Algorithms and Lower Bounds in the Streaming and Sparse Recovery Models

by

Khanh Do Ba

B.A., Dartmouth College (2006)

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

June 2012

Author ....
Department of Electrical Engineering and Computer Science
May 23, 2012

Certified by ....
Piotr Indyk
Professor
Thesis Supervisor

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

Algorithms and Lower Bounds
in the Streaming and Sparse Recovery Models
by
Khanh Do Ba

Submitted to the Department of Electrical Engineering and Computer Science
on May 23, 2012, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computer Science

Abstract

In the *data stream computation* model, input data is given to us sequentially (the data stream), and our goal is to compute or approximate some function or statistic on that data using a sublinear (in both the length of the stream and the size of the universe of items that can appear in the stream) amount of space; in particular, we can store neither the entire stream nor a counter for each possible item we might see.

In the *sparse recovery* model (also known as *compressed sensing*), input data is a large but sparse vector $x \in \mathbb{R}^n$, and our goal is to design an $m \times n$ matrix Φ, where $m \ll n$, such that for any sufficiently sparse x we can efficiently recover a good approximation of x from Φx.

Although at first glance these two models may seem quite different, they are in fact intimately related. In the streaming model, most statistics of interest are order-invariant, meaning they care only about the frequency of each item in the stream and not their position. For these problems, the data in the stream can be viewed as an n-dimensional vector x, where x_i is the number of occurrences of item i. Using this representation, one of the high-level tools that have proven most popular has been the *linear sketch*, where for some $m \times n$ matrix Φ, we maintain Φx (the sketch) for the partial vector x as we progress along the stream. The linearity of the mapping Φ allows us to efficiently do incremental updates on our sketch, and as in its use in sparse recovery, the linear sketch turns out to be surprisingly powerful. In this thesis, we try to answer some questions of interest in each model, illustrating both the power and the limitations of the linear sketch.

In Chapter 2, we provide an efficient sketch for estimating the (planar) *Earth-Mover Distance* (EMD) between two multisets of points. The EMD between point sets $A, B \subseteq \mathbb{R}^2$ of the same size is defined as the minimum cost of a perfect matching between them, with each edge contributing a cost equal to its (Euclidean) length. As immediate consequences, we give an improved algorithm for estimating EMD between point sets given over a stream, and an improved algorithm for the approximate nearest neighbor problem under EMD.

In Chapter 3, we prove tight lower bounds for sparse recovery in the number of rows in the matrix Φ (i.e., the number of measurements) in order to achieve any of the three most studied recovery guarantees. Specifically, consider a matrix Φ and an algorithm $\mathcal{R}$ such that for any signal x, $\mathcal{R}$ can recover an approximation $\hat{x}$ from Φx satisfying

$$\|x - \hat{x}\|_p \le C \min_{k\text{-sparse } x'} \|x - x'\|_q ,$$

where (1) $p = q = 1$ and $C = O(1)$, (2) $p = q = 2$ and $C = O(1)$, or (3) $p = 2$, $q = 1$ and

$C = O(k^{-1/2})$. We show that any such Φ must have at least $\Omega(k \log(n/k))$ rows. This is known to be optimal in cases (1) and (2), and near optimal for (3).

In Chapter 4, we propose a variant of sparse recovery that incorporates some additional knowledge about the signal that allows the above lower bound to be broken. In particular, we consider the scenario where, after measurements are taken, we are given a set S of size $s \ll n