will now show that this means the appropriate "region" $Q_{q'}$ gets a "vote" with "large" probability.

For the q' with $\tau \in [l_j + \frac{q'-1}{t}w, l_j + \frac{q'}{t}w]$, we have that $m_{j,q'} = l_j + \frac{q'-1/2}{t}w$ satisfies

$$|\tau - m_{j,q'}| \leq \frac{w}{2t}$$

so

$$|\theta^*_\tau - \theta_{j,q'}| \leq \frac{2\pi}{n}\frac{w}{2t}.$$

Hence by Equation (4.4), the triangle inequality, and the choice of $B \leq \frac{snt}{2w}$,

$$\begin{aligned}\|c_j - \beta\theta_{j,q'}\|_\bigcirc &\leq \|c_j - \beta\theta^*_\tau\|_\bigcirc + \|\beta\theta^*_\tau - \beta\theta_{j,q'}\|_\bigcirc \\ &< \frac{s\pi}{2} + \frac{\beta\pi w}{nt} \\ &\leq \frac{s\pi}{2} + \frac{s\pi}{2} \\ &= s\pi.\end{aligned}$$

Thus, $v_{j,q'}$ will increase in each round with probability at least $1 - 2p$.

Now, consider q with $|q - q'| > 3$. Then $|\tau - m_{j,q}| \geq \frac{7w}{2t}$, and (from the definition of $\beta > \frac{snt}{4w}$) we have

$$\beta|\tau - m_{j,q}| \geq \frac{7sn}{8} > \frac{3sn}{4}. \tag{4.5}$$

We now consider two cases. First, suppose that $|\tau - m_{j,q}| \leq \frac{w}{st}$. In this case, from the definition of β it follows that

$$\beta|\tau - m_{j,q}| \leq n/2.$$

Together with Equation (4.5) this implies

$$\Pr[\beta(\tau - m_{j,q}) \bmod n \in [-3sn/4, 3sn/4]] = 0.$$

On the other hand, suppose that $|\tau - m_{j,q}| > \frac{w}{st}$. In this case, we use Lemma 4.4.3 with

parameters $l = 3sn/2$, $m = \frac{snt}{2w}$, $t = \frac{snt}{4w}$, $i = (\tau - m_{j,q})$ and $n = n$, to conclude that

$$\begin{aligned}\Pr[\beta(\tau - m_{j,q}) \bmod n \in [-3sn/4, 3sn/4]] &\leq \frac{4w}{snt} + 2\frac{|\tau - m_{j,q}|}{n} + 3s + \frac{3sn}{2}\frac{st}{w}\frac{4w}{snt} \\ &\leq \frac{4w}{snt} + \frac{2w}{n} + 9s \\ &< \frac{6}{sB} + 9s \\ &< 10s\end{aligned}$$

where we used that $|i| \leq w \leq n/B$, the assumption $\frac{w}{st} < |i|$, $t \geq 1$, $s < 1$, and that $s^2 > 6/B$ (because $s = \Theta(g)$ and $B = \omega(1/g^3)$).

Thus in either case, with probability at least $1 - 10s$ we have

$$\|\beta\theta_{j,q} - \beta\theta^*_\tau\|_\bigcirc = \|\frac{2\pi\beta(m_{j,q} - \tau)}{n}\|_\bigcirc > \frac{2\pi}{n}\frac{3sn}{4} = \frac{3}{2}s\pi$$

for any q with $|q - q'| > 3$. Therefore we have

$$\|c_j - \beta\theta_{j,q}\|_\bigcirc \geq \|\beta\theta_{j,q} - \beta\theta^*_\tau\|_\bigcirc - \|c_j - \beta\theta^*_\tau\|_\bigcirc > s\pi$$

with probability at least $1 - 10s - 2p$, and $v_{j,q}$ is not incremented.

To summarize: in each round, $v_{j,q'}$ is incremented with probability at least $1 - 2p$ and $v_{j,q}$ is incremented with probability at most $10s + 2p$ for $|q - q'| > 3$. The probabilities corresponding to different rounds are independent.

Set $s = g/20$ and $p = g/4$. Then $v_{j,q'}$ is incremented with probability at least $1 - g$ and $v_{j,q}$ is incremented with probability less than g. Then after R_{loc} rounds, if $|q - q'| > 3$,

$$\Pr[v_{j,q} > R_{loc}/2] \leq \binom{R_{loc}}{R_{loc}/2} g^{R_{loc}/2} \leq (4g)^{R_{loc}/2} = f^{\Omega(R_{loc})}$$

for $g = f^{1/3}/4$. Similarly,

$$\Pr[v_{j,q'} < R_{loc}/2] \leq f^{\Omega(R_{loc})}.$$

Hence with probability at least $1 - tf^{\Omega(R_{loc})}$ we have $q' \in Q^*$ and $|q - q'| \leq 3$ for all $q \in Q^*$. But then $\tau - l'_j \in [0, 4w/t]$ as desired.

Because $\mathbb{E}[|\{i \in \mathrm{supp}(\chi) \mid E_{off}(i)\}|] = \alpha\|\chi\|_0$, the expected running time is $O(R_{loc}Bt + R_{loc}\frac{B}{\alpha}\log(n/\delta) + R_{loc}\|\chi\|_0(1 + \alpha\log(n/\delta)))$. □

Lemma 4.4.5. *Suppose $B = \frac{Ck}{\alpha^2\epsilon}$ for C larger than some fixed constant. The procedure* LocateSignal *returns a set L of size $|L| \leq B$ such that for any $i \in S$, $\Pr[i \in L] \geq 1 - O(\alpha)$. Moreover the procedure runs in expected time*

$$O((\frac{B}{\alpha}\log(n/\delta) + \|\chi\|_0(1 + \alpha\log(n/\delta)))\log(n/B)).$$

Proof. Consider any $i \in S$ such that none of $E_{coll}(i), E_{off}(i)$, and $E_{noise}(i)$ hold, as happens with probability $1 - O(\alpha)$.

Set $t = \log n, t' = t/4$ and $R_{loc} = O(\log_{1/\alpha}(t/\alpha))$. Let $w_0 = n/B$ and $w_D = w_0/(t')^{D-1}$, so $w_{D_{max}+1} < 1$ for $D_{max} = \log_{t'}(w_0 + 1) < t$. In each round D, Lemma 4.4.4 implies that if $\pi_{\sigma,b}(i) \in [l_j^{(D)}, l_j^{(D)} + w_D]$ then $\pi_{\sigma,b}(i) \in [l_j^{(D+1)}, l_j^{(D+1)} + w_{D+1}]$ with probability at least $1 - \alpha^{\Omega(R_{loc})} = 1 - \alpha/t$.

By a union bound, with probability at least $1-\alpha$ we have $\pi_{\sigma,b}(i) \in [l_j^{(D_{max}+1)}, l_j^{(D_{max}+1)} + w_{D_{max}+1}] = \{l_j^{(D_{max}+1)}\}$. Thus $i = \pi_{\sigma,b}^{-1}(l_j^{(D_{max}+1)}) \in L$.

Since $R_{loc} D_{max} = O(\log_{1/\alpha}(t/\alpha) \log_t(n/B)) = O(\log(n/B))$, the running time is

$$\begin{aligned} &O(D_{max}(R_{loc}\frac{B}{\alpha}\log(n/\delta) + R_{loc}\|\chi\|_0(1+\alpha\log(n/\delta)))) \\ &= O((\frac{B}{\alpha}\log(n/\delta) + \|\chi\|_0(1+\alpha\log(n/\delta)))\log(n/B)). \end{aligned}$$

□

4.4.4 Properties of EstimateValues

Lemma 4.4.6. *For any $i \in L$,*

$$\Pr[|w_i - x_i|^2 > \mu^2] < e^{-\Omega(R_{est})}$$

if $B > \frac{Ck}{\alpha\epsilon}$ for some constant C.

Proof. Define $e_r = u_j^{(r)} \omega^{-a_r \sigma i} - x_i$ in each round r. Suppose none of $E_{coll}^{(r)}(i), E_{off}^{(r)}(i)$, and $E_{noise}^{(r)}(i)$ hold, as happens with probability $1 - O(\alpha)$. Then by Lemma 4.4.2,

$$\mathop{\mathbb{E}}_{a_r}[|e_r|^2] \le 2\frac{\rho^2}{\alpha B} = \frac{2k}{\alpha\epsilon B}\mu^2 < \frac{2}{C}\mu^2$$

Hence with $3/4 - O(\alpha) > 5/8$ probability in total,

$$|e_r|^2 < \frac{8}{C}\mu^2 < \mu^2/2$$

for sufficiently large C. Then with probability at least $1 - e^{-\Omega(R_{est})}$, both of the following occur:

$$\begin{aligned} |\operatorname{median}_r \operatorname{real}(e_r)|^2 &< \mu^2/2 \\ |\operatorname{median}_r \operatorname{imag}(e_r)|^2 &< \mu^2/2. \end{aligned}$$

If this is the case, then $|\operatorname{median}_r e_r|^2 < \mu^2$. Since $w_i = x_i + \operatorname{median} e_r$, the result follows. □

Lemma 4.4.7. *Let $R_{est} \ge C \log \frac{B}{\gamma f k}$ for some constant C and parameters $\gamma, f > 0$. Then if* EstimateValues *is run with input $k' = 3k$, it returns w_J for $|J| = 3k$ satisfying*

$$\operatorname{Err}_{fk}^2(x_L - w_J) \le \operatorname{Err}_k^2(x_L) + O(k\mu^2)$$

with probability at least $1 - \gamma$.

Proof. By Lemma 4.4.6, each index $i \in L$ has

$$\Pr[|w_i - x_i|^2 > \mu^2] < \frac{\gamma f k}{B}.$$

Let $U = \{i \in L \mid |w_i - x_i|^2 > \mu^2\}$. With probability $1 - \gamma$, $|U| \le fk$; assume this happens. Then

$$\|(x - w)_{L \setminus U}\|_\infty^2 \le \mu^2. \tag{4.6}$$

Let T contain the top $2k$ coordinates of $w_{L\setminus U}$. By Lemma 7.2.1, the ℓ_∞ guarantee (4.6) means that

$$\|x_{L\setminus U} - w_T\|_2^2 \le \mathrm{Err}_k^2(x_{L\setminus U}) + 3k\mu^2. \tag{4.7}$$

Because J is the top $3k > (2+f)k$ coordinates of w_L, $T \subset J$. Let $J' = J \setminus (T \cup U)$, so $|J'| \le k$. Then

$$\begin{aligned}
\mathrm{Err}_{fk}^2(x_L - w_J) &\le \|x_{L\setminus U} - w_{J\setminus U}\|_2^2 \\
&= \|x_{L\setminus(U\cup J')} - w_T\|_2^2 + \|(x-w)_{J'}\|_2^2 \\
&\le \|x_{L\setminus U} - w_T\|_2^2 + |J'|\|(x-w)_{J'}\|_\infty^2 \\
&\le \mathrm{Err}_k^2(x_{L\setminus U}) + 3k\mu^2 + k\mu^2 \\
&= \mathrm{Err}_k^2(x_{L\setminus U}) + O(k\mu^2)
\end{aligned}$$

where we used Equations (4.6) and (4.7). □

4.4.5 Properties of SparseFFT

We will show that $x^* - \chi^{(r)}$ gets sparser as r increases, with only a mild increase in the error.

Lemma 4.4.8. *Define $x^{(r)} = x^* - \chi^{(r)}$. Consider any one loop r of* SparseFFT*, running with parameters $(B, k, \alpha) = (B_r, k_r, \alpha_r)$ such that $B \ge \frac{Ck}{\alpha^2\epsilon}$ for some C larger than some fixed constant. Then for any $f > 0$,*

$$\mathrm{Err}_{fk}^2(x^{(r+1)}) \le (1+\epsilon)\,\mathrm{Err}_k^2(x^{(r)}) + O(\epsilon\delta^2 n\|x^*\|_1^2)$$

with probability $1 - O(\alpha/f)$, and the running time is

$$O((\|\chi^{(r)}\|_0(1+\alpha\log(n/\delta)) + \frac{B}{\alpha}\log(n/\delta))(\log\frac{1}{\alpha\epsilon} + \log(n/B))).$$

Proof. We use $R_{est} = O(\log\frac{B}{\alpha k}) = O(\log\frac{1}{\alpha\epsilon})$ rounds inside EstimateValues.

The running time for LocateSignal is

$$O((\frac{B}{\alpha}\log(n/\delta) + \|\chi^{(r)}\|_0(1+\alpha\log(n/\delta)))\log(n/B)),$$

and for EstimateValues is

$$O((\frac{B}{\alpha}\log(n/\delta) + \|\chi^{(r)}\|_0(1+\alpha\log(n/\delta)))\log\frac{1}{\alpha\epsilon})$$

for a total running time as given.

Recall that in round r, $\mu^2 = \frac{\epsilon}{k}(\mathrm{Err}_k^2(x^{(r)}) + \delta^2 n\|x^*\|_1^2)$ and $S = \{i \in [n] \mid |x_i^{(r)}|^2 > \mu^2\}$. By Lemma 4.4.5, each $i \in S$ lies in L_r with probability at least $1 - O(\alpha)$. Hence $|S \setminus L| < fk$ with probability at least $1 - O(\alpha/f)$. Then

$$\begin{aligned}
\mathrm{Err}_{fk}^2(x_{[n]\setminus L}^{(r)}) &\le \|x_{[n]\setminus(L\cup S)}^{(r)}\|_2^2 \\
&\le \mathrm{Err}_k^2(x_{[n]\setminus(L\cup S)}^{(r)}) + k\|x_{[n]\setminus(L\cup S)}^{(r)}\|_\infty^2 \\
&\le \mathrm{Err}_k^2(x_{[n]\setminus L}^{(r)}) + k\mu^2.
\end{aligned} \tag{4.8}$$

Let $w = \chi^{(r+1)} - \chi^{(r)} = x^{(r)} - x^{(r+1)}$ by the vector recovered by EstimateValues. Then $\operatorname{supp}(w) \subset L$, so

$$\begin{aligned}\operatorname{Err}^2_{2fk}(x^{(r+1)}) &= \operatorname{Err}^2_{2fk}(x^{(r)} - w) \\ &\le \operatorname{Err}^2_{fk}(x^{(r)}_{[n]\setminus L}) + \operatorname{Err}^2_{fk}(x^{(r)}_L - w) \\ &\le \operatorname{Err}^2_{fk}(x^{(r)}_{[n]\setminus L}) + \operatorname{Err}^2_k(x^{(r)}_L) + O(k\mu^2)\end{aligned}$$

by Lemma 4.4.7. But by Equation (4.8), this gives

$$\begin{aligned}\operatorname{Err}^2_{2fk}(x^{(r+1)}) &\le \operatorname{Err}^2_k(x^{(r)}_{[n]\setminus L}) + \operatorname{Err}^2_k(x^{(r)}_L) + O(k\mu^2) \\ &\le \operatorname{Err}^2_k(x^{(r)}) + O(k\mu^2) \\ &= (1 + O(\epsilon)) \operatorname{Err}^2_k(x^{(r)}) + O(\epsilon\delta^2 n \|x^*\|_1^2).\end{aligned}$$

The result follows from rescaling f and ϵ by constant factors. □

As in previous chapters, repeating this lemma leads to a general sparse recovery algorithm:

Theorem 4.4.9. *With 2/3 probability,* SparseFFT *recovers* $\chi^{(R+1)}$ *such that*

$$\|x^* - \chi^{(R+1)}\|_2 \le (1+\epsilon) \operatorname{Err}_k(x) + \delta \|x\|_2$$

in $O(\frac{k}{\epsilon} \log(n/k) \log(n/\delta))$ *time.*

Proof. Define $f_r = \Theta(1/r^2)$ so $\sum f_r < 1/4$. Choose R so $\prod_{r \le R} f_r < 1/k \le \prod_{r<R} f_r$. Then $R = O(\log k / \log\log k)$, since $\prod_{r \le R} f_r < (f_{R/2})^{R/2} = (2/R)^R$.

Set $\epsilon_r = f_r \epsilon$, $\alpha_r = \Theta(f_r^2)$, $k_r = k \prod_{i<r} f_i$, $B_r = O(\frac{k}{\epsilon} \alpha_r f_r)$. Then $B_r = \omega(\frac{k_r}{\alpha_r^2 \epsilon_r})$, so for sufficiently large constant the constraint of Lemma 4.4.8 is satisfied. For appropriate constants, Lemma 4.4.8 says that in each round r,

$$\operatorname{Err}^2_{k_{r+1}}(x^{(r+1)}) = \operatorname{Err}^2_{f_r k_r}(x^{(r+1)}) \le (1 + f_r\epsilon) \operatorname{Err}^2_{k_r}(x^{(r)}) + O(f_r \epsilon \delta^2 n \|x^*\|_1^2) \tag{4.9}$$

with probability at least $1 - f_r$. The error accumulates, so in round r we have

$$\operatorname{Err}^2_{k_r}(x^{(r)}) \le \operatorname{Err}^2_k(x) \prod_{i<r} (1 + f_i\epsilon) + \sum_{i<r} O(f_r \epsilon \delta^2 n \|x^*\|_1^2) \prod_{i<j<r} (1 + f_j \epsilon)$$

with probability at least $1 - \sum_{i<r} f_i > 3/4$. Hence in the end, since $k_{R+1} = k \prod_{i \le R} f_i < 1$,

$$\|x^{(R+1)}\|_2^2 = \operatorname{Err}^2_{k_{R+1}}(x^{(R+1)}) \le \operatorname{Err}^2_k(x) \prod_{i \le R} (1 + f_i\epsilon) + O(R\epsilon\delta^2 n \|x^*\|_1^2) \prod_{i \le R} (1 + f_i \epsilon)$$

with probability at least 3/4. We also have

$$\prod_i (1 + f_i\epsilon) \le e^{\epsilon \sum_i f_i} \le e$$

making

$$\prod_i (1 + f_i\epsilon) \le 1 + e \sum_i f_i \epsilon < 1 + 2\epsilon.$$

Thus we get the approximation factor

$$\|x^* - \chi^{(R+1)}\|_2^2 \leq (1+2\epsilon)\operatorname{Err}_k^2(x) + O((\log k)\epsilon\delta^2 n\|x^*\|_1^2)$$

with at least 3/4 probability. Rescaling δ by $\text{poly}(n)$, using $\|x^*\|_1^2 \leq n\|x\|_2$, and taking the square root gives the desired

$$\|x^* - \chi^{(R+1)}\|_2 \leq (1+\epsilon)\operatorname{Err}_k(x) + \delta\|x\|_2.$$

Now we analyze the running time. The update $\chi^{(r+1)} - \chi^{(r)}$ in round r has support size $3k_r$, so in round r

$$\|\chi^{(r)}\|_0 \leq \sum_{i<r} 3k_r \lesssim k.$$

Thus the expected running time in round r is order

$$\begin{aligned}
&(k(1+\alpha_r\log(n/\delta)) + \frac{B_r}{\alpha_r}\log(n/\delta))(\log\frac{1}{\alpha_r\epsilon_r} + \log(n/B_r)) \\
\lesssim\ &(k + \frac{k}{r^4}\log(n/\delta) + \frac{k}{\epsilon r^2}\log(n/\delta))(\log\frac{r^2}{\epsilon} + \log(n\epsilon/k) + \log r) \\
\lesssim\ &(k + \frac{k}{\epsilon r^2}\log(n/\delta))(\log r + \log(n/k))
\end{aligned}$$

We split the terms multiplying k and $\frac{k}{\epsilon r^2}\log(n/\delta)$, and sum over r. First,

$$\begin{aligned}
\sum_{r=1}^{R}(\log r + \log(n/k)) &\leq R\log R + R\log(n/k) \\
&\lesssim \log k + \log k\log(n/k) \\
&\lesssim \log k\log(n/k).
\end{aligned}$$

Next,

$$\sum_{r=1}^{R}\frac{1}{r^2}(\log r + \log(n/k)) \lesssim \log(n/k)$$

Thus the total running time is order

$$k\log k\log(n/k) + \frac{k}{\epsilon}\log(n/\delta)\log(n/k) \lesssim \frac{k}{\epsilon}\log(n/\delta)\log(n/k).$$

□

4.5 Reducing the full k-dimensional DFT to the exact k-sparse case in n dimensions

In this section we show the following lemma. Assume that k divides n.

Lemma 4.5.1. *Suppose that there is an algorithm A that, given an n-dimensional vector $\widehat{y}$ such that y is k-sparse, computes y in time $T(k)$. Then there is an algorithm A' that given a k-dimensional vector $\widehat{x}$ computes x in time $O(T(k))$.*

Proof. Given a k-dimensional vector $\widehat{x}$, we define $\widehat{y}_i = \widehat{x}_{i \bmod k}$, for $i \in [n]$. Whenever A requests a sample $\widehat{y}_i$, we compute it from $\widehat{x}$ in constant time. Moreover, we have that $y_i = x_{i/(n/k)}$ if i is a multiple of (n/k), and $y_i = 0$ otherwise. Thus y is k-sparse. Since x can be immediately recovered from y, the lemma follows. □

Corollary 4.5.2. *Assume that the n-dimensional DFT cannot be computed in $o(n \log n)$ time. Then any algorithm for the k-sparse DFT (for vectors of arbitrary dimension) must run in $\Omega(k \log k)$ time.*

4.6 Efficient Constructions of Window Functions

Claim 4.6.1. *Let* cdf *denote the standard Gaussian cumulative distribution function. Then:*

1. $\mathrm{cdf}(t) = 1 - \mathrm{cdf}(-t)$.
2. $\mathrm{cdf}(t) \leq e^{-t^2/2}$ *for* $t < 0$.
3. $\mathrm{cdf}(t) < \delta$ *for* $t < -\sqrt{2\log(1/\delta)}$.
4. $\int_{x=-\infty}^{t} \mathrm{cdf}(x)dx < \delta$ *for* $t < -\sqrt{2\log(3/\delta)}$.
5. *For any δ, there exists a function $\widetilde{\mathrm{cdf}}_\delta(t)$ computable in $O(\log(1/\delta))$ time such that $\|\mathrm{cdf} - \widetilde{\mathrm{cdf}}_\delta\|_\infty < \delta$.*

Proof.

1. Follows from the symmetry of Gaussian distribution.
2. Follows from a standard moment generating function bound on Gaussian random variables.
3. Follows from (2).
4. Property (2) implies that $\mathrm{cdf}(t)$ is at most $\sqrt{2\pi} < 3$ times larger than the Gaussian pdf. Then apply (3).
5. By (1) and (3), $\mathrm{cdf}(t)$ can be computed as $\pm\delta$ or $1 \pm \delta$ unless $|t| < \sqrt{2(\log(1/\delta))}$. But then an efficient expansion around 0 only requires $O(\log(1/\delta))$ terms to achieve precision $\pm\delta$.

   For example, we can truncate the representation [Mar04]

$$\mathrm{cdf}(t) = \frac{1}{2} + \frac{e^{-t^2/2}}{\sqrt{2\pi}}\left(t + \frac{t^3}{3} + \frac{t^5}{3\cdot 5} + \frac{t^7}{3\cdot 5\cdot 7} + \cdots\right)$$

   at $O(\log(1/\delta))$ terms.

□

Claim 4.6.2. *Define the continuous Fourier transform of $f(t)$ by*

$$\widehat{f}(s) = \int_{-\infty}^{\infty} e^{-2\pi \mathbf{i} s t} f(t)dt.$$

For $t \in [n]$, define

$$g_t = \sqrt{n} \sum_{j=-\infty}^{\infty} f(t + nj)$$

and

$$g'_t = \sum_{j=-\infty}^{\infty} \widehat{f}(t/n + j).$$

Then $\widehat{g} = g'$, where $\widehat{g}$ is the n-dimensional DFT of g.

Proof. Let $\Delta_1(t)$ denote the Dirac comb of period 1: $\Delta_1(t)$ is a Dirac delta function when t is an integer and zero elsewhere. Then $\widehat{\Delta}_1 = \Delta_1$. For any $t \in [n]$, we have

$$\begin{aligned}
\widehat{g}_t &= \sum_{s=1}^{n} \sum_{j=-\infty}^{\infty} f(s + nj) e^{-2\pi \mathbf{i} ts/n} \\
&= \sum_{s=1}^{n} \sum_{j=-\infty}^{\infty} f(s + nj) e^{-2\pi \mathbf{i} t(s+nj)/n} \\
&= \sum_{s=-\infty}^{\infty} f(s) e^{-2\pi \mathbf{i} ts/n} \\
&= \int_{-\infty}^{\infty} f(s) \Delta_1(s) e^{-2\pi \mathbf{i} ts/n} ds \\
&= \widehat{(f \cdot \Delta_1)}(t/n) \\
&= (\widehat{f} * \Delta_1)(t/n) \\
&= \sum_{j=-\infty}^{\infty} \widehat{f}(t/n + j) \\
&= g'_t.
\end{aligned}$$

□

Lemma 4.6.3. *For any parameters $B \geq 1, \delta > 0$, and $\alpha > 0$, there exist flat window functions G' and $\widehat{G}$ such that $\widehat{G}$ can be computed in $O(\frac{B}{\alpha} \log(n/\delta))$ time, and for each i, G'_i can be evaluated in $O(\log(n/\delta))$ time.*

Proof. We will show this for a function G' that is a Gaussian convolved with a rectangular filter. First we construct analogous window functions for the continuous Fourier transform. We then show that discretizing these functions gives the desired result.

For some parameters σ and 1 with $1 < \sigma \leq n$ and $C < 1$ to be determined later, define D and F to be Gaussian and rectangular filters, respectively, according to:

- $D(s) = \frac{\sigma}{\sqrt{2\pi}} e^{-\sigma^2 s^2/2}$ is a Gaussian pdf with standard deviation $1/\sigma$.
- $\widehat{D}(t) = e^{-2\pi^2 t^2/\sigma^2}$ is $\sigma/\sqrt{2\pi}$ times a Gaussian pdf with standard deviation $\sigma/2\pi$
- $F(s) = \text{rect}(s/(2C))$ is 1 if $|s| < C$ and 0 otherwise, is a rectangular filter of length $2C$.
- $\widehat{F}(t) = 2C\text{sinc}(2Ct) = \frac{\sin(2\pi Ct)}{\pi t}$.

Consider the filter

$$G^* = D * F$$
$$\widehat{G}^* = \widehat{D} \cdot \widehat{F}.$$

We have $|\widehat{G}^*(t)| \leq 2C|\widehat{D}(t)| < 2C\delta$ for $|t| > \frac{\sigma}{2\pi}\sqrt{2\log(1/\delta)}$. Furthermore, $\widehat{G}^*$ is computable in $O(1)$ time.

Its inverse Fourier transform is $G^*(s) = \mathrm{cdf}(\sigma(s+C)) - \mathrm{cdf}(\sigma(s-C))$. By Claim 4.6.1 we have for $|s| > C + \sqrt{2\log(1/\delta)}/\sigma$ that $G^*(s) = \pm\delta$. We also have, for $|s| < C - \sqrt{2\log(1/\delta)}/\sigma$, that $G^*(s) = 1 \pm 2\delta$.

This gives us efficient *continuous* flat window functions. To get *discrete* ones, for $i \in [n]$ let $\widehat{H}_i = \sqrt{n}\sum_{j=\infty}^{\infty} \widehat{G}^*(i+nj)$. By Claim 4.6.2 it has DFT $H_i = \sum_{j=\infty}^{\infty} G^*(i/n + j)$.

We show how to approximate H and $\widehat{H}$ efficiently. First, $\widehat{H}$:

$$\begin{aligned}
\sum_{|i|>1+\frac{\sigma}{2\pi}\sqrt{2\log(1/\delta)}} |\widehat{G}^*(i)| &\leq 4C \sum_{i<-1-\frac{\sigma}{2\pi}\sqrt{2\log(1/\delta)}} |\widehat{D}(i)| \\
&\leq 4C \int_{-\infty}^{-\frac{\sigma}{2\pi}\sqrt{2\log(1/\delta)}} |\widehat{D}(x)|dx \\
&\leq 4C \frac{\sigma}{\sqrt{2\pi}} \mathrm{cdf}(-\sqrt{2\log(1/\delta)}) \\
&< 2C\sigma\delta \leq 2n\delta.
\end{aligned}$$

Thus if we let

$$\widehat{G}_i = \sqrt{n} \sum_{\substack{|j|<\frac{\sigma}{2\pi}\sqrt{2\log(1/\delta)} \\ j\equiv i \pmod n}} \widehat{G}^*(j)$$

for $|i| < \frac{\sigma}{2\pi}\sqrt{2\log(1/\delta)}$ and $\widehat{G}_i = 0$ otherwise, then $\|\widehat{G} - \widehat{H}\|_1 \leq 2\delta n^{3/2}$.

Now consider approximating H. Note that for integer i with $|i| \leq n/2$,

$$\begin{aligned}
H_i - G^*(i/n) &= \sum_{\substack{j\in\mathbb{Z} \\ j\neq 0}} G^*(i/n + j) \\
|H_i - G^*(i/n)| &\leq 2\sum_{j=0}^{\infty} G^*(-1/2 - j) \\
&\leq 2\sum_{j=0}^{\infty} \mathrm{cdf}(\sigma(-1/2 - j + C)) \\
&\leq 2\int_{-\infty}^{-1/2} \mathrm{cdf}(\sigma(x+C))dx + 2\,\mathrm{cdf}(\sigma(-1/2+C)) \\
&\leq 2\delta/\sigma + 2\delta \leq 4\delta
\end{aligned}$$

by Claim 4.6.1, as long as

$$\sigma(1/2 - C) > \sqrt{2\log(3/\delta)}. \tag{4.10}$$

Let

$$G'_i = \begin{cases} 1 & |i| \le n(C - \sqrt{2\log(1/\delta)}/\sigma) \\ 0 & |i| \ge n(C + \sqrt{2\log(1/\delta)}/\sigma) \\ \widetilde{\mathrm{cdf}}_\delta(\sigma(i/n + C)) - \widetilde{\mathrm{cdf}}_\delta(\sigma(i/n - C)) & \text{otherwise} \end{cases}$$

where $\widetilde{\mathrm{cdf}}_\delta(t)$ computes $\mathrm{cdf}(t)$ to precision $\pm\delta$ in $O(\log(1/\delta))$ time, as per Claim 4.6.1. Then $G'_i = G^*(i/n) \pm 2\delta = H_i \pm 6\delta$. Hence

$$\begin{aligned} \|G - G'\|_\infty &\le \|G' - H\|_\infty + \|G - H\|_\infty \\ &\le \|G' - H\|_\infty + \|G - H\|_2 \\ &= \|G' - H\|_\infty + \|\widehat{G} - \widehat{H}\|_2 \\ &\le \|G' - H\|_\infty + \|\widehat{G} - \widehat{H}\|_1 \\ &\le (2n^{3/2} + 6)\delta. \end{aligned}$$

Replacing δ by δ/n^2 and plugging in $\sigma = \frac{4B}{\alpha}\sqrt{2\log(n/\delta)} > 1$ and $C = (1 - \alpha/2)/(2B) < 1$, we have the required properties of flat window functions:

- $|\widehat{G}_i| = 0$ for $|i| \ge \Omega(\frac{B}{\alpha}\log(n/\delta))$
- $G'_i = 1$ for $|i| \le (1 - \alpha)n/(2B)$
- $G'_i = 0$ for $|i| \ge n/(2B)$
- $G'_i \in [0, 1]$ for all i.
- $\|G' - G\|_\infty < \delta$.
- We can compute $\widehat{G}$ over its entire support in $O(\frac{B}{\alpha}\log(n/\delta))$ total time.
- For any i, G'_i can be computed in $O(\log(n/\delta))$ time for $|i| \in [(1 - \alpha)n/(2B), n/(2B)]$ and $O(1)$ time otherwise.

The only requirement was Equation (4.10), which is that

$$\frac{4B}{\alpha}\sqrt{2\log(n/\delta)}(1/2 - \frac{1 - \alpha/2}{2B}) > \sqrt{2\log(3n/\delta)}.$$

This holds if $B \ge 2$. The $B = 1$ case is trivial using the constant function $G'_i = 1$. □

Chapter 5

Sparse Fourier Transforms: Optimizing Measurements

(Based on parts of [IKP13])

This chapter revisits the sparse Fourier transform problem. We give a randomized algorithm that takes $O(k \log n (\log \log n)^{O(1)})$ samples and uses $O(k \log^2 n (\log \log n)^{O(1)})$ time, assuming that the entries of the signal are polynomially bounded. The sampling complexity improves over the $O(k \log n \log(n/k))$ bound in Chapter 4, and matches the lower bound of $\Omega(k \log(n/k)/\log \log n)$ from Chapter 6 up to $\text{poly}(\log \log n)$ factors when $k = O(n^{1-\delta})$ for a constant $\delta > 0$.

As a reminder to set notation, our algorithm has access to the Fourier transform[1] $\widehat{x}$ of x. We would like to compute an approximation x' to x such that

$$\|x - x'\|_2 \leq C \min_{k\text{-sparse } y} \|x - y\|_2. \tag{5.1}$$

for some approximation factor $C = 1 + \epsilon \lesssim 1$.

5.1 Techniques

Our algorithm follows a similar approach to [GMS05] and our time-optimizing Chapter 4, which try to adapt the methods of [CCF02, GLPS10] from arbitrary linear measurements to Fourier ones. We use a "filter" that lets us "hash" the k large frequencies to $B = O(k)$ buckets. This lets us "locate"—i.e., find the indices of—many of the large frequencies. We then "estimate" the value of x at these frequencies, giving a sparse estimate χ of x. To improve this estimate, we can repeat the process on $x - \chi$ by subtracting the influence of χ during hashing. This repetition will yield a good sparse approximation χ of x.

The methods of [CCF02, GLPS10] will, multiple times, take a set of B linear measurements of the form

$$\tilde{u}_j = \sum_{i:h(i)=j} s_i x_i$$

for random hash functions $h : [n] \to [B]$ and random sign changes s_i with $|s_i| = 1$. This denotes *hashing* to B *buckets*. With such ideal linear measurements, $O(\log(n/k))$ hashes suffice for sparse recovery, giving an $O(k \log(n/k))$ sample complexity.

To perform sparse Fourier transforms, both [GMS05] and our Chapter 4 approximate $\tilde{u}$ using

[1] This is the *inverse* discrete Fourier transform problem. It is equivalent to the forward direction modulo some conjugation and has simpler notation.

linear combinations of Fourier samples. They use *filters* to compute $u \approx \tilde{u}$ using somewhat more than B *Fourier* measurements. Choosing a filter involves a tradeoff between the approximation quality and increase in number of samples. As described in Section 5.3, for any parameter $R > 2$, using $O(B \log R)$ Fourier measurements we can get (very roughly) that $\|u - \tilde{u}\|_2 \leq \|x\|_2/R$. We refer to this error $(u - \tilde{u})$, which is mostly caused by elements x_i contributing to buckets other than $h(i)$, as "leakage."

The difference between [GMS05] and Chapter 4 is largely driven by a different choice of filters. [GMS05] uses a filter with $R = O(1)$, which gives efficient sample complexity per hashing but involves lots of leakage. Dealing with this leakage requires multiple logarithmic factors of overhead in the number of hashes. By contrast, Chapter 4 uses a filter with $R = n^{O(1)}$. This filter loses one logarithmic factor in sample complexity, but makes leakage negligible for polynomially bounded inputs. The rest of the algorithm then can proceed somewhat similarly to [GLPS10] and be optimal, giving $O(k \log n \log(n/k))$ sample complexity.

In this chapter we observe that setting $R = n^{O(1)}$ is often overkill: in many cases the post-filtering parts of Chapter 4 can tolerate a larger amount of leakage (and hence use a filter that performs fewer measurements). Moreover, the situations where R must be large are precisely the situations where the post-filtering parts of Chapter 4 can be made more efficient and use $o(\log(n/k))$ hashings. We give a broad outline of our analysis, starting with a special case.

Similar magnitude heavy hitters. Even with the "ideal" hashing $\tilde{u}$, we expect an average of around $\mu^2 = \mathrm{Err}_k^2(x)/B$ "noise" from the tail in each of the $B = O(k)$ buckets, where $\mathrm{Err}_k(x)$ denotes $\min_{k\text{-sparse } y} \|x - y\|_2$. This means that the post-filtering steps of the algorithm must already tolerate average noise of order μ^2.

For intuition, it is useful to consider recovery of a signal where the largest k coordinates are all between $\sqrt{R}\mu$ and $R\mu$ for a parameter $R \geq 2$. Then choosing the filter with $O(\log R)$ overhead, i.e. performing $O(B \log R)$ Fourier measurements, the average leakage will be

$$\frac{1}{B}\|\tilde{u} - u\|_2^2 \leq \frac{1}{R^2 B}\|x\|_2^2 \leq \frac{k \cdot (R\mu)^2 + \mathrm{Err}_k^2(x)}{R^2 B} < \mu^2.$$

This means that the post-filtering steps of the algorithm will succeed, giving a sample complexity of $O(k \log R \log(n/k))$. This is a great improvement over the $O(k \log n \log(n/k))$ sampling complexity of Chapter 4 when R is small, but if R is polynomially large we have not gained anything.

The next insight is that we can use fewer than $\log(n/k)$ hashings if the smallest heavy hitter has value $\sqrt{R}\mu^2 \gg \mu^2$. Indeed, the bottleneck in these algorithms is the location phase, where we need to recover $\log(n/k)$ bits about each large frequency (in order to identify it among the n/k different frequencies in the bucket). While [GMS05] and Chapter 4 recover $O(1)$ of these bits per hashing, their methods can actually recover $\Omega(\log R)$ bits per hashing in this case because the expected signal to noise ratio in each bucket is $\Omega(R)$. This gives a sample complexity of $O(k \log R \log_R(n/k)) = O(k \log(Rn/k))$.

Our algorithm uses the approach we just outlined, but also needs to cope with additional difficulties that we ignored in the sketch above. First, in the general case we cannot expect all heavy hitters to be in the range $[\sqrt{R}\mu^2, R\mu^2]$, and the argument above does not give any guarantees on recovery of smaller elements. Additionally, the sketch above ignores collisions during hashing, which cause us to only recover a constant fraction of the heavy hitters in each round. We now give an outline of our approach to the general problem.

General vectors. The above algorithm finds most of the large frequencies if they all have value between $\sqrt{R}\mu^2$ and $R\mu^2$ for a known R. More generally, if $\|x\|_2^2 \leq Rk\mu^2$, the same techniques can recover most of the frequencies of magnitude larger than $R^\delta\mu^2$ with sample complexity $O(\frac{1}{\delta}k\log(Rn/k))$, for a parameter $\delta > 0$: we perform $O(\log_{R^\delta}(n/k))$ hashings that each take $O(k\log R)$ samples. Call this algorithm $\mathcal{A}(R,\delta)$.

Our algorithm will repeat $\mathcal{A}(R,\delta)$ multiple times for some δ. After enough repetitions, we will recover almost every coordinate larger than $\sqrt{R}\mu^2$. The residual will then have norm bounded by $O(\sqrt{R}k\mu^2)$. Our algorithm takes the following form: we repeat $\mathcal{A}(\sqrt{R},\delta)$ multiple times, then $\mathcal{A}(R^{1/4},\delta)$, and so on. After $\log\log R$ rounds of this, the residual will have norm $O(k\mu^2)$ and we can perform recovery directly. For this technique to work with $(\log\log(Rn))^c$ overhead, we will show that $\log\log R$ repetitions of $\mathcal{A}(R,\delta)$ suffice to reduce the residual norm to $\sqrt{R}k\mu^2$, for some $\delta = \Omega(1/\log\log R)$.

A first attempt might be to set $\delta = 1/2$, thus recovering most of the coordinates larger than $\sqrt{R}k\mu^2$ in each stage. This leads to problems if, for example, the vector has $k/2$ elements of value $R^{.4}\mu^2$ and $k/2$ elements of value $R^{.6}\mu^2$. Then $\mathcal{A}(R,1/2)$ will never recover the first $k/2$ coordinates, and collisions with those coordinates mean it will only recover a constant fraction of the second $k/2$ coordinates. So it takes $\Omega(\log R) \gg \log\log R$ repetitions to reduce the residual from $R^{.6}k\mu^2$ to $\sqrt{R}k\mu^2$. This is too slow; we need to make the number of elements above $\sqrt{R}\mu^2$ decay doubly exponentially.

This suggests that we need a more delicate characterization of $\mathcal{A}(r,\delta)$. We show in our analysis that coordinates are recovered with high probability if they are "well-hashed," meaning that the total noise in the bucket is R^δ smaller than the value of the coordinate. Coordinates of magnitude $R^\delta\mu^2$ have a constant chance of being well-hashed (leading to singly exponential decay), and coordinates that are much larger than $R^\delta\mu^2$ have a higher chance of being well-hashed (ultimately yielding the required doubly exponential decay). Our analysis follows this outline, but has to handle further complications that arise from imperfect estimation phase. For simplicity, we first present the analysis assuming perfect estimation, and then give the proof without any assumptions.

General vectors: perfect estimation. We classify the elements of the signal into $1/\delta$ "levels" of elements between $[R^{\delta j}\mu^2, R^{\delta(j+1)}\mu^2]$ for $j = 0,\ldots,1/\delta - 1$, as opposed to a single range like $[\sqrt{R}\mu^2, R\mu^2]$. We then bound the success rate of recovery at each level in terms of the number of elements in various levels above and below it.

To first approximation, coordinates are recovered and eliminated from the residual if they are well-hashed, and are not recovered if they are not well-hashed. And in most cases the probability that a large coordinate j is not well-hashed is dominated by the probability that it collides with a coordinate of magnitude at least $R^{-\delta}|x_j|^2$. In this approximation, if we set $m_\ell(t)$ to be the number of $|x_j|^2$ larger than $R^{\ell\delta}\mu^2$ after t rounds of the algorithm, then $\mathbb{E}[m_\ell(t+1)] \leq m_\ell(t)m_{\ell-1}(t)/B$. Then m_0 doesn't decay—coordinates less than $R^\delta\mu^2$ will not be recovered by $\mathcal{A}(R,\delta)$—but m_1 decays exponentially, m_2 will then decay as 2^{-t^2}, and in general m_ℓ will decay as $2^{-\binom{t}{\ell}}$. With $\delta = 1/\log\log R$, we find that $m_{1/\delta-1}$ (which contains all coordinates larger than $\sqrt{R}\mu^2$) will decay to $1/R^c$ in $O(\log\log R)$ rounds. As a result, the squared norm of the residual will be at most $O(\sqrt{R}\mu^2)$. The details of this part of the analysis are presented in Section 5.6.

General vectors: actual behavior. In the actual algorithm, coordinates do not just disappear if they are located, but are estimated with some error. This means large components can appear in the residual where no component was before, if lots of small components were hashed to a certain bucket. This causes the m_ℓ to not obey the nice recurrence in the previous paragraph. To deal

with this, we introduce the notion of *splittings* of the residual. For analysis purposes, we split each component of the residual into multiple terms whose total magnitude is the same. We define the m_ℓ in terms of the number of components in the splitting, not the actual residual.

The intuition is that the residual error when estimating an element x_i is approximately $\|x_C\|_2$, where $C \subset [n]$ is the set that "collides" with i. Rather than thinking of the residual as a single coordinate with value $\|x_C\|_2$, we "split" it and imagine duplicating x_j for each $j \in C$. Because $j \in C$ was not recovered from the bucket, j was (most likely) not well-hashed. So the contribution of the duplicated x_j to m_ℓ is comparable to the contribution of the x_j that remain after not being well-hashed. Hence the m_ℓ obey almost the same recurrence as in the perfect estimation setting above.

As a result, $O(\log\log R)$ repetitions of $\mathcal{A}(R, 1/\log\log R)$ reduce the residual norm to $\sqrt{R}k\mu^2$. Repeating for $\log\log n$ rounds decreases R from n^c to $O(1)$, and we can finish off by accepting a $\log R$ loss. The details of this part of the analysis are presented in Section 5.7.

5.2 Notation and definitions

We will use the orthonormal version of the Fourier transform. For $x \in \mathbb{R}^n$

$$\widehat{x}_j = \frac{1}{\sqrt{n}} \sum_{i \in [n]} \omega^{ij} x_i, \tag{5.2}$$

where ω is a root of unity of order n. The inverse transform is given by

$$x_j = \frac{1}{\sqrt{n}} \sum_{i \in [n]} \omega^{-ij} \widehat{x}_i. \tag{5.3}$$

We assume that n is a power of 2.

5.2.1 Notation

We will use the same pseudorandom spectrum permutation as Chapter 4, which we now define.

Definition 4.2.1. *Suppose σ^{-1} exists mod n. We define the* permutation $P_{\sigma,a,b}$ *by*

$$(P_{\sigma,a,b}\widehat{x})_i = \widehat{x}_{\sigma(i+a)}\omega^{-\sigma b i}.$$

We also define $\pi_{\sigma,b}(i) = \sigma(i-b) \bmod n$.

Claim 4.2.2. *Let $\mathcal{F}^{-1}(x)$ denote the inverse Fourier transform of x. Then*

$$(\mathcal{F}^{-1}(P_{\sigma,a,b}\widehat{x}))_{\pi_{\sigma,b}(i)} = x_i \omega^{a\sigma i}.$$

Also, define

- $h_{\sigma,b}(i) = \text{round}(\pi_{\sigma,b}(i)n/B)$ to be an $[n] \to [B]$ "hash function" .
- $o_i(j) = \pi(j) - (n/B)h(i)$ to be the "offset" of j relative to i.

This "hashing" h is approximately pairwise independent in the following sense:

Lemma 4.2.4. (Lemma 3.6 of [HIKP12c]) *If $j \neq 0$, n is a power of two, and σ is a uniformly random odd number in $[n]$, then* $\Pr[\sigma j \in [-C, C] \pmod n] \leq 4C/n$.

In much of the chapter, we use $|i|$ for $i \in [n]$ to denote $\min_{z \in \mathbb{Z}} |i + zn|$; this is the "absolute value modulo n." So the above lemma, for example, bounds $\Pr[|\sigma j| \leq C]$.

Our algorithm will start with an input $\widehat{x}^*$ and find progressively better approximations χ to x^*. Most of the analysis will depend only on $x := x^* - \chi$. Our algorithm will involve decreasing the "signal to noise ratio" $R \approx \|x\|_2^2 / \operatorname{Err}_k^2(x^*)$.

Pseudocode for our algorithm is given below. Due to space constraints, the pseudocode for the function LocateSignal appears in Section 5.10.

```
1: procedure SparseFFT(x̂, k, ε, R, p)
2:     χ^(0) ← 0                                              ▷ in ℂ^n.
3:     R_0 ← R
4:     r ← Θ(log log R)
5:     for i = 0, 1, ..., r − 1 do
6:         χ' ← ReduceSNR(x̂, χ^(i), 3k, R_i, p/(2r))
7:         χ^(i+1) ← Sparsify(χ^(i) + χ', 2k)              ▷ Zero out all but top 2k entries
8:         R_{i+1} ← c√R_i                                 ▷ For some constant c
9:     end for
10:    χ' ← RecoverAtConstantSNR(x̂, χ^(r), 3k, ε, p/2)
11:    return χ^(r) + χ'
12: end procedure
```

Algorithm 5.2.1: Overall algorithm: perform Sparse Fourier Transform

```
1: procedure ReduceSNR(x̂, χ, k, R, p)
2:     B ← (1/α)k for sufficiently small α > 0.
3:     χ^(1) ← χ
4:     N ← Θ(log_2 log_2 R)
5:     for t = 0, 1, ..., N − 1 do
6:         k_t ← O(k4^{−t})
7:         L ← LocateSignal(x̂, χ^(i), B, σ, b, R, αR^{−20})
8:         x̃ ← EstimateValues(x̂, χ^(i), L, B, 3k_t, 13, R)
9:         χ^(i+1) ← χ^(i) + x̃
10:    end for
11:    return χ^N − χ
12: end procedure
```

Algorithm 5.2.2: Reduce the SNR $\|x\|_2^2/\xi^2$ from R to $O(\sqrt{R})$

5.2.2 Glossary of terms in ReduceSNR and RecoverAtConstantSNR

In ReduceSNR and RecoverAtConstantSNR, there are a lot of variables with common names and similar purposes. This section provides a glossary, which may be useful for reference. We have globally:

- $\widehat{x}^* \in \mathbb{C}^n$ is the original input, where we want to recover an approximation to x^*.
- k^* is the original value of k, for which we expect $\widehat{x}^*$ to be approximately k^*-sparse.
- $R^* \geq \|x^*\|_2^2 / \operatorname{Err}_k^2(x^*)$ is an upper bound on the SNR for the original signal. We have $R^* = O(\text{poly}(n))$ by the input assumption.

procedure RecoverAtConstantSNR($\widehat{x}, \chi, k, \epsilon, p$)
    $R \leftarrow 20$
    $B \leftarrow Rk/(\epsilon\alpha p)$ for a sufficiently small constant $\alpha > 0$
    Choose σ, b uniformly at random in $[n]$, σ odd.
    $L \leftarrow$ LocateSignal($\widehat{x}, \chi, B, \sigma, b, R, \alpha\epsilon p$)
    $\chi' \leftarrow$ EstimateValues($\widehat{x}, \chi, L, B, 3k, \log(B/(4k)), R$)
    return χ'
end procedure

Algorithm 5.2.3: Recovery when $\|x - \chi\|_2 \lesssim \mathrm{Err}_k^2(x)$

procedure EstimateValues($\widehat{x}, \chi, L, B, k, T, R$)
    for $t = 1$ to T **do**
        Choose $\sigma, b, a \in [n]$ uniformly at random, σ odd
        $u \leftarrow$ HashToBins($x, \chi, P_{\sigma,a,b}, B, R$)
        $\tilde{x}_i^{(t)} \leftarrow G_{o_i(i)}^{-1} u_{h_{\sigma,b}(i)} \omega^{-a\sigma i}$ for all $i \in L$. ▷ Note that $G_{o_i(i)}$ depends on σ, b, a
    end for
    $\tilde{x}_i \leftarrow \mathrm{median}_t(\tilde{x}_i^{(t)})$ for all $i \in L$. ▷ Median in real and imaginary axis separately
    return Sparsify($\tilde{x}, k$).
end procedure

Algorithm 5.2.4: Estimation: estimate $(x - \chi)_L$ using T rounds of B-bucket, contrast R hashing.

procedure HashToBins($\widehat{x}, \chi, P_{\sigma,a,b}, B, R$)
    $G \leftarrow$ flat window function with B buckets and contrast R.
    Compute $y' = \widehat{G} \cdot P_{\sigma,a,b}(\widehat{x} - \widehat{\chi}')$, $\|\widehat{\chi} - \widehat{\chi}'\|_\infty < \frac{\|\chi\|_2}{R^* n^{12}}$ ▷ Have $\|y'\|_0 \lesssim B \log R$
    Compute $u_j = \widehat{y'}_{jn/B}$ for $j \in [B]$, where $\|\Delta\|_\infty \leq \frac{\|\chi\|_2}{R^2 n^{11}}$
    return u.
end procedure

Algorithm 5.2.5: Hashing using Fourier samples (analyzed in Lemma 5.11.3)

and for each different call to ReduceSNR and RecoverAtConstantSNR, we have

- $\chi \in \mathbb{C}^n$ is our current best guess at x, which will (in all calls) be $2k^*$-sparse.
- $x = x^* - \chi$ is the "residual" signal that we want to recover in this call. (When analyzing ReduceSNR, we set $x = x^* - \chi^{(i)}$ in each inner loop.)
- $k = 3k^*$ is the approximate sparsity of x.
- $\alpha > 0$ is sufficiently small, but at least $1/(\log\log n)^c$.
- $B > k/\alpha$ to be the number of buckets in each hashing.
- T to be the number of hashings done inside EstimateValues.
- $\tilde{x}^{(t)} \in \mathbb{C}^n$ is the estimation of x in EstimateValues for each $t \in [T]$.
- $\tilde{x} \in \mathbb{C}^n$ is the median of $\tilde{x}^{(t)}$ over t.
- R will is a parameter (in ReduceSNR) and sufficiently large constant (in RecoverAtConstantSNR). It roughly represents the "signal-to-noise ratio".
- $\delta = 1/(40\log_2\log_2 R)$.
- $\gamma = R^{-\delta}$ to be the "contrast" our LocateSignal requires.

5.3 Properties of the bucketing scheme

Our algorithm uses filters and various choices of σ, b, a to "hash" the coordinates of x into buckets. For each (σ, b, a) and each bucket $j \in [B]$ we recover an estimate i^* of the heavy coordinate in that bucket. Also, for each $i \in [n]$ we can recover an estimate $\tilde{x}_i$ of x_i.

We use a modification of the filters from Chapter 4 that allows the noise in a single bucket to depend on the energy of the signal as well as the energy of the colliding elements.

Definition 5.3.1 (Flat Window Functions)**.** *A flat window function G over $\mathbb{C}^n$ has B* buckets *and* contrast *R if, for $|i| \le n/2$, we have*

- *$G_i \ge 1/3$ for $|i| \le n/(2B)$.*
- *$0 \le G_i \le 1$ for all i.*
- *$G_i \le (\frac{cn}{|i|B})^{\log R}$ for all i for some constant c*

The filters in Chapter 4 were more stringent, roughly corresponding to the $R = n^{O(1)}$ case. We will prove in Section 5.11 that

Lemma 5.3.2. *There exist flat window functions where $|\operatorname{supp}(\widehat{G})| \lesssim B\log R$. Moreover, $\operatorname{supp}(\widehat{G}) \subset [-O(B\log R), O(B\log R)]$.*

Most of the analysis in this chapter will assume we have precomputed $\widehat{G}$ and G and may access them with unit cost. This is unnecessary: in Section 5.12.1 we describe how to compute them on the fly to $1/n^c$ precision without affecting our overall running time. This precision is sufficient for our purposes.

Lemma 5.3.3. *Let $(\sigma, a, b) \in [n]$ be uniform subject to σ being odd. Let $u \in \mathbb{C}^B$ denote the result of* HashToBins*$(\widehat{x}^*, \chi, P_{\sigma,a,b}, B, R)$. Fix a coordinate $i \in [n]$ and define $x = x^* - \chi$. For each (σ, b), we can define variables $C \subset [n]$ and $w > 0$ (and in particular, $C = \{j \ne i : |\sigma(i-j) \bmod n| \le cn/B\}$ for some constant c,) so that*

- *For all j, as a distribution over (σ, b),*

$$\Pr[j \in C] \lesssim 1/B.$$

- *As a distribution over (σ, b),*

$$\mathbb{E}[w^2] \lesssim \frac{\|x\|_2^2}{R^2 B} + \frac{\|x^*\|_2^2}{R^* n^{11}}$$

- *Conditioned on (σ, b) and as a distribution over a,*

$$\mathop{\mathbb{E}}_{a}[|G^{-1}_{o_i(i)}\omega^{-a\sigma i}u_{h(i)} - x_i|^2] \lesssim w^2 + \|x_C\|_2^2.$$

Intuitively, C denotes the elements of x that collide with i, and w denotes the rest of the noise. The two terms of w correspond to leakage of x from other hash locations and to errors in the subtraction of χ, respectively. This latter term should be thought of as negligible.

We also define the notion of being "well-hashed," which depends on another parameter $\gamma = R^\delta$ from the glossary:

Definition 5.3.4. *Let $\sigma, b \in [n], \sigma$ odd. An element i is well-hashed for a particular σ, b and filter G if over uniformly random $a \in [n]$,*

$$\mathop{\mathbb{E}}_{a}[|G^{-1}_{o_i(i)}\omega^{-a\sigma i}u_{h(i)} - x'_i|^2] \le \gamma^{1/2}|x'_i|^2.$$

Intuitively, a well-hashed element contains little noise in the bucket that it hashed to, relative to its own energy, and will hence be likely to be recovered in LocateSignal. This is formalized in Lemma 5.10.2.

5.4 Proof overview

This section gives the key lemmas that are proven in later sections. Our procedures try to reduce the ℓ_2 norm of the residual to the "noise level" $\xi^2 := \mathrm{Err}^2_{k^*}(x^*) + \|x^*\|_2^2/(R^* n^{10})$. The polynomial n^{10} can be arbitrary, and only affects the running time of the algorithm; we choose a specific constant for simplicity of notation. The $\|x^*\|_2^2/(R^* n^{10})$ term is essentially irrelevant to the behavior of the algorithm, and will be ignored when discussing intuition.

First, we give an algorithm RecoverAtConstantSNR that is efficient when $\|x\|_2^2 \lesssim \mathrm{Err}^2_k(x)$.

Lemma 5.5.1. *For $x^*, \chi \in \mathbb{C}^n$ define $x = x^* - \chi$. Then* RecoverAtConstantSNR$(\widehat{x}^*, \chi, k, \epsilon, p)$ *returns χ' such that*

$$\|x - \chi'\|_2^2 \le \mathrm{Err}^2_k(x) + \epsilon\|x\|_2^2 + \frac{\|x^*\|_2^2}{n^{10}}$$

with probability $1 - p$, using $O(\frac{1}{p\epsilon} k \log(n/k) \log\log(n/k) \log(1/(\epsilon p)))$ measurements and (assuming $\|\chi\|_0 \lesssim k$) a $\log n$ factor more time.

This is relatively straightforward. To see why it is useful, for $k = 3k^*$ we have $\mathrm{Err}^2_k(x) \le \mathrm{Err}_{k^*}(x^*)$. Therefore, once χ is close enough to x^* that $x = x^* - \chi$ has $\|x\|_2^2 \lesssim \mathrm{Err}^2_{k^*}(x^*)$, this lemma gives that $\chi + \chi'$ is within $(1+\epsilon)\,\mathrm{Err}^2_{k^*}(x^*)$ of x^* using only $O^*(\frac{1}{p\epsilon} k \log(n/k) \log(1/(\epsilon p)))$ measurements. (As stated above, for intuition we are ignoring the negligible $\frac{\|x^*\|_2^2}{n^{10}}$ term.)

We then show how to quickly reduce $\|x\|_2^2$ to $O(\mathrm{Err}^2_{k^*}(x^*))$:

Lemma 5.7.11. *For $x^*, \chi \in \mathbb{C}^n$ define $x = x^* - \chi$ and*

$$\xi^2 = \mathrm{Err}_k^2(x) + \frac{\|x\|_2^2}{R} + \frac{\|x^*\|_2^2}{R^* n^{10}}.$$

Suppose $\|\chi\|_0 \le k$. Then ReduceSNR$(\widehat{x}^*, \chi, k, R, p)$ *returns χ' such that*

$$\|x - \chi'\|_2^2 \lesssim \sqrt{R}\xi^2$$

with probability $1-p$, using $O(\frac{1}{p^2} k \log(Rn/k)(\log\log(Rn/k))^c)$ measurements and a $\log(Rn)$ factor more time.

This is the most technical part of the chapter. By iterating it $\log\log R$ times and finishing off with Lemma 5.5.1, we get the final result:

Theorem 5.8.1. *Let $x \in \mathbb{C}^n$ satisfy $\|x\|_2^2 \le R\,\mathrm{Err}_k^2(x)$. Then* SparseFFT$(\widehat{x}, k, R, p)$ *returns a χ' such that*

$$\|x - \chi'\|_2^2 \le (1+\epsilon)\,\mathrm{Err}_k^2(x) + \|x\|_2^2/(R^* n^{10})$$

with probability $1-p$ and using $O(\frac{1}{p^2\epsilon} k \log(Rn/k)(\log\log(Rn/k))^c \log(1/\epsilon))$ measurements and a $\log(Rn)$ factor more time.

We now summarize the proof of Lemma 5.7.11. Let $x = x^* - \chi$, and define

$$\mu^2 = \frac{1}{k}\xi^2 \ge \left(\mathrm{Err}_k^2(x) + \frac{\|x\|_2^2}{R}\right)/k.$$

If we hash to $B = k/\alpha$ buckets with flat window functions of contrast R, then the expected magnitude of the contribution of the tail of x to any bucket is $O(\alpha\mu^2)$.

ReduceSNR involves $O(\log\log R)$ stages. In each stage, we hash to B bins and call LocateSignal to get a set L of candidate locations for heavy hitters. We then estimate x_i for each $i \in L$ as the median $\tilde{x}_i$ of $O(1)$ random hashings to B bins. We then subtract off $\tilde{x}_{L'}$, where L' contains the largest k' coordinates of $\tilde{x}$ and k' starts out $\Theta(k)$ in the first stage and decreases exponentially. So the recurrence in each stage is $x \to x - \tilde{x}_{L'}$.

This process is somewhat complicated, so we start by analyzing a simpler process in each stage. Let S denote the set of "well-hashed" coordinates $i \in [n]$, i.e. coordinates that are hashed to bins with noise less than $\gamma^{1/2}|x_i|^2$. In Section 5.6 we analyze the recurrence $x \to x - x_S$. Generally, we expect larger elements to be more likely to be well-hashed, and so the number of them to decay more quickly. We analyze the number $m_\ell(t)$ of i with $|x_i| > \mu^2\gamma^{-\ell}$ that remain at each stage t, for each level ℓ. We show that these quantities obey a nice system of equations, causing the $m_\ell(t)$ to decay doubly exponentially for the first ℓ rounds. Then after $t = O(\log\log R)$ rounds, an R^{-10} fraction of the coordinates larger than $\mu^2\sqrt{R}$ remain. This means that the recurrence $x \to x - x_S$ would leave a remainder of norm $O(k\mu^2\sqrt{R})$ as desired.

In Section 5.7, we relate this to the true recurrence $x \to x - \tilde{x}_{L'}$. We study recurrences that are *admissible*, meaning that they satisfy a similar system of equations to that in Section 5.6. Admissible recurrences satisfy composition rules that let us find them sequentially, and using Section 5.6 we can show the remainder after $\log\log R$ iterations of any admissible recurrence has small norm. In a series of results, we show that $x \to x - \tilde{x}_S$, $x \to x - x_{L'}$, and finally $x \to x - \tilde{x}_{L'}$ are admissible. This then proves Lemma 5.7.11.

5.5 Constant SNR

Our procedure for recovery at constant SNR is given by Algorithm 5.2.3. In this section we prove

Lemma 5.5.1. *For $x^*, \chi \in \mathbb{C}^n$ define $x = x^* - \chi$. Then* RecoverAtConstantSNR$(\widehat{x}^*, \chi, k, \epsilon, p)$ *returns χ' such that*

$$\|x - \chi'\|_2^2 \leq \mathrm{Err}_k^2(x) + \epsilon\|x\|_2^2 + \frac{\|x^*\|_2^2}{n^{10}}$$

with probability $1 - p$, using $O(\frac{1}{p\epsilon} k \log(n/k) \log\log(n/k) \log(1/(\epsilon p)))$ measurements and (assuming $\|\chi\|_0 \lesssim k$) a $\log n$ factor more time.

In what follows we define

$$\xi^2 = \|x\|_2^2 + \|x^*\|_2^2/(R^* n^{11}).$$

and $\mu^2 = \xi^2/k$. By definition of the algorithm, $B = Rk/(\epsilon\alpha p)$ for some constants R, α. We will show that, if R is a sufficiently large constant, then with probability $1 - p$,

$$\|x - \chi'\|_2^2 - \mathrm{Err}_k^2(x) \lesssim \alpha\epsilon\xi^2.$$

For sufficiently small α this gives the result.

A simple consequence of Lemma 5.3.3 is that for each i, and for random (σ, a, b), we have

$$\mathop{\mathbb{E}}_{a,\sigma,b}[|G^{-1}_{o_i(i)}\omega^{-a\sigma i} u_{h(i)} - x_i|^2] \lesssim \|x\|_2^2/B + \|x^*\|_2^2/(R^* n^{11}) \leq \xi^2/B = \epsilon\alpha p\mu^2/R. \tag{5.4}$$

There are two sources of error in RecoverAtConstantSNR, coming from location and estimation respectively. The proof of Lemma 5.5.1 proceeds in two stages, bounding the error introduced in both steps.

5.5.1 Energy lost from LocateSignal

Let S contain the largest k coordinates of x and L be the list of locations output by LocateSignal. In this section we bound the energy of the vector $x_{S\setminus L}$. Define

$$A_{large} = \{i \in S : |x_i|^2 \geq \alpha\epsilon\mu^2/R\}$$
$$A_{small} = \{i \in S : |x_i|^2 \leq \alpha\epsilon\mu^2/R\},$$

so that

$$||x_{A_{small}}||^2 \leq \alpha\epsilon\mu^2 k/R \leq \alpha\epsilon\xi^2. \tag{5.5}$$

For each $i \in [n]$ by (5.4) we have

$$\mathop{\mathbb{E}}_{a,\sigma,b}[|G^{-1}_{o_i(i)}\omega^{-a\sigma i} u_{h(i)} - x_i|^2] \lesssim \alpha\epsilon p\mu^2/R.$$

Consider $i \in A_{large}$, and recall Definition 5.3.4 of being well-hashed. By Markov's inequality applied to (5.4) and $R > \gamma^{-1/2}$, the probability that $i \in A_{large}$ is not well-hashed is bounded by

$$\frac{\epsilon\alpha p\mu^2/R}{\gamma^{1/2}|x_i|^2} \leq \frac{\epsilon\alpha p\mu^2}{|x_i|^2}. \tag{5.6}$$

Each well-hashed element is then located in LocateSignal with probability at least $1 - \alpha\epsilon p$

by our choice of parameters. Thus, for $i \in A_{large}$ one has

$$\Pr[i \notin L] \leq \frac{\epsilon\alpha p\mu^2}{|x_i|^2} + O(\alpha\epsilon p).$$

It then follows that

$$\begin{aligned}\mathbb{E}[||x_{S\setminus L}||^2 - ||x_{A_{small}}||^2] &= \mathbb{E}[||x_{A_{large}\setminus L}||^2] \\ &\leq \sum_{i\in A_{large}} \frac{\epsilon\alpha p\mu^2}{|x_i|^2}|x_i|^2 + \alpha\epsilon p\|x\|_2^2 \leq \alpha\epsilon p\xi^2 \\ &\leq 2\alpha\epsilon p\xi^2.\end{aligned} \tag{5.7}$$

Combined with (5.5) one has

$$||x_{S\setminus L}||^2 \lesssim \alpha\epsilon\xi^2 \tag{5.8}$$

with probability at least $1-p/2$ by Markov's inequality. It remains to consider the effect of pruning in EstimateValues.

5.5.2 Energy of $x - \chi'$

We now analyze the errors introduced in the estimation step. These errors come from two sources: estimation noise and the pruning step in EstimateValues. Let $\tilde{x}^{(t)}$ denote the estimate in each hashing in EstimateValues (defined to be zero outside L), and $\tilde{x}$ denote the coordinate-wise median over t of $\tilde{x}^{(t)}$. By definition, $\chi' = \tilde{x}_{L'}$ where L' denotes the largest $3k$ elements of $\tilde{x}$. By (5.4), for each $i \in L$ and $t \in [T]$ during estimation we have

$$\mathbb{E}[|\tilde{x}_i^{(t)} - x_i|^2] \lesssim \epsilon\alpha p\mu^2/R \leq \epsilon\alpha p\mu^2,$$

and so by properties of the median (Lemma 5.9.5),

$$\mathbb{E}[|\tilde{x}_i - x_i|^2] \leq 4\,\mathbb{E}[|\tilde{x}_i^{(t)} - x_i|^2] \lesssim \epsilon\alpha p\mu^2 \tag{5.9}$$

for all i. Now, by Lemma 5.9.1,

$$\|x - \chi'\|_2^2 = \|x - \tilde{x}_{L'}\|_2^2 \leq \mathrm{Err}_k^2(x) + 4\|(x-\tilde{x})_{S\cup L'}\|_2^2. \tag{5.10}$$

The first term appears in our output, so it is sufficient to upper bound the last term by $O(\alpha\epsilon\xi^2)$ with probability $1-p$. We write

$$\|(x-\tilde{x})_{S\cup L'}\|_2^2 \leq \|(x-\tilde{x})_{S\setminus L}\|_2^2 + \|(x-\tilde{x})_{(S\cap L)\cup L'}\|_2^2. \tag{5.11}$$

The first term is bounded by (5.8). It remains to bound this last bit, which is entirely the effect of estimation error since $(S\cap L)\cup L' \subseteq L$. By the fact that $|(S\cap L)\cup L'| \leq 4k$, Lemma 5.9.4 with

$T = O(\log(B/4k))$, and (5.9),

$$\begin{aligned}\mathbb{E}[\|(x-\tilde{x})_{(S\cap L)\cup L'}\|_2^2] &\le \max_{A\subseteq L,|A|=4k}\|(x-\tilde{x})_A\|_2^2 \\ &\lesssim 4k\cdot(B/4k)^{\Theta(1/T)}\cdot\max_i \mathbb{E}[|x_i-\tilde{x}_i^{(t)}|^2] \\ &\lesssim k\cdot 1\cdot\epsilon\alpha p\mu^2 \\ &= \epsilon\alpha p\xi^2.\end{aligned}$$

Hence by Markov's inequality, with probability at least $1-p/2$ one has $\|(x-\tilde{x})_{(S\cap L)\cup L'}\|_2^2 \lesssim \alpha\epsilon\xi^2$, and putting this together with (5.10) and (5.11), we get

$$\begin{aligned}\|x-\chi'\|_2^2 &\le \mathrm{Err}_k^2(x) + O(\alpha\epsilon)\xi^2 \\ &\le \mathrm{Err}_k^2(x) + \epsilon\xi^2\end{aligned} \tag{5.12}$$

with probability at least $1-p$, for sufficiently small constant α.

Proof of Lemma 5.5.1. The guarantee on the residual error is provided by (5.12), so it remains to verify sampling complexity. The call to LocateSignal takes order

$$B\log(Rn/B)\log\log R\log\log(n/B)\log(1/(\alpha\epsilon p)) \lesssim \frac{1}{p\epsilon}k\log(n/k)\log\log(n/k)\log(1/(\epsilon p))$$

samples by Lemma 5.10.2. The call to EstimateValues takes order

$$\log(B/4k)B\log R \lesssim \frac{1}{\epsilon p}k\log(1/(\epsilon p))$$

samples, giving the desired total sample complexity. □

5.6 Reducing SNR: idealized analysis

5.6.1 Dynamics of the process with simplifying assumptions

The goal of this section and the next is to prove the following lemma:

Lemma 5.7.11. *For $x^*, \chi \in \mathbb{C}^n$ define $x = x^* - \chi$ and*

$$\xi^2 = \mathrm{Err}_k^2(x) + \frac{\|x\|_2^2}{R} + \frac{\|x^*\|_2^2}{R^* n^{10}}.$$

Suppose $\|\chi\|_0 \le k$. Then ReduceSNR$(\widehat{x}^*, \chi, k, R, p)$ *returns χ' such that*

$$\|x-\chi'\|_2^2 \lesssim \sqrt{R}\xi^2$$

with probability $1-p$, using $O(\frac{1}{p^2}k\log(Rn/k)(\log\log(Rn/k))^c)$ measurements and a $\log(Rn)$ factor more time.

In this section we give a description of iterative process in ReduceSNR under simplifying assumptions, demonstrating the basic dynamics of the process. We will later give a general analysis.

Define $\mu^2 = \xi^2/k$, and from the glossary (Section 5.2.2) recall the definitions

$$\delta = \frac{1}{40 \log_2 \log_2 R}, \qquad \gamma = R^{-\delta}.$$

We define the following *energy levels*. For each $j = 1, \ldots, 1/\delta - 1$ let

$$L_j = [\mu^2 \cdot \gamma^{-j}, \mu^2 \gamma^{-(j+1)}],$$

and let $L_0 := [0, \mu^2 \gamma^{-(j+1)}]$ and $L_{1/\delta} := [\mu^2 \gamma^{-1/\delta}, \infty) = [\mu^2 R, \infty)$.

Simplifying assumptions. Recall the notion of well-hashedness (Definition 5.3.4). The crucial property of well-hashed elements is that if $i \in [n]$ is well-hashed, then an invocation of LocateSignal locates it with probability at least $1 - 1/\operatorname{poly}(R)$. This property is proved in Lemma 5.10.2. In this section we make the following simplifying assumption: we assume that each well-hashed element $i \in [n]$ is estimated with zero error and removed from the signal. The elements that are not well-hashed, on the other hand, we assume simply remain in the system untouched. Let H denote the set of well-hashed elements (which is close to the list of locations output by LocateSignal). In this section, therefore, we analyze the recursion $x \to x - x_H$.

For each $x_i \in L_j$ and each $t \geq 1$ let $\mathbf{1}_{i,t}$ denote the indicator variable equal to 1 if x_i survived up to the t-th round of the process and 0 otherwise. For each $j \in [1 : 1/\delta]$ and $t \geq 1$ let

$$m_j(t) = \frac{1}{k} \sum_{j' \geq j} \sum_{i \in [k] : (x_i)^2 \in L_{j'}} \mathbf{1}_{i,t}.$$

Recall that by Definition 5.3.4 an element $i \in [n]$ is well-hashed for a particular choice of $\sigma, b \in [n], \sigma$ odd, and filter G if over uniformly random $a \in [n]$,

$$\mathop{\mathbb{E}}_{a}[|G^{-1}_{o_i(i)} \omega^{-a\sigma i} u_{h(i)} - x_i|^2] \leq \gamma^{1/2} |x_i|^2.$$

Lemma 5.6.1. *Let $\sigma, b \in [n]$ be chosen uniformly at random, σ odd. Let $i \in [n]$ denote an index such that $|x_i|^2 \in L_j$. Then the probability that i is not well-hashed at time t is at most of order*

$$\alpha \left(\gamma^{j-1/2} + \sum_{j' < j-1} \gamma^{j-j'-3/2} m_{j'}(t) \right) + \alpha m_{j-1}(t),$$

where the number of buckets B satisfies $B \geq k/\alpha$.

Proof. Let $x^{(h)}$ denote all elements of x in levels L_{j-1} and above. Denote the set of such elements by S^+. Let $x^{(t)}$ denote all elements of x in $L_{j'}, j' < j - 1$. Since $|x_i|^2 \in L_j$, we have $|x_i|^2 \geq \gamma^{-j} \mu^2$.

Define C to be the indices that "collide with" i as in Lemma 5.3.3. We have that

$$\Pr[C \cap S^+ \neq \{\}] \lesssim |S^+|/B = \alpha m_{j-1}(t).$$

Condition on the event that $C \cap S^+ = \{\}$; since this happens with more than $1/2$ probability, the conditioning only loses a constant factor in expectations and we may neglect this influence. We

have by Lemma 5.3.3 that

$$\mathop{\mathbb{E}}_{\sigma,b,a}[|G^{-1}_{o_i(i)}\omega^{-a\sigma i}u_{h(i)} - x_i|^2] \lesssim \|x\|_2^2/(R^2B) + \mathop{\mathbb{E}}_{\sigma,b}[\|x_C\|_2^2] + \frac{1}{R^*n^{11}}\|x^*\|_2^2. \tag{5.13}$$

$$\tag{5.14}$$

Recall that by the definition of $\mu^2 = \xi^2/k$,

$$\|x\|_2^2/(R^2B) + \frac{1}{R^*n^{11}}\|x^*\|_2^2 \le \|x\|_2^2/(RB) + \frac{1}{BR^*n^{10}}\|x^*\|_2^2 \le \alpha\mu^2.$$

Furthermore, recall that by Lemma 5.3.3, (1) any given element belongs to C with probability $O(1/B)$. Since the energy of an element in $L_{j'}$ is bounded above by $\gamma^{-(j'+1)}\mu^2$ by definition of $L_{j'}$, we get that

$$\mathop{\mathbb{E}}_{\sigma,b}[\|x_C\|_2^2 | C \cap S^+ = \{\}] \le \alpha\mu^2 \sum_{j'<j-1} \gamma^{-(j'+1)} m_{j'}(t).$$

Putting these two estimates together, we get that the rhs of (5.13) is bounded by

$$\alpha\mu^2 + \alpha\mu^2 \sum_{j'<j-1} \gamma^{-(j'+1)} m_{j'}(t),$$

Therefore, conditioned on $C \cap S^+ = \{\}$, we have

$$\begin{aligned}
\Pr[i \text{ not well-hashed}] &= \Pr_{\sigma,b}[\mathop{\mathbb{E}}_{a}[|G^{-1}_{o_i(i)}\omega^{-a\sigma i}u_{h(i)} - x_i|^2] \ge \gamma^{1/2}|x_i|^2] \\
&\le \frac{\mathbb{E}_{\sigma,b,a}[|G^{-1}_{o_i(i)}\omega^{-a\sigma i}u_{h(i)} - x_i|^2]}{\gamma^{1/2}|x_i|^2} \\
&\lesssim \frac{1}{\gamma^{1/2-j}\mu^2}\alpha\mu^2\left(1 + \mu^2 \sum_{j'<j-1} \gamma^{-(j'+1)} m_{j'}(t)\right) \\
&= \alpha\left(\gamma^{j-1/2} + \sum_{j'<j-1} \gamma^{j-j'-3/2} m_{j'}(t)\right).
\end{aligned}$$

Adding the $\alpha m_{j-1}(1)$ chance that $C \cap S^+ \neq \{\}$ in a union bound gives the result. □

Let S_t denote the state of the system at time t. By Lemma 5.6.1 at each time step t we have

$$\begin{aligned}
\mathbb{E}[m_1(t+1)|S_t] &\le \alpha m_1(t) \cdot (m_0(t)) \\
\mathbb{E}[m_2(t+1)|S_t] &\le \alpha m_2(t) \cdot (\gamma^{1/2} m_0(t) + m_1(t) + R^{-20}) \\
\mathbb{E}[m_3(t+1)|S_t] &\le \alpha m_3(t) \cdot (\gamma^{3/2} m_0(t) + \gamma^{1/2} m_1(t) + m_2(t) + R^{-20}) \\
&\vdots \\
\mathbb{E}[m_j(t+1)|S_t] &\le \alpha m_j(t) \cdot (\gamma^{j-3/2} m_0(t) + \ldots + \gamma^{1/2} m_{j-2}(t) + m_{j-1}(t) + R^{-20}).
\end{aligned} \tag{5.15}$$

Note that Lemma 5.6.1 in fact yields the bound without the additive term of R^{-20}. We analyze the weaker recurrence (5.15) in what follows, since the additional term of R^{-20} will be useful later

in section 5.7 for handling location errors. Lemma 5.6.1 does not provide any guarantees on the evolution of $m_0(t)$. It will be convenient to assume that $m_0(t)$ is chosen arbitrarily from the range $[0, C]$ for a constant $C > 0$ and all $t \geq 1$. Note that the contribution of μ^2 to the rhs in Lemma 5.6.1 disappeared since it is dominated by the contribution of $m_0(t)$.

In what follows we first analyze a related deterministic process, and then show that the randomized process closely follows its deterministic version with high probability.

5.6.2 Deterministic process

Let $m_j(1) \in [0, C]$ for a constant $C > 0$, and let $m_0^{det}(t) \in [0, C]$ be chosen arbitrarily for every t. Further, let for each $t \geq 1$ and $j \in [1 : 1/\delta]$

$$\begin{aligned}
m_1^{det}(t+1) &= \alpha m_1^{det}(t) \cdot (m_0^{det}(t)) \\
m_2^{det}(t+1) &= \alpha m_2^{det}(t) \cdot (\gamma^{1/2} m_0^{det}(t) + m_1^{det}(t) + R^{-20}) \\
m_3^{det}(t+1) &= \alpha m_3^{det}(t) \cdot (\gamma^{3/2} m_0^{det}(t) + \gamma^{1/2} m_1^{det}(t) + m_2^{det}(t) + R^{-20}) \\
&\vdots \\
m_j^{det}(t+1) &= \alpha m_j^{det}(t) \cdot (\gamma^{j-1/2} m_0^{det}(t) + \ldots + \gamma^{1/2} m_{j-2}^{det}(t) + m_{j-1}^{det}(t) + R^{-20}).
\end{aligned} \tag{5.16}$$

We now analyze the evolution of solutions to (5.16):

Lemma 5.6.2. *For each $j = 1, \ldots, 1/\delta$ and $t \leq j$ one has either $m_j^{det}(t) \leq 2^{-2^t}$ or $m_{j-1}^{det}(t-1) = O(\gamma^{1/2})$.*

The same conclusion holds if the equations for $m_j^{det}(t)$ are modified to include a R^{-20} additive term to obtain

$$m_j^{det}(t+1) = \alpha m_j^{det}(t) \cdot (\gamma^{j-1/2} m_0^{det}(t) + \ldots + \gamma^{1/2} m_{j-2}^{det}(t) + m_{j-1}^{det}(t) + R^{-20}).$$

for $j = 1, \ldots, 1/\delta$.

Proof. Induction on j and t.

Base: $j = 1$**,** $t = 1$ Trivial by appropriate choice of α.

Inductive step:(j, t) Suppose that $m_{j'}^{det}(t') \leq 2^{-2^{t'}}$ for all $j' < j$ and $t' \leq j'$. Then we have

$$\begin{aligned}
m_j^{det}(t+1) &\leq \alpha m_j^{det}(t) \cdot (\gamma^{j-1/2} m_0^{det}(t) + \ldots + \gamma^{1/2} m_{j-2}^{det}(t) + m_{j-1}^{det}(t)) \\
&\leq \alpha m_j^{det}(t) \cdot (O(\gamma^{1/2}) + m_{j-1}^{det}(t) + R^{-20}) \\
&\leq \alpha m_j^{det}(t) \cdot (O(\gamma^{1/2}) + m_{j-1}^{det}(t)),
\end{aligned}$$

where we used the fact that $R^{-20} = O(\gamma^{1/2})$.

Thus, if t is the first index such that $m_{j-1}^{det}(t) = O(\gamma^{1/2})$, we are done since $m_{j-1}^{det}(t)$ is non-increasing in t; Otherwise by the inductive hypothesis $m_j^{det}(t) \leq 2^{-2^t}$, so

$$\begin{aligned}
m_j^{det}(t+1) &\leq \alpha m_j^{det}(t) \cdot (\gamma^{j-1/2} m_0^{det}(t) + \ldots + \gamma^{1/2} m_{j-2}^{det}(t) + m_{j-1}^{det}(t)) \\
&\leq \alpha m_j^{det}(t) \cdot (O(\gamma^{1/2}) + m_{j-1}^{det}(t)) \\
&\leq m_j^{det}(t) \cdot 2^{-2^t} \leq 2^{-2^{t+1}}
\end{aligned}$$

as long as α is smaller than an appropriate constant.

□

Thus, we obtain

Lemma 5.6.3. *One has for all $j \geq 1/(4\delta)$ and any $t \geq c \log\log R$*

$$m_j^{det}(t) \leq R^{-10},$$

where $c > 0$ is a sufficiently large constant.

Proof. We use Lemma 5.6.2. First note that for $t \geq 1/(4\delta) \geq 10\log_2\log_2 R$ one has $2^{-2^t} < 2^{-2^{10\log_2\log_2 R}} < 2^{-\log_2^{10} R} < R^{-10}$. Thus, if the first case in Lemma 5.6.2 holds for $m_j(t)$, $j = t = (1/(2\delta))$, we are done. Otherwise if the second case holds, we have

$$\begin{aligned} m_j^{det}(t+1) &\leq \alpha m_j^{det}(t) \cdot (\gamma^{j-1/2} m_0^{det}(t) + \ldots + \gamma^{1/2} m_{j-2}^{det}(t) + m_{j-1}^{det}(t)) \\ &\leq \alpha m_j^{det}(t) \cdot (O(\gamma^{1/2}) + m_{j-1}^{det}(t)) = \alpha m_j^{det}(t) \cdot (O(\gamma^{1/2}) + m_{j-1}^{det}(t)), \end{aligned}$$

and then $m_j^{det}(t+t') = \gamma^{O(t')} = R^{O(t'/\log_2\log_2 R)}$, and hence $m_j^{det}(t+t') \leq R^{-10}$ for $t' = O(\log_2\log_2 R)$. □

We have proved

Lemma 5.6.4. *Let $\gamma = R^{-\delta}$ for some parameters $R > 1$ and $\delta = \Theta(1/\log\log R)$. Let $m_\ell(t) \in [0, C]$ be defined for some constant C, integer $\ell \in [0, 1/\delta - 1]$, and integer $t > 0$. Suppose it satisfies*

$$m_{\ell+1}(t+1) \leq \alpha m_{\ell+1}(t) \left(m_\ell(t) + \sum_{i=1}^{\ell} \gamma^{i-1/2} m_{\ell-i}(t) + R^{-20} \right) \text{ for } \ell \geq 0.$$

for some sufficiently small constant α. Then there exists a universal constant c such that for all $t > c\log\log R$ and $\ell \geq 1/(4\delta)$,

$$m_\ell(t) \leq R^{-10}.$$

5.6.3 Bound in expectation

In this section we show similar convergence if the decay is only in expectation, and using a continuous version of the recurrence.

For all $\eta \geq 0$, define the function $f_\eta : \mathbb{C}^n \to [0, \infty)$ by

$$f_\eta(x) = \frac{1}{k} |\{i : |x^i|^2 \geq \eta\}|$$

to be roughly the "fraction" of heavy hitters that remain above η.

Lemma 5.6.5. *Let k, R, μ^2 be arbitrary with $\delta = \Theta(\log\log R)$ and $\gamma = R^\delta$. Consider a recursion $x \to x'$ of vectors $x \in \mathbb{C}^n$ that is repeated $N = \Theta(\log\log R)$ times as $x^0 \to x^1 \to \cdots \to x^N$, and for all $\ell \geq 0$ and all inputs x satisfies*

$$\mathbb{E}[f_\eta(x')] \lesssim \alpha f_\eta(x) \left(R^{-20} + \frac{\mu^2}{\gamma\eta} + \frac{1}{\gamma\eta} \int_0^{\gamma\eta} f_t(x) dt \right) \tag{5.17}$$

for some sufficiently small parameter α. Suppose that $\|x^0\|_2^2 \lesssim Rk\mu^2$ and we know for all $i \in [0, N]$ that $f_0(x^i) \lesssim 1$. Then

$$\|x^N\|_2^2 \lesssim \sqrt{R}k\mu^2$$

with probability $1 - O(\alpha N^2)$. Furthermore, with the same probability we also have for all $i \leq N$ that

$$\|x^i\|_2^2 \lesssim Rk\mu^2$$
$$f_{\mu^2/\gamma}(x^i) \lesssim 1/4^i.$$

Proof. For simplicity of notation, we will prove the result about x^{N+1} rather than x^N; adjusting N gives the lemma statement.

The only properties of f that we use are (5.17), $f_a(x) \geq f_b(x)$ for $a \leq b$, and that

$$\|x\|_2^2 = k \int_0^\infty f_\eta(x) d\eta.$$

The desired claims are made more difficult by increasing the $f_\eta(x)$. Since we know that $f_\eta(x) \leq f_0(x) \leq C$ for some constant C, we may set

$$f_\eta(x) = C \quad \text{for } \eta < \mu^2/\gamma$$

for all x without loss of generality.

Then the μ^2 term in (5.17) may be absorbed by the integral, giving for each $x \to x'$ that:

$$\text{for any } \eta \geq \mu^2/\gamma, \qquad \mathbb{E}[f_\eta(x')] \lesssim \alpha f_\eta(x) \left(R^{-20} + \frac{1}{\gamma\eta} \int_0^{\gamma\eta} f_t(x) dt \right) \tag{5.18}$$
$$\lesssim \alpha f_\eta(x) \tag{5.19}$$

where the last step uses that $f_t(x) \leq C \lesssim 1$.

For $i \geq 1$ we have

$$\mathbb{E}[f_{\mu^2/\gamma}(x^i)] \leq (O(\alpha))^i f_{\mu^2/\gamma}(x^0) \lesssim \alpha/4^i.$$

for sufficiently small α, so by Markov's inequality and a union bound, with $1 - O(\alpha N)$ probability we have $f_{\mu^2/\gamma}(x^i) \leq 1/4^i$ for all $i \leq N + 1$. This gives the last claim in the lemma statement.

Part 1:. We know prove that $\|x^i\|_2^2 \lesssim Rk\mu^2$ for all i. We have that

$$\frac{1}{k}\|x'\|_2^2 = \int_0^\infty f_\eta(x') d\eta \lesssim \mu^2/\gamma + \int_{\mu^2/\gamma}^\infty f_\eta(x') d\eta$$

and by (5.19),

$$\begin{aligned}\mathbb{E}[\int_{\mu^2/\gamma}^\infty f_\eta(x') d\eta] &= \int_{\mu^2/\gamma}^\infty \mathbb{E}[f_\eta(x')] d\eta \\ &\lesssim \alpha \int_{\mu^2/\gamma}^\infty f_\eta(x) d\eta \\ &\leq \alpha \|x\|_2^2/k.\end{aligned}$$

Hence with probability $1 - O(\alpha N^2)$, in all N stages this is at most $\|x\|_2^2/(Nk)$ and we have

$$\frac{1}{k}\|x'\|_2^2 \lesssim \mu^2/\gamma + \|x\|_2^2/(Nk).$$

Hence for all $i \leq N$,

$$\|x^i\|_2^2 \lesssim k\mu^2/\gamma + \|x^0\|_2^2 \lesssim Rk\mu^2. \tag{5.20}$$

Part 2:. We now prove that

$$f_{R^{1/4}\mu^2}(x^N) \lesssim R^{-10} \tag{5.21}$$

with the desired probability.

Define the functions $m_\ell : \mathbb{C}^n \to [0, C]$ for integer ℓ by

$$\begin{aligned} m_0(x) &= f_0(x) \\ m_\ell(x) &= f_{\gamma^{-2\ell}\mu^2}(x) \qquad \text{for } \ell > 0 \end{aligned}$$

We will show that they satisfy the recurrence in Lemma 5.6.4 with γ^2 replacing γ. By (5.18), for $\ell \geq 1$ we have

$$\mathbb{E}[m_\ell(x)] \lesssim \alpha m_\ell(x)\left(R^{-20} + \frac{\gamma^{2\ell-1}}{\mu^2}\int_0^{\gamma^{1-2\ell}\mu^2} f_t(x)dt\right)$$

and we know that

$$\begin{aligned} \int_0^{\gamma^{1-2\ell}\mu^2} f_t(x)dt &\leq C\mu^2/\gamma^2 + \sum_{i=1}^{\ell-2}\int_{\mu^2\gamma^{-2i}}^{\mu^2\gamma^{-2i-2}} f_t(x)dt + \int_{\mu^2\gamma^{2-2\ell}}^{\mu^2\gamma^{1-2\ell}} f_t(x)dt \\ &\leq C\mu^2/\gamma^2 + \sum_{i=1}^{\ell-2}\mu^2\gamma^{-2i-2}m_i(x) + \mu^2\gamma^{1-2\ell}m_{\ell-1}(x) \\ &= \mu^2\gamma^{1-2\ell}m_{\ell-1}(x) + \sum_{i=0}^{\ell-2}\gamma^{-2i-2}\mu^2 m_i(x) \end{aligned}$$

so

$$\begin{aligned} \mathbb{E}[m_\ell(x)] &\lesssim \alpha m_\ell(x)(R^{-20} + m_{\ell-1}(x) + \sum_{i=0}^{\ell-2}\gamma^{2\ell-2i-3}m_i(x)) \\ &= \alpha m_\ell(x)(R^{-20} + m_{\ell-1}(x) + \sum_{i=2}^{\ell}(\gamma^2)^{i-3/2}m_{\ell-i}(x)) \end{aligned}$$

for $\ell \geq 1$. But for the expectation, this is precisely the recurrence of Lemma 5.6.4 after substituting γ^2 for γ. Since Lemma 5.6.4 only considers $N/\delta \lesssim N^2$ different ℓ and x^i, by Markov's inequality the recurrence will hold in all instances for a sufficiently small constant α' with probability $1 - O(\alpha N^2)$. Assume this happens. Since Lemma 5.6.4 is applied with $\gamma \to \gamma^2$, $\delta \to \delta/2$, this implies

$$m_{1/8\delta}(x^N) \lesssim R^{-10}.$$

This gives (5.21), because

$$f_{R^{1/4}\mu^2}(x^N) = f_{\gamma^{2\cdot 1/(8\delta)}\mu^2}(x^N) = m_{1/8\delta}(x^N) \lesssim R^{-10}.$$

Part 3:. We now prove that $\|x^{N+1}\|_2^2 \lesssim \sqrt{R}k\mu^2$ with $1 - O(\alpha)$ probability conditioned on the above.

We have

$$\frac{1}{k}\|x^{N+1}\|_2^2 = \int_0^\infty f_\eta(x^{N+1})d\eta \lesssim R^{1/4}\mu^2 + \int_{R^{1/4}\mu^2}^\infty f_\eta(x^{N+1})d\eta$$

Define V to be the latter term. We have by (5.18) and (5.21) that

$$\begin{aligned}
\mathbb{E}[V] &= \int_{R^{1/4}\mu^2}^\infty \mathbb{E}[f_\eta(x^{N+1})]d\eta \\
&\lesssim \int_{R^{1/4}\mu^2}^\infty \alpha f_\eta(x^N)(R^{-20} + \frac{1}{\gamma\eta}\int_0^{\gamma\eta} f_t(x^N)dt)d\eta \\
&= \int_{R^{1/4}\mu^2}^\infty \alpha f_\eta(x^N)(R^{-20} + \frac{1}{\gamma\eta}(\int_0^{R^{1/4}\mu^2} f_t(x^N)dt + \int_{R^{1/4}\mu^2}^{\gamma\eta} f_t(x^N)dt))d\eta \\
&\lesssim \int_{R^{1/4}\mu^2}^\infty \alpha f_\eta(x^N)(R^{-20} + \frac{1}{\gamma\eta}(R^{1/4}\mu^2 + \gamma\eta R^{-10}))d\eta \\
&\lesssim \alpha R^{-10}\|x^N\|_2^2/k + \int_{R^{1/4}\mu^2}^\infty \alpha f_\eta(x^N)\frac{R^{1/4}\mu^2}{\gamma\eta}d\eta
\end{aligned}$$

In the latter term, for fixed $\int_0^\infty f_\eta(x^N)d\eta = \|x^N\|_2^2/k$ this is maximized when the mass of f_η is pushed towards smaller η. Hence

$$\begin{aligned}
\int_{R^{1/4}\mu^2}^\infty \alpha f_\eta(x^N)\frac{R^{1/4}\mu^2}{\gamma\eta}d\eta &\le \int_{R^{1/4}\mu^2}^{R^{1/4}\mu^2 + \|x^N\|_2^2/k} \alpha \cdot C \cdot \frac{R^{1/4}\mu^2}{\gamma\eta}d\eta \\
&= C\alpha R^{1/4}\mu^2\gamma^{-1}\log(1 + \frac{\|x^N\|_2^2/k}{R^{1/4}\mu^2}) \\
&\le C\alpha R^{1/4+\delta}\mu^2\log R \\
&\lesssim \alpha\sqrt{R}\mu^2.
\end{aligned}$$

by (5.20). But then $\mathbb{E}[V] \lesssim \alpha\sqrt{R}\mu^2$, so with $1 - O(\alpha)$ probability $V \lesssim \sqrt{R}\mu^2$ and

$$\|x^{N+1}\|_2^2 \lesssim \sqrt{R}k\mu^2$$

as desired. □

5.7 Reducing SNR: general analysis

Recall our goal:

Lemma 5.7.11. *For $x^*, \chi \in \mathbb{C}^n$ define $x = x^* - \chi$ and*

$$\xi^2 = \mathrm{Err}_k^2(x) + \frac{\|x\|_2^2}{R} + \frac{\|x^*\|_2^2}{R^* n^{10}}.$$

Suppose $\|\chi\|_0 \le k$. *Then* ReduceSNR$(\widehat{x}^*, \chi, k, R, p)$ *returns* χ' *such that*

$$\|x - \chi'\|_2^2 \lesssim \sqrt{R}\xi^2$$

with probability $1-p$, *using* $O(\frac{1}{p^2} k \log(Rn/k)(\log\log(Rn/k))^c)$ *measurements and a* $\log(Rn)$ *factor more time.*

We will show that each inner loop of ReduceSNR satisfies some nice properties (similar to those of Lemma 5.6.4) that cause the residual to reduce from signal-to-noise ratio R to $\sqrt{R}$. As in ReduceSNR and 5.2.2, we define

- $B = k/\alpha$ to be the size of each hash table, where $\alpha = O(1/\log\log^c n)$
- $T = O(1)$ to be the number of hashings done in each EstimateValues
- $\xi^2 = \mathrm{Err}_k^2(x^* - \chi) + \frac{\|x^* - \chi\|_2^2}{R} + \|x^*\|_2^2/(R^* n^{10})$.
- $\mu^2 = \xi^2/k \ge \frac{1}{k}(\mathrm{Err}_k^2(x^* - \chi) + \|x^* - \chi\|_2^2/R)$ to be the "noise level."
- $\delta = 1/(40\log_2\log_2 R)$.
- $\gamma = R^{-\delta}$ to be the "contrast" our LocateSignal requires.

In round t of the inner loop, we define the following variables:

- $\chi^{(t)} \in \mathbb{C}^n$: the estimate of x^* recovered so far.
- $x = x^* - \chi^{(t)} \in \mathbb{C}^n$: The vector we want to recover.
- $k' = k_t = \Theta(k4^{-t})$: The number of coordinates to update this round.
- $L \subset [n]$: Indices located by LocateSignal (with $|L| \le B$)
- $\tilde{x}^{(t)} \in \mathbb{C}^n$ for $t \in T$: The estimations of x in each inner loop of EstimateValues.
- $\tilde{x} \in \mathbb{C}^n$: $\tilde{x}_i = \mathrm{median}_t\, \tilde{x}_i^{(t)}$ is the estimation of x that would result from EstimateValues (although the algorithm only computes $\tilde{x}_L$).
- $S \subseteq L$ contains the largest $k'/4$ coordinates of x_L.
- $L' \subseteq L$: The indices of the largest k' coordinates of $\tilde{x}_L$

In the algorithm ReduceSNR, the inner loop replaces x with $x - \tilde{x}_{L'}$. This is then repeated $N = O(\log\log R)$ times. We say that this is a "recurrence" $x \to x - \tilde{x}_{L'}$, and will prove that the final result x^N has $\|x^N\|_2^2 \lesssim \sqrt{R}\xi^2$.

We will split our analysis of ReduceSNR into stages, where the earlier stages analyze the algorithm with the inner loop giving a simpler recurrence. In subsequent sections, we will consider the following different recurrences:

1. $x \to x - x_S$
2. $x \to x - \tilde{x}_S$
3. $x \to x - x_{L'}$
4. $x \to x - \tilde{x}_{L'}$

and show that each would reduce the noise level after $O(\log\log R)$ repetitions.

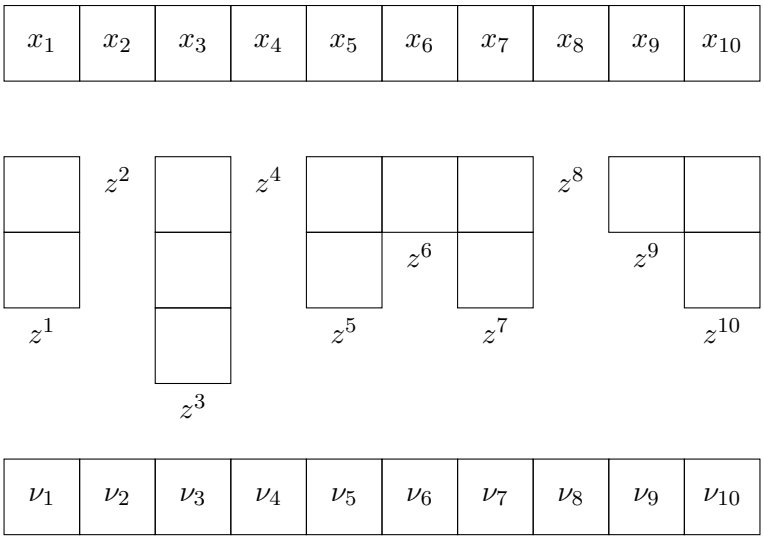


Figure 5-1: A representation of a splitting of x. In each column, $|x_i|^2 \leq \|z^i\|_2^2 + \nu_i^2$.

5.7.1 Splittings and admissibility

We introduce the notion of *splittings*. These allow us to show that the error introduced by the estimation phase is of the same order as the error from coordinates that are not well hashed. Since that level of error is tolerable according to Section 5.6, we get that the total error is also tolerable.

Definition 5.7.1. *For $x \in \mathbb{C}^n$, (z, ν) is a* splitting *of x if, for all $i \in [n]$, z^i is a vector and $\nu_i \in \mathbb{R}$ with*

$$\|z^i\|_2^2 + \nu_i^2 \geq |x_i|^2.$$

Analogously to the previous section, we can measure the number of elements of z above any value $\eta \geq 0$:

$$f_\eta(z) = \frac{1}{k}|\{(i,j) : |z_j^i|^2 \geq \eta\}|$$

We will want to deal with "nice" splittings that satisfy additional properties, as described below.

Definition 5.7.2. *We say (z, ν) is a* concise *splitting of x if (z, ν) is a splitting of x and also*

$$\begin{aligned} \|z^i\|_2^2 + \nu_i^2 &= |x_i|^2 \text{ for all } i \\ f_0(z) &\lesssim 1 \\ \|\nu\|_2^2 &\lesssim k\mu^2 \\ f_{\mu^2/\gamma}(z) &\leq k'/(4k) \\ \sum_i \|z^i\|_2^2 &\lesssim R^2 k\mu^2 \end{aligned}$$

For various recurrences $x \to x'$ and any concise splitting (z, ν) of x, we would like to find methods of assigning splittings (z', ν') to x' that satisfy some nice properties. In particular, we will

want decay as in Lemma 5.6.5:

$$\mathbb{E}[f_\eta(z')] \lesssim \alpha f_\eta(z)\left(R^{-20} + \frac{\mu^2}{\gamma\eta} + \frac{1}{\gamma\eta}\int_0^{\gamma\eta} f_t(z)dt\right) \tag{D}$$

and we will want relatively slow growth in the size of the splitting:

$$\begin{aligned}\mathbb{E}[\max\left(0, (\sum_i \|(z')^i\|_0) - (\sum_i \|z^i\|_0)\right)] &\lesssim \sqrt{\alpha k' k} \\ \mathbb{E}[\max\left(0, \sum_i (\nu_i')^2 - \sum_i \nu_i^2.\right)] &\lesssim \sqrt{\alpha k' k}\mu^2\end{aligned} \tag{G}$$

For some recurrences, we will get the stronger condition:

$$\begin{aligned}\mathbb{E}[\sum_i \|(z')^i\|_0] &\lesssim \sqrt{\alpha k' k} \\ \mathbb{E}[\sum_i (\nu_i')^2] &\lesssim \sqrt{\alpha k' k}\mu^2.\end{aligned} \tag{G'}$$

Definition 5.7.3. *A recurrence $x \to x'$ is*

- admissible *if for any concise splitting (z, ν) of x, we can assign splittings (z', ν') to x' that satisfy* (D) *and* (G).
- fully admissible *if (z', ν') can also satisfy* (G').

Note that we require the input (z, ν) to be a concise splitting, but the result (z', ν') may not be concise.

Analyzing repeated applications of (D) gives the following lemma:

Lemma 5.7.4. *Suppose $x \to x'$ is admissible. Then consider $r = \log\log R$ repetitions of the recurrence, i.e. $x^0 \to x^1 \to \cdots \to x^r$, where the jth round is run with $k' = k_j = k/4^j$ and x^0 has a concise splitting and $\|x^0\|_2^2 \lesssim Rk\mu^2$. Then for any parameter p, as long as $\alpha = \Theta(p^2/(\log\log R)^2)$ is sufficiently small, we get*

$$\|x^r\|_2^2 \lesssim \sqrt{R}k\mu^2$$

with $1 - p$ probability.

Proof. Because $x \to x'$ is admissible, there is a corresponding recurrence

$$(z, \nu) \to (z', \nu')$$

of splittings of x and x' that satisfies (D) and (G) whenever (z, ν) is concise. For this analysis, we will suppose that it satisfies (D) and (G) unconditionally, and bound the probability that (z, ν) is ever not concise.

The conditions (D) and (G) are only made more true by decreasing z' and ν' in absolute value, so we may also assume the (z', ν') resulting from the recurrence satisfies the first requirement of concise splittings,

$$\|(z')^i\|_2^2 + (\nu_i')^2 = |x_i|^2.$$

At each stage, with probability $1 - O(\sqrt{\alpha})$ we have by Markov's inequality and a union bound that

$$\begin{aligned} \max\left(0, (\sum_i \|(z')^i\|_0) - (\sum_i \|z^i\|_0)\right) &\leq \sqrt{k'k} \\ \max\left(0, \sum_i (\nu'_i)^2 - \sum_i \nu_i^2.\right) &\leq \sqrt{k'k}\mu^2 \end{aligned} \tag{5.22}$$

Hence with $1 - O((\log\log R)\sqrt{\alpha}) > 1 - p/2$ probability, equation set (5.22) holds for all r stages. Assume this happens.

Then at any stage j, the resulting (z', ν') has $f_0(z') = \frac{1}{k}\sum_i \|(z')^i\|_0 \leq \frac{1}{k}(k + \sum_{t\leq j}\sqrt{k_t k}) \leq 3$ and $\|\nu'\|_2^2 \leq k\mu^2 + \sum_{t\leq j}\sqrt{k_t k}\mu^2 \leq 3k\mu^2$. Therefore the second and third requirements for conciseness are satisfied in every stage.

Now, we apply Lemma 5.6.5 to observe that with $1 - O(\alpha N^2) > 1 - p/2$ probability, the remaining two requirements for conciseness are satisfied in all stages and the final splitting (z, ν) of x^r satisfies

$$\sum_i \|z^i\|_2^2 \lesssim \sqrt{R}k\mu^2.$$

Therefore with probability $1 - p$, our supposition of conciseness is correct in all stages and the final x^r satisfies

$$\|x^r\|_2^2 \leq \sum_i \|z^i\|_2^2 + \nu_i^2 \lesssim (\sqrt{R} + 3)k\mu^2 \lesssim \sqrt{R}k\mu^2$$

which is our result. □

Given that admissibility is a sufficient condition, we construct tools to prove that recurrences are admissible.

Lemma 5.7.5. *If $x \to x'$ is admissible, $x \to x^\#$ is fully admissible, and $x'_{\mathrm{supp}(x^\#)}$ is identically zero then $x \to x' + x^\#$ is admissible.*

Proof. For any splitting (z, ν) of x, we have splittings (z', ν') and $(z^\#, \nu^\#)$ of x' and $x^\#$. We would like to combine them for a splitting of $x' + x^\#$.

Let $A = \mathrm{supp}(x^\#)$. For $i \notin A$, we use $((z')^i, \nu'_i)$. For $i \in A$, we use $((z^\#)^i, \nu_i^\#)$. This is a valid splitting of $x' + x^\#$.

By linearity it satisfies (D) and (G) with a minor loss in the constants. □

Lemma 5.7.6. *If $x \to x'$ and $x \to x^\#$ are both fully admissible, then $x \to x' + x^\#$ is fully admissible.*

Proof. For any splitting (z, ν) of x, we have splittings (z', ν') and $(z^\#, \nu^\#)$ of x' and $x^\#$. We would like to combine them for a splitting of $x' + x^\#$.

For each coordinate i, let $u = (z')^i$ and $v = (z^\#)^i$, and $a = |x'_i + x_i^\#|$. We will find a vector w and scalar g such that

$$\begin{aligned} \|w\|_2^2 + g^2 &\geq a^2 \\ \|w\|_0 &\lesssim \|u\|_0 + \|v\|_0 \\ g^2 &\lesssim (\nu'_i)^2 + (\nu_i^\#)^2 \\ |\{i \mid w_i \geq \eta\}| &\lesssim |\{i \mid u_i \geq \eta\}| + |\{i \mid v_i \geq \eta\}|. \end{aligned}$$

for all thresholds η. This will only lose a constant factor in (G') and (D). In particular, we set w to be the concatenation of two copies of u and two copies of v, and $g^2 = 2(\nu_i')^2 + 2(\nu_i^\#)^2$. Then

$$\|w\|_2^2 + g^2 = 2(\|u\|_2^2 + (\nu_i')^2) + 2(\|v\|_2^2 + (\nu_i^\#)^2) \geq 2|x_i'|^2 + 2|x_i^\#|^2 \geq a^2,$$

so (w, g) is a valid splitting for each coordinate, giving us that $x' + x^\#$ is fully admissible. □

5.7.2 Recurrence $x \to x - x_S$

Lemma 5.7.7. *Let S contain the largest $k'/4$ coordinates of L. Then $x \to x - x_S$ is admissible.*

Proof. Consider any concise splitting (z, ν) of x. Let $S' = \{i \in L : \|z^i\|_\infty^2 \geq \mu^2\gamma^{-1}\}$.

We have $|S'| \leq k f_{\mu^2/\gamma}(z) \leq k'/4$ because (z, ν) is concise. Since $x - x_{S'}$ can be permuted to be coordinate-wise dominated by $x - x_S$, it suffices to split $x - x_{S'}$.

For $i \in S'$, we set $(z')^i = \{\}$ and $\nu_i' = 0$; for $i \notin S'$, we set $((z')^i, \nu_i') = (z^i, \nu_i)$. We must only show (D) holds, because (G) is trivial (the growth is zero). That is, we must show that if $|z_j^i|^2 \geq \eta$ then

$$\Pr[i \notin S'] \lesssim \alpha \left(R^{-20} + \frac{\mu^2}{\gamma\eta} + \frac{1}{\gamma\eta} \int_0^{\gamma\eta} f_t(x)dt \right) =: M. \tag{5.23}$$

Let M denote the right hand side of (5.23). For such an i, $|x_i|^2 \geq |z_j^i|^2 \geq \mu^2\gamma^{-1}$, and

$$\Pr[i \notin S'] = \Pr[i \notin L] \leq \Pr[i \text{ not well-hashed}] + \Pr[i \notin L | i \text{ well-hashed}]$$

Define $H = \{i : \|z^i\|_\infty^2 \geq \gamma\eta\}$. Then from Lemma 5.3.3 we get a subset $C \subset [n]$ and variable w so that i is well-hashed if

$$w^2 + \|x_C\|_2^2 \leq c\gamma^{1/2}|x_i|^2$$

for some constant c, which is implied by $w^2 + \|x_C\|_2^2 \leq \gamma\eta$. We have that

$$\Pr[H \cap C \neq \{\}] \lesssim |H|/B \leq k f_{\gamma\eta}(z)/B = \alpha f_{\gamma\eta}(z)$$

and that

$$\begin{aligned}\mathbb{E}[w^2 + \|x_{C\setminus H}\|_2^2] &\lesssim \frac{\|x\|_2^2}{R^2 B} + \frac{\|x^*\|_2^2}{R^* n^{11}} + \|x_{\overline{H}}\|_2^2/B \\ &\lesssim \alpha\mu^2 + \|x_{\overline{H}}\|_2^2/B\end{aligned}$$

by the definition of μ^2. We know that

$$\|x_{\overline{H}}\|_2^2 \leq \sum_{(i,j):|z_j^i|^2 \leq \gamma\eta} |z_j^i|^2 = k \int_0^{\gamma\eta} (f_t(z) - f_{\gamma\eta}(z))dt.$$

Therefore by Markov's inequality,

$$\begin{aligned}\Pr[i \text{ not well-hashed}] &\leq \Pr[C \cap H \neq \{\}] + \Pr[w^2 + \|x_{C\setminus H}\|_2^2 \geq \gamma\eta] \\ &\lesssim \alpha f_{\gamma\eta}(z) + \frac{1}{\gamma\eta}(\alpha\mu^2 + \alpha \int_0^{\gamma\eta} (f_t(z) - f_{\gamma\eta}(z))dt) \\ &= \frac{1}{\gamma\eta}(\alpha\mu^2 + \alpha \int_0^{\gamma\eta} f_t(z)dt) < M.\end{aligned}$$

Next, by Lemma 5.10.2, since we call LocateSignal with failure probability αR^{-20}, we have

$$\Pr[i \notin L | i \text{ well-hashed}] \lesssim \alpha R^{-20} < M.$$

giving $\Pr[i \notin S'] \lesssim M$ for each i, as desired. □

5.7.3 Recurrence $x \to x - \tilde{x}_S$

Lemma 5.7.8. *Let L be independent of the estimation phase with $|L| \le B$, and $A \subseteq L$ be possibly dependent on the estimation phase with $|A| \lesssim k'$. Then $x \to x_A - \tilde{x}_A$ is fully admissible.*

Proof. Let (z, ν) be a concise splitting of x. For $i \in L$, we have

$$|\tilde{x}_i - x_i|^2 = |\operatorname*{median}_t \tilde{x}_i^{(t)} - x_i|^2 \le 2 \operatorname*{median}_t |\tilde{x}_i^{(t)} - x_i|^2 \tag{5.24}$$

because we take the median in real and imaginary components separately. We have by $\tilde{x}_i^{(t)} = G_{o_i(i)}^{-1} \omega^{-a\sigma i} u_{h(i)}$ and Lemma 5.3.3 that

$$\mathop{\mathbb{E}}_a[|\tilde{x}_i^{(t)} - x_i|^2] \lesssim w_i^2 + \|x_{C_i^t}\|_2^2$$

for some C with $\Pr[j \in C] \lesssim 1/B$ for all j, and some w with

$$\mathbb{E}[w_i^2] \lesssim \frac{\|x\|_2^2}{R^2 B} + \|x^*\|_2^2/(R^* n^{11}) \lesssim \alpha\mu^2, \tag{5.25}$$

where the last step uses that $\|x\|_2^2 \lesssim R^2 k \mu^2$ because a concise splitting (z, ν) of x exists. Then

$$\mathop{\mathbb{E}}_a[|\tilde{x}_i^{(t)} - x_i|^2] \lesssim w_i^2 + \sum_{j \in C_i^t} \|z^j\|_2^2 + \nu_j^2.$$

Define

$$(y_i^t)^2 := w_i^2 + \sum_{j \in C_i^t} \|z^j\|_2^2 + \nu_j^2$$
$$\tau_i^t := \lceil 2|\tilde{x}_i^{(t)} - x_i|^2/(y_i^t)^2 \rceil$$

so

$$\mathop{\mathbb{E}}_a[\tau_i^t] \lesssim 1 \tag{5.26}$$

even after conditioning on the hash function (σ, b).

For any $t \in [T]$ and $i \in L$, let $U^{(t),i}$ be the concatenation of τ_i^t copies of z^j for each $j \in C_i^t$ and $\nu_i^{(t)} = \sqrt{\tau_i^t}((w_i^t)^2 + \sum_{j \in C_i^t} \nu_j^2)$. Then we have that

$$\|U^{(t),i}\|_2^2 + (\nu_i^{(t)})^2 \ge \tau_i^t (y_i^t)^2 \ge 2|\tilde{x}_i^{(t)} - x_i|^2$$

and so by (5.24), for at least $1 + \lfloor T/2 \rfloor$ different $t \in [T]$ we have

$$|\tilde{x}_i - x_i| \le \|U^{(t),i}\|_2^2 + (\nu_i^{(t)})^2. \tag{5.27}$$

For each $i \in A$, our $(\tilde{z}^i, \tilde{\nu}_i)$ will equal $(U^{(t^*),i}, \nu_i^{(t^*)})$ for a t^* satisfying (5.27) as well as

$$\begin{aligned} \|\tilde{z}^i\|_\infty &\le \operatorname*{quant}_t{}^{1/6} \|U^{(t),i}\|_\infty \\ \|\tilde{z}^i\|_0 &\le \operatorname*{quant}_t{}^{1/6} \|U^{(t),i}\|_0 \\ \tilde{\nu}_i^2 &\le \operatorname*{quant}_t{}^{1/6} (\nu_i^{(t)})^2 \end{aligned} \tag{5.28}$$

where $\operatorname{quant}^{1/6}$ is the "quantile" defined in Section 5.9.1. This is always possible, because the number of t excluded by these additional conditions is at most $3\lfloor T/6 \rfloor \le \lfloor T/2 \rfloor$. Choosing such a t^* for each i gives us a splitting $(\tilde{z}, \tilde{\nu})$ of $x_A - \tilde{x}_A$.

To show (D), for any $i \in L$ and threshold η define

$$\begin{aligned} m &= |\{(\ell, j) : |z_j^\ell| \ge \eta\}| \\ \tilde{m} &= |\{(\ell, j) : |\tilde{z}_j^\ell| \ge \eta\}| \\ m_t^i &= |\{j : U_j^{(t),i} \ge \eta\}| \end{aligned}$$

We bound $\mathbb{E}[m_{t^*}^i]$ using Lemma 5.9.3. Since $\Pr[j \in C_i^t] \lesssim 1/B$ and $\mathbb{E}[\tau_i^t] \lesssim 1$ after conditioning on (σ, b) and so fixing C_i^t, for fixed i and t we have

$$\begin{aligned} \mathbb{E}[m_t^i] &= \mathbb{E}[(|\{(\ell, j) : |z_j^\ell| \ge \eta, \ell \in C_i^t\}|)\tau_i^t] \\ &= \mathbb{E}[(|\{(\ell, j) : |z_j^\ell| \ge \eta, \ell \in C_i^t\}|)\, \mathbb{E}[\tau_i^t \mid (\sigma, b)]] \\ &\lesssim \mathbb{E}[|\{(\ell, j) : |z_j^\ell| \ge \eta, \ell \in C_i^t\}|] \\ &= \sum_{(\ell,j):|z_j^\ell| \ge \eta} \Pr[\ell \in C_i^t] \\ &\lesssim \sum_{(\ell,j):|z_j^\ell| \ge \eta} 1/B \\ &= m/B \end{aligned}$$

We also have $m_{t^*}^i = 0$ if $\operatorname{quant}^{1/6}{}_t\, m_t^i = 0$ and $m_{t^*}^i \le \sum_t m_t^i$ always; hence for each fixed index $i \in L$, by Lemma 5.9.3

$$\mathbb{E}[m_{t^*}^i] \lesssim (m/B)^{T/6}.$$

But then for $T \ge 12$ we have

$$\begin{aligned} \mathbb{E}[\tilde{m}] = \mathbb{E}[\sum_{i \in A} m_{t^*}^i] \le \sum_{i \in L} \mathbb{E}[m_{t^*}^i] &\lesssim B(m/B)^2 = m^2/B \\ \mathbb{E}[\tilde{m}/k] &\lesssim \alpha(m/k)^2 \end{aligned}$$

which says that

$$\mathbb{E}[f_\eta(z')] \lesssim \alpha(f_\eta(z))^2 \le \alpha f_\eta(\frac{1}{\gamma\eta} \int_0^{\gamma\eta} f_t(z) dt)$$

for each η, implying (D).

We now show (G'). For any nonnegative random variable $X_{i,t}$ (which will be either $\|U^{(t),i}\|_0$ or $(\nu_i^{(t)})^2$) and $Y_i \le \operatorname{quant}^{1/6}{}_t\, X_{i,t}$ (which will be $\|\tilde{z}^i\|_0$ or $\tilde{\nu}_i^2$), for sufficiently large constant T we

have by Lemma 5.9.4 with $\delta = 1/2$ that

$$\mathbb{E}[\max_{|A'|\lesssim k'} \sum_{i\in A'} Y_i] \lesssim \sqrt{k'B} \max_{i,t} \mathbb{E}[X_{i,t}]$$

Plugging in that, for each fixed i, by (5.26) and conciseness we have

$$\mathop{\mathbb{E}}_{\sigma,b,a}[\|U^{(t),i}\|_0] = \mathop{\mathbb{E}}_{\sigma,b,a}\left[\sum_{j\in C_i^t} \|z^j\|_0 \tau_i^t\right] \le \mathop{\mathbb{E}}_{\sigma,b}\left[\sum_{j\in C_i^t} \|z^j\|_0 \max_{\sigma,b} \mathbb{E}[\tau_i^t \mid (\sigma,b)]\right] \lesssim \mathop{\mathbb{E}}_{\sigma,b}\left[\sum_{j\in C_i^t} \|z^j\|_0\right] \lesssim k/B = \alpha$$

gives

$$\mathbb{E}[\max_{|A'|\lesssim k'} \sum_{i\in A'} \|\tilde{z}^i\|_0] \lesssim \sqrt{k'B}\alpha = \sqrt{\alpha k' k},$$

which is the first part of (G'). Similarly, plugging in

$$\mathbb{E}[(\nu_i^{(t)})^2] \lesssim (\mathop{\mathbb{E}}_{\sigma,b}[w_i^2] + \|\nu\|_2^2/B) \max_{\sigma,b} \mathbb{E}[\tau_i^t | (\sigma,b)] \lesssim \alpha\mu^2$$

gives

$$\mathbb{E}[\max_{|A'|\lesssim k'} \sum_{i\in A'} \tilde{\nu}_i^2] \lesssim \sqrt{k'B}\alpha\mu^2 = \sqrt{\alpha k' k}\mu^2,$$

which is the second part.

Therefore $(\tilde{z}, \tilde{\nu})$ is a splitting of $x_A - \tilde{x}_A$ that satisfies (D) and (G'), so $x \to x_A - \tilde{x}_A$ is fully admissible. □

5.7.4 Recurrence $x \to x - \tilde{x}_{L'}$

The following lemma has a similar proof structure to Lemma 5.9.1.

Lemma 5.7.9. *The recurrence $x \to x - x_{L'}$ is admissible.*

Proof. Recall the set S from Lemma 5.7.7, which has $|S| = k'/4 \le |L'|/4$ and for which $x - x_S$ is admissible. Let $A = L' \setminus S$ and $B = S \setminus L'$. We have $|A| \ge 4|B|$. Furthermore $\min_{i\in A} |\tilde{x}_i| \ge \max_{i\in B} |\tilde{x}_i|$ because EstimateValues chose A over B.

By Lemma 5.7.7, $x \to x - x_S$ is admissible. Let $y = (x - \tilde{x})_A + 2(\tilde{x} - x)_B$. Using Lemma 5.7.8, $x \to y$ is admissible. Hence for every splitting (z, ν) of x there are splittings (z^S, ν^S) of $x - x_S$ that satisfies (D) and (G) and (z^{AB}, ν^{AB}) of y that satisfies (D) and (G').

For $i \notin A \cup B$, we set $((z')^i, \nu_i') = ((z^S)^i, \nu_i^S)$. For $i \in A$, we set $((z')^i, \nu_i') = (\{\}, 0)$. Finally, we want to fill $((z')^i, \nu_i')$ for $i \in B$. To do this, pair up each $i \in B$ with a disjoint set P_i of four elements in A. We know that

$$\begin{aligned} 2|\tilde{x}_i| &\le \|\tilde{x}_{P_i}\|_2^2 \\ |2x_i + y_i| &\le \|x_{P_i} + y_{P_i}\|_2 \\ 2|x_i| &\le |y_i| + \|x_{P_i}\|_2 + \|y_{P_i}\|_2 \\ 4|x_i|^2 &\le 3(|y_i|^2 + \|x_{P_i}\|_2^2 + \|y_{P_i}\|_2^2) \\ |x_i|^2 &\le |y_i|^2 + \|x_{P_i}\|_2^2 + \|y_{P_i}\|_2^2 \end{aligned} \tag{5.29}$$

Set $(z')^i$ to the concatenation of $(z^{AB})^i$ and, for all $j \in P_i$, $(z^S)^j$ and $(z^{AB})^j$. Similarly, set $(\nu_i')^2 = (\nu_i^{AB})^2 + \sum_{j\in P_i} (\nu_j^S)^2 + (\nu_j^{AB})^2$. By (5.29), this makes (z', ν') be a valid splitting of $x - x_{L'}$.

Then each element of z^S, z^{AB}, ν^S, and ν^{AB} appears exactly once in (z', ν'); hence (z', ν') satisfies (D) and (G) so $x \to x - x_{L'}$ is admissible. □

Lemma 5.7.10. *The recurrence $x \to x - \tilde{x}_{L'}$ is admissible.*

Proof. We have

$$x - \tilde{x}_{L'} = (x - x_{L'}) + (x_{L'} - \tilde{x}_{L'}).$$

The first term is admissible by Lemma 5.7.9 and zero over L'. The second is fully admissible by Lemma 5.7.8, with support inside L'. Hence $x \to x - \tilde{x}_{L'}$ is admissible by Lemma 5.7.5. □

Lemma 5.7.11. *For $x^*, \chi \in \mathbb{C}^n$ define $x = x^* - \chi$ and*

$$\xi^2 = \mathrm{Err}_k^2(x) + \frac{\|x\|_2^2}{R} + \frac{\|x^*\|_2^2}{R^* n^{10}}.$$

Suppose $\|\chi\|_0 \le k$. Then ReduceSNR$(\widehat{x}^*, \chi, k, R, p)$ *returns χ' such that*

$$\|x - \chi'\|_2^2 \lesssim \sqrt{R}\xi^2$$

with probability $1 - p$, using $O(\frac{1}{p^2} k \log(Rn/k)(\log\log(Rn/k))^c)$ measurements and a $\log(Rn)$ factor more time.

Proof. The following is a concise splitting (z, ν) of $x = x^* - \chi^{(0)}$: place the largest k coordinates of x into z, and the rest into ν. By Lemma 5.7.10, $x^* - \chi^{(i)} \to x^* - \chi^{(i+1)}$ is admissible. Therefore, by Lemma 5.7.4, χ^N satisfies

$$\|x^* - \chi^{(N)}\|_2^2 \lesssim \sqrt{R}\xi^2.$$

as desired for correctness.

In each of $O(\log\log R)$ rounds we call LocateSignal and EstimateValues with $B = k/\alpha = O(k(\log\log R)^2/p^2)$ and failure probability R^{-20}. The sampling complexity of each LocateSignal is

$$O(B\log(Rn/B)\log\log R\log\log(n/B)\max(1, \log_R(1/p))) \lesssim \frac{1}{p^2} k\log(Rn/B)(\log\log(Rn/B))^4$$

by Lemma 5.10.2. The complexity of EstimateValues is bounded by $O(B\log R) = O(k\log R)$ since we perform $O(1)$ bucketings using a filter with contrast R. The overall sampling complexity over $O(\log\log R)$ rounds is hence bounded by

$$O\left(\frac{1}{p^2} k\log(Rn/B)(\log\log(Rn/B))^c\right)$$

for a constant $c > 0$. □

5.8 Final Result

Repeating Lemma 5.7.11 $\log\log R$ times and applying Lemma 5.5.1 gives the result:

Theorem 5.8.1. *Let $x \in \mathbb{C}^n$ satisfy $\|x\|_2^2 \le R\,\mathrm{Err}_k^2(x)$. Then* SparseFFT$(\widehat{x}, k, R, p)$ *returns a χ' such that*

$$\|x - \chi'\|_2^2 \le (1 + \epsilon)\,\mathrm{Err}_k^2(x) + \|x\|_2^2/(R^* n^{10})$$

with probability $1 - p$ *and using* $O(\frac{1}{p^2\epsilon} k \log(Rn/k)(\log\log(Rn/k))^c \log(1/\epsilon))$ *measurements and a* $\log(Rn)$ *factor more time.*

Proof. During this proof, we define $x^* := x$.

The algorithm performs $r = O(\log\log R)$ rounds of ReduceSNR. We may assume that all the calls succeed, as happens with $1 - p/2$ probability. We will show that at each stage i of the algorithm,

$$\|x^* - \chi^{(i)}\|_2^2 \le R_i \xi^2$$

for $\xi^2 = \mathrm{Err}_k^2(x^*) + \|x^*\|_2^2/(R^* n^{10})$. This is true by assumption at $i = 0$, and by Lemma 5.7.11, in each iteration ReduceSNR causes

$$\begin{aligned}\|x^* - \chi^{(i)} - \chi'\|_2^2 &\le c\sqrt{R_i}(\mathrm{Err}_{3k}^2(x^* - \chi^{(i)}) + \xi^2) \\ &\le c\sqrt{R_i}(\mathrm{Err}_k^2(x^*) + \xi^2) \\ &\le 2c\sqrt{R_i}\xi^2\end{aligned}$$

for some constant c. By Lemma 5.9.1,

$$\begin{aligned}\|x^* - \mathrm{Sparsify}(\chi^{(i)} + \chi', 2k)\|_2^2 &\le \mathrm{Err}_k^2(x^*) + 4\|x^* - \chi^{(i)} - \chi'\|_2^2 \\ &\le (1 + 8c\sqrt{R_i})\xi^2 \\ &\le R_{i+1}\xi^2\end{aligned}$$

for sufficient constant in the recurrence for R. This proves the induction.

For some $r = O(\log\log R)$, we have $R_r = O(1)$ and so

$$\|x^* - \chi^{(r)}\|_2^2 \lesssim \xi^2.$$

Then Lemma 5.5.1 shows that the χ' given by RecoverAtConstantSNR satisfies

$$\begin{aligned}\|x^* - \chi^{(r)} - \chi'\|_2^2 &\le \mathrm{Err}_{3k}^2(x^* - \chi^{(r)}) + \epsilon\|x - \chi^{(r)}\|_2^2 + \|x^*\|_2^2/n^{10} \\ &\le \mathrm{Err}_k^2(x^*) + O(\epsilon\xi^2) \\ &\le (1 + O(\epsilon))\,\mathrm{Err}_k^2(x^*) + \|x^*\|_2^2/n^9\end{aligned}$$

which is the desired bound after rescaling ϵ. □

5.9 Utility Lemmas

This section proves a few standalone technical lemmas.

Lemma 5.9.1. *Let* $x, z \in \mathbb{C}^n$ *and* $k \le n$. *Let* S *contain the largest* k *terms of* x *and* T *contain the largest* $2k$ *terms of* z. *Then*

$$\|x - z_T\|_2^2 \le \|x - x_S\|_2^2 + 4\|(x - z)_{S \cup T}\|_2^2.$$

Proof. Note that each term in $\overline{S \cup T}$ and T appears exactly once on each side. Hence it suffices to show that

$$\|x_{S \setminus T}\|_2^2 \le \|x_{T \setminus S}\|_2^2 + 4\|(x - z)_{S \setminus T}\|_2^2 + 3\|(x - z)_T\|_2^2.$$

Consider any $i \in S \setminus T$ and $j \in T \setminus S$. Then $|z_j| \geq |z_i|$ by the choice of T, so by the triangle inequality

$$\begin{aligned} |x_i| &\leq |x_i - z_i| + |z_i| \\ &\leq |x_i - z_i| + |z_j| \\ &\leq |x_i - z_i| + |x_j - z_j| + |x_j| \end{aligned}$$

and so by Cauchy-Schwarz inequality

$$|x_i|^2 \leq 2|x_j|^2 + 4|x_i - z_i|^2 + 4|x_j - z_j|^2. \tag{5.30}$$

Since $|T| = 2|S|$, we can match up each element $i \in S \setminus T$ with a distinct pair P_i of two elements of $T \setminus S$. Summing up (5.30) for $j \in P_i$ and dividing by two,

$$|x_i|^2 \leq \|x_{P_i}\|_2^2 + 4|x_i - z_i|^2 + 2\|(x - z)_{P_i}\|_2^2.$$

Summing up over $i \in S \setminus T$, we have

$$\|x_{S \setminus T}\|_2^2 \leq \|x_{T \setminus S}\|_2^2 + 4\|(x - z)_{S \setminus T}\|_2^2 + 2\|(x - z)_{T \setminus S}\|_2^2$$

which gives the desired result. □

5.9.1 Lemmas on quantiles

This section proves some elementary lemmas on quantiles of random variables, which are a generalization of the median.

Definition 5.9.2. *For $f \geq 0$, we define the $1 - f$ quantile* quant^f *of any list $x_1, \ldots, x_n \in R$ to be the $\lceil (1 - f)n \rceil$th smallest element in the list.*

Then median = $\text{quant}^{1/2}$ for lists of odd length.

Lemma 5.9.3. *Let $f > 0$ and T be constants. Let $X_1, \ldots, X_T$ be independent nonnegative integer random variables with $\mathbb{E}[X_i] \leq \epsilon < 1$ for all i. Let Y satisfy*

$$Y \leq \begin{cases} 0 & \text{if } \text{quant}^f X_i = 0 \\ \sum X_i & \text{otherwise} \end{cases}$$

Then $\mathbb{E}[Y] \lesssim \epsilon^{fT}$.

Proof. For each i, we have $\Pr[X_i = 0] \geq 1 - \epsilon$. Let B_i be a $\{0, 1\}$-valued random variable with $\Pr[B_i = 1] = \epsilon$ and jointly distributed with X_i such that $X_i = 0$ whenever $B_i = 0$. Then let X_i' be a random variable distributed according to $(X_i \mid B_i = 1)$ independent of B_i, so that $X_i = B_i X_i'$. Then $\mathbb{E}[X_i'] = \mathbb{E}[X_i] / \mathbb{E}[B_i] \leq 1$, and we have

$$Y \leq \begin{cases} 0 & \text{if } \text{quant}^f B_i = 0 \\ \sum X_i' & \text{otherwise.} \end{cases}$$

Therefore

$$\begin{aligned}\mathbb{E}[Y] &\le \mathbb{E}[\sum_i X_i'] \Pr[\operatorname{quant}^f B_i \neq 0] \\ &\le T \Pr[\sum B_i \ge 1 + \lfloor fT \rfloor] \\ &\le T \binom{T}{1+\lfloor fT \rfloor} \epsilon^{fT} \lesssim \epsilon^{fT}.\end{aligned}$$

□

Lemma 5.9.4. *Let $f, \delta > 0$ be constants and T be a sufficiently large constant (depending on f and δ). Let $X^1, \dots, X^T \in \mathbb{R}^n$ be independent random variables with nonnegative coordinates and $\mathbb{E}[X_i^t] \le \mu$ independent of i and t. Then for any $k \le n$,*

$$\mathbb{E}[\max_{|A|=k} \sum_{i \in A} \operatorname{quant}_t^f X_i^t] \lesssim k\mu(n/k)^\delta$$

Proof. Let $Y_i = \operatorname{quant}^f{}_t X_i^t$. We have for any threshold η that

$$\begin{aligned}\Pr[Y_i \ge \eta] &= \Pr[|\{t : X_i^t \ge \eta\}| \ge 1 + \lfloor fT \rfloor] \\ &\le \binom{T}{1+\lfloor fT \rfloor} (\mu/\eta)^{fT} \\ &\lesssim (\mu/\eta)^{fT}.\end{aligned}$$

Therefore $\mathbb{E}[|\{i : Y_i \ge \eta\}|] \le n(\mu/\eta)^{fT}$. But then

$$\begin{aligned}\mathbb{E}[\max_{|A|=k} \sum_{i \in A} Y_i] &= \mathbb{E} \int_0^\infty \min(k, |\{i : Y_i \ge \eta\}|) d\eta \\ &\le \int_0^\infty \min(k, n(\mu/\eta)^{fT}) d\eta \\ &= k\mu(n/k)^{1/fT} \int_0^\infty \min(1, u^{-fT}) du \\ &= k\mu(n/k)^{1/fT} \left(1 + \frac{1}{fT-1}\right).\end{aligned}$$

If $T > 1/(\delta f)$ and $T > 2/f$ this gives the result. □

Lemma 5.9.5. *For any $x_1, \dots, x_n \in \mathbb{C}$ with n odd we have*

$$\mathbb{E}[|\operatorname*{median}_t x_t|^2] \le 4 \max_t \mathbb{E}[|x_t|^2]$$

where the median is taken separately in the real and imaginary axes.

Proof. We will show that if $x_i \in \mathbb{R}$ then

$$\mathbb{E}[(\operatorname*{median}_t x_t)^2] \le 2 \max_t \mathbb{E}[x_t^2].$$

applying this separately to the real and imaginary axes gives the result.

Let S be jointly distributed with x as a set of $(n+1)/2$ coordinates i with $x_i^2 \geq \text{median}_t\, x_t^2$. This must exist by choosing coordinates less than or greater than x_i. Then

$$\mathbb{E}[(\underset{t}{\text{median}}\, x_t)^2] \leq \underset{i \in S}{\text{mean}}\, x_i^2 \leq \frac{2}{n+1} \sum_i x_i^2 \leq 2 \underset{i \in [n]}{\text{mean}}\, x_i^2 \leq 2 \max_i x_i^2.$$

□

5.10 1-sparse recovery

```
1: procedure LocateSignal(x̂, χ, B, σ, b, R, p)
2:     n ← Dim(x̂).                                          ▷ Dimension of vector
3:     γ ← R^{1/40 log2 log2 R}
4:     c ← O(log log(n/B) log(1/p)).
5:     T ← Locate1SparseSamples(n, γ, c, n/B).
6:     u^a_[B] ← HashToBins(x̂, χ, P_{σ,a,b}, B, R) for a ∈ T.
7:     v̂^j_a := u^a_j for a ∈ T and j ∈ [B].
8:     L ← {}
9:     for j ∈ [B] do
10:        L ← L ∪ {σ^{-1}(Locate1Sparse(v̂^j, T, γ, jn/B, n/B))}
11:    end for
12:    return L
13: end procedure
14: procedure Locate1SparseSamples(n, γ, c, w)
15:    δ ← γ^{1/10}
16:    t_max ← O(log_{1/δ} w).
17:    g_{i,t} ∈ [n] uniformly for i ∈ [c], t ∈ [t_max].
18:    f_t ∈ [δ^{1−t}/8, δ^{1−t}/4] an arbitrary integer, for all t ∈ [t_max].
19:    return T = ∪_{t∈[t_max],i∈[c]}{g_{i,t}, g_{i,t} + f_t} for all i, t.
20: end procedure
21: procedure Locate1Sparse(v̂_T, T, γ, l, w)
22:    δ ← γ^{1/10}
23:    w_t defined to be wδ^{t−1}.
24:    f_t defined to be any integer in [(n/w_t)/8, (n/w_t)/4].
25:    Expects T = ∪_{t∈[t_max],i∈[c]}{(g_{i,t}, g_{i,t} + f_t)} for t_max = O(log_{1/γ} w)
26:    Define m^{(i)}_t = φ(v̂_{g_{i,t}+f_t} / v̂_{g_{i,t}}).          ▷ Estimates of f_t i* 2π/n
27:    Define m_t = median_i m^{(i)}_t.
28:    l_1 ← l, w_1 ← w.                                    ▷ Location in l_1 − w_1/2, l_1 + w_1/2
29:    for t = 1, ..., t_max do
30:        o_t ← (m_t n/(2π) − (f_t l_t mod n)) / f_t          ▷ Within [−n/2f_t, n/2f_t]
31:        l_{t+1} ← l_t + o_t
32:    end for
33:    return ROUND(l_{t_max+1}).
34: end procedure
```

Algorithm 5.10.1: Fourier 1-sparse recovery

We first show that LOCATE1SPARSE solves the 1-sparse recovery problem. This result is independent of the rest of the machinery in this chapter: if v has a single component with $1-\gamma^{1/2}$ of the mass, we find it with $\widetilde{O}(\log_{1/\gamma} n)$ samples of $\widehat{v}$.

Lemma 5.10.1. *Let $1/\gamma, c$ be larger than a sufficiently large constant. Let $\widehat{v} \in \mathbb{C}^n$, and suppose that there exists an $i^* \in [l - w/2, l + w/2]$ such that*

$$\gamma^{1/2}|v_{i^*}|^2 \geq \sum_{j \neq i^*} |v_j|^2.$$

Then LOCATE1SPARSE*($\widehat{v}_T$, T, γ, l_1, l_1+w_1) returns i^* with all but $\gamma^{\Omega(c)} \log w$ probability, where the set T is the output of* LOCATE1SPARSESAMPLES*(n, γ, c, w) and has size $|T| = O(c(1+\log_{1/\gamma} w))$. The time taken is $O(|T|) = O(c(1+\log_{1/\gamma} w))$.*

Proof. Note that for uniformly random $g \in [n]$, by Parseval's theorem

$$\mathbb{E}[|\sqrt{n}\widehat{v}_g - \omega^{gi^*} v_{i^*}|^2] = \sum_{j \neq i^*} |v_j|^2 \leq \gamma^{1/2}|v_{i^*}|^2$$

Set $b = \gamma^{1/20}$. By Markov's inequality, with $1-b$ probability

$$|\sqrt{n}\widehat{v}_g - \omega^{gi^*} v_{i^*}| \leq \sqrt{\gamma^{1/2}/b}|v_{i^*}|$$

and so

$$\|\phi(\widehat{v}_g) - (\frac{2\pi}{n} g i^* + \phi(v_{i^*}))\|_{\bigcirc} = \|\phi(\sqrt{n}\widehat{v}_g) - \phi\omega^{gi^*} v_{i^*}\|_{\bigcirc} \leq \sin^{-1}(\sqrt{\gamma^{1/2}/b}) \leq 2\sqrt{\gamma^{1/2}/b}$$

where $\|a-b\|_{\bigcirc} = \min_{i \in \mathbb{Z}}(|a-b-2\pi i|)$ denotes the "circular distance" between a and b. Hence for any $(g_{i,t}, g_{i,t} + f_t)$, we have that

$$m_t^{(i)} = \phi(\widehat{v}_{g_{i,t}+f_t} / \widehat{v}_{g_{i,t}})$$

satisfies

$$\|m_t^{(i)} - f_t i^* 2\pi/n\|_{\bigcirc} \leq 4\sqrt{\gamma^{1/2}/b} \tag{5.31}$$

with probability $1-2b$ as a distribution over $g_{i,t}$. Because this is independent for different i, for any t by a Chernoff bound we have that (5.31) holds for at least $3c/4$ of the $m_t^{(i)}$ with probability at least

$$1 - \binom{c}{c/4}(2b)^{c/4} \geq 1 - 2^c(2b)^{c/4} = 1 - (32b)^{c/4} = 1 - \gamma^{\Omega(c)}.$$

If so, their median satisfies the same property[2]

$$\|m_t - f_t i^* 2\pi/n\|_{\bigcirc} \leq 4\sqrt{\gamma^{1/2}/b} \leq 2\pi b\delta. \tag{5.32}$$

Since there are $\log_{1/\gamma^{1/2}} w < \log w$ different t, by a union bound (5.32) holds for all t with the desired probability

$$1 - \gamma^{-\Omega(c)} \log w.$$

[2]To define a median over the circle, we need to cut the circle somewhere; we may do so at any position not within $4\sqrt{\gamma^{1/2}/b}$ of at least $c/4$ of the points.

We will show that this implies that i^* is recovered by the algorithm.

We will have by induction that, for all t, $i^* \in [l_t - w_t/2, l_t + w_t/2]$. This certainly holds at $t = 1$. Recall that $4w_t \le n/f_t \le 8w_t$ by the construction of f_t.

For any t, by (5.32) we have that $o_t f_t$ lies within δbn of $(f_t i^* - f_t l_t)$ (modulo n). Hence $(i^* - l_t)$ lies within $\delta bn/f_t$ of $o_t + zn/f_t$ for $|o_t| < n/(2f_t)$ and some integer z. But since $|i^* - l_t| \le w_t/2 \le n/(8f_t)$ and $\delta bn/f_t < n/(4f_t)$, this means that $z = 0$ and we have that $(i^* - l_t)$ lies within $\delta bn/f_t$ of o_t. Since

$$\delta bn/f_t \le \delta b 8 w_t \le \delta w_t/2 \le w_{t+1}/2,$$

i^* lies within $w_{t+1}/2$ of $l_{t+1} = l_t + o_t$ and the inductive step holds.

In the end, therefore, i^* lies within $w_{t_{max}}/2 = w\delta^{t_{max}-1}/2 < 1/2$ of l, so it is returned by the algorithm. □

We now relate Lemma 5.10.1, which guarantees 1-sparse recovery, to k-sparse recovery of well-hashed signals.

Lemma 5.10.2. *Let x be a signal, and B and R larger than sufficiently large constants. An invocation of* LocateSignal *returns a list L of size B such that each well-hashed (per Definition 5.3.4) $i \in [n]$ is present in L with probability at least $1 - p$. The sample complexity is $O(B \log(Rn/B) \log\log R \log\log(n/B) \max(1, \log_R(1/p))$, and the time complexity is $O(\log R)$ larger.*

Proof. Consider any well-hashed i and $j = h(i)$. We define the vector $y^j \in \mathbb{C}^n$ by

$$y^j_{\sigma\ell} = x_\ell G_{\pi(\ell) - jn/B} = x_\ell G_{o_i(\ell)}.$$

Then

$$u^a_j = \sum_\ell \omega^{a\ell} y^j_\ell = \sqrt{n}\widehat{y^j}_a,$$

i.e. $\widehat{v^j} = \widehat{y^j}/\sqrt{n}$, so $v^j_{\sigma\ell} = x_\ell G_{o_i(\ell)}/\sqrt{n}$.

By the definition 5.3.4 of well-hashedness, over uniformly random $a \in [n]$,

$$\gamma^{1/2} x_i^2 \ge \mathop{\mathbb{E}}_a [|G^{-1}_{o_i(i)} \omega^{-a\sigma i} v^j_a - x_i|^2]$$

If we define $v_{-\sigma i} = v_{[n]\setminus\{\sigma i\}}$, we have after multiplying by $G^2_{o_i(i)}$ that

$$\begin{aligned}
\gamma^{1/2}|v^j_{\sigma i}|^2/n = G^2_{o_i(i)}\gamma^{1/2}|x_i|^2 &\ge \mathop{\mathbb{E}}_a[|\widehat{v^j}_a - \omega^{a\sigma i} G_{o_i(i)} x_i|^2] \\
&= \mathop{\mathbb{E}}_a[|\widehat{v^j}_a - \frac{1}{\sqrt{n}}\omega^{a\sigma i} v^j_{\sigma i}|^2] \\
&= \mathop{\mathbb{E}}_a[|\widehat{(v^j_{-\sigma i})}_a|^2]
\end{aligned}$$

Therefore by Parseval's inequality,

$$\gamma^{1/2}|v^j_{\sigma i}|^2 \ge \|v^j_{-\sigma i}\|_2^2.$$

This is precisely the requirement of Lemma 5.10.1. Hence Locate1Sparse will return σi with all but $\gamma^{\Omega(c)} \log(n/B)$ probability, in which case i will be in the output set L.

Recall that $\log_{1/\gamma} R \lesssim \log\log R$. Setting

$$\begin{aligned} c &= \Theta(\max(1, \log_\gamma(\log(n/B)/p))) \\ &\lesssim \max(1, (\log_R \log(n/B) + \log_R(1/p)) \log\log R) \\ &\lesssim \log\log(n/B) \max(1, \log_R(1/p)) \end{aligned}$$

gives the desired probability $1 - p$. , the number of samples is

$$\begin{aligned} |T| B \log R &= cB \log R \max(1, \log_{1/\gamma}(n/B)) \\ &\lesssim B \log(Rn/B) \log\log R \log\log(n/B) \max(1, \log_R(1/p)). \end{aligned}$$

The time taken is dominated by HASHTOBINS, which takes sample complexity times $\log R$ time. □

5.11 Filter construction

Lemma 5.11.1. *If G is a flat window function with B buckets and contrast $R > 2$, then for some constant c,*

$$\sum_{|i|>cn/2B} G_i^2 \lesssim \frac{n}{R^2 B}$$

Proof. Let c be the constant such that $G_i \le (\frac{cn}{|i|B})^t$ for $t = \log R$. Then

$$\begin{aligned} \sum_{|i|>2cn/B} G_i^2 &\le 2 \sum_{i=2cn/B}^{\infty} (\frac{cn}{|i|B})^{2\log R} \\ &\le \frac{4cn}{B} \sum_{i=1}^{\infty} (\frac{1}{2i})^{2\log R} \\ &= \frac{4cn}{R^2 B} \sum_{i=1}^{\infty} i^{-2\log R} \\ &\lesssim \frac{n}{R^2 B} \end{aligned}$$

and rescaling c gives the result. □

Lemma 5.3.2. *There exist flat window functions where $|\operatorname{supp}(\widehat{G})| \lesssim B \log R$. Moreover, $\operatorname{supp}(\widehat{G}) \subset [-O(B \log R), O(B \log R)]$.*

Proof. Suppose B is an odd integer; otherwise, replace B with $B' = B - 1$. The properties for B' will imply the properties for B, albeit with a worse constant in the third property.

Define $\widehat{F}$ to be a rectangular filter of length B, scaled so F is the Dirichlet kernel

$$F_i = \frac{\sin(\pi B i/n)}{B \sin(\pi i/n)}.$$

Noting that $2|x| \le |\sin(\pi x)| \le \pi|x|$ for $|x| \le 1/2$, we have for all i that

$$|F_i| \le \frac{|\sin(\pi B i/n)|}{2Bi/n} \le \frac{n}{2B|i|} \tag{5.33}$$

and for $i \in [-n/2B, n/2B]$ that

$$|F_i| \geq |\frac{2Bi/n}{B\pi i/n}| = \frac{2}{\pi}. \tag{5.34}$$

Define $\widehat{F'}$ to be $\widehat{F}$ convolved with itself $t = \Theta(\log R)$ times for an even integer t, so $\|\widehat{F'}\|_0 \lesssim B \log R$ and $F'_i = F_i^t$, and by (5.33)

$$0 \leq F'_i \leq \left(\frac{n}{2Bi}\right)^t. \tag{5.35}$$

Now, define G to be F' convolved with a length $\ell = 2\lfloor n/(2B) \rfloor + 1$ rectangular filter, i.e.

$$G_i = \sum_{|j-i| \leq n/(2B)} F'_j,$$

so $\widehat{G}$ is $\widehat{F'}$ multiplied by a scaled Dirichlet kernel. By the last equation, it follows that $\|\widehat{G}\|_0 \leq \|\widehat{F'}\|_0 \lesssim B \log R$. We would just like to show that $G/\|G\|_\infty$ satisfies the flat window function requirements.

Since $F'_i \geq 0$ per (5.35), we have $0 \leq G_i/\|G\|_\infty \leq 1$ so $G/\|G\|_\infty$ passes the second property of a flat window function.

For the first property of flat window functions, let $a = \sum_{i=0}^{\lfloor n/(2B) \rfloor} F'_i$. We have that $G_i \geq a$ for $|i| \leq n/(2B)$ because each of those terms (or their symmetries F'_{-i}) appear in the summation that forms G_i. So it suffices to show that $G_i \leq 3a$ for all i.

Define $S_k = \mathbb{Z} \cap [kn/(2B), (k+1)n/(2B)]$ for $k \in \mathbb{Z}$, so $|S_k| \leq \lceil n/(2B) \rceil$ for all k. For any i, $\{j : |j - i| \leq n/(2B)\}$ has nonzero intersection with at most 3 different S_k. Hence it suffices to show for all k that

$$a \geq \sum_{j \in S_k} F'_j.$$

To do this, we extend the definition of F'_x to all $x \in \mathbb{R}$. By symmetry, it suffices to consider $k \geq 0$. We have that $\sin(\pi x/n)$ is increasing on $[0, n/2]$, so for $0 \leq x \leq n/2 - n/B$ we have

$$F'_{x+n/B}/F'_x = \left(\frac{\sin(\pi x/n)}{\sin(\pi(x + n/B)/n)}\right)^t < 1.$$

Therefore, for each $j \in S_k$,

$$F'_j \leq F'_{j - \lceil k/2 \rceil (n/B)} = F'_{|j - \lceil k/2 \rceil (n/B)|}.$$

Let $T_k = \{|j - \lceil k/2 \rceil (n/B)| : j \in S_k\}$. By considering the even and odd cases for k, we conclude that $T_k \subset [0, n/(2B)]$ and that for some parameter $\theta \geq 0$ we have

$$T_k = \{\theta, \theta + 1, \ldots, \theta + |T| - 1\}.$$

Since F' is decreasing on $[0, n/(2B)]$ we have that

$$\sum_{j \in S_k} F'_j \leq \sum_{j \in T_k} F'_j = \sum_{j=0}^{|T|-1} F'_{\theta+j} \leq \sum_{j=0}^{|T|-1} F'_j \leq \sum_{j=0}^{\lfloor n/(2B) \rfloor} F'_j = a.$$

Therefore $G/\|G\|_\infty$ satisfies the first property of a flat window function.

Lastly, the third property of flat window functions. Consider $i = \alpha n/2B$ with $\alpha \geq 2$ (for smaller i, $G_i \leq 1$ suffices as a bound). We have by (5.35) that

$$G_i \leq \ell \max_{|j-i|\leq n/2B} F'_j \leq \ell(\frac{n}{2B(|i| - n/(2B))})^t = \ell(\frac{1}{\alpha - 1})^t.$$

We also have by (5.34) that

$$\|G\|_\infty \geq G_0 \geq \ell \min_{|i|\leq n/(2B)} F'_i \geq \ell(2/\pi)^t.$$

Hence

$$G_i/\|G\|_\infty \leq (\frac{\pi}{2(\alpha - 1)})^t = (O(1/\alpha))^t = (O(\frac{n}{B|i|}))^t$$

which is the third property of flat window functions. Thus $G/\|G\|_\infty$ is the desired flat window function. □

For a bucketing (σ, b), each coordinate j is permuted to an index $\pi(j) = \sigma j - b$, with nearest multiple of (n/B) being $(n/B)h(j)$. Define the offset of j relative to i to be $o_i(j) = \pi(j) - (n/B)h(i)$.

Given a bucketing (σ, b), for each bucket $j \in [B]$ we define the associated "bucket vectors" $v^{(j)}$ given by

$$v^{(j)}_{\sigma i} := x_i G_{\pi(i)-(n/B)j}.$$

This has the property that running the algorithm with offset a yields $u \in \mathbb{R}^B$ given by

$$u_j = \sum_i v^{(j)}_i \omega^{ia} = \widehat{v^{(j)}}_a.$$

For any bucketing (σ, b), we say that a bucket j has *noise at most* μ^2 if $\|v^{(j)}\|_2^2 \leq \mu^2$. We say that an index i is hashed with noise at most μ^2 if, for $j = h(i)$, we have

$$\|v^{(j)} - x_i G_{\pi(i)-(n/B)j}\|_2^2 \leq \mu^2.$$

We show how to relate the pairwise independence property 4.2.4 to flat window functions:

Lemma 5.11.2. *Let G be a flat window function with B buckets and contrast R. Then for $i \neq j$, there exists a constant c such that*

$$\mathbb{E}[G^2_{o_i(j)} \cdot I[|o_i(j)| > cn/B]] \lesssim \frac{1}{R^2 B}.$$

where $I[a > b]$ is 1 when $a > b$ and 0 otherwise.

Proof. Note that $o_i(j) = \pi(j) - (n/B)h(i)$ is within $n/(2B)$ of $\pi(j) - \pi(i) = \sigma(j - i)$. Let $f \geq 1$ be the constant such that

$$G_{o_i(j)} \leq \left(\frac{f}{B|o_i(j)|/n}\right)^{\log R}.$$

Then

$$\begin{aligned} G_{o_i(j)} &\le \max_{|a-\sigma(i-j)|<n/(2B)} G_a \\ &\le \max_{|a-\sigma(i-j)|<n/(2B)} \left(\frac{f}{B|a|/n}\right)^{\log R} \\ &\le \left|\frac{f}{B|\sigma(i-j)|/n - 1/2}\right|^{\log R} \end{aligned}$$

as well as $G_{o_i(j)} \le 1$. Define

$$y_b = \min\left(1, \left|\frac{f}{B|b|/n - 1/2}\right|^{\log R}\right).$$

It suffices to show that, for any $a \neq 0$ and as a distribution over σ,

$$\mathbb{E}[y_{\sigma a}^2 \cdot I[|\sigma a| > cn/B]] \lesssim \frac{1}{R^2 B}.$$

Let $D = 3fn/B \lesssim n/B$. Note that, for $d \ge 1$ and $|b| \ge dD > (2df + 1/2)n/B$,

$$y_b \le \left(\frac{1}{2d}\right)^{\log R} = \frac{1}{R}\frac{1}{R^{\log d}}.$$

Consider the "levels sets" $S_l : \{b \mid 2^l D \le |b| < 2^{l+1} D\}$, for $l \ge 0$. Then by Lemma 4.2.4,

$$\Pr[\sigma a \in S_l] \le 4 \cdot 2^{l+1} D/n \lesssim 2^l D/n$$

and

$$\max_{b \in S_l} y_b \le \frac{1}{R^{l+1}}.$$

Hence

$$\begin{aligned} \mathbb{E}[y_{\sigma a}^2 \cdot I[|\sigma a| \ge D]] &\lesssim \sum_{l=0}^{\infty} (2^l D/n) R^{-2l-2} \\ &\lesssim D/(R^2 n) \lesssim 1/(R^2 B) \end{aligned}$$

because $R^2 > 2$. Since $D \lesssim n/B$, this gives the result. □

Lemma 5.11.3. HashToBins$(\widehat{x}, \chi, P_{\sigma,a,b}, B, R)$ *computes u such that for any $i \in [n]$,*

$$u_{h(i)} = \Delta_{h(i)} + \sum_j G_{o_i(j)} (x - \chi)_j \omega^{a\sigma j}$$

where G is the flat window function with B buckets and contrast R from Lemma 5.3.2, and $\Delta_{h(i)}^2 \le \|\chi\|_2^2/(R^* n^{11})$ is a negligible error term. It takes $O(B \log R)$ samples, and if $\|\chi\|_0 \lesssim B$, it takes and $O(B \log R \log(Rn)$ time.

Proof. Let $S = \mathrm{supp}(\widehat{G})$, so $|S| \lesssim B \log R$ and in fact $S \subset [-O(B \log R), O(B \log R)]$.

First, HashToBins computes

$$y' = \widehat{G} \cdot P_{\sigma,a,b}\widehat{x - \chi'} = \widehat{G} \cdot P_{\sigma,a,b}\widehat{x - \chi} + \widehat{G} \cdot P_{\sigma,a,b}\widehat{\chi - \chi'},$$

for an approximation $\widehat{\chi}'$ to $\widehat{\chi}$. This is efficient because one can compute $(P_{\sigma,a,b}\widehat{x})_S$ with $O(|S|)$ time and samples, and $P_{\sigma,a,b}\widehat{\chi}'_S$ is easily computed from $\widehat{\chi}'_T$ for $T = \{\sigma(j-b) : j \in S\}$. Since T is an arithmetic sequence and χ is B-sparse, by Corollary 5.12.2, an approximation $\widehat{\chi}'$ to $\widehat{\chi}$ can be computed in $O(B \log R \log(Rn))$ time such that

$$|\widehat{\chi}_i - \widehat{\chi}'_i| < \frac{\|\chi\|_2}{R^* n^{12}}$$

for all $i \in T$. Since $\|\widehat{G}\|_1 \le \sqrt{n}\|\widehat{G}\|_2 = \sqrt{n}\|G\|_2 \le n\|G\|_\infty \le n$ and $\widehat{G}$ is 0 outside S, this implies that

$$\|\widehat{G} \cdot P_{\sigma,a,b}(\widehat{\chi - \chi'})\|_2 \le \|\widehat{G}\|_1 \max_{i \in S} |(P_{\sigma,a,b}(\widehat{\chi - \chi'}))_i| = \|\widehat{G}\|_1 \max_{i \in T} |(\widehat{\chi - \chi'})_i| \le \frac{\|\chi\|_2}{R^* n^{11}}. \tag{5.36}$$

Define Δ by $\widehat{\Delta} = \widehat{G} \cdot P_{\sigma,a,b}(\widehat{\chi - \chi'})$. Next, HashToBins computes $u \in \mathbb{C}^B$ such that for all i,

$$u_{h(i)} = \widehat{y'}_{h(i)n/B} = \widehat{y}_{h(i)n/B} + \Delta_{h(i)n/B},$$

for $y = \widehat{G} \cdot P_{\sigma,a,b}\widehat{x - \chi}$. This computation takes $O(\|y'\|_0 + B \log B) \lesssim B \log(Rn)$ time. Let $\mathcal{F}(x) = \widehat{x}$ denote the Fourier transform of x. We have

$$\begin{aligned} u_{h(i)} &= (G * \mathcal{F}(P_{\sigma,a,b}(\widehat{x - \chi})))_{h(i)n/B} + \Delta_{h(i)n/B} \\ &= \sum_{\pi(j) \in [n]} G_{h(i)n/B - \pi(j)} \mathcal{F}(P_{\sigma,a,b}(\widehat{x - \chi}))_{\pi(j)} + \Delta_{h(i)n/B} \\ &= \sum_{i \in [n]} G_{o_i(j)} (x - \chi)_j \omega^{a\sigma j} + \Delta_{h(i)n/B} \end{aligned}$$

where the last step is the definition of $o_i(j)$ and Claim 4.2.2.

Finally, we note that

$$|\Delta_{h(i)n/B}| \le \|\Delta\|_2 = \|\widehat{\Delta}\|_2 = \|\widehat{G} \cdot P_{\sigma,a,b}(\widehat{\chi - \chi'})\|_2 \le \frac{\|\chi\|_2}{R^* n^{11}},$$

where we used (5.36) in the last step. This completes the proof. □

Lemma 5.3.3. *Let $(\sigma, a, b) \in [n]$ be uniform subject to σ being odd. Let $u \in \mathbb{C}^B$ denote the result of* HashToBins$(\widehat{x}^*, \chi, P_{\sigma,a,b}, B, R)$. *Fix a coordinate $i \in [n]$ and define $x = x^* - \chi$. For each (σ, b), we can define variables $C \subset [n]$ and $w > 0$ (and in particular, $C = \{j \ne i : |\sigma(i-j) \bmod n| \le cn/B\}$ for some constant c,) so that*

- *For all j, as a distribution over (σ, b),*

$$\Pr[j \in C] \lesssim 1/B.$$

- *As a distribution over (σ, b),*

$$\mathbb{E}[w^2] \lesssim \frac{\|x\|_2^2}{R^2 B} + \frac{\|x^*\|_2^2}{R^* n^{11}}$$

- *Conditioned on (σ, b) and as a distribution over a,*

$$\mathop{\mathbb{E}}_{a}[|G^{-1}_{o_i(i)}\omega^{-a\sigma i}u_{h(i)} - x_i|^2] \lesssim w^2 + \|x_C\|_2^2.$$

Proof. By Lemma 4.2.4, for any fixed i and j,

$$\Pr[j \in C] = \Pr[|\sigma(i-j)| \le cn/B] \lesssim 1/B$$

which gives the first part.

Define $x' = x - \chi$. Per Lemma 5.11.3, HashToBins computes the vector $u \in \mathbb{C}^B$ given by

$$u_{h(i)} - \Delta_{h(i)} = \sum_j G_{o_i(j)} x'_j \omega^{a\sigma j}$$

for some Δ with $\|\Delta\|_\infty^2 \le \|x\|_2^2/(R^* n^{11})$. We define the vector $v \in \mathbb{C}^n$ by $v_{\sigma j} = x'_j G_{o_i(j)}$, so that

$$u_{h(i)} - \Delta_{h(i)} = \sum_j \omega^{aj} v_j = \sqrt{n}\widehat{v}_a$$

so

$$u_{h(i)} - \omega^{a\sigma i} G_{o_i(i)} x'_i - \Delta_{h(i)} = \sqrt{n}(\widehat{v_{\overline{\{\sigma i\}}}})_a.$$

By Parseval's theorem, therefore,

$$\begin{aligned}
\mathop{\mathbb{E}}_{a}[|G^{-1}_{o_i(i)}\omega^{-a\sigma i}u_{h(i)} - x'_i|^2] &\le 2G^{-2}_{o_i(i)}(\mathop{\mathbb{E}}_{a}[|u_{h(i)} - \omega^{a\sigma i}G_{o_i(i)}x'_i - \Delta_{h(i)}|^2] + \mathop{\mathbb{E}}_{a}[\Delta^2_{h(i)}]) \\
&= 2G^{-2}_{o_i(i)}(\|v_{\overline{\{\sigma i\}}}\|_2^2 + \Delta^2_{h(i)}) \\
&\lesssim \frac{\|\chi\|_2^2}{R^* n^{11}} + \sum_{j \ne i} |x'_j G_{o_i(j)}|^2 \\
&\le \frac{\|\chi\|_2^2}{R^* n^{11}} + \sum_{j \notin C \cup \{i\}} |x'_j G_{o_i(j)}|^2 + \sum_{j \in C} |x'_j|^2
\end{aligned}$$

If we define w^2 to be the first two terms, we satisfy the third part of the lemma statement. Next, we have that

$$\frac{\|\chi\|_2^2}{R^* n^{11}} \le 2(\frac{\|x\|_2^2 + \|x-\chi\|_2^2}{R^* n^{11}}) \lesssim \frac{\|x\|_2^2}{R^* n^{11}} + \frac{\|x-\chi\|_2^2}{R^2 B}.$$

From the other term, for $j \notin C \cup \{i\}$, $|\sigma(i-j)| > cn/B$ so $o_i(j) > (c-1)n/B$. Hence for sufficiently large c, by Lemma 5.11.2,

$$\mathbb{E}[\sum_{j \notin C \cup \{i\}} |x'_j G_{o_i(j)}|^2] \le \sum_{j \ne i} |x_j - \chi_j|^2 \, \mathbb{E}[G^2_{o_i(j)} \cdot I[o_i(j) > (c-1)n/B]] \le \frac{\|x-\chi\|_2^2}{R^2 B}.$$

Hence their sum has

$$\mathbb{E}[w^2] \lesssim \frac{\|x\|_2^2}{R^* n^{11}} + \frac{\|x-\chi\|_2^2}{R^2 B}.$$

This proves the second part of the lemma statement, completing the proof. □

5.12 Semi-equispaced Fourier transform

```
1: procedure SemiEquispaceFFT(x, c)                    ▷ x ∈ ℂ^n is k-sparse
2:     Round k up to a factor of n.
3:     G, Ĝ' ← Filters(n, k, c).
4:     y_i ← (x * G)_{in/2k} for i ∈ [2k].
5:     ŷ ← FFT(y)                                      ▷ 2k dimensional
6:     x̂'_i ← ŷ_i for |i| ≤ k/2.
7:     return x̂'
8: end procedure
```

Algorithm 5.12.1: Semi-equispaced Fourier Transform in $O(k \log(n/\delta))$ time

The following is similar to results of [DR93, PST01].

Lemma 5.12.1. *Let n be a power of two and $c \geq 1$. Suppose $x \in \mathbb{C}^n$ is k-sparse for some k. We can compute $\widehat{x}'_i$ for all $|i| \leq k/2$ in $O(ck \log n)$ time such that*

$$|\widehat{x}'_i - \widehat{x}_i| \leq \|x\|_2 / n^c.$$

Proof. Without loss of generality k is a power of two (round up), so $2k$ divides n.

Let $G, \widehat{G'}$ be the flat window functions of [HIKP12c], so that $G_i = 0$ for all $|i| \gtrsim (n/k)c \log n$, $\|G - G'\|_2 \leq n^{-c}$,

$$\widehat{G'}_i = \begin{cases} 1 & \text{if } |i| \leq k/2 \\ 0 & \text{if } |i| \geq k \end{cases},$$

and $\widehat{G'}_i \in [0, 1]$ everywhere. The construction is that G approximates a Gaussian convolved with a rectangular filter and G is a (truncated) Gaussian times a sinc function, and is efficiently computable.

Define

$$z = x * G.$$

We have that $\widehat{z}_i = \widehat{x}_i \widehat{G}_i$ for all i. Furthermore, because subsampling and aliasing are dual under the Fourier transform, since $y_i = z_{in/(2k)}$ is a subsampling of z we have for $|i| \leq k/2$ that

$$\begin{aligned}
\widehat{x}'_i = \widehat{y}_i &= \sum_{j=0}^{n/2k-1} \widehat{z}_{i+2kj} \\
&= \sum_{j=0}^{n/2k-1} \widehat{x}_{i+2kj} \widehat{G}_{i+2kj} \\
&= \sum_{j=0}^{n/2k-1} \widehat{x}_{i+2kj} \widehat{G'}_{i+2kj} + \sum_{j=0}^{n/2k-1} \widehat{x}_{i+2kj} (\widehat{G}_{i+2kj} - \widehat{G'}_{i+2kj}) \\
&= \widehat{x}_i + \sum_{j=0}^{n/2k-1} \widehat{x}_{i+2kj} (\widehat{G}_{i+2kj} - \widehat{G'}_{i+2kj})
\end{aligned}$$

and so

$$|\widehat{x}'_i - \widehat{x}_i| \leq \|\widehat{x}\|_2 \|\widehat{G} - \widehat{G'}\|_2 \leq \|x\|_2 n^{-c}$$

as desired.

The time complexity is $O(k \log k)$ for a $2k$-dimensional FFT, plus the time to construct y. Because G_i has a localized support, each nonzero coordinate i of x only contributes to $O(c \log n)$ entries of y. Hence the time to construct y is $O(ck \log n)$ times the time to evaluate G at an arbitrary position. Because G is a Gaussian times a sinc function, assuming we can evaluate exponentials in unit time this is $O(ck \log n)$ total. □

This can be easily generalized to arbitrary arithmetic sequences of length k:

Corollary 5.12.2. *Let n be a power of two, $c \geq 1$, and σ odd. Suppose $x \in \mathbb{C}^n$ is k-sparse for some k, and $S = \{\sigma(i-b) : i \in \mathbb{Z}, |i| \leq k\}$. Then we can compute $\widehat{x}'_i$ for all $i \in S$ in $O(ck \log n)$ time such that*

$$|\widehat{x}'_i - \widehat{x}_i| \leq \|x\|_2 / n^c.$$

Proof. Let σ^{-1} denote the inverse of σ modulo n. Define $x^*_j = \omega^{-bj} x_{\sigma^{-1}j}$. Then for all $i \in [n]$,

$$\begin{aligned}
\widehat{x}_{\sigma(i-b)} &= \frac{1}{\sqrt{n}} \sum_{j \in [n]} \omega^{\sigma(i-b)j} x_j \\
&= \frac{1}{\sqrt{n}} \sum_{j \in [n]} \omega^{i\sigma j} \omega^{-b\sigma j} x_j \\
&= \frac{1}{\sqrt{n}} \sum_{j' = \sigma j \in [n]} \omega^{ij'} \omega^{-bj'} x_{\sigma^{-1}j'} \\
&= \widehat{x}^*_i.
\end{aligned}$$

We can sample from $\widehat{x}^*_i$ with $O(1)$ overhead, so by Lemma 5.12.1 we can approximate $\widehat{x}_{\sigma(i-b)} = \widehat{x}^*_i$ for $|i| \leq k$ in $O(ck \log n)$ time. □

To compute $G_{o_i(i)}$, we take the opposite semi-equispaced Fourier transform.

```
1: procedure ConverseSemiEquispaceFFT(x̂, S, c)        ▷ supp(x) ∈ [−k/2, k/2]
2:     Round k up to a factor of n.
3:     G, Ĝ′ ← Filters(n, k, c).
4:     u ← InvFFT(x̂_[−k,k])                            ▷ 2k dimensional
5:     y_in/(2k) ← u_i for i ∈ [2k].
6:     x′_i ← Σ_{j∈supp(G): i+j≡0 mod n/(2k)} G_j y_{i+j} for i ∈ S.
7:     return x′
8: end procedure
```

Algorithm 5.12.2: Converse semi-equispaced Fourier Transform in $O(k \log(n/\delta))$ time

Lemma 5.12.3. *Let n be a power of two and $c \geq 1$. Suppose $\widehat{x} \in \mathbb{C}^n$ has $\operatorname{supp}(x) \in [-k/2, k/2]$, and let $S \subset [n]$ have $|S| = k$. We can compute x'_i for all $i \in S$ in $O(ck \log n)$ time such that*

$$|x'_i - x_i| \leq \|x\|_2 / n^c.$$

Proof. Without loss of generality k is a power of two (round up), so $2k$ divides n.

Let $G, \widehat{G'}$ be the flat window functions of [HIKP12c], so that $G_i = 0$ for all $|i| \gtrsim (n/k)c \log n$, $\|G - G'\|_2 \leq n^{-c}$,

$$\widehat{G'}_i = \begin{cases} 1 & \text{if } |i| \leq k/2 \\ 0 & \text{if } |i| \geq k \end{cases},$$

and $\widehat{G'}_i \in [0, 1]$ everywhere. The construction is that G approximates a Gaussian convolved with a rectangular filter and G is a (truncated) Gaussian times a sinc function, and is efficiently computable.

For the y defined in the algorithm, we have that $y_{in/(2k)} = x_{in/(2k)}\sqrt{n/(2k)}$ by the definition of the Fourier transform. Setting $y_j = 0$ elsewhere, y is a scaled subsampling x. Since subsampling and aliasing are dual under the Fourier transform, we have that $\widehat{y}_i = \sum_{j=-\infty}^{\infty} \widehat{x}_{i+2kj}$.

Therefore $\widehat{x} = \widehat{y} \cdot \widehat{G'}$, so $x = y * G'$. Then for any i,

$$\begin{aligned} |x'_i - x_i| &= |\sum_j (G_j - G'_j) y_{i+j}| \\ &\leq \|G - G'\|_2 \|y\|_2 \\ &\lesssim n^{-c}\sqrt{n/(2k)}\|x\|_2. \end{aligned}$$

Rescaling c gives the result.

The time complexity is $O(k \log k)$ for the Fourier transform and $O(ck \log n)$ for the summation to form x', giving $O(ck \log n)$ time total. □

5.12.1 Computing G, $\widehat{G}$

Our algorithm needs to know, for each R, both $\widehat{G}_i$ for $|i| \leq B \log R$ and $G_{o_i(j)}$ for various j. Here we show how to compute these up to $1/n^c$ precision for an arbitrary constant c with no additional time overhead beyond the already existing $\log(Rn)$ factor.

Computing $\widehat{G}_i$ for all i is possible in $O(B \log^2 R)$ time, because it is a sinc function times a degree $\log R$ polynomial at each position i. Since we only need this once for each R, the total time is at most a $\log R$ factor above the sample complexity.

For each hashing in estimation phases, we will need to compute $G_{o_i(j)}$ for the set L of $O(B)$ coordinates. We will already know $\widehat{G}$, which is $O(B \log R)$ sparse and dense around the origin. Hence Lemma 5.12.3 can compute $G_{o_i(j)}$ in $O(B \log R \log n)$ time, which is only $\log n$ more than the sample complexity to perform the hashing.

Part II

Impossibility Results

Chapter 6

Gaussian Channel-Based Lower Bounds: ℓ_2

(Based on parts of [PW11, HIKP12a, PW12])

This section gives lower bounds for the measurement complexity required to achieve the ℓ_2/ℓ_2 recovery guarantee, in the three sensing modalities considered in Part I. Namely, we show:

- **Nonadaptive, arbitrary measurements**: $\Omega(k \log_C(n/k))$ for C-approximate recovery.
- **Adaptive, Fourier measurements**: $\Omega(k \log(n/k)/\log\log n)$ for $O(1)$-approximate recovery.
- **Adaptive, arbitrary measurements**: $\Omega(\log\log n)$ for $O(1)$-approximate recovery.

These proofs all involve the information capacity of the additive white Gaussian noise channel.

6.1 Notation

We use $\log x$ to denote $\log_2 x$, and $\ln x$ to denote $\log_e x$. For a discrete random variable X with probability distribution p, we use $H(X)$ or $H(p)$ to denote its entropy

$$H(X) = H(p) = \sum -p(x) \log p(x).$$

For a continuous random variable X with pdf p, we use $h(X)$ to denote its differential entropy

$$h(X) = \int_{x \in X} -p(x) \log p(x) dx.$$

Let y be drawn from a random variable Y. Then $(X \mid y) = (X \mid Y = y)$ denotes the random variable X conditioned on $Y = y$. We define $h(X \mid Y) = \mathbb{E}_{y \sim Y} h(X \mid y)$. The mutual information between X and Y is denoted $I(X;Y) = h(X) - h(X \mid Y)$.

We will also use the following lemma, which shows that mutual information is additive for linear measurements with independent noise. This means that it suffices to bound the mutual information of a individual measurements in isolation.

Lemma 6.1.1. *Suppose $a_i = b_i + w_i$ for $i \in [s]$ and the w_i are independent of each other and the b_i. Then*

$$I(a;b) \leq \sum_i I(a_i;b_i)$$

Proof. Note that $h(a \mid b) = h(a - b \mid b) = h(w \mid b) = h(w)$. Thus

$$\begin{aligned} I(a;b) &= h(a) - h(a \mid b) = h(a) - h(w) \\ &\leq \sum_i h(a_i) - h(w_i) \\ &= \sum_i h(a_i) - h(a_i \mid b_i) = \sum_i I(a_i;b_i) \end{aligned}$$

□

6.2 Nonadaptive, arbitrary linear measurements

Our goal in this section is to lower bound the number of rows of a distribution of matrices A if one can recover x' from Ax such that

$$\|x' - x\|_2 \leq C \min_{k\text{-sparse } y} \|x - y\|_2$$

with at least $3/4$ probability for all x. We will give a lower bound of $\Omega(k \log_C(n/k))$, for all $C > 1$.

Define $\epsilon = C - 1$.

We will set up a communication game. Let $\mathcal{F} \subset \{S \subset [n] \mid |S| = k\}$ be a family of k-sparse supports such that:

- $|S \oplus S'| \geq k$ for $S \neq S' \in \mathcal{F}$, where $\oplus$ denotes the exclusive difference between two sets,
- $\Pr_{S \in \mathcal{F}}[i \in S] = k/n$ for all $i \in [n]$, and
- $\log |\mathcal{F}| = \Omega(k \log(n/k))$.

This is possible; for example, a random linear code on $[n/k]^k$ with relative distance $1/2$ has these properties [Gur10].[1]

Let $X = \{x \in \{0, \pm 1\}^n \mid \text{supp}(x) \in \mathcal{F}\}$. Let $w \sim N(0, \alpha \frac{k}{n} I_n)$ be i.i.d. normal with variance $\alpha k/n$ in each coordinate for some parameter α which will be $\Theta(1/\epsilon)$. Consider the following process:

Procedure. First, Alice chooses $S \in \mathcal{F}$ uniformly at random, then $x \in X$ uniformly at random subject to $\text{supp}(x) = S$. Alice independently chooses $w \sim N(0, \alpha \frac{k}{n} I_n)$. She sets $y = A(x + w)$ and sends y to Bob. Bob performs sparse recovery on y to recover $x' \approx x$, rounds to X by $\widehat{x} = \arg\min_{x^* \in X} \|x^* - x'\|_2$, and sets $S' = \text{supp}(\widehat{x})$. This gives a Markov chain $S \to x \to y \to x' \to \widehat{x} \to S'$.

We will show that deterministic sparse recovery algorithms require large m to succeed on this input distribution $x + w$ with $3/4$ probability. By Yao's minimax principle, this means randomized sparse recovery algorithms also require large m to succeed with $3/4$ probability.

Our strategy is to give upper and lower bounds on $I(S; S')$, the mutual information between S and S'.

Lemma 6.2.1. $I(S; S') = O(m \log(1 + \frac{1}{\alpha}))$.

[1] This assumes n/k is a prime power larger than 2. If n/k is not prime, we can choose $n' \in [n/2, n]$ to be a prime multiple of k, and restrict to the first n' coordinates. This works unless $n/k < 3$, in which case a bound of $\Theta(\min(n, k \log_C(n/k))) = \Theta(k)$ is trivial.

Proof. Let the columns of A^T be $v^1, \ldots, v^m$. We may assume that the v^i are orthonormal, because this can be accomplished via invertible postprocessing of A, by multiplying Ax on the left to orthogonalize A. Then we have that $y_i = \langle v^i, x + w\rangle = \langle v^i, x\rangle + w_i'$, where $w_i' \sim N(0, \alpha k\|v^i\|_2^2/n) = N(0, \alpha k/n)$. Let $z = Ax$, so we have that

$$\mathop{\mathbb{E}}_{x}[z_i^2] = \mathop{\mathbb{E}}_{x}[\langle v^i, x\rangle^2] = \mathop{\mathbb{E}}_{S}[\sum_{j\in S}(v_j^i)^2] = \frac{k}{n}.$$

Hence $y_i = z_i + w_i'$ is a Gaussian channel with power constraint $\mathbb{E}[z_i^2] \le \frac{k}{n}$ and noise variance $\mathbb{E}[(w_i')^2] = \alpha\frac{k}{n}$. Hence by the Shannon-Hartley theorem this channel has information capacity

$$\max_{v_i} I(z_i; y_i) = C \le \frac{1}{2}\log(1 + \frac{1}{\alpha}).$$

Then by the data processing inequality for Markov chains and Lemma 6.1.1, this means

$$I(S; S') \le I(z; y) \le \sum_i I(z_i; y_i) \le \frac{m}{2}\log(1 + \frac{1}{\alpha}).$$

□

We will show that successful recovery either recovers most of x, in which case $I(S; S') = \Omega(k\log(n/k))$, or recovers an ϵ fraction of w. First we show that recovering w requires $m = \Omega(\epsilon n)$.

Lemma 6.2.2. *Suppose $w \in \mathbb{R}^n$ with $w_i \sim N(0, \sigma^2)$ for all i and $n = \Omega(\frac{1}{\epsilon^2}\log(1/\delta))$, and $A \in \mathbb{R}^{m\times n}$ for $m < \delta\epsilon n$. Then any algorithm that finds w' from Aw must have $\|w' - w\|_2^2 > (1-\epsilon)\|w\|_2^2$ with probability at least $1 - O(\delta)$.*

Proof. Note that Aw merely gives the projection of w onto m dimensions, giving no information about the other $n - m$ dimensions. Since w and the ℓ_2 norm are rotation invariant, we may assume WLOG that A gives the projection of w onto the first m dimensions, namely $T = [m]$. By the norm concentration of Gaussians, with probability $1 - \delta$ we have $\|w\|_2^2 < (1+\epsilon)n\sigma^2$, and by Markov with probability $1 - \delta$ we have $\|w_T\|_2^2 \le m\sigma^2/\delta < \epsilon n\sigma^2$. Assume both of these events happen.

For any fixed value d, since w is uniform Gaussian and $w'_{\overline{T}}$ is independent of $w_{\overline{T}}$,

$$\Pr[\|w' - w\|_2^2 < d] \le \Pr[\|(w' - w)_{\overline{T}}\|_2^2 < d] \le \Pr[\|w_{\overline{T}}\|_2^2 < d].$$

Therefore

$$\begin{aligned}\Pr[\|w' - w\|_2^2 < (1-4\epsilon)\|w\|_2^2] &\le \Pr[\|w' - w\|_2^2 < (1-2\epsilon)n\sigma^2] \\ &\le \Pr[\|w_{\overline{T}}\|_2^2 < (1-2\epsilon)n\sigma^2] \\ &\le \Pr[\|w_{\overline{T}}\|_2^2 < (1-\epsilon)(n-m)\sigma^2] \le \delta\end{aligned}$$

as desired. Rescaling ϵ gives the result. □

Lemma 6.2.3. *Suppose $n = \Omega(1/\epsilon^2 + (k/\epsilon)\log(k/\epsilon))$ and $m = O(\epsilon n)$. Then $I(S; S') = \Omega(k\log(n/k))$ for some $\alpha = \Omega(1/\epsilon)$.*

Proof. Consider the x' recovered from $A(x+w)$, and let $T = S \cup S'$. Suppose that $\|w\|_\infty^2 \lesssim \frac{\alpha k}{n}\log n$ and $\|w\|_2^2/(\alpha k) \in [1 \pm \epsilon]$, as happens with probability at least (say) 3/4. Then we claim that if

recovery is successful, one of the following must be true:

$$\|x'_T - x\|_2^2 \le 9\epsilon\|w\|_2^2 \tag{6.1}$$
$$\|x'_{\overline{T}} - w\|_2^2 \le (1-2\epsilon)\|w\|_2^2 \tag{6.2}$$

To show this, suppose $\|x'_T - x\|_2^2 > 9\epsilon\|w\|_2^2 \ge 9\|w_T\|_2^2$ (the last by $|T| = 2k = O(\epsilon n/\log n)$). Then

$$\begin{aligned}\|(x' - (x+w))_T\|_2^2 &\ge (\|x' - x\|_2 - \|w_T\|_2)^2 \\ &> (2\|x' - x\|_2/3)^2 \ge 4\epsilon\|w\|_2^2.\end{aligned}$$

Because recovery is successful,

$$\|x' - (x+w)\|_2^2 \le (1+\epsilon)\|w\|_2^2.$$

Therefore

$$\begin{aligned}\|x'_{\overline{T}} - w_{\overline{T}}\|_2^2 + \|x'_T - (x+w)_T\|_2^2 &= \|x' - (x+w)\|_2^2 \\ \|x'_{\overline{T}} - w_{\overline{T}}\|_2^2 + 4\epsilon\|w\|_2^2 &< (1+\epsilon)\|w\|_2^2 \\ \|x'_{\overline{T}} - w\|_2^2 - \|w_T\|_2^2 &< (1-3\epsilon)\|w\|_2^2 \le (1-2\epsilon)\|w\|_2^2\end{aligned}$$

as desired. Thus with 3/4 probability, at least one of (6.1) and (6.2) is true.

Suppose Equation (6.2) holds with at least 1/4 probability. There must be some x and S such that the same equation holds with 1/4 probability. For this S, given x' we can find T and thus $x'_{\overline{T}}$. Hence for a uniform Gaussian $w_{\overline{T}}$, given $Aw_{\overline{T}}$ we can compute $A(x + w_{\overline{T}})$ and recover $x'_{\overline{T}}$ with $\|x'_{\overline{T}} - w_{\overline{T}}\|_2^2 \le (1-\epsilon)\|w_{\overline{T}}\|_2^2$. By Lemma 6.2.2 this is impossible, since $n - |T| = \Omega(\frac{1}{\epsilon^2})$ and $m = \Omega(\epsilon n)$ by assumption.

Therefore Equation (6.1) holds with at least 1/2 probability, namely $\|x'_T - x\|_2^2 \le 9\epsilon\|w\|_2^2 \le 9\epsilon(1-\epsilon)\alpha k < k/2$ for appropriate α. But if the nearest $\widehat{x} \in X$ to x is not equal to x,

$$\begin{aligned}\|x' - \widehat{x}\|_2^2 =& \|x'_{\overline{T}}\|_2^2 + \|x'_{\overline{T}} - \widehat{x}\|_2^2 \ge \|x'_{\overline{T}}\|_2^2 + (\|x - \widehat{x}\|_2 - \|x'_{\overline{T}} - x\|_2)^2 \\ >& \|x'_{\overline{T}}\|_2^2 + (k - k/2)^2 > \|x'_{\overline{T}}\|_2^2 + \|x'_{\overline{T}} - x\|_2^2 = \|x' - x\|_2^2,\end{aligned}$$

a contradiction. Hence $S' = S$. Then Fano's inequality states $H(S|S') \le 1 + \Pr[S' \ne S]\log|\mathcal{F}|$ and hence

$$I(S;S') = H(S) - H(S|S') \ge -1 + \frac{1}{4}\log|\mathcal{F}| = \Omega(k\log(n/k))$$

as desired. □

Theorem 6.2.4. *Any nonadaptive $(1+\epsilon)$-approximate ℓ_2/ℓ_2 recovery scheme with $\epsilon > \sqrt{\frac{k\log n}{n}}$ and failure probability $\delta < 1/2$ requires $m = \Omega(k\log_{1+\epsilon}(n/k))$.*

Proof. Combine Lemmas 6.2.3 and 6.2.1 with $\alpha = 1/\epsilon$ to get $m = \Omega(\frac{k\log(n/k)}{\log(1+\epsilon)})$, $m = \Omega(\epsilon n)$, or $n = O(\frac{1}{\epsilon}k\log(k/\epsilon))$. For ϵ as in the theorem statement, the first bound is the relevant one. □

6.3 Fourier lower bound

In this section, we show any ℓ_2 recovery algorithm using (possibly adaptive) Fourier samples of x must take $\Omega(k\log(n/k)/\log\log n)$ samples:

Theorem 6.3.1. *Let $F \in \mathbb{C}^{n\times n}$ be orthonormal and satisfy $|F_{i,j}| = 1/\sqrt{n}$ for all i, j. Suppose an algorithm takes m adaptive samples of Fx and computes x' with*

$$\|x - x'\|_2 \le 2 \min_{k\text{-sparse } x^*} \|x - x^*\|_2,$$

with probability at least $3/4$ for any x. Then it must have $m = \Omega(k \log(n/k)/\log\log n)$.

Corollary 6.3.2. *Any algorithm computing the $O(1)$-approximate sparse Fourier transform must access $\Omega(k \log(n/k)/\log\log n)$ samples.*

If the samples were chosen nonadaptively, we would immediately have $m = \Omega(k \log(n/k))$ by Theorem 6.2.4. However, an algorithm could choose samples based on the values of previous samples. In the sparse recovery framework allowing general linear measurements, this adaptivity can decrease the number of measurements to $O(k \log\log(n/k))$ (see Chapter 2); in this section, we show that adaptivity is much less effective in our setting where adaptivity only allows the choice of Fourier coefficients.

In essence, in the previous section we showed that any nonadaptive measurement has constant signal-to-noise ratio for the hard instance. In this section, we show that with high probability all n Fourier measurements have $O(\log n)$ signal-to-noise ratio.

As in the previous section, let $\mathcal{F} \subset \{S \subset [n] : |S| = k\}$ be a family of k-sparse supports such that:

- $|S \oplus S'| \ge k$ for $S \ne S' \in \mathcal{F}$, where $\oplus$ denotes the exclusive difference between two sets, and
- $\log |\mathcal{F}| = \Omega(k \log(n/k))$.

For each $S \in \mathcal{F}$, let $X^S = \{x \in \{0, \pm 1\}^n \mid \operatorname{supp}(x^S) = S\}$. Let $x \in X^S$ uniformly at random. The variables x_i, $i \in S$, are i.i.d. subgaussian random variables with parameter $\sigma^2 = 1$, so for any row F_j of F, $F_j x$ is subgaussian with parameter $\sigma^2 = k/n$. Therefore

$$\Pr_{x \in X^S}[|F_j x| > t\sqrt{k/n}] < 2e^{-t^2/2}.$$

By a union bound over the n rows of F, for each S we can choose an $x^S \in X^S$ with

$$\|Fx^S\|_\infty < O(\sqrt{\frac{k \log n}{n}}). \tag{6.3}$$

Let $X = \{x^S \mid S \in \mathcal{F}\}$ be the set of such x^S.

Let $w \sim N(0, \alpha \frac{k}{n} I_n)$ be i.i.d. normal with variance $\alpha k/n$ in each coordinate.

Consider the following process:

Procedure. First, Alice chooses $S \in \mathcal{F}$ uniformly at random, then selects the $x \in X$ with $\operatorname{supp}(x) = S$. Alice independently chooses $w \sim N(0, \alpha \frac{k}{n} I_n)$ for a parameter $\alpha = \Theta(1)$ sufficiently small. For $j \in [m]$, Bob chooses $i_j \in [n]$ and observes $y_j = F_{i_j}(x + w)$. He then computes the result $x' \approx x$ of sparse recovery, rounds to X by $\widehat{x} = \arg\min_{x^* \in X} \|x^* - x'\|_2$, and sets $S' = \operatorname{supp}(\widehat{x})$. This gives a Markov chain $S \to x \to y \to x' \to \widehat{x} \to S'$.

We will show that deterministic sparse recovery algorithms require large m to succeed on this input distribution $x + w$ with $3/4$ probability. By Yao's minimax principle, this means randomized sparse recovery algorithms also require large m to succeed with $3/4$ probability.

Our strategy is to give upper and lower bounds on $I(S; S')$, the mutual information between S and S'.

First we show an analog of Lemma 6.2.3 in this setting. The previous proof actually applies in this setting, but because we only consider $\epsilon = \Theta(1)$ a simpler proof is possible.

Lemma 6.3.3 (Lemma 6.2.3, simplified for $\epsilon = \Theta(1)$)**.** *There exists a constant $\alpha' > 0$ such that if $\alpha < \alpha'$, then $I(S; S') = \Omega(k \log(n/k))$.*

Proof. Assuming the sparse recovery succeeds (as happens with 3/4 probability), we have $\|x' - (x + w)\|_2 \leq 2\|w\|_2$, which implies $\|x' - x\|_2 \leq 3\|w\|_2$. Therefore

$$\begin{aligned}\|\widehat{x} - x\|_2 &\leq \|\widehat{x} - x'\|_2 + \|x' - x\|_2 \\ &\leq 2\|x' - x\|_2 \\ &\leq 6\|w\|_2.\end{aligned}$$

We also know $\|x' - x''\|_2 \geq \sqrt{k}$ for all distinct $x', x'' \in X$ by construction. Because $\mathbb{E}[\|w\|_2^2] = \alpha k$, with probability at least 3/4 we have $\|w\|_2 \leq \sqrt{4\alpha k} < \sqrt{k}/6$ for sufficiently small α. But then $\|\widehat{x} - x\|_2 < \sqrt{k}$, so $\widehat{x} = x$ and $S = S'$. Thus $\Pr[S \neq S'] \leq 1/2$.

Fano's inequality states $H(S \mid S') \leq 1 + \Pr[S \neq S'] \log |\mathcal{F}|$. Thus

$$I(S; S') = H(S) - H(S \mid S') \geq -1 + \frac{1}{2} \log |\mathcal{F}| = \Omega(k \log(n/k))$$

as desired. □

We next show an upper bound on $I(S; S')$, the analog of Lemma 6.2.1 for adaptive measurements of bounded ℓ_∞ norm. The proof is similar, but is more careful about dependencies and needs the ℓ_∞ bound on Fx.

Lemma 6.3.4.

$$I(S; S') \lesssim m \log(1 + \frac{1}{\alpha} \log n).$$

Proof. Let $A_j = F_{i_j}$ for $j \in [m]$, and let $w'_j = A_j w$. The w'_j are independent normal variables with variance $\alpha \frac{k}{n}$. Because the A_j are orthonormal and w is drawn from a rotationally invariant distribution, the w' are also independent of x. We have $y_j = A_j x + w'_j$. By the data processing inequality for Markov chains and Lemma 6.1.1,

$$I(S; S') \leq I(Ax; y) \leq \sum_j I(A_j x; y_j)$$

By the Shannon-Hartley theorem and Equation (6.3), for all j we have

$$I(A_j x; A_j x + w'_j) \leq \frac{1}{2} \log(1 + \frac{\mathbb{E}[|A_j x|^2]}{\mathbb{E}[(w'_j)^2]}) \leq \frac{1}{2} \log(1 + \frac{\|Fx\|_\infty^2}{\alpha k/n}) \leq \frac{1}{2} \log(1 + \frac{1}{\alpha} \log n).$$

Combining these two equations gives the result. □

Theorem 6.3.1 follows from Lemma 6.3.3 and Lemma 6.3.4, with $\alpha = \Theta(1)$.

6.4 Adaptive lower bound for $k = 1$

In this section we give an $\Omega(\log \log n)$ lower bound for ℓ_2 approximation with arbitrary adaptive linear measurements, showing that Lemma 2.2.4 is tight. More generally, given R "rounds" of adaptivity, we show that $\Omega(R \log^{1/R} n)$ measurements are necessary; this is tight, since the techniques used to show Lemma 2.2.4 can get a matching upper bound.

The main previous lower bound for adaptive sparse recovery gets $m = \Omega(k/\epsilon)$ [ACD11] in an essentially equivalent setting[2]. They consider going down a similar path to our $\Omega(\log \log n)$ lower bound, but ultimately reject it as difficult to bound in the adaptive setting. Combining their result with ours gives a $\Omega(\frac{1}{\epsilon}k + \log \log n)$ lower bound, compared with the $O(\frac{1}{\epsilon}k \cdot \log \log n)$ upper bound.

Setting. One would like to estimate a vector $x \in \mathbb{R}^n$ from m linear measurements $A_1 x, \ldots, A_m x$. One may choose each vector A_i based on $A_1 x, \ldots, A_{i-1} x$, and must output x' satisfying

$$\|x' - x\|_2 \leq O(1) \cdot \min_{k\text{-sparse } y} \|x - y\|_2$$

Intuition. As in the rest of this chapter, we reduce to the information capacity of a Gaussian channel. We consider recovery of the vector $x = e_{i^*} + w$, for $i^* \in [n]$ uniformly and $w \sim N(0, I_n/\Theta(n))$. Correct recovery must find i^*, so the mutual information $I(i^*; Ax)$ is $\Omega(\log n)$. On the other hand, in the nonadaptive case (Section 6.2) we showed that each measurement $A_j x$ is a power-limited Gaussian channel with constant signal-to-noise ratio, and therefore has $I(i^*; A_j x) = O(1)$. Linearity gives that $I(i^*; Ax) = O(m)$, so $m = \Omega(\log n)$ in the nonadaptive case. In the adaptive case, later measurements may "align" the row A_j with i^*, to increase the signal-to-noise ratio and extract more information—this is exactly how the upper bound works in Chapter 2. To deal with this, we bound how much information we can extract as a function of how much we know about i^*. In particular, we show that given a small number b bits of information about i^*, the posterior distribution of i^* remains fairly well "spread out". We then show that any measurement row A_j can only extract $O(b + 1)$ bits from such a spread out distribution on i^*. This shows that the information about i^* increases at most exponentially, so $\Omega(\log \log n)$ measurements are necessary.

Proof. We may assume that the measurements are orthonormal, since this can be performed in post-processing of the output, by multiplying Ax on the left by a lower triangular matrix to orthogonalize A. We will give a lower bound for the following instance:

Alice chooses random $i^* \in [n]$ and i.i.d. Gaussian noise $w \in \mathbb{R}^n$ with $\mathbb{E}[\|w\|_2^2] = \sigma^2 = \Theta(1)$, then sets $x = e_{i^*} + w$. Bob performs R rounds of adaptive measurements on x, getting $y^r = A^r x = (y_1^r, \ldots, y_{m_r}^r)$ in each round r. Let I^* and Y^r denote the random variables from which i^* and y^r are drawn, respectively. We will bound $I(I^*; Y^1, Y^2, \ldots, Y^r)$.

We may assume Bob is deterministic, since we are giving a lower bound for a distribution over inputs—for any randomized Bob that succeeds with probability $1 - \delta$, there exists a choice of random seed such that the corresponding deterministic Bob also succeeds with probability $1 - \delta$.

First, we give a bound on the information received from any single measurement, depending on Bob's posterior distribution on I^* at that point:

[2]Both our result and their result apply in both settings. See Section 6.5 for a more detailed discussion of the relationship between the two settings.

Lemma 6.4.1. *Let I^* be a random variable over $[n]$ with probability distribution $p_i = \Pr[I^* = i]$, and define*

$$b = \sum_{i=1}^{n} p_i \log(np_i) = \log n - H(p).$$

Define $X = e_{I^*} + N(0, I_n\sigma^2/n)$. Consider any fixed vector $v \in \mathbb{R}^n$ independent of X with $\|v\|_2 = 1$, and define $Y = v \cdot X$. Then

$$I(v_{I^*}; Y) \leq C(b+1)$$

for some constant C.

Proof. Let $S_i = \{j \mid 2^i \leq np_j < 2^{i+1}\}$ for $i > 0$ and $S_0 = \{i \mid np_i < 2\}$. Define $t_i = \sum_{j \in S_i} p_j = \Pr[I^* \in S_i]$. Then

$$\begin{aligned}
\sum_{i=0}^{\infty} it_i &= \sum_{i>0} \sum_{j \in S_i} p_j \cdot i \\
&\leq \sum_{i>0} \sum_{j \in S_i} p_j \log(np_j) \\
&= b - \sum_{j \in S_0} p_j \log(np_j) \\
&\leq b - t_0 \log(nt_0/|S_0|) \\
&\leq b + |S_0|/(ne)
\end{aligned}$$

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

$$\sum_{i=0}^{\infty} it_i < b + 1. \tag{6.4}$$

Let $W = N(0, \sigma^2/n)$. For any measurement vector v, let $Y = v \cdot X \sim v_{I^*} + W$. Let $Y_i = (Y \mid I^* \in S_i)$. Because $\sum v_j^2 = 1$,

$$\mathbb{E}[Y_i^2] = \sigma^2/n + \sum_{j \in S_i} v_j^2 p_j/t_i \leq \sigma^2/n + \|p_{S_i}\|_\infty/t_i \leq \sigma^2/n + 2^{i+1}/(nt_i). \tag{6.5}$$

Let T be the (discrete) random variable denoting the i such that $I^* \in S_i$. Then Y is drawn from Y_T, and T has probability distribution t. Hence

$$\begin{aligned}
h(Y) &\leq h((Y,T)) \\
&= H(T) + h(Y_T \mid T) \\
&= H(t) + \sum_{i \geq 0} t_i h(Y_i) \\
&\leq H(t) + \sum_{i \geq 0} t_i h(N(0, \mathbb{E}[Y_i^2]))
\end{aligned}$$

because the Gaussian distribution maximizes entropy subject to a power constraint. Using the

same technique as the Shannon-Hartley theorem,

$$\begin{aligned} I(v_{I^*}, Y) = I(v_{I^*}; v_{I^*} + W) &= h(v_{I^*} + W) - h(v_{I^*} + W \mid v_{I^*}) \\ &= h(Y) - h(W) \\ &\leq H(t) + \sum_{i \geq 0} t_i (h(N(0, \mathbb{E}[Y_i^2])) - h(W)) \\ &= H(t) + \frac{1}{2} \sum_{i \geq 0} t_i \ln(\frac{\mathbb{E}[Y_i^2]}{\mathbb{E}[W^2]}) \end{aligned}$$

and hence by Equation (6.5),

$$I(v_{I^*}; Y) \leq H(t) + \frac{\ln 2}{2} \sum_{i \geq 0} t_i \log(1 + \frac{2^{i+1}}{t_i \sigma^2}). \tag{6.6}$$

All that requires is to show that this is $O(1 + b)$. Since $\sigma = \Theta(1)$, we have

$$\begin{aligned} \sum_i t_i \log(1 + \frac{2^i}{\sigma^2 t_i}) &\leq \log(1 + 1/\sigma^2) + \sum_i t_i \log(1 + \frac{2^i}{t_i}) \\ &\leq O(1) + \sum_i t_i \log(1 + 2^i) + \sum_i t_i \log(1 + 1/t_i). \end{aligned} \tag{6.7}$$

Now, $\log(1 + 2^i) \lesssim i$ for $i > 0$ and is $O(1)$ for $i = 0$, so by Equation (6.4),

$$\sum_i t_i \log(1 + 2^i) \lesssim 1 + \sum_{i > 0} i t_i < 2 + b.$$

Next, $\log(1 + 1/t_i) \lesssim \log(1/t_i)$ for $t_i \leq 1/2$, so

$$\sum_i t_i \log(1 + 1/t_i) \lesssim \sum_{i | t_i \leq 1/2} t_i \log(1/t_i) + \sum_{i | t_i > 1/2} 1 \leq H(t) + 1.$$

Plugging into Equations (6.7) and (6.6),

$$I(v_{I^*}, Y) \lesssim 1 + b + H(t). \tag{6.8}$$

To bound $H(t)$, we consider the partition $T_+ = \{i \mid t_i > 1/2^i\}$ and $T_- = \{i \mid t_i \leq 1/2^i\}$. Then

$$\begin{aligned} H(t) &= \sum_i t_i \log(1/t_i) \\ &\leq \sum_{i \in T_+} i t_i + \sum_{t \in T_-} t_i \log(1/t_i) \\ &\leq 1 + b + \sum_{t \in T_-} t_i \log(1/t_i) \end{aligned}$$

But $x \log(1/x)$ is increasing on $[0, 1/e]$, so

$$\sum_{t \in T_-} t_i \log(1/t_i) \leq t_0 \log(1/t_0) + t_1 \log(1/t_1) + \sum_{i \geq 2} \frac{1}{2^i} \log(1/2^i) \leq 2/e + 3/2 = O(1)$$

and hence $H(t) \leq b + O(1)$. Combining with Equation (6.8) gives that

$$I(v_{I^*}; Y) \lesssim b + 1$$

as desired. □

Theorem 6.4.2. *Any scheme using R rounds with number of measurements $m_1, m_2, \ldots, m_R > 0$ in each round has*

$$I(I^*; Y^1, \ldots, Y^R) \leq C^R \prod_i m_i$$

for some constant $C > 1$.

Proof. Let the signal in the absence of noise be $Z^r = A^r e_{I^*} \in \mathbb{R}^{m_r}$, and the signal in the presence of noise be $Y^r = A^r(e_{I^*} + N(0, \sigma^2 I_n/n)) = Z^r + W^r$ where $W^r = N(0, \sigma^2 I_{m_r}/n)$ independently. In round r, after observations $y^1, \ldots, y^{r-1}$ of $Y^1, \ldots, Y^{r-1}$, let p^r be the distribution on $(I^* \mid y^1, \ldots, y^{r-1})$. That is, p^r is Bob's posterior distribution on I^* at the beginning of round r.

We define

$$\begin{aligned} b_r &= H(I^*) - H(I^* \mid y^1, \ldots, y^{r-1}) \\ &= \log n - H(p^r) \\ &= \sum p_i^r \log(n p_i^r). \end{aligned}$$

Because the rows of A^r are deterministic given $y^1, \ldots, y^{r-1}$, Lemma 6.4.1 shows that any single measurement $j \in [m_r]$ satisfies

$$I(Z_j^r; Y_j^r \mid y^1, \ldots, y^{r-1}) \leq C(b_r + 1).$$

for some constant C. Thus by Lemma 6.1.1

$$I(Z^r; Y^r \mid y^1, \ldots, y^{r-1}) \leq C m_r (b_r + 1).$$

There is a Markov chain $(I^* \mid y^1, \ldots, y^{r-1}) \to (Z^r \mid y^1, \ldots, y^{r-1}) \to (Y^r \mid y^1, \ldots, y^{r-1})$, so

$$I(I^*; Y^r \mid y^1, \ldots, y^{r-1}) \leq I(Z^r; Y^r \mid y^1, \ldots, y^{r-1}) \leq C m_r (b_r + 1).$$

We define $B_r = I(I^*; Y^1, \ldots, Y^{r-1}) = \mathbb{E}_y\, b_r$. Therefore

$$\begin{aligned}
B_{r+1} &= I(I^*; Y^1, \ldots, Y^r) \\
&= I(I^*; Y^1, \ldots, Y^{r-1}) + I(I^*; Y^r \mid Y^1, \ldots, Y^{r-1}) \\
&= B_r + \mathop{\mathbb{E}}_{y^1,\ldots,y^{r-1}} I(I^*; Y^r \mid y^1, \ldots, y^{r-1}) \\
&\leq B_r + Cm_r \mathop{\mathbb{E}}_{y^1,\ldots,y^{r-1}} (b_r + 1) \\
&= (B_r + 1)(Cm_r + 1) - 1 \\
&\leq C' m_r (B_r + 1)
\end{aligned}$$

for some constant C'. Then for some constant $D \geq C'$,

$$I(I^*; Y^1, \ldots, Y^R) = B_{R+1} \leq D^R \prod_i m_i$$

as desired. □

Corollary 6.4.3. *Any scheme using R rounds with m measurements has*

$$I(I^*; Y^1, \ldots, Y^R) \leq (Cm/R)^R$$

for some constant C. Thus for sparse recovery, $m = \Omega(R \log^{1/R} n)$. Minimizing over R, we find that $m = \Omega(\log \log n)$ independent of R.

Proof. The equation follows from the AM-GM inequality. Furthermore, our setup is such that Bob can recover I^* from Y with large probability, so $I(I^*; Y) = \Omega(\log n)$; this is Lemma 6.3.3 in the $k = 1$ case. The result follows. □

6.5 Relationship between post-measurement and pre-measurement noise

In the setting of [ACD11], the goal is to recover a k-sparse x from observations of the form $Ax + w$, where A has unit norm rows and w is i.i.d. Gaussian with variance $\|x\|_2^2/\epsilon^2$. By ignoring the (irrelevant) component of w orthogonal to A, this is equivalent to recovering x from observations of the form $A(x + w)$. By contrast, our goal is to recover $x + w$ from observations of the form $A(x + w)$, and for general w rather than only for Gaussian w.

As shown in Sections 6.2 and 6.3, for Gaussian w the difference between recovering x and recovering $x + w$ is minor. Hence any lower bound of m in the [ACD11] setting implies a lower bound of $\min(m, \epsilon n)$ in our setting. The converse is also true for proofs that use Gaussian w, such as ours.

Chapter 7

Communication Complexity-Based Lower Bounds

(Based on parts of [PW11, PW12])

This chapter considers variations on the recovery guarantee

$$\|x' - x\|_p \leq (1+\epsilon) \min_{k\text{-sparse } y} \|x - y\|_p. \tag{7.1}$$

Previous chapters consider (7.1) for $p = 2$. In this chapter, we consider $p = 1$ and give improved upper and lower bounds, showing that the complexity is $k/\sqrt{\epsilon}$ up to logarithmic factors in both the nonadaptive and adaptive cases.

We also consider the requirement that the result x' must itself be exactly k-sparse. In this case, for $p = 1$ and $p = 2$, we show that k/ϵ^p is tight up to logarithmic factors in the nonadaptive setting.

7.1 Introduction

The difficulty of $(1+\epsilon)$-approximate recovery has seemed to depend on whether the output x' is required to be k-sparse or can have more than k elements in its support. Having k-sparse output is important for some applications (e.g. the MapReduce `top` table [PDGQ05]) but not for others (e.g. imaging). Algorithms that output a k-sparse x' have used $\Theta(\frac{1}{\epsilon^p} k \log n)$ measurements [CCF02, CM04, CM06]. In contrast, [GLPS10] uses only $\Theta(\frac{1}{\epsilon} k \log(n/k))$ measurements for $p = 2$ and outputs a non-k-sparse x'.

		Lower bound	Upper bound
Non-k-sparse output	ℓ_1	$\Omega(\frac{1}{\sqrt{\epsilon}\log(k/\epsilon)}k)$ [†]	$O(\frac{\log^3(1/\epsilon)}{\sqrt{\epsilon}} k \log n)$
	ℓ_2	$\Omega(\frac{1}{\epsilon} k \log(n/k))$ [Ch. 6]	$O(\frac{1}{\epsilon} k \log(n/k))$[GLPS10]
k-sparse output	ℓ_1	$\Omega(\frac{1}{\epsilon}(k \log \frac{1}{\epsilon} + \log \frac{1}{\delta}))$	$O(\frac{1}{\epsilon} k \log n)$[CM04]
	ℓ_2	$\Omega(\frac{1}{\epsilon^2}(k + \log \frac{1}{\delta}))$	$O(\frac{1}{\epsilon^2} k \log n)$[CCF02, CM06]

Figure 7-1: Our results in this chapter, along with existing upper bounds. The bound [†] applies to the adaptive setting as well.

Our results. We show that the apparent distinction between complexity of sparse and non-sparse outputs is fundamental, for both $p = 1$ and $p = 2$. We show that for sparse output, $\Omega(k/\epsilon^p)$ measurements are necessary, matching the upper bounds up to a $\log n$ factor. For general output and $p = 2$, Chapter 6 showed $\Omega(\frac{1}{\epsilon}k\log(n/k))$ measurements are necessary, matching the upper bound up to a constant factor. In the remaining case of general output and $p = 1$, we show $\widetilde{\Omega}(k/\sqrt{\epsilon})$ measurements are necessary. We then give a novel algorithm that uses $O(\frac{\log^3(1/\epsilon)}{\sqrt{\epsilon}}k\log n)$ measurements, beating the $1/\epsilon$ dependence given by all previous algorithms. As a result, all our bounds are tight up to factors logarithmic in n. The full results are shown in Figure 7-1.

In addition, for $p = 2$ and general output, we show that thresholding the top $2k$ elements of a Count-Sketch [CCF02] estimate gives $(1+\epsilon)$-approximate recovery with $\Theta(\frac{1}{\epsilon}k\log n)$ measurements. This is interesting because it highlights the distinction between sparse output and non-sparse output: [CM06] showed that thresholding the top k elements of a Count-Sketch estimate works for $m = O(\frac{1}{\epsilon^2}k\log n)$. While [GLPS10] achieves $m = \Theta(\frac{1}{\epsilon}k\log(n/k))$ for the same regime, it only succeeds with constant probability while ours succeeds with probability $1 - n^{-\Omega(1)}$; hence ours is the most efficient known algorithm when $\delta = o(1), \epsilon = o(1)$, and $k < n^{0.9}$.

Related work. Much of the work on sparse recovery has relied on the Restricted Isometry Property [CRT06b]. None of this work has been able to get better than 2-approximate recovery, so there are relatively few papers achieving $(1 + \epsilon)$-approximate recovery. The existing ones with $O(k\log n)$ measurements are surveyed above (except for [IR08], which has worse dependence on ϵ than [CM04] for the same regime).

Previous work [CM05] has studied lower bounds for the ℓ_∞/ℓ_p problem, where every coordinate must be estimated with small error. This problem is harder than ℓ_p/ℓ_p sparse recovery with sparse output, so lower bounds are easier. For $p = 1$, they showed that any sketch requires $\Omega(k/\epsilon)$ bits (rather than measurements).

Our techniques. For the upper bounds for non-sparse output, we observe that the hard case for sparse output is when the noise is fairly concentrated, in which the estimation of the top k elements can have $\sqrt{\epsilon}$ error. Our goal is to recover enough mass from outside the top k elements to cancel this error. The upper bound for $p = 2$ is a fairly straightforward analysis of the top $2k$ elements of a Count-Sketch data structure.

The upper bound for $p = 1$ proceeds by subsampling the vector at rate 2^{-i} and performing a Count-Sketch with size proportional to $\frac{1}{\sqrt{\epsilon}}$, for $i \in \{0, 1, \ldots, O(\log(1/\epsilon))\}$. The intuition is that if the noise is well spread over many (more than $k/\epsilon^{3/2}$) coordinates, then the ℓ_2 bound from the first Count-Sketch gives a very good ℓ_1 bound, so the approximation is $(1 + \epsilon)$-approximate. However, if the noise is concentrated over a small number k/ϵ^c of coordinates, then the error from the first Count-Sketch is proportional to $1 + \epsilon^{c/2+1/4}$. But in this case, one of the subsamples will only have $O(k/\epsilon^{c/2-1/4}) < k/\sqrt{\epsilon}$ of the coordinates with large noise. We can then recover those coordinates with the Count-Sketch for that subsample. Those coordinates contain an $\epsilon^{c/2+1/4}$ fraction of the total noise, so recovering them decreases the approximation error by exactly the error induced from the first Count-Sketch.

The lower bounds use substantially different techniques for sparse output and for non-sparse output. For sparse output, we use reductions from communication complexity to show a lower bound in terms of bits. Then, as in [DIPW10], we embed $\Theta(\log n)$ copies of this communication problem into a single vector. This multiplies the bit complexity by $\log n$; we also show we can round Ax to $\log n$ bits per measurement without affecting recovery, giving a lower bound in terms of measurements.

We illustrate the lower bound on bit complexity for sparse output using $k = 1$. Consider a vector x containing $1/\epsilon^p$ ones and zeros elsewhere, such that $x_{2i} + x_{2i+1} = 1$ for all i. For any i, set $z_{2i} = z_{2i+1} = 1$ and $z_j = 0$ elsewhere. Then successful $(1 + \epsilon/3)$-approximate sparse recovery from $A(x + z)$ returns z' with $\text{supp}(z') = \text{supp}(x) \cap \{2i, 2i + 1\}$. Hence we can recover each bit of x with probability $1 - \delta$, requiring $\Omega(1/\epsilon^p)$ bits[1]. We can generalize this to k-sparse output for $\Omega(k/\epsilon^p)$ bits, and to δ failure probability with $\Omega(\frac{1}{\epsilon^p} \log \frac{1}{\delta})$. However, the two generalizations do not seem to combine.

For non-sparse output, the hard instances for $k = 1$ must have one large value (or else 0 is a valid output) but small other values (or else the 2-sparse approximation is significantly better than the 1-sparse approximation). Suppose x has one value of size ϵ and d values of size $1/d$ spread through a vector of size d^2. Then a $(1 + \epsilon/2)$-approximate recovery scheme must either locate the large element or guess the locations of the d values with $\Omega(\epsilon d)$ more correct than incorrect. The former requires $1/(d\epsilon^2)$ bits by the difficulty of a novel version of the Gap-ℓ_∞ problem that we call $\mathsf{Multi}\ell_\infty$. The latter requires ϵd bits because it allows recovering an error correcting code. Setting $d = \epsilon^{-3/2}$ balances the terms at $\epsilon^{-1/2}$ bits. We then convert to a bound on measurement complexity, losing a $\log n$ factor in a universe of size $n = \text{poly}(k/\epsilon)$.

Intuition for $\mathsf{Multi}\ell_\infty$. In the $\mathsf{Gap}\ell_\infty$ problem, the two players are given x and y respectively, and they want to approximate $\|x-y\|_\infty$ given the promise that all entries of $x-y$ are small in magnitude or there is a single large entry. The $\mathsf{Multi}\ell_\infty$ problem consists of solving multiple independent instances of $\mathsf{Gap}\ell_\infty$ in parallel. Intuitively, the sparse recovery algorithm needs to determine if there are entries of $x - y$ that are large, which corresponds to solving multiple instances of $\mathsf{Gap}\ell_\infty$. We prove a multiround direct sum theorem for a distributional version of $\mathsf{Gap}\ell_\infty$, thereby giving a distributional lower bound for $\mathsf{Multi}\ell_\infty$. We use the information complexity framework [BJKS04] to lower bound the conditional mutual information between the inputs to $\mathsf{Gap}\ell_\infty$ and the transcript of any correct protocol for $\mathsf{Gap}\ell_\infty$ under a certain input distribution, and prove a direct sum theorem for solving k instances of this problem. We need to condition on "help variables" in the mutual information which enable the players to embed single instances of $\mathsf{Gap}\ell_\infty$ into $\mathsf{Multi}\ell_\infty$ in a way in which the players can use a correct protocol on our input distribution for $\mathsf{Multi}\ell_\infty$ as a correct protocol on our input distribution for $\mathsf{Gap}\ell_\infty$; these help variables are in addition to help variables used for proving lower bounds for $\mathsf{Gap}\ell_\infty$, which is itself proved using information complexity. We also look at the conditional mutual information with respect to an input distribution which doesn't immediately fit into the information complexity framework. We relate the conditional information of the transcript with respect to this distribution to that with respect to a more standard distribution.

7.2 Relevant upper bounds

The algorithms in this section are indifferent to permutation of the coordinates. Therefore, for simplicity of notation in the analysis, we assume the coefficients of x are sorted such that $|x_1| \geq |x_2| \geq \ldots \geq |x_n| \geq 0$.

Count-Sketch. Both our upper bounds use the Count-Sketch [CCF02] data structure. The structure consists of $c \log n$ hash tables of size $O(q)$, for $O(cq \log n)$ total space; it can be represented

[1]For $p = 1$, we can actually set $|\text{supp}(z)| = 1/\epsilon$ and search among a set of $1/\epsilon$ candidates. This gives $\Omega(\frac{1}{\epsilon} \log(1/\epsilon))$ bits.

as Ax for a matrix A with $O(cq \log n)$ rows. Given Ax, one can construct x^* with

$$\|x^* - x\|_\infty^2 \le \frac{1}{q}\|x_{\overline{[q]}}\|_2^2 \tag{7.2}$$

with failure probability n^{1-c}.

7.2.1 ℓ_2

It was shown in [CM06] that, if x^* is the result of a Count-Sketch with hash table size $O(k/\epsilon^2)$, then outputting the top k elements of x^* gives a $(1+\epsilon)$-approximate ℓ_2/ℓ_2 recovery scheme. Here we show that a seemingly minor change—selecting $2k$ elements rather than k elements—turns this into a $(1+\epsilon^2)$-approximate ℓ_2/ℓ_2 recovery scheme.

Lemma 7.2.1. *For any $x, x^* \in \mathbb{R}^n$ let S contain the largest $2k$ elements of x^*. Then*

$$\|x_S^* - x\|_2^2 \le \mathrm{Err}_k^2(x) + 3k\|x^* - x\|_\infty^2.$$

Proof. Since $\mathrm{Err}_k^2(x) = \|x_{\overline{[k]}}\|_2^2$ by our WLOG assumption on the ordering,

$$\begin{aligned}\|x_S^* - x\|_2^2 - \mathrm{Err}_k^2(x) &\le \|(x^* - x)_S\|_2^2 + \|x_{[n]\setminus S}\|_2^2 - \|x_{\overline{[k]}}\|_2^2 \\ &\le 2k\|x^* - x\|_\infty^2 + \|x_{[k]\setminus S}\|_2^2 - \|x_{S\setminus[k]}\|_2^2 \end{aligned} \tag{7.3}$$

Let $a = \max_{i\in[k]\setminus S} x_i$ and $b = \min_{i\in S\setminus[k]} x_i$, and let $d = |[k] \setminus S|$. The algorithm passes over an element of value a to choose one of value b, so

$$a \le b + 2\|x^* - x\|_\infty$$

Then

$$\begin{aligned}\|x_{[k]\setminus S}\|_2^2 - \|x_{S\setminus[k]}\|_2^2 &\le da^2 - (k+d)b^2 \\ &\le d(b + 2\|x^* - x\|_\infty)^2 - (k+d)b^2 \\ &= -kb^2 + 4db\|x^* - x\|_\infty + 4d\|x^* - x\|_\infty^2 \\ &= -k(b - 2(d/k)\|x^* - x\|_\infty)^2 + 4d\|x^* - x\|_\infty^2 \frac{k-d}{k} \\ &\le \frac{4d(k-d)}{k}\|x^* - x\|_\infty^2 \le k\|x^* - x\|_\infty^2\end{aligned}$$

and combining this with (7.3) gives

$$\|x_S^* - x\|_2^2 - \mathrm{Err}_k^2(x) \le 3k\|x^* - x\|_\infty^3$$

as desired. □

Theorem 7.2.2. *Let x' be the top $2k$ estimates from a Count-Sketch structure with hash table size $O(k/\epsilon)$. Then with failure probability $n^{-\Omega(1)}$,*

$$\|x' - x\|_2 \le (1+\epsilon)\,\mathrm{Err}_k^2(x).$$

Therefore, there is a $1+\epsilon$-approximate ℓ_2/ℓ_2 recovery scheme with $O(\frac{1}{\epsilon}k\log n)$ rows.

Proof. Let the hash table size be $O(k/\epsilon)$, and let x^* be the vector of estimates for each coordinate. By (7.2), the standard analysis of Count-Sketch,

$$\|x^* - x\|_\infty^2 \le \frac{\epsilon}{k} \operatorname{Err}_{ck/\epsilon}^2(x) \le \frac{\epsilon}{k} \operatorname{Err}_k^2(x).$$

with the appropriate failure probability. Then by Lemma 7.2.1,

$$\|x' - x\|_2^2 \le \operatorname{Err}_k^2(x) + 3k\frac{\epsilon}{k} \operatorname{Err}_k^2(x) \le (1 + 3\epsilon) \operatorname{Err}_k^2(x).$$

Rescaling ϵ gives the result. □

7.2.2 ℓ_1

Theorem 7.2.3. *There exists a $(1+\epsilon)$-approximate ℓ_1/ℓ_1 recovery scheme with $O(\frac{\log^3 1/\epsilon}{\sqrt{\epsilon}} k \log n)$ measurements and failure probability $e^{-\Omega(k/\sqrt{\epsilon})} + n^{-\Omega(1)}$.*

Set $f = \sqrt{\epsilon}$, so our goal is to get $(1+f^2)$-approximate ℓ_1/ℓ_1 recovery with $O(\frac{\log^3 1/f}{f} k \log n)$ measurements.

For intuition, consider 1-sparse recovery of the following vector x: let $c \in [0,2]$ and set $x_1 = 1/f^9$ and $x_2, \ldots, x_{1+1/f^{1+c}} \in \{\pm 1\}$. Then we have

$$\|x_{\overline{[1]}}\|_1 = 1/f^{1+c}$$

and by (7.2), a Count-Sketch with $O(1/f)$-sized hash tables returns x^* with

$$\|x^* - x\|_\infty \le \sqrt{f}\|x_{\overline{[1/f]}}\|_2 \approx 1/f^{c/2} = f^{1+c/2}\|x_{\overline{[1]}}\|_1.$$

The reconstruction algorithm therefore cannot reliably find any of the x_i for $i > 1$, and its error on x_1 is at least $f^{1+c/2}\|x_{\overline{[1]}}\|_1$. Hence the algorithm will not do better than a $f^{1+c/2}$-approximation.

However, consider what happens if we subsample an f^c fraction of the vector. The result probably has about $1/f$ nonzero values, so a $O(1/f)$-width Count-Sketch can reconstruct it exactly. Putting this in our output improves the overall ℓ_1 error by about $1/f = f^c\|x_{\overline{[1]}}\|_1$. Since $c < 2$, this more than cancels the $f^{1+c/2}\|x_{\overline{[1]}}\|_1$ error the initial Count-Sketch makes on x_1, giving an approximation factor better than 1.

This tells us that subsampling can help. We don't need to subsample at a scale below k/f (where we can reconstruct well already) or above k/f^3 (where the ℓ_2 bound is small enough already), but in the intermediate range we need to subsample. Our algorithm subsamples at all $\log 1/f^2$ rates in between these two endpoints, and combines the heavy hitters from each.

First we analyze how subsampled Count-Sketch works.

Lemma 7.2.4. *Suppose we subsample with probability p and then apply Count-Sketch with $\Theta(\log n)$ rows and $\Theta(q)$-sized hash tables. Let y be the subsample of x. Then with failure probability $e^{-\Omega(q)} + n^{-\Omega(1)}$ we recover a y^* with*

$$\|y^* - y\|_\infty \le \sqrt{p/q}\|x_{\overline{[q/p]}}\|_2.$$

Proof. Recall the following form of the Chernoff bound: if $X_1, \ldots, X_m$ are independent with $0 \le X_i \le M$, and $\mu \ge \mathbb{E}[\sum X_i]$, then

$$\Pr[\sum X_i \ge \frac{4}{3}\mu] \le e^{-\Omega(\mu/M)}.$$

Let T be the set of coordinates in the sample. Then $\mathbb{E}[|T \cap [\frac{3q}{2p}]|] = 3q/2$, so

$$\Pr\left[|T \cap [\frac{3q}{2p}]| \geq 2q\right] \leq e^{-\Omega(q)}.$$

Suppose this event does not happen, so $|T \cap [\frac{3q}{2p}]| < 2q$. We also have

$$\|x_{\overline{[q/p]}}\|_2 \geq \sqrt{\frac{q}{2p}}|x_{\frac{3q}{2p}}|.$$

Let $Y_i = 0$ if $i \notin T$ and $Y_i = x_i^2$ if $i \in T$. Then

$$\mathbb{E}[\sum_{i > \frac{3q}{2p}} Y_i] = p\|x_{\overline{[\frac{3q}{2p}]}}\|_2^2 \leq p\|x_{\overline{[q/p]}}\|_2^2$$

For $i > \frac{3q}{2p}$ we have

$$Y_i \leq |x_{\frac{3q}{2p}}|^2 \leq \frac{2p}{q}\|x_{\overline{[q/p]}}\|_2^2$$

giving by Chernoff that

$$\Pr[\sum Y_i \geq \frac{4}{3}p\|x_{\overline{[q/p]}}\|_2^2] \leq e^{-\Omega(q/2)}$$

But if this event does not happen, then

$$\|y_{\overline{[2q]}}\|_2^2 \leq \sum_{i \in T, i > \frac{3q}{2p}} x_i^2 = \sum_{i > \frac{3q}{2p}} Y_i \leq \frac{4}{3}p\|x_{\overline{[q/p]}}\|_2^2$$

By (7.2), using $O(2q)$-size hash tables gives a y^* with

$$\|y^* - y\|_\infty \leq \frac{1}{\sqrt{2q}}\|y_{\overline{[2q]}}\|_2 \leq \sqrt{p/q}\|x_{\overline{[q/p]}}\|_2$$

with failure probability $n^{-\Omega(1)}$, as desired. □

Let $r = 2\log 1/f$. Our algorithm is as follows: for $j \in \{0, \ldots, r\}$, we find and estimate the $2^{j/2}k$ largest elements not found in previous j in a subsampled Count-Sketch with probability $p = 2^{-j}$ and hash size $q = ck/f$ for some parameter $c = \Theta(r^2)$. We output x', the union of all these estimates. Our goal is to show

$$\|x' - x\|_1 - \|x_{\overline{[k]}}\|_1 \leq O(f^2)\|x_{\overline{[k]}}\|_1.$$

For each level j, let S_j be the $2^{j/2}k$ largest coordinates in our estimate not found in $S_1 \cup \cdots \cup S_{j-1}$. Let $S = \cup S_j$. By Lemma 7.2.4, for each j we have (with failure probability $e^{-\Omega(k/f)} + n^{-\Omega(1)}$) that

$$\begin{aligned}\|(x' - x)_{S_j}\|_1 &\leq |S_j|\sqrt{\frac{2^{-j}f}{ck}}\|x_{\overline{[2^j ck/f]}}\|_2 \\ &\leq 2^{-j/2}\sqrt{\frac{fk}{c}}\|x_{\overline{[2k/f]}}\|_2\end{aligned}$$

and so

$$\|(x'-x)_S\|_1 = \sum_{j=0}^{r} \|(x'-x)_{S_j}\|_1 \leq \frac{1}{(1-1/\sqrt{2})\sqrt{c}} \sqrt{fk} \|x_{\overline{[2k/f]}}\|_2 \tag{7.4}$$

By standard arguments, the ℓ_∞ bound for S_0 gives

$$\|x_{[k]}\|_1 \leq \|x_{S_0}\|_1 + k\|x'_{S_0} - x_{S_0}\|_\infty \leq \sqrt{fk/c} \|x_{\overline{[2k/f]}}\|_2 \tag{7.5}$$

Combining Equations (7.4) and (7.5) gives

$$\begin{aligned}
\|x'-x\|_1 - \|x_{\overline{[k]}}\|_1 &= \|(x'-x)_S\|_1 + \|x_{\overline{S}}\|_1 - \|x_{\overline{[k]}}\|_1 \\
&= \|(x'-x)_S\|_1 + \|x_{[k]}\|_1 - \|x_S\|_1 \\
&= \|(x'-x)_S\|_1 + (\|x_{[k]}\|_1 - \|x_{S_0}\|_1) - \sum_{j=1}^{r} \|x_{S_j}\|_1 \\
&\leq \left(\frac{1}{(1-1/\sqrt{2})\sqrt{c}} + \frac{1}{\sqrt{c}} \right) \sqrt{fk} \|x_{\overline{[2k/f]}}\|_2 - \sum_{j=1}^{r} \|x_{S_j}\|_1 \\
&= O(\frac{1}{\sqrt{c}}) \sqrt{fk} \|x_{\overline{[2k/f]}}\|_2 - \sum_{j=1}^{r} \|x_{S_j}\|_1
\end{aligned} \tag{7.6}$$

We would like to convert the first term to depend on the ℓ_1 norm. For any u and s we have, by splitting into chunks of size s, that

$$\|u_{\overline{[2s]}}\|_2 \leq \sqrt{\frac{1}{s}} \|u_{\overline{[s]}}\|_1$$
$$\|u_{\overline{[s]} \cap [2s]}\|_2 \leq \sqrt{s} |u_s|.$$

Along with the triangle inequality, this gives us that

$$\begin{aligned}
\sqrt{kf} \|x_{\overline{[2k/f]}}\|_2 &\leq \sqrt{kf} \|x_{\overline{[2k/f^3]}}\|_2 + \sqrt{kf} \sum_{j=1}^{r} \|x_{\overline{[2^j k/f]} \cap [2^{j+1} k/f]}\|_2 \\
&\leq f^2 \|x_{\overline{[k/f^3]}}\|_1 + \sum_{j=1}^{r} k 2^{j/2} |x_{2^j k/f}|
\end{aligned}$$

so

$$\|x'-x\|_1 - \|x_{\overline{[k]}}\|_1 \leq O(\frac{1}{\sqrt{c}}) f^2 \|x_{\overline{[k/f^3]}}\|_1 + \sum_{j=1}^{r} O(\frac{1}{\sqrt{c}}) k 2^{j/2} |x_{2^j k/f}| - \sum_{j=1}^{r} \|x_{S_j}\|_1 \tag{7.7}$$

Define $a_j = k2^{j/2}|x_{2^j k/f}|$. The first term grows as f^2 so it is fine, but a_j can grow as $f2^{j/2} > f^2$. We need to show that they are canceled by the corresponding $\|x_{S_j}\|_1$. In particular, we will show that $\|x_{S_j}\|_1 \geq \Omega(a_j) - O(2^{-j/2} f^2 \|x_{\overline{[k/f^3]}}\|_1)$ with high probability—at least wherever $a_j \geq \|a\|_1/(2r)$.

Let $U \in [r]$ be the set of j with $a_j \geq \|a\|_1/(2r)$, so that $\|a_U\|_1 \geq \|a\|_1/2$. We have

$$\|x_{\overline{[2^j k/f]}}\|_2^2 = \|x_{\overline{[2k/f^3]}}\|_2^2 + \sum_{i=j}^{r} \|x_{\overline{[2^j k/f]} \cap [2^{j+1} k/f]}\|_2^2$$
$$\leq \|x_{\overline{[2k/f^3]}}\|_2^2 + \frac{1}{kf} \sum_{i=j}^{r} a_j^2 \tag{7.8}$$

For $j \in U$, we have

$$\sum_{i=j}^{r} a_i^2 \leq a_j \|a\|_1 \leq 2r a_j^2$$

so, along with $(y^2 + z^2)^{1/2} \leq y + z$, we turn Equation (7.8) into

$$\|x_{\overline{[2^j k/f]}}\|_2 \leq \|x_{\overline{[2k/f^3]}}\|_2 + \sqrt{\frac{1}{kf} \sum_{i=j}^{r} a_j^2}$$
$$\leq \sqrt{\frac{f^3}{k}} \|x_{\overline{[k/f^3]}}\|_1 + \sqrt{\frac{2r}{kf}} a_j$$

When choosing S_j, let $T \in [n]$ be the set of indices chosen in the sample. Applying Lemma 7.2.4 the estimate x^* of x_T has

$$\|x^* - x_T\|_\infty \leq \sqrt{\frac{f}{2^j ck}} \|x_{\overline{[2^j k/f]}}\|_2$$
$$\leq \sqrt{\frac{1}{2^j c}} \frac{f^2}{k} \|x_{\overline{[k/f^3]}}\|_1 + \sqrt{\frac{2r}{2^j c}} \frac{a_j}{k}$$
$$= \sqrt{\frac{1}{2^j c}} \frac{f^2}{k} \|x_{\overline{[k/f^3]}}\|_1 + \sqrt{\frac{2r}{c}} |x_{2^j k/f}|$$

for $j \in U$.

Let $Q = [2^j k/f] \setminus (S_0 \cup \cdots \cup S_{j-1})$. We have $|Q| \geq 2^{j-1} k/f$ so $\mathbb{E}[|Q \cap T|] \geq k/2f$ and $|Q \cap T| \geq k/4f$ with failure probability $e^{-\Omega(k/f)}$. Conditioned on $|Q \cap T| \geq k/4f$, since x_T has at least $|Q \cap T| \geq k/(4f) = 2^{r/2} k/4 \geq 2^{j/2} k/4$ possible choices of value at least $|x_{2^j k/f}|$, x_{S_j} must have at least $k 2^{j/2}/4$ elements at least $|x_{2^j k/f}| - \|x^* - x_T\|_\infty$. Therefore, for $j \in U$,

$$\|x_{S_j}\|_1 \geq -\frac{1}{4\sqrt{c}} f^2 \|x_{\overline{[k/f^3]}}\|_1 + \frac{k 2^{j/2}}{4} (1 - \sqrt{\frac{2r}{c}}) |x_{2^j k/f}|$$

and therefore

$$\sum_{j=1}^{r}\|x_{S_j}\|_1 \geq \sum_{j\in U}\|x_{S_j}\|_1 \geq \sum_{j\in U} -\frac{1}{4\sqrt{c}}f^2\|x_{\overline{[k/f^3]}}\|_1 + \frac{k2^{j/2}}{4}(1-\sqrt{\frac{2r}{c}})|x_{2^jk/f}|$$
$$\geq -\frac{r}{4\sqrt{c}}f^2\|x_{\overline{[k/f^3]}}\|_1 + \frac{1}{4}(1-\sqrt{\frac{2r}{c}})\|a_U\|_1$$
$$\geq -\frac{r}{4\sqrt{c}}f^2\|x_{\overline{[k/f^3]}}\|_1 + \frac{1}{8}(1-\sqrt{\frac{2r}{c}})\sum_{j=1}^{r}k2^{j/2}|x_{2^jk/f}| \qquad (7.9)$$

Using (7.7) and (7.9) we get

$$\|x'-x\|_1 - \|x_{\overline{[k]}}\|_1 \leq \left(\frac{r}{4\sqrt{c}}+O(\frac{1}{\sqrt{c}})\right)f^2\|x_{\overline{[k/f^3]}}\|_1 + \sum_{j=1}^{r}\left(O(\frac{1}{\sqrt{c}})+\frac{1}{8}\sqrt{\frac{2r}{c}}-\frac{1}{8}\right)k2^{j/2}|x_{2^jk/f}|$$
$$\leq f^2\|x_{\overline{[k/f^3]}}\|_1 \leq f^2\|x_{\overline{[k]}}\|_1$$

for some $c \lesssim r^2$. Hence we use a total of $\frac{rc}{f}k\log n = \frac{\log^3 1/f}{f}k\log n$ measurements for $1+f^2$-approximate ℓ_1/ℓ_1 recovery.

For each $j \in \{0,\ldots,r\}$ we had failure probability $e^{-\Omega(k/f)}+n^{-\Omega(1)}$ (from Lemma 7.2.4 and $|Q\cap T| \geq k/2f$). By the union bound, our overall failure probability is at most

$$(\log\frac{1}{f})(e^{-\Omega(k/f)}+n^{-\Omega(1)}) \leq e^{-\Omega(k/f)}+n^{-\Omega(1)},$$

proving Theorem 7.2.3.

7.3 Adaptive ℓ_1 lower bound

This section proves the following theorem:

Theorem 7.3.1. *Any, possibly adaptive, $(1+\epsilon)$-approximate ℓ_1/ℓ_1 recovery scheme with sufficiently small constant failure probability δ must make $\Omega(\frac{1}{\sqrt{\epsilon}}k/\log(k/\epsilon))$ measurements.*

Setting. One would like to estimate a vector $x \in \mathbb{R}^n$ from m linear measurements $A_1x,\ldots,A_mx$. One may choose each vector A_i based on $A_1x,\ldots,A_{i-1}x$, and must output x' satisfying

$$\|x'-x\|_1 \leq O(1)\cdot\min_{k\text{-sparse } y}\|x-y\|_1$$

We will show that $\widetilde{\Omega}(k/\sqrt{\epsilon})$ adaptive measurements are necessary, which shows that our $\widetilde{O}(\frac{1}{\sqrt{\epsilon}}k\log n)$ nonadaptive upper bound (Lemma 7.2.3) is tight up to logarithmic factors—even in the adaptive setting.

A recent lower bound showed that $\Omega(k/\epsilon)$ measurements are necessary in the ℓ_2 setting [ACD11]. Our result can be seen as the ℓ_1 analog of this result. Their techniques rely on special properties of the 2-norm; namely, that it is a rotationally invariant inner product space and that the Gaussian is both 2-stable and a maximum entropy distribution. Such techniques do not seem useful for proving lower bounds for ℓ_1.

Overview of section. In Section 7.3.1 we show how to relate a lower bound on bit complexity to a lower bound on measurement complexity, with only a $\log n$ loss. In Section 7.3.2, we show that recovering the locations of an ϵ fraction of d ones in a vector of size $n > d/\epsilon$ requires $\widetilde{\Omega}(\epsilon d)$ bits. In Section 7.3.3 we establish a new lower bound on the communication complexity of a two-party communication problem that we call $\mathsf{Multi}\ell_\infty$. In Section 7.3.4 we show that successful sparse recovery must solve one of the previous problems, giving a lower bound in bit complexity and hence measurement complexity.

Section 7.3.5 is essentially an appendix to Section 7.3.3, showing that the $\mathsf{Gap}\ell_\infty$ problem is hard on a particular distribution.

7.3.1 Bit complexity to measurement complexity

This section describes how a lower bound on bit complexity implies a lower bound on the number of measurements.

Let $X \subset \mathbb{R}^n$ be a distribution with $x_i \in \{-n^d, \ldots, n^d\}$ for all $i \in [n]$ and $x \in X$. Here $d = \Theta(1)$ is a parameter. Given an adaptive compressed sensing scheme $\mathcal{A}$, we define a $(1+\epsilon)$-approximate ℓ_1/ℓ_1 sparse recovery *multiround bit scheme* on X as follows.

Let A^i be the i-th (possibly adaptively chosen) measurement matrix of the compressed sensing scheme. We may assume that the union of rows in matrices $A^1, \ldots, A^r$ generated by $\mathcal{A}$ is an orthonormal system, since the rows can be orthogonalized in a post-processing step. We can assume that $r \leq n$.

Choose a random $u \in \mathbb{R}^n$ from the distribution $\mathcal{N}(0, \frac{1}{n^c} \cdot I_{n\times n})$, where $c = \Theta(1)$ is a parameter. We require that the compressed sensing scheme outputs a valid result of $(1+\epsilon)$-approximate recovery on $x+u$ with probability at least $1-\delta$, over the choice of u and its random coins. By Yao's minimax principle, we can fix the randomness of the compressed sensing scheme and assume that the scheme is deterministic.

Let B^1 be the matrix A^1 with entries rounded to $t \log n$ bits for a parameter $t = \Theta(1)$. We compute $B^1 x$. Then, we compute $B^1 x + A^1 u$. From this, we compute A^2, using the algorithm specified by $\mathcal{A}$ as if $B^1 x + A^1 u$ were equal to $A^1 x'$ for some x'. For this, we use the following lemma, which is Lemma 5.1 of [DIPW10].

Lemma 7.3.2. *Consider any $m \times n$ matrix A with orthonormal rows. Let B 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 $Bv = A(v-s)$ and $\|s\|_1 < n^2 2^{-b} \|v\|_1$.*

In general for $i \geq 2$, given $B^1 x + A^1 u, B^2 x + A^2 u, \ldots, B^{i-1} x + A^{i-1} u$ we compute A^i, and round to $t \log n$ bits per entry to get B^i. The output of the multiround bit scheme is the same as that of the compressed sensing scheme. If the compressed sensing scheme uses r rounds, then the multiround bit scheme uses r rounds. Let b denote the total number of bits in the concatenation of discrete vectors $B^1 x, B^2 x, \ldots, B^r x$.

Lemma 7.3.3. *For $t = O(1+c+d)$, a lower bound of $\Omega(b)$ bits for a multiround bit scheme with error probability at most $\delta + 1/n$ implies a lower bound of $\Omega(b/((1+c+d)\log n))$ measurements for $(1+\epsilon)$-approximate sparse recovery schemes with failure probability at most δ.*

Proof. Let $\mathcal{A}$ be a $(1+\epsilon)$-approximate adaptive compressed sensing scheme with failure probability δ. We will show that the associated multiround bit scheme has failure probability $\delta + 1/n$.

By Lemma 7.3.2, for any vector $x \in \{-n^d, \ldots, n^d\}$ we have $B^1 x = A^1(x+s)$ for a vector s with $\|s\|_1 \leq n^2 2^{-t \log n} \|x\|_1$, so $\|s\|_2 \leq n^{2.5-t} \|x\|_2 \leq n^{3.5+d-t}$. Notice that $u + s \sim \mathcal{N}(s, \frac{1}{n^c} \cdot I_{n\times n})$. We

use the following quick suboptimal upper bound on the statistical distance between two univariate normal distributions, which suffices for our purposes.

Fact 7.3.4. *(see section 3 of [Pol05]) The variation distance between $\mathcal{N}(\theta_1, 1)$ and $\mathcal{N}(\theta_2, 1)$ is $\frac{4\tau}{\sqrt{2\pi}} + O(\tau^2)$, where $\tau = |\theta_1 - \theta_2|/2$.*

It follows by Fact 7.3.4 and independence across coordinates, that the variation distance between $\mathcal{N}(0, \frac{1}{n^c} \cdot I_{n\times n})$ and $\mathcal{N}(s, \frac{1}{n^c} \cdot I_{n\times n})$ is the same as that between $\mathcal{N}(0, I_{n\times n})$ and $\mathcal{N}(s \cdot n^{c/2}, I_{n\times n})$, which can be upper-bounded as

$$\begin{aligned}\sum_{i=1}^{n} \cdot \frac{2n^{c/2}|s_i|}{\sqrt{2\pi}} + O(n^c s_i^2) &= O(n^{c/2}\|s\|_1 + n^c\|s\|_2^2) \\ &= O(n^{c/2} \cdot \sqrt{n}\|s\|_2 + n^c\|s\|_2^2) \\ &= O(n^{c/2+4+d-t} + n^{c+7+2d-2t}).\end{aligned}$$

It follows that for $t = O(1 + c + d)$, the variation distance is at most $1/n^2$.

Therefore, if $\mathcal{T}^1$ is the algorithm which takes $A^1(x+u)$ and produces A^2, then $\mathcal{T}^1(A^1(x+u)) = \mathcal{T}^1(B^1x + A^1u)$ with probability at least $1 - 1/n^2$. This follows since $B^1x + A^1u = A^1(x + u + s)$ and $u + s$ and u have variation distance at most $1/n^2$.

In the second round, $B^2x + A^2u$ is obtained, and importantly we have for the algorithm $\mathcal{T}^2$ in the second round, $\mathcal{T}^2(A^2(x+u)) = \mathcal{T}^2(B^2x+A^2u)$ with probability at least $1-1/n^2$. This follows since A^2 is a deterministic function of A^1u, and A^1u and A^2u are independent since A^1 and A^2 are orthonormal while u is a vector of i.i.d. Gaussians (here we use the rotational invariance / symmetry of Gaussian space). It follows by induction that with probability at least $1 - r/n^2 \geq 1 - 1/n$, the output of the multiround bit scheme agrees with that of $\mathcal{A}$ on input $x + u$.

Hence, if m_i is the number of measurements in round i, and $m = \sum_{i=1}^{r} m_i$, then we have a multiround bit scheme using a total of $b = mt \log n = O(m(1 + c + d) \log n)$ bits and with failure probability $\delta + 1/n$. □

7.3.2 Information lower bound for recovering noise bits

Definition 7.3.5. *We say a set $C \subset [q]^d$ is a (d, q, ϵ) code if any two distinct $c, c' \in C$ agree in at most ϵd positions. We say a set $X \subset \{0,1\}^{dq}$ represents C if X is C concatenated with the trivial code $[q] \to \{0,1\}^q$ given by $i \to e_i$.*

Claim 7.3.6. *For $\epsilon \geq 2/q$, there exist (d, q, ϵ) codes C of size $q^{\Omega(\epsilon d)}$ by the Gilbert-Varshamov bound (details in [DIPW10]).*

Lemma 7.3.7. *Let $X \subset \{0,1\}^{dq}$ represent a (d, q, ϵ) code. Suppose $y \in \mathbb{R}^{dq}$ satisfies $\|y - x\|_1 \leq (1 - \epsilon)\|x\|_1$ for some $x \in X$. Then we can recover x uniquely from y.*

Proof. We assume $y_i \in [0, 1]$ for all i; thresholding otherwise decreases $\|y - x\|_1$. We will show that there exists no other $x' \in X$ with $\|y - x\|_1 \leq (1 - \epsilon)\|x\|_1$; thus choosing the nearest element of X is a unique decoder. Suppose otherwise, and let $S = \text{supp}(x), T = \text{supp}(x')$. Then

$$\begin{aligned}(1 - \epsilon)\|x\|_1 &\geq \|x - y\|_1 \\ &= \|x\|_1 - \|y_S\|_1 + \|y_{\overline{S}}\|_1 \\ \|y_S\|_1 &\geq \|y_{\overline{S}}\|_1 + \epsilon d\end{aligned}$$

Since the same is true relative to x' and T, we have

$$\begin{aligned}\|y_S\|_1 + \|y_T\|_1 &\geq \|y_{\overline{S}}\|_1 + \|y_{\overline{T}}\|_1 + 2\epsilon d \\ 2\|y_{S\cap T}\|_1 &\geq 2\|y_{\overline{S\cup T}}\|_1 + 2\epsilon d \\ \|y_{S\cap T}\|_1 &\geq \epsilon d \\ |S \cap T| &\geq \epsilon d\end{aligned}$$

This violates the distance of the code represented by X. □

Lemma 7.3.8. *Let $R = [s, cs]$ for some constant c and parameter s. Let X be a permutation independent distribution over $\{0,1\}^n$ with $\|x\|_1 \in R$ with probability p. If y satisfies $\|x - y\|_1 \leq (1-\epsilon)\|x\|_1$ with probability p' with $p' - (1-p) = \Omega(1)$, then $I(x;y) = \Omega(\epsilon s \log(n/s))$.*

Proof. For each integer $i \in R$, let $X_i \subset \{0,1\}^n$ represent an $(i, n/i, \epsilon)$ code. Let $p_i = \Pr_{x\in X}[\|x\|_1 = i]$. Let S_n be the set of permutations of $[n]$. Then the distribution X' given by (a) choosing $i \in R$ proportional to p_i, (b) choosing $\sigma \in S_n$ uniformly, (c) choosing $x_i \in X_i$ uniformly, and (d) outputting $x' = \sigma(x_i)$ is equal to the distribution $(x \in X \mid \|x\|_1 \in R)$.

Now, because $p' \geq \Pr[\|x\|_1 \notin R] + \Omega(1)$, x' chosen from X' satisfies $\|x' - y\|_1 \leq (1-\epsilon)\|x'\|_1$ with $\delta \geq p' - (1-p)$ probability. Therefore, with at least $\delta/2$ probability, i and σ are such that $\|\sigma(x_i) - y\|_1 \leq (1-\epsilon)\|\sigma(x_i)\|_1$ with $\delta/2$ probability over uniform $x_i \in X_i$. But given y with $\|y - \sigma(x_i)\|_1$ small, we can compute $y' = \sigma^{-1}(y)$ with $\|y' - x_i\|_1$ equally small. Then by Lemma 7.3.7 we can recover x_i from y with probability $\delta/2$ over $x_i \in X_i$. Thus for this i and σ, $I(x; y \mid i, \sigma) \geq \Omega(\log |X_i|) = \Omega(\delta\epsilon s \log(n/s))$ by Fano's inequality. But then $I(x;y) = \mathbb{E}_{i,\sigma}[I(x; y \mid i, \sigma)] = \Omega(\delta^2 \epsilon s \log(n/s)) = \Omega(\epsilon s \log(n/s))$. □

7.3.3 Direct sum for distributional ℓ_∞

This section shows that the multiround communication complexity of r instances of the $\mathsf{Gap}\ell_\infty$ problem is r times that of one instance.

We assume basic familiarity with communication complexity; see the textbook of Kushilevitz and Nisan [KN97] for further background. Our reason for using communication complexity is to prove lower bounds, and we will do so by using information-theoretic arguments. We refer the reader to the thesis of Bar-Yossef [Bar02] for a comprehensive introduction to information-theoretic arguments used in communication complexity.

We consider two-party randomized communication complexity. There are two parties, Alice and Bob, with input vectors x and y respectively, and their goal is to solve a promise problem $f(x,y)$. The parties have private randomness. The communication cost of a protocol is its maximum transcript length, over all possible inputs and random coin tosses. The randomized communication complexity $R_\delta(f)$ is the minimum communication cost of a randomized protocol Π which for every input (x,y) outputs $f(x,y)$ with probability at least $1-\delta$ (over the random coin tosses of the parties). We also study the distributional complexity of f, in which the parties are deterministic and the inputs (x,y) are drawn from distribution μ, and a protocol is correct if it succeeds with probability at least $1-\delta$ in outputting $f(x,y)$, where the probability is now taken over $(x,y) \sim \mu$. We define $D_{\mu,\delta}(f)$ to be the minimum communication cost of a correct protocol Π.

We consider the following promise problem $\mathsf{Gap}\ell_\infty^B$, where B is a parameter, which was studied in [SS02, BJKS04]. The inputs are pairs (x,y) of m-dimensional vectors, with $x_i, y_i \in \{0,1,2,\ldots,B\}$ for all $i \in [m]$, with the promise that (x,y) is one of the following types of instance:

- NO instance: for all i, $|x_i - y_i| \in \{0,1\}$, or

- YES instance: there is a unique i for which $|x_i - y_i| = B$, and for all $j \neq i$, $|x_j - y_j| \in \{0, 1\}$.

The goal of a protocol is to decide which of the two cases (NO or YES) the input is in.

Consider the distribution σ: for each $j \in [m]$, choose a random pair $(Z_j, P_j) \in \{0, 1, 2, \ldots, B\} \times \{0, 1\} \setminus \{(0, 1), (B, 0)\}$. If $(Z_j, P_j) = (z, 0)$, then $X_j = z$ and Y_j is uniformly distributed in $\{z, z+1\}$; if $(Z_j, P_j) = (z, 1)$, then $Y_j = z$ and X_j is uniformly distributed on $\{z-1, z\}$. Let $Z = (Z_1, \ldots, Z_m)$ and $P = (P_1, \ldots, P_m)$. Next choose a random coordinate $S \in [m]$. For coordinate S, replace (X_S, Y_S) with a uniform element of $\{(0, 0), (0, B)\}$. Let $X = (X_1, \ldots, X_m)$ and $Y = (Y_1, \ldots, Y_m)$.

Using similar arguments to those in [BJKS04], we can show that there are positive, sufficiently small constants δ_0 and C so that for any randomized protocol Π which succeeds with probability at least $1 - \delta_0$ on distribution σ,

$$I(X, Y; \Pi | Z, P) \geq \frac{Cm}{B^2}, \tag{7.10}$$

where, with some abuse of notation, Π is also used to denote the transcript of the corresponding randomized protocol, and here the input (X, Y) is drawn from σ conditioned on (X, Y) being a NO instance. Here, Π is randomized, and succeeds with probability at least $1 - \delta_0$, where the probability is over the joint space of the random coins of Π and the input distribution.

Our starting point for proving (7.10) is Jayram's lower bound for the conditional mutual information when the inputs are drawn from a related distribution (reference [70] on p.182 of [Bar02]), but we require several nontrivial modifications to his argument in order to apply it to bound the conditional mutual information for our input distribution, which is σ conditioned on (X, Y) being a NO instance. Essentially, we are able to show that the variation distance between our distribution and his distribution is small, and use this to bound the difference in the conditional mutual information between the two distributions. The proof is rather technical, and we postpone it to Section 7.3.5.

We make a few simple refinements to (7.10). Define the random variable W which is 1 if (X, Y) is a YES instance, and 0 if (X, Y) is a NO instance. Then by definition of the mutual information, if (X, Y) is drawn from σ without conditioning on (X, Y) being a NO instance, then we have

$$\begin{aligned} I(X, Y; \Pi | W, Z, P) &\geq \frac{1}{2} I(X, Y; \Pi | Z, P, W = 0) \\ &= \Omega(m/B^2). \end{aligned}$$

Observe that

$$I(X, Y; \Pi | S, W, Z, P) \geq I(X, Y; \Pi | W, Z, P) - H(S) = \Omega(m/B^2), \tag{7.11}$$

where we assume that $\Omega(m/B^2) - \log m = \Omega(m/B^2)$. Define the constant $\delta_1 = \delta_0/4$. We now define a problem which involves solving r copies of $\mathsf{Gap}\ell_\infty^B$.

Definition 7.3.9 ($\mathsf{Multi}\ell_\infty^{r,B}$ Problem)**.** *There are r pairs of inputs $(x^1, y^1), (x^2, y^2), \ldots, (x^r, y^r)$ such that each pair (x^i, y^i) is a legal instance of the $\mathsf{Gap}\ell_\infty^B$ problem. Alice is given $x^1, \ldots, x^r$. Bob is given $y^1, \ldots, y^r$. The goal is to output a vector $v \in \{NO, YES\}^r$, so that for at least a $1 - \delta_1$ fraction of the entries i, $v_i = \mathsf{Gap}\ell_\infty^B(x^i, y^i)$.*

Remark. *Notice that Definition 7.3.9 is defining a promise problem. We will study the distributional complexity of this problem under the distribution σ^r, which is a product distribution on the r instances $(x^1, y^1), (x^2, y^2), \ldots, (x^r, y^r)$.*

Theorem 7.3.10. $D_{\sigma^r,\delta_1}(\mathsf{Multi}\ell_\infty^{r,B}) = \Omega(rm/B^2)$.

Proof. Let Π be any deterministic protocol for $\mathsf{Multi}\ell_\infty^{r,B}$ which succeeds with probability at least $1-\delta_1$ in solving $\mathsf{Multi}\ell_\infty^{r,B}$ when the inputs are drawn from σ^r, where the probability is taken over the input distribution. We show that Π has communication cost $\Omega(rm/B^2)$.

Let $X^1, Y^1, S^1, W^1, Z^1, P^1 \ldots, X^r, Y^r, S^r, W^r, Z^r$, and P^r be the random variables associated with σ^r, i.e., X^j, Y^j, S^j, W^j, P^j and Z^j correspond to the random variables X, Y, S, W, Z, P associated with the j-th independent instance drawn according to σ, defined above. We let $X = (X^1, \ldots, X^r)$, $X^{<j} = (X^1, \ldots, X^{j-1})$, and X^{-j} equal X without X^j. Similarly we define these vectors for Y, S, W, Z and P.

By the chain rule for mutual information, $I(X^1, \ldots, X^r, Y^1, \ldots, Y^r; \Pi | S, W, Z, P)$ is equal to $\sum_{j=1}^r I(X^j, Y^j; \Pi | X^{<j}, Y^{<j}, S, W, Z, P)$. Let V be the output of Π, and V_j be its j-th coordinate. For a value $j \in [r]$, we say that j is *good* if $\Pr_{X,Y}[V_j = \mathsf{Gap}\ell_\infty^B(X^j, Y^j)] \geq 1 - \frac{2\delta_0}{3}$. Since Π succeeds with probability at least $1 - \delta_1 = 1 - \delta_0/4$ in outputting a vector with at least a $1 - \delta_0/4$ fraction of correct entries, the expected probability of success over a random $j \in [r]$ is at least $1 - \delta_0/2$, and so by a Markov argument, there are $\Omega(r)$ good indices j.

Fix a value of $j \in [r]$ that is good, and consider $I(X^j, Y^j; \Pi | X^{<j}, Y^{<j}, S, W, Z, P)$. By expanding the conditioning, $I(X^j, Y^j; \Pi | X^{<j}, Y^{<j}, S, W, Z, P)$ is equal to

$$\mathbf{E}_{x,y,s,w,z,p}[I(X^j, Y^j; \Pi \mid (X^{<j}, Y^{<j}, S^{-j}, W^{-j}, Z^{-j}, P^{-j}) = (x, y, s, w, z, p), S^j, W^j, Z^j, P^j)]. \quad (7.12)$$

For each x, y, s, w, z, p, define a randomized protocol $\Pi_{x,y,s,w,z,p}$ for $\mathsf{Gap}\ell_\infty^B$ under distribution σ. Suppose that Alice is given a and Bob is given b, where $(a, b) \sim \sigma$. Alice sets $X^j = a$, while Bob sets $Y^j = b$. Alice and Bob use x, y, s, w, z and p to set their remaining inputs as follows. Alice sets $X^{<j} = x$ and Bob sets $Y^{<j} = y$. Alice and Bob can randomly set their remaining inputs without any communication, since for $j' > j$, conditioned on $S^{j'}, W^{j'}, Z^{j'}$, and $P^{j'}$, Alice and Bob's inputs are independent. Alice and Bob run Π on inputs X, Y, and define $\Pi_{x,y,s,w,z,p}(a, b) = V_j$. We say a tuple (x, y, s, w, z, p) is *good* if

$$\Pr_{X,Y}[V_j = \mathsf{Gap}\ell_\infty^B(X^j, Y^j) \ \mid \ X^{<j} = x, Y^{<j} = y, S^{-j} = s, W^{-j} = w, Z^{-j} = z, P^{-j} = p] \geq 1 - \delta_0.$$

By a Markov argument, and using that j is good, we have $\Pr_{x,y,s,w,z,p}[(x, y, s, w, z, p) \text{ is good }] = \Omega(1)$. Plugging into (7.12), $I(X^j, Y^j; \Pi | X^{<j}, Y^{<j}, S, W, Z, P)$ is at least a constant times

$$\mathbf{E}_{x,y,s,w,z,p}[I(X^jY^j; \Pi | (X^{<j}, Y^{<j}, S^{-j}, W^{-j}, Z^{-j}, P^{-j}) = (x, y, s, w, z, p), \\ S^j, W^j, Z^j, P^j, (x, y, s, w, z, p) \text{ is good})].$$

For any (x, y, s, w, z, p) that is good, $\Pi_{x,y,s,w,z,p}(a, b) = V_j$ with probability at least $1 - \delta_0$, over the joint distribution of the randomness of $\Pi_{x,y,s,w,z,p}$ and $(a, b) \sim \sigma$. By (7.11),

$$\mathbf{E}_{x,y,s,w,z,p}[I(X^j, Y^j; \Pi | (X^{<j}, Y^{<j}, S^{-j}, W^{-j}, Z^{-j}, P^{-j}) = (x, y, s, w, z, p), \\ S^j, W^j, Z^j, P^j, (x, y, s, w, z, p) \text{ is good}] = \Omega\left(\frac{m}{B^2}\right).$$

Since there are $\Omega(r)$ good indices j, we have $I(X^1, \ldots, X^r; \Pi | S, W, Z, P) = \Omega(mr/B^2)$. Since the distributional complexity $D_{\sigma^r,\delta_1}(\mathsf{Multi}\ell_\infty^{r,B})$ is at least the minimum of $I(X^1, \ldots, X^r; \Pi | S, W, Z, P)$ over deterministic protocols Π which succeed with probability at least $1 - \delta_1$ on input distribution σ^r, it follows that $D_{\sigma^r,\delta_1}(\mathsf{Multi}\ell_\infty^{r,B}) = \Omega(mr/B^2)$. $\square$

7.3.4 The overall lower bound

This section gives a lower bound for multiround bit schemes.

Fix parameters $B = \Theta(1/\epsilon^{1/2})$, $r = k$, $m = 1/\epsilon^{3/2}$, and $n = k/\epsilon^3$. Given an instance $(x^1, y^1), \ldots, (x^r, y^r)$ of $\mathsf{Multi}\ell_\infty^{r,B}$ we define the input signal z to a sparse recovery problem. We allocate a set S^i of m disjoint coordinates in a universe of size n for each pair (x^i, y^i), and on these coordinates place the vector $y^i - x^i$. The locations turn out to be essential for the proof of Lemma 7.3.12 below, and are placed uniformly at random among the n total coordinates (subject to the constraint that the S^i are disjoint). Let ρ be the induced distribution on z.

Fix a $(1+\epsilon)$-approximate k-sparse recovery multiround bit scheme Alg that uses b bits and succeeds with probability at least $1 - \delta_1/4$ over $z \sim \rho$. Let S be the set of top k coordinates in z. Alg has the guarantee that if it succeeds for $z \sim \rho$, then there exists a small u with $\|u\|_1 < n^{-2}$ so that $w = Alg(z)$ satisfies

$$\begin{aligned} \|w - z - u\|_1 &\leq (1+\epsilon)\|(z+u)_{[n]\setminus S}\|_1 \\ \|w - z\|_1 &\leq (1+\epsilon)\|z_{[n]\setminus S}\|_1 + (2+\epsilon)/n^2 \\ &\leq (1+2\epsilon)\|z_{[n]\setminus S}\|_1 \end{aligned}$$

and thus

$$\|(w-z)_S\|_1 + \|(w-z)_{[n]\setminus S}\|_1 \leq (1+2\epsilon)\|z_{[n]\setminus S}\|_1. \tag{7.13}$$

We will show that satisfying (7.13) with $1 - \delta_1/4$ probability requires $b = \Omega(k/\epsilon^{1/2})$.

Lemma 7.3.11. *For $B = \Theta(1/\epsilon^{1/2})$ sufficiently large, suppose that*

$$\Pr_{z\sim\rho}[\|(w-z)_S\|_1 \leq 10\epsilon \cdot \|z_{[n]\setminus S}\|_1] \geq 1 - \frac{\delta_1}{2}.$$

Then Alg requires $b = \Omega(k/\epsilon^{1/2})$.

Proof. We show how to use Alg to solve instances of $\mathsf{Multi}\ell_\infty^{r,B}$ with probability at least $1 - \delta_1$, where the probability is over input instances to $\mathsf{Multi}\ell_\infty^{r,B}$ distributed according to σ^r, inducing the distribution ρ on z. The lower bound will follow by Theorem 7.3.10. Let w be the output of Alg.

Given $x^1, \ldots, x^r$, Alice places $-x^i$ on the appropriate coordinates in the set S^i used in defining z, obtaining a vector z_{Alice}. Given $y^1, \ldots, y^r$, Bob places the y^i on the appropriate coordinates in S^i. He thus creates a vector z_{Bob} for which $z_{Alice} + z_{Bob} = z$. In round i, Alice transmits $B^i z_{Alice}$ to Bob, who computes $B^i(z_{Alice} + z_{Bob})$ and transmits it back to Alice. Alice can then compute $B^i(z) + A^i(u)$ for a random $u \sim \mathcal{N}(0, \frac{1}{n^c} \cdot I_{n\times n})$. We can assume all coordinates of the output vector w are in the real interval $[0, B]$, since rounding the coordinates to this interval can only decrease the error.

For each i we say that S^i is *bad* if either

- there is no coordinate j in S^i for which $|w_j| \geq \frac{B}{2}$ yet (x^i, y^i) is a YES instance of $\mathsf{Gap}\ell_\infty^B$, or
- there is a coordinate j in S^i for which $|w_j| \geq \frac{B}{2}$ yet either (x^i, y^i) is a NO instance of $\mathsf{Gap}\ell_\infty^B$ or j is not the unique j^* for which $y^i_{j^*} - x^i_{j^*} = B$.

The ℓ_1-error incurred by a bad block is at least $B/2 - 1$. Hence, if there are t bad blocks, the total error is at least $t(B/2-1)$, which must be smaller than $10\epsilon \cdot \|z_{[n]\setminus S}\|_1$ with probability $1 - \delta$. Suppose this happens.

We bound t. All coordinates in $z_{[n]\setminus S}$ have value in the set $\{0,1\}$. Hence, $\|z_{[n]\setminus S}\|_1 < rm$. So $t \leq 20\epsilon rm/(B-2)$. For $B \geq 6$, $t \leq 30\epsilon rm/B$. Plugging in r, m and B, $t \leq Ck$, where $C > 0$ is a constant that can be made arbitrarily small by increasing $B = \Theta(1/\epsilon^{1/2})$.

Here we choose $C = \delta_1$. We also condition on $\|u\|_2 \leq n^{-c}$ for a sufficiently large constant $c > 0$, which occurs with probability at least $1 - 1/n$. Hence, with probability at least $1 - \delta_1/2 - 1/n > 1 - \delta_1$, we have a $1 - \delta_1$ fraction of indices i for which the following algorithm correctly outputs $\mathsf{Gap}\ell_\infty(x^i, y^i)$: if there is a $j \in S^i$ for which $|w_j| \geq B/2$, output YES, otherwise output NO. It follows by Theorem 7.3.10 that Alg requires $b = \Omega(k/\epsilon^{1/2})$, independent of the number of rounds. □

Lemma 7.3.12. *Suppose* $\Pr_{z\sim\rho}[\|(w-z)_{[n]\setminus S}\|_1 \leq (1-8\epsilon)\cdot\|z_{[n]\setminus S}\|_1] \geq \delta_1/4$. *Then Alg requires* $b = \Omega(k\log(1/\epsilon)/\epsilon^{1/2})$.

Proof. By Lemma 7.3.8, we have $I(w;z) = \Omega(\epsilon mr\log(n/(mr)))$, which implies that $b = \Omega(\epsilon mr\log(n/(mr)))$, independent of the number r of rounds used by Alg, since the only information about the signal is in the concatenation of $B^1z, \ldots, B^rz$. □

Finally, we combine our Lemma 7.3.11 and Lemma 7.3.12.

Theorem 7.3.1. *Any, possibly adaptive,* $(1+\epsilon)$*-approximate* ℓ_1/ℓ_1 *recovery scheme with sufficiently small constant failure probability* δ *must make* $\Omega(\frac{1}{\sqrt{\epsilon}}k/\log(k/\epsilon))$ *measurements.*

Proof. We will lower bound the number of bits used by any ℓ_1/ℓ_1 multiround bit scheme Alg. In order to satisfy (7.13), we must either have $\|(w-z)_S\|_1 \leq 10\epsilon\cdot\|z_{[n]\setminus S}\|_1$ or $\|(w-z)_{[n]\setminus S}\|_1 \leq (1-8\epsilon)\|z_{[n]\setminus S}\|_1$. Since Alg succeeds with probability at least $1-\delta_1/4$, it must either satisfy the hypothesis of Lemma 7.3.11 or Lemma 7.3.12. But by these two lemmas, it follows that $b = \Omega(k/\epsilon^{1/2})$. Therefore by Lemma 7.3.3, any $(1+\epsilon)$-approximate ℓ_1/ℓ_1 sparse recovery algorithm succeeding with probability at least $1 - \delta_1/4 + 1/n = 1 - \Omega(1)$ requires $\Omega(k/(\epsilon^{1/2}\cdot\log(k/\epsilon)))$ measurements. □

7.3.5 Switching distributions from Jayram's distributional bound

We first sketch a proof of Jayram's lower bound on the distributional complexity of $\mathsf{Gap}\ell_\infty^B$ [Jay02], then change it to a different distribution that we need for our sparse recovery lower bounds. Let $X, Y \in \{0,1,\ldots,B\}^m$. Define distribution $\mu^{m,B}$ as follows: for each $j \in [m]$, choose a random pair $(Z_j, P_j) \in \{0,1,2,\ldots,B\}\times\{0,1\}\setminus\{(0,1),(B,0)\}$. If $(Z_j,P_j) = (z,0)$, then $X_j = z$ and Y_j is uniformly distributed in $\{z, z+1\}$; if $(Z_j,P_j) = (z,1)$, then $Y_j = z$ and X_j is uniformly distributed on $\{z-1, z\}$. Let $X = (X_1,\ldots,X_m)$, $Y = (Y_1,\ldots,Y_m)$, $Z = (Z_1,\ldots,Z_m)$ and $P = (P_1,\ldots,P_m)$.

The other distribution we define is $\sigma^{m,B}$, which is the same as distribution σ in Section 7.3.3 (we include m and B in the notation here for clarity). This is defined by first drawing X and Y according to distribution $\mu^{m,B}$. Then, we pick a random coordinate $S \in [m]$ and replace (X_S, Y_S) with a uniformly random element in the set $\{(0,0),(0,B)\}$.

Let Π be a deterministic protocol that errs with probability at most δ on input distribution $\sigma^{m,B}$.

By the chain rule for mutual information, when X and Y are distributed according to $\mu^{m,B}$,

$$I(X,Y;\Pi|Z,P) \;=\; \sum_{j=1}^{m} I(X_j,Y_j;\Pi|X^{<j},Y^{<j},Z,P),$$

which is equal to

$$\sum_{j=1}^{m} \mathbf{E}_{x,y,z,p}[I(X_j, Y_j; \Pi \mid Z_j, P_j, X^{<j} = x, Y^{<j} = y, Z^{-j} = z, P^{-j} = p)].$$

Say that an index $j \in [m]$ is *good* if conditioned on $S = j$, Π succeeds on $\sigma^{m,B}$ with probability at least $1 - 2\delta$. By a Markov argument, at least $m/2$ of the indices j are good. Fix a good index j.

We say that the tuple (x, y, z, p) is *good* if conditioned on $S = j$, $X^{<j} = x$, $Y^{<j} = y$, $Z^{-j} = z$, and $P^{-j} = p$, Π succeeds on $\sigma^{m,B}$ with probability at least $1 - 4\delta$. By a Markov bound, with probability at least $1/2$, (x, y, z, p) is good. Fix a good (x, y, z, p).

We can define a single-coordinate protocol $\Pi_{x,y,z,p,j}$ as follows. The parties use x and y to fill in their input vectors X and Y for coordinates $j' < j$. They also use $Z^{-j} = z$, $P^{-j} = p$, and private randomness to fill in their inputs without any communication on the remaining coordinates $j' > j$. They place their single-coordinate input (U, V) on their j-th coordinate. The parties then output whatever Π outputs.

It follows that $\Pi_{x,y,z,p,j}$ is a single-coordinate protocol Π' which distinguishes $(0, 0)$ from $(0, B)$ under the uniform distribution with probability at least $1-4\delta$. For the single-coordinate problem, we need to bound $I(X_j, Y_j; \Pi' | Z_j, P_j)$ when (X_j, Y_j) is uniformly random from the set $\{(Z_j, Z_j), (Z_j, Z_j + 1)\}$ if $P_j = 0$, and (X_j, Y_j) is uniformly random from the set $\{(Z_j, Z_j), (Z_j - 1, Z_j)\}$ if $P_j = 1$. By the same argument as in the proof of Lemma 8.2 of [BJKS04], if $\Pi'_{u,v}$ denotes the distribution on transcripts induced by inputs u and v and private coins, then we have

$$I(X_j, Y_j; \Pi' | Z_j, P_j) \geq \Omega(1/B^2) \cdot (h^2(\Pi'_{0,0}, \Pi'_{0,B}) + h^2(\Pi'_{B,0}, \Pi'_{B,B})), \tag{7.14}$$

where

$$h(\alpha, \beta) = \sqrt{\frac{1}{2} \sum_{\omega \in \Omega} (\sqrt{\alpha(\omega)} - \sqrt{\beta(\omega)})^2}$$

is the Hellinger distance between distributions α and β on support Ω. For any two distributions α and β, if we define

$$D_{TV}(\alpha, \beta) = \frac{1}{2} \sum_{\omega \in \Omega} |\alpha(\omega) - \beta(\omega)|$$

to be the variation distance between them, then $\sqrt{2} \cdot h(\alpha, \beta) \geq D_{TV}(\alpha, \beta)$ (see Proposition 2.38 of [Bar02]).

Finally, since Π' succeeds with probability at least $1 - 4\delta$ on the uniform distribution on input pair in $\{(0, 0), (0, B)\}$, we have

$$\sqrt{2} \cdot h(\Pi'_{0,0}, \Pi'_{0,B}) \geq D_{TV}(\Pi'_{0,0}, \Pi'_{0,B}) = \Omega(1).$$

Hence,

$$\begin{aligned} I(X_j, Y_j; \Pi | Z_j, P_j, X^{<j} = x, Y^{<j} = y, Z^{-j} = z, P^{-j} = p) \\ = \Omega(1/B^2) \end{aligned}$$

for each of the $\Omega(m)$ good j. Thus $I(X, Y; \Pi | Z, P) = \Omega(m/B^2)$ when inputs X and Y are distributed according to $\mu^{m,B}$, and Π succeeds with probability at least $1 - \delta$ on X and Y distributed according to $\sigma^{m,B}$.

Changing the distribution. Consider the distribution

$$\zeta^{m,B} = (\sigma^{m,B} \mid (X_S, Y_S) = (0,0)).$$

We show $I(X, Y; \Pi|Z) = \Omega(m/B^2)$ when X and Y are distributed according to $\zeta^{m,B}$ rather than according to $\mu^{m,B}$.

For X and Y distributed according to $\zeta^{m,B}$, by the chain rule we again have that $I(X, Y; \Pi|Z, P)$ is equal to

$$\sum_{j=1}^{m} \mathbf{E}_{x,y,z,p}[I(X_j, Y_j; \Pi|Z_j, P_j, X^{<j} = x, Y^{<j} = y, Z^{-j} = z, P^{-j} = p)].$$

Again, say that an index $j \in [m]$ is good if conditioned on $S = j$, Π succeeds on $\sigma^{m,B}$ with probability at least $1 - 2\delta$. By a Markov argument, at least $m/2$ of the indices j are good. Fix a good index j.

Again, we say that the tuple (x, y, z, p) is good if conditioned on $S = j$, $X^{<j} = x$, $Y^{<j} = y$, $Z^{-j} = z$ and $P^{-j} = p$, Π succeeds on $\sigma^{m,B}$ with probability at least $1 - 4\delta$. By a Markov bound, with probability at least $1/2$, (x, y, z, p) is good. Fix a good (x, y, z, p).

As before, we can define a single-coordinate protocol $\Pi_{x,y,z,p,j}$. The parties use x and y to fill in their input vectors X and Y for coordinates $j' < j$. They can also use $Z^{-j} = z$, $P^{-j} = p$, and private randomness to fill in their inputs without any communication on the remaining coordinates $j' > j$. They place their single-coordinate input (U, V), uniformly drawn from $\{(0,0), (0,B)\}$, on their j-th coordinate. The parties output whatever Π outputs. Let Π' denote $\Pi_{x,y,z,p,j}$ for notational convenience.

The first issue is that unlike before Π' is not guaranteed to have success probability at least $1 - 4\delta$ since Π is not being run on input distribution $\sigma^{m,B}$ in this reduction. The second issue is in bounding $I(X_j, Y_j; \Pi'|Z_j, P_j)$ since (X_j, Y_j) is now drawn from the marginal distribution of $\zeta^{m,B}$ on coordinate j.

Notice that $S \neq j$ with probability $1 - 1/m$, which we condition on. This immediately resolves the second issue since now the marginal distribution on (X_j, Y_j) is the same under $\zeta^{m,B}$ as it was under $\sigma^{m,B}$; namely it is the following distribution: (X_j, Y_j) is uniformly random from the set $\{(Z_j, Z_j), (Z_j, Z_j + 1)\}$ if $P_j = 0$, and (X_j, Y_j) is uniformly random from the set $\{(Z_j, Z_j), (Z_j - 1, Z_j)\}$ if $P_j = 1$.

We now address the first issue. After conditioning on $S \neq j$, we have that (X^{-j}, Y^{-j}) is drawn from $\zeta^{m-1,B}$. If instead (X^{-j}, Y^{-j}) were drawn from $\mu^{m-1,B}$, then after placing (U, V) the input to Π would be drawn from $\sigma^{m,B}$ conditioned on a good tuple. Hence in that case, Π' would succeed with probability $1 - 4\delta$. Thus for our actual distribution on (X^{-j}, Y^{-j}), after conditioning on $S \neq j$, the success probability of Π' is at least

$$1 - 4\delta - D_{TV}(\mu^{m-1,B}, \zeta^{m-1,B}).$$

Let $C^{\mu,m-1,B}$ be the random variable which counts the number of coordinates i for which $(X_i, Y_i) = (0,0)$ when X and Y are drawn from $\mu^{m-1,B}$. Let $C^{\zeta,m-1,B}$ be a random variable which counts the number of coordinates i for which $(X_i, Y_i) = (0,0)$ when X and Y are drawn from $\zeta^{m-1,B}$. Observe that $(X_i, Y_i) = (0,0)$ in μ only if $P_i = 0$ and $Z_i = 0$, which happens with probability $1/(2B)$. Hence, $C^{\mu,m-1,B}$ is distributed as Binomial$(m-1, 1/(2B))$, while $C^{\zeta,m-1,B}$ is distributed as Binomial$(m-2, 1/(2B))+1$. We use μ' to denote the distribution of $C^{\mu,m-1,B}$ and ζ' to denote the distribution of $C^{\zeta,m-1,B}$. Also, let ι denote the Binomial$(m-2, 1/(2B))$ distribution.

Conditioned on $C^{\mu,m-1,B} = C^{\zeta,m-1,B}$, we have that $\mu^{m-1,B}$ and $\zeta^{m-1,B}$ are equal as distributions, and so

$$D_{TV}(\mu^{m-1,B}, \zeta^{m-1,B}) \leq D_{TV}(\mu', \zeta').$$

We use the following fact:

Fact 7.3.13. *(see, e.g., Fact 2.4 of [GMRZ11]). Any binomial distribution X with variance equal to σ^2 satisfies $D_{TV}(X, X+1) \leq 2/\sigma$.*

By definition,

$$\mu' = (1 - 1/(2B)) \cdot \iota + 1/(2B) \cdot \zeta'.$$

Since the variance of the Binomial$(m-2, 1/(2B))$ distribution is

$$(m-2)/(2B) \cdot (1 - 1/(2B)) = m/(2B)(1 - o(1)),$$

applying Fact 7.3.13 we have

$$\begin{aligned} D_{TV}(\mu', \zeta') &= D_{TV}((1 - 1/(2B)) \cdot \iota + (1/(2B)) \cdot \zeta', \zeta') \\ &= \frac{1}{2} \cdot \|(1 - 1/(2B)) \cdot \iota + (1/(2B)) \cdot \zeta' - \zeta'\|_1 \\ &= (1 - 1/(2B)) \cdot D_{TV}(\iota, \zeta') \\ &\leq \frac{2\sqrt{2B}}{\sqrt{m}} \cdot (1 + o(1)) \\ &= O\left(\sqrt{\frac{B}{m}}\right). \end{aligned}$$

It follows that the success probability of Π' is at least

$$1 - 4\delta - O\left(\sqrt{\frac{B}{m}}\right) \geq 1 - 5\delta.$$

Let E be an indicator random variable for the event that $S \neq j$. Then $H(E) = O((\log m)/m)$. Hence,

$$\begin{aligned} I(X_j, Y_j; \Pi' | Z_j, P_j) &\geq I(X_j, Y_j; \Pi' | Z_j, P_j, E) - O((\log m)/m) \\ &\geq (1 - 1/m) \cdot I(X_j, Y_j; \Pi' | Z_j, P_j, S \neq j) - O((\log m)/m) \\ &= \Omega(1/B^2), \end{aligned}$$

where we assume that $\Omega(1/B^2) - O((\log m)/m) = \Omega(1/B^2)$.

Hence, $I(X, Y; \Pi | Z, P) = \Omega(m/B^2)$ when inputs X and Y are distributed according to $\zeta^{m,B}$, and Π succeeds with probability at least $1 - \delta$ on X and Y distributed according to $\sigma^{m,B}$.

7.4 Lower bounds for k-sparse output

Theorem 7.4.1. *Any $1+\epsilon$-approximate ℓ_1/ℓ_1 nonadaptive recovery scheme with k-sparse output and failure probability δ requires $m = \Omega(\frac{1}{\epsilon}(k \log \frac{1}{\epsilon} + \log \frac{1}{\delta}))$, for $32 \leq \frac{1}{\delta} \leq n\epsilon^2/k$.*

Theorem 7.4.2. *Any $1+\epsilon$-approximate ℓ_2/ℓ_2 nonadaptive recovery scheme with k-sparse output and failure probability δ requires $m = \Omega(\frac{1}{\epsilon^2}(k + \log \frac{\epsilon^2}{\delta}))$, for $32 \leq \frac{1}{\delta} \leq n\epsilon^2/k$.*

These two theorems correspond to four statements: one for large k and one for small δ for both ℓ_1 and ℓ_2.

All the lower bounds proceed by reductions from communication complexity. The following lemma (implicit in [DIPW10]) shows that lower bounding the number of bits for approximate recovery is sufficient to lower bound the number of measurements.

Lemma 7.4.3. *Let $p \in \{1, 2\}$ and $\alpha = \Omega(1) < 1$. Suppose $X \subset \mathbb{R}^n$ has $\|x\|_p \le D$ and $\|x\|_\infty \le D'$ for all $x \in X$, and all coefficients of elements of X are expressible in $O(\log n)$ bits. Further suppose that we have a recovery algorithm that, for any ν with $\|\nu\|_p < \alpha D$ and $\|\nu\|_\infty < \alpha D'$, recovers $x \in X$ from $A(x + \nu)$ with constant probability. Then A must have $\Omega(\log |X|)$ measurements.*

Proof. First, we may assume that $A \in \mathbb{R}^{m\times n}$ has orthonormal rows (otherwise, if $A = U\Sigma V^T$ is its singular value decomposition, $\Sigma^+ U^T A$ has this property and can be inverted before applying the algorithm). Let A' be A rounded to $c \log n$ bits per entry. By Lemma 7.3.2, for any v we have $A'v = A(v - s)$ for some s with $\|s\|_1 \le n^2 2^{-c \log n}\|v\|_1$, so $\|s\|_p \le n^{2.5-c}\|v\|_p$.

Suppose Alice has a bit string of length $r \log |X|$ for $r = \Theta(\log n)$. By splitting into r blocks, this corresponds to $x_1, \dots, x_r \in X$. Let β be a power of 2 between $\alpha/2$ and $\alpha/4$, and define

$$z_j = \sum_{i=j}^{r} \beta^i x_i.$$

Alice sends $A'z_1$ to Bob; this is $O(m \log n)$ bits. Bob will solve the *augmented indexing problem*—given $A'z_1$, arbitrary $j \in [r]$, and $x_1, \dots, x_{j-1}$, he must find x_j with constant probability. This requires $A'z_1$ to have $\Omega(r \log |X|)$ bits, giving the result.

Bob receives $A'z_1 = A(z_1 + s)$ for $\|s\|_1 \le n^{2.5-c}\|z_1\|_p \le n^{2.5-c}D$. Bob then chooses $u \in B_p^n(n^{4.5-c}D)$ uniformly at random. With probability at least $1 - 1/n$, $u \in B_p^n((1 - 1/n^2)n^{4.5-c}D)$ by a volume argument. In this case $u + s \in B_p^n(n^{4.5-c}D)$; hence the random variables u and $u + s$ overlap in at least a $1 - 1/n$ fraction of their volumes, so $z_j + s + u$ and $z_j + u$ have statistical distance at most $1/n$. The distribution of $z_j + u$ is independent of A (unlike $z_j + s$) so running the recovery algorithm on $A(z_j + s + u)$ succeeds with constant probability as well.

We also have $\|z_j\|_p \le \frac{\beta^j - \beta^{r+1}}{1-\beta} D < 2(\beta^j - \beta^{r+1})D$. Since $r = O(\log n)$ and β is a constant, there exists a $c = O(1)$ with

$$\|z_j + s + u\|_p < (2\beta^j + n^{4.5-c} + n^{2.5-c} - 2\beta^r)D \le \beta^{j-1}\alpha D$$

for all j.

Therefore, given $x_1, \dots, x_{j-1}$, Bob can compute

$$\frac{1}{\beta^j}(A'z_1 + Au - A' \sum_{i<j} \beta^i x_i) = A(x_j + \frac{1}{\beta^j}(z_{j+1} + s + u)) = A(x_j + y)$$

for some y with $\|y\|_p \le \alpha D$. Hence Bob can use the recovery algorithm to recover x_j with constant probability. Therefore Bob can solve augmented indexing, so the message $A'z_1$ must have $\Omega(\log n \log |X|)$ bits, so $m = \Omega(\log |X|)$. □

We will now prove another lemma that is useful for all four theorem statements.

Let $x \in \{0, 1\}^n$ be k-sparse with $\text{supp}(x) \subseteq S$ for some known S. Let $\nu \in \mathbb{R}^n$ be a noise vector that roughly corresponds to having $O(k/\epsilon^p)$ ones for $p \in \{1, 2\}$, all located outside of S.

We consider under what circumstances we can use a $(1+\epsilon)$-approximate ℓ_p/ℓ_p recovery scheme to recover $\mathrm{supp}(x)$ from $A(x+\nu)$ with (say) 90% accuracy.

Lemma 7.4.4 shows that this is possible for $p=1$ when $|S| \lesssim k/\epsilon$ and for $p=2$ when $|S| \leq 2k$. The algorithm in both instances is to choose a parameter μ and perform sparse recovery on $A(x+\nu+z)$, where $z_i = \mu$ for $i \in S$ and $z_i = 0$ otherwise. The support of the result will be very close to $\mathrm{supp}(x)$.

Lemma 7.4.4. *Let $S \subset [n]$ have $|S| \leq s$, and suppose $x \in \{0,1\}^n$ satisfies $\mathrm{supp}(x) \subseteq S$ and $\|x_S\|_1 = k$. Let $p \in \{1,2\}$, and $\nu \in \mathbb{R}^n$ satisfy $\|\nu_S\|_\infty \leq \alpha$, $\|\nu\|_p^p \leq r$, and $\|\nu\|_\infty \leq D$ for some constants $\alpha \leq 1/4$ and $D = O(1)$. Suppose $A \in \mathbb{R}^{m\times n}$ is part of a $(1+\epsilon)$-approximate k-sparse ℓ_p/ℓ_p recovery scheme with failure probability δ.*

Then, given $A(x_S+\nu)$, Bob can with failure probability δ recover $\widehat{x_S}$ that differs from x_S in at most k/c locations, as long as either

$$p = 1, s = \Theta(\frac{k}{c\epsilon}), r = \Theta(\frac{k}{c\epsilon}) \tag{7.15}$$

or

$$p = 2, s = 2k, r = \Theta(\frac{k}{c^2\epsilon^2}) \tag{7.16}$$

Proof. For some parameter $\mu \geq D$, let $z_i = \mu$ for $i \in S$ and $z_i = 0$ elsewhere. Consider $y = x_S + \nu + z$. Let $U = \mathrm{supp}(x_S)$ have size k. Let $V \subset [n]$ be the support of the result of running the recovery scheme on $Ay = A(x_S+\nu) + Az$. Then we have that $x_S + z$ is $\mu+1$ over U, μ over $S \setminus U$, and zero elsewhere. Since $\|u+v\|_p^p \leq p(\|u\|_p^p + \|v\|_p^p)$ for any u and v, we have

$$\begin{aligned}\|y_{\overline{U}}\|_p^p &\leq p(\|(x_S+z)_{\overline{U}}\|_p^p + \|\nu\|_p^p) \\ &\leq p((s-k)\mu^p + r) \\ &< p(r + s\mu^p).\end{aligned}$$

Since $\|\nu_S\|_\infty \leq \alpha$ and $\|\nu_{\overline{S}}\|_\infty < \mu$, we have

$$\begin{aligned}\|y_U\|_\infty &\geq \mu + 1 - \alpha \\ \|y_{\overline{U}}\|_\infty &\leq \mu + \alpha\end{aligned}$$

We then get

$$\begin{aligned}\|y_{\overline{V}}\|_p^p &= \|y_{\overline{U}}\|_p^p + \|y_{U\setminus V}\|_p^p - \|y_{V\setminus U}\|_p^p \\ &\geq \|y_{\overline{U}}\|_p^p + |V\setminus U|((\mu+1-\alpha)^p - (\mu+\alpha)^p) \\ &= \|y_{\overline{U}}\|_p^p + |V\setminus U|(1+(2p-2)\mu)(1-2\alpha)\end{aligned}$$

where the last step can be checked for $p \in \{1,2\}$. So

$$\|y_{\overline{V}}\|_p^p \geq \|y_{\overline{U}}\|_p^p(1 + |V\setminus U|\frac{(1+(2p-2)\mu)(1-2\alpha)}{p(r+s\mu^p)})$$

However, V is the result of $1+\epsilon$-approximate recovery, so

$$\|y_{\overline{V}}\|_p \le \|y-\widehat{y}\|_p \le (1+\epsilon)\|y_{\overline{U}}\|_p$$
$$\|y_{\overline{V}}\|_p^p \le (1+(2p-1)\epsilon)\|y_{\overline{U}}\|_p^p$$

for $p \in \{1,2\}$. Hence

$$|V \setminus U| \frac{(1+(2p-2)\mu)(1-2\alpha)}{p(r+s\mu^p)} \le (2p-1)\epsilon$$

for $\alpha \le 1/4$, this means

$$|V \setminus U| \le \frac{2\epsilon(2p-1)p(r+s\mu^p)}{1+(2p-2)\mu}.$$

Plugging in the parameters $p=1, s=r=\frac{k}{d\epsilon}, \mu = D$ gives

$$|V \setminus U| \le \frac{2\epsilon((1+D^2)r)}{1} = O(\frac{k}{d}).$$

Plugging in the parameters $p=2, q=2, r=\frac{k}{d^2\epsilon^2}, \mu=\frac{1}{d\epsilon}$ gives

$$|V \setminus U| \le \frac{12\epsilon(3r)}{2\mu} = \frac{18k}{d}.$$

Hence, for $d = O(c)$, we get the parameters desired in the lemma statement, and

$$|V \setminus U| \le \frac{k}{2c}.$$

Bob can recover V with probability $1-\delta$. Therefore he can output x' given by $x'_i = 1$ if $i \in V$ and $x'_i = 0$ otherwise. This will differ from x_S only within $(V \setminus U \cup U \setminus V)$, which is at most k/c locations. □

7.4.1 $k > 1$

Suppose $p, s, 3r$ satisfy Lemma 7.4.4 for some parameter c, and let $q = s/k$. The Gilbert-Varshamov bound implies that there exists a code $V \subset [q]^r$ with $\log|V| = \Omega(r \log q)$ and minimum Hamming distance $r/4$. Let $X \subset \{0,1\}^{qr}$ be in one-to-one correspondence with V: $x \in X$ corresponds to $v \in V$ when $x_{(a-1)q+b} = 1$ if and only if $v_a = b$.

Let x and v correspond. Let $S \subset [r]$ with $|S| = k$, so S corresponds to a set $T \subset [n]$ with $|T| = kq = s$. Consider arbitrary ν that satisfies $\|\nu\|_p < \alpha\|x\|_p$ and $\|\nu\|_\infty \le \alpha$ for some small constant $\alpha \le 1/4$. We would like to apply Lemma 7.4.3, so we just need to show we can recover x from $A(x+\nu)$ with constant probability. Let $\nu' = x_{\overline{T}} + \nu$, so

$$\|\nu'\|_p^p \le p(\|x_{\overline{T}}\|_p^p + \|\nu\|_p^p) \le p(r-k+\alpha^p r) \le 3r$$
$$\|\nu'_{\overline{T}}\|_\infty \le 1+\alpha$$
$$\|\nu'_T\|_\infty \le \alpha$$

Therefore Lemma 7.4.4 implies that with probability $1-\delta$, if Bob is given $A(x_T+\nu') = A(x+\nu)$

he can recover x' that agrees with x_T in all but k/c locations. Hence in all but k/c of the $i \in S$, $x_{\{(i-1)q+1,\ldots,iq\}} = x'_{\{(i-1)q+1,\ldots,iq\}}$, so he can identify v_i. Hence Bob can recover an estimate of v_S that is accurate in $(1-1/c)k$ characters with probability $1-\delta$, so it agrees with v_S in $(1-1/c)(1-\delta)k$ characters in expectation. If we apply this in parallel to the sets $S_i = \{k(i-1)+1,\ldots,ki\}$ for $i \in [r/k]$, we recover $(1-1/c)(1-\delta)r$ characters in expectation. Hence with probability at least $1/2$, we recover more than $(1-2(1/c+\delta))r$ characters of v. If we set δ and $1/c$ to less than $1/32$, this gives that we recover all but $r/8$ characters of v. Since V has minimum distance $r/4$, this allows us to recover v (and hence x) exactly. By Lemma 7.4.3 this gives a lower bound of $m = \Omega(\log|V|) = \Omega(r\log q)$. Hence $m = \Omega(\frac{1}{\epsilon}k\log\frac{1}{\epsilon})$ for ℓ_1/ℓ_1 recovery and $m = \Omega(\frac{1}{\epsilon^2}k)$ for ℓ_2/ℓ_2 recovery.

7.4.2 $k = 1, \delta = o(1)$

To achieve the other half of our lower bounds for sparse outputs, we restrict to the $k = 1$ case. A k-sparse algorithm implies a 1-sparse algorithm by inserting $k-1$ dummy coordinates of value ∞, so this is valid.

Let $p, s, 51r$ satisfy Lemma 7.4.4 for some α and D to be determined, and let our recovery algorithm have failure probability δ. Let $C = 1/(2r\delta)$ and $n = Cr$. Let $V = [(s-1)C]^r$ and let $X' \in \{0,1\}^{(s-1)Cr}$ be the corresponding binary vector. Let $X = \{0\} \times X'$ be defined by adding $x_0 = 0$ to each vector.

Now, consider arbitrary $x \in X$ and noise $\nu \in \mathbb{R}^{1+(s-1)Cr}$ with $\|\nu\|_p < \alpha\|x\|_p$ and $\|\nu\|_\infty \le \alpha$ for some small constant $\alpha \le 1/20$. Let $e^0/5$ be the vector that is $1/5$ at 0 and 0 elsewhere. Consider the sets $S_i = \{0, (s-1)(i-1)+1, (s-1)(i-1)+2, \ldots, (s-1)i\}$. We would like to apply Lemma 7.4.4 to recover $(x+\nu+e^0/5)_{S_i}$ for each i.

To see what it implies, there are two cases: $\|x_{sSi}\|_1 = 1$ and $\|x_{S_i}\|_1 = 0$ (since S_i lies entirely in one character, $\|x_{S_i}\|_1 \in \{0,1\}$). In the former case, we have $\nu' = x_{\overline{S_i}} + \nu + e^0/5$ with

$$\begin{aligned}
\|\nu'\|_p^p &\le (2p-1)(\|x_{\overline{S_i}}\|_p^p + \|\nu\|_p^p + \|e^0/5\|_p^p) \le 3(r + \alpha^p r + 1/5^p) < 4r \\
\|\nu'_{\overline{S_i}}\|_\infty &\le 1+\alpha \\
\|\nu'_{S_i}\|_\infty &\le 1/5 + \alpha \le 1/4
\end{aligned}$$

Hence Lemma 7.4.4 will, with failure probability δ, recover x'_{S_i} that differs from x_{S_i} in at most $1/c < 1$ positions, so x_{S_i} is correctly recovered.

Now, suppose $\|x_{S_i}\|_1 = 0$. Then we observe that Lemma 7.4.4 would apply to recovery from $5A(x+\nu+e^0/5)$, with $\nu' = 5x + 5\nu$ and $x' = e^0$, so

$$\begin{aligned}
\|\nu'\|_p^p &\le 5^p p(\|x\|_p^p + \|\nu\|_p^p) \le 5^p p(r + \alpha^p r) < 51r \\
\|\nu'_{\overline{S_i}}\|_\infty &\le 5 + 5\alpha \\
\|\nu'_{S_i}\|_\infty &\le 5\alpha.
\end{aligned}$$

Hence Lemma 7.4.4 would recover, with failure probability δ, an x'_{S_i} with support equal to $\{0\}$.

Now, we observe that the algorithm in Lemma 7.4.4 is robust to scaling the input $A(x'+\nu')$ by 5; the only difference is that the effective μ changes by the same factor, which increases the number of errors k/c by a factor of at most 5. Hence if $c > 5$, we can apply the algorithm once and have it work regardless of whether $\|x_{S_i}\|_1$ is 0 or 1: if $\|x_{S_i}\|_1 = 1$ the result has support $\mathrm{supp}(x_i)$, and if $\|x_{S_i}\|_1 = 0$ the result has support $\{0\}$. Thus we can recover x_{S_i} exactly with failure probability δ.

If we try this to the $Cr = 1/(2\delta)$ sets S_i, we recover all of x correctly with failure probability

at most 1/2. Hence Lemma 7.4.3 implies that $m = \Omega(\log |X|) = \Omega(r \log \frac{s}{r\delta})$. For ℓ_1/ℓ_1, this means $m = \Omega(\frac{1}{\epsilon} \log \frac{1}{\delta})$; for ℓ_2/ℓ_2, this means $m = \Omega(\frac{1}{\epsilon^2} \log \frac{\epsilon^2}{\delta})$.

Bibliography

[ACD11] E. Arias-Castro, E.J. Candes, and M. Davenport. On the fundamental limits of adaptive sensing. *Arxiv preprint arXiv:1111.4646*, 2011.

[AFS93] R. Agrawal, C. Faloutsos, and A. Swami. Efficient similarity search in sequence databases. *Int. Conf. on Foundations of Data Organization and Algorithms*, pages 69–84, 1993.

[AGS03] A. Akavia, S. Goldwasser, and S. Safra. Proving hard-core predicates using list decoding. *FOCS*, 44:146–159, 2003.

[Aka10] A. Akavia. Deterministic sparse Fourier approximation via fooling arithmetic progressions. *COLT*, pages 381–393, 2010.

[ASZ10] S. Aeron, V. Saligrama, and M. Zhao. Information theoretic bounds for compressed sensing. *Information Theory, IEEE Transactions on*, 56(10):5111–5130, 2010.

[AWZ08] A. Aldroubi, H. Wang, and K. Zarringhalam. Sequential adaptive compressed sampling via huffman codes. *Preprint*, 2008.

[Bar02] Ziv Bar-Yossef. *The Complexity of Massive Data Set Computations*. PhD thesis, UC Berkeley, 2002.

[BCG+12] P. Boufounos, V. Cevher, A. C. Gilbert, Y. Li, and M. J. Strauss. What's the frequency, Kenneth?: Sublinear Fourier sampling off the grid. *RANDOM/APPROX*, 2012.

[BDDW08] Richard Baraniuk, Mark Davenport, Ronald DeVore, and Michael Wakin. A simple proof of the restricted isometry property for random matrices. *Constr. Approx.*, 28:253–263, 2008.

[BJCC12] Mayank Bakshi, Sidharth Jaggi, Sheng Cai, and Minghua Chen. SHO-FA: Robust compressive sensing with order-optimal complexity, measurements, and bits. In *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, pages 786–793. IEEE, 2012.

[BJKS04] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *J. Comput. Syst. Sci.*, 68(4):702–732, 2004.

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

[CD11] E.J. Candès and M.A. Davenport. How well can we estimate a sparse vector? *Arxiv preprint arXiv:1104.5246*, 2011.

[CDD09] A. Cohen, W. Dahmen, and R. DeVore. Compressed sensing and best k-term approximation. *J. Amer. Math. Soc*, 22(1):211–231, 2009.

[CGV12] Mahdi Cheraghchi, Venkatesan Guruswami, and Ameya Velingker. Restricted isometry of Fourier matrices and list decodability of random linear codes. *SODA*, 2012.

[CGX96] A. Chandrakasan, V. Gutnik, and T. Xanthopoulos. Data driven signal processing: An approach for energy efficient computing. *International Symposium on Low Power Electronics and Design*, 1996.

[CHNR08] R. Castro, J. Haupt, R. Nowak, and G. Raz. Finding needles in noisy haystacks. *Proc. IEEE Conf. Acoustics, Speech, and Signal Proc.*, page 5133¡D0¿5136, 2008.

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

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

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

[CP10] E. Candes and Y. Plan. A probabilistic and ripless theory of compressed sensing. *IEEE Transactions on Information Theory*, 2010.

[CRT06a] E. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52:489–509, 2006.

[CRT06b] E. J. Candès, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Comm. Pure Appl. Math.*, 59(8):1208–1223, 2006.

[CSN09] Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. Power-law distributions in empirical data. *SIAM review*, 51(4):661–703, 2009.

[CT65] James W Cooley and John W Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematics of computation*, 19(90):297–301, 1965.

[CT06] E. Candes and T. Tao. Near optimal signal recovery from random projections: Universal encoding strategies. *IEEE Trans. on Info.Theory*, 2006.

[DDT+08a] 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.

[DDT+08b] Marco F Duarte, Mark A Davenport, Dharmpal Takhar, Jason N Laska, Ting Sun, Kevin F Kelly, and Richard G Baraniuk. Single-pixel imaging via compressive sampling. *Signal Processing Magazine, IEEE*, 25(2):83–91, 2008.

[Def10] Defense Sciences Office. Knowledge enhanced compressive measurement. *Broad Agency Announcement*, DARPA-BAA-10-38, 2010.

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

[Don06] D. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[DR93] A. Dutt and V. Rokhlin. Fast Fourier transforms for nonequispaced data. *SIAM J. Sci. Comput.*, 14(6):1368–1393, November 1993.

[DRZ07] I. Daubechies, O. Runborg, and J. Zou. A sparse spectral method for homogenization multiscale problems. *Multiscale Model. Sim.*, 6(3):711–740, 2007.

[ECG+09] Yaniv Erlich, Kenneth Chang, Assaf Gordon, Roy Ronen, Oron Navon, Michelle Rooks, and Gregory J Hannon. Dna sudokuharnessing high-throughput sequencing for multiplexed specimen analysis. *Genome research*, 19(7):1243–1253, 2009.

[EG07] David Eppstein and Michael T Goodrich. Space-efficient straggler identification in round-trip data streams via newtons identities and invertible bloom filters. In *Algorithms and Data Structures*, pages 637–648. Springer, 2007.

[FJ98] Matteo Frigo and Steven G Johnson. Fftw: An adaptive software architecture for the fft. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 3, pages 1381–1384. IEEE, 1998.

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

[GGI+02a] A. Gilbert, S. Guha, P. Indyk, M. Muthukrishnan, and M. Strauss. Near-optimal sparse Fourier representations via sampling. *STOC*, 2002.

[GGI+02b] A. C. Gilbert, S. Guha, P. Indyk, Y. Kotidis, S. Muthukrishnan, and M. J. Strauss. Fast, small-space algorithms for approximate histogram maintenance. In *ACM Symposium on Theoretical Computer Science*, 2002.

[GL89] O. Goldreich and L. Levin. A hard-corepredicate for allone-way functions. *STOC*, pages 25–32, 1989.

[GLPS10] Anna C. Gilbert, Yi Li, Ely Porat, and Martin J. Strauss. Approximate sparse recovery: optimizing time and measurements. In *STOC*, pages 475–484, 2010.

[GMRZ11] Parikshit Gopalan, Raghu Meka, Omer Reingold, and David Zuckerman. Pseudorandom generators for combinatorial shapes. In *STOC*, pages 253–262, 2011.

[GMS05] A. Gilbert, M. Muthukrishnan, and M. Strauss. Improved time bounds for near-optimal space Fourier representations. *SPIE Conference, Wavelets*, 2005.

[GMS11] Ran Gelles, Ankur Moitra, and Amit Sahai. Efficient and explicit coding for interactive communication. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 768–777. IEEE, 2011.

[GST08] A.C. Gilbert, M.J. Strauss, and J. A. Tropp. A tutorial on fast Fourier sampling. *Signal Processing Magazine*, 2008.

[Gur10] V. Guruswami. Introduction to coding theory. *Graduate course notes, available at* `http://www.cs.cmu.edu/~venkatg/teaching/codingtheory/`, 2010.

[HAKI12] Haitham Hassanieh, Fadel Adib, Dina Katabi, and Piotr Indyk. Faster gps via the sparse fourier transform. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 353–364. ACM, 2012.

[HBCN09] J. Haupt, R. Baraniuk, R. Castro, and R. Nowak. Compressive distilled sensing. *Asilomar*, 2009.

[HCN09] J. Haupt, R. Castro, and R. Nowak. Adaptive sensing for sparse signal recovery. *Proc. IEEE 13th Digital Sig. Proc./5th Sig. Proc. Education Workshop*, page 702¡D0¿707, 2009.

[HIKP12a] H. Hassanieh, P. Indyk, D. Katabi, and E. Price. Nearly optimal sparse Fourier transform. *STOC*, 2012.

[HIKP12b] H. Hassanieh, P. Indyk, D. Katabi, and E. Price. sFFT: Sparse Fast Fourier Transform. *http://groups.csail.mit.edu/netmit/sFFT/*, 2012.

[HIKP12c] H. Hassanieh, P. Indyk, D. Katabi, and E. Price. Simple and practical algorithm for sparse Fourier transform. *SODA*, 2012.

[HKPV13] Sabine Heider, Stefan Kunis, Daniel Potts, and Michael Veit. A sparse prony fft. *SAMPTA*, 2013.

[How08] Rodney R Howell. On asymptotic notation with multiple variables. Technical report, Citeseer, 2008.

[HT01] Juha Heiskala and John Terry, Ph.D. *OFDM Wireless LANs: A Theoretical and Practical Guide*. Sams, Indianapolis, IN, USA, 2001.

[IKP13] P. Indyk, M. Kapralov, and E. Price. (Nearly) sample-optimal sparse Fourier transform. *Manuscript*, 2013.

[IPW11] P. Indyk, E. Price, and D. Woodruff. On the power of adaptivity in sparse recovery. *FOCS*, 2011.

[IR08] Piotr Indyk and Milan Ruzic. Near-optimal sparse recovery in the l1 norm. In *FOCS*, pages 199–207, 2008.

[IT10] MA Iwen and AH Tewfik. Adaptive group testing strategies for target detection and localization in noisy environments. *IMA Preprint Series*, 2311, 2010.

[Iwe10] M. A. Iwen. Combinatorial sublinear-time Fourier algorithms. *Foundations of Computational Mathematics*, 10:303–338, 2010.

[Jay02] T.S. Jayram. Unpublished manuscript, 2002.

[JXC08] S. Ji, Y. Xue, and L. Carin. Bayesian compressive sensing. *IEEE Trans. Signal Processing*, 56(6):23462356, 2008.

[KK07] Richard M Karp and Robert Kleinberg. Noisy binary search and its applications. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 881–890. Society for Industrial and Applied Mathematics, 2007.

[KKL88] J. Kahn, G. Kalai, and N. Linial. The influence of variables on boolean functions. *FOCS*, 1988.

[KM91] E. Kushilevitz and Y. Mansour. Learning decision trees using the Fourier spectrum. *STOC*, 1991.

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, 1997.

[LDSP08] M. Lustig, D.L. Donoho, J.M. Santos, and J.M. Pauly. Compressed sensing mri. *Signal Processing Magazine, IEEE*, 25(2):72–82, 2008.

[LMN93] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, Fourier transform, and learnability. *Journal of the ACM (JACM)*, 1993.

[LVS11] Mengda Lin, A. P. Vinod, and Chong Meng Samson See. A new flexible filter bank for low complexity spectrum sensing in cognitive radios. *Journal of Signal Processing Systems*, 62(2):205–215, 2011.

[LWC12] D. Lawlor, Y. Wang, and A. Christlieb. Adaptive sub-linear time Fourier algorithms. *arXiv:1207.6368*, 2012.

[Man92] Y. Mansour. Randomized interpolation and approximation of sparse polynomials. *ICALP*, 1992.

[Mar04] G. Marsaglia. Evaluating the normal distribution. *Journal of Statistical Software*, 11(4):1–7, 2004.

[McG09] A. McGregor. Graph mining on streams. *Encyclopedia of Database Systems*, page 12711275, 2009.

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

[MNL10] A. Mueen, S. Nath, and J. Liu. Fast approximate correlation for massive time-series data. In *Proceedings of the 2010 international conference on Management of data*, pages 171–182. ACM, 2010.

[MSW08] D. M. Malioutov, S. Sanghavi, and A. S. Willsky. Compressed sensing with sequential observations. *ICASSP*, 2008.

[O'D08] R. O'Donnell. Some topics in analysis of boolean functions (tutorial). *STOC*, 2008.

[PDGQ05] R. Pike, S. Dorward, R. Griesemer, and S. Quinlan. Interpreting the data: Parallel analysis with sawzall. *Scientific Programming*, 13(4):277, 2005.

[PIF+12] Jonathan Perry, Peter A Iannucci, Kermin E Fleming, Hari Balakrishnan, and Devavrat Shah. Spinal codes. In *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*, pages 49–60. ACM, 2012.

[Pol05] Pollard. Total variation distance between measures. 2005. http://www.stat.yale.edu/ pollard/Courses/607.spring05/handouts/Totalvar.pdf.

[PR13] Sameer Pawar and Kannan Ramchandran. Computing a k-sparse n-length discrete Fourier transform using at most 4k samples and o (k log k) complexity. *arXiv preprint arXiv:1305.0870*, 2013.

[PST01] Daniel Potts, Gabriele Steidl, and Manfred Tasche. Fast Fourier transforms for nonequispaced data: A tutorial. In *Modern sampling theory*, pages 247–270. Springer, 2001.

[PW11] E. Price and D. P. Woodruff. $(1 + \epsilon)$-approximate sparse recovery. *FOCS*, 2011.

[PW12] Eric Price and David P Woodruff. Lower bounds for adaptive sparse recovery. *arXiv preprint arXiv:1205.3518*, 2012.

[RV08] M. Rudelson and R. Vershynin. On sparse reconstruction from Fourier and Gaussian measurements. *CPAM*, 61(8):1025–1171, 2008.

[Sch93] Leonard J Schulman. Deterministic coding for interactive communication. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 747–756. ACM, 1993.

[Sha48] C.E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[SS02] Michael E. Saks and Xiaodong Sun. Space lower bounds for distance approximation in the data stream model. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 360–369, 2002.

[TDB09] J. Treichler, M. Davenport, and R. Baraniuk. Application of compressive sensing to the design of wideband signal acquisition receivers. *In Proc. U.S./Australia Joint Work. Defense Apps. of Signal Processing (DASP)*, 2009.

[Wai09] Martin J. Wainwright. Information-theoretic limits on sparsity recovery in the high-dimensional and noisy setting. *IEEE Transactions on Information Theory*, 55(12):5728–5741, 2009.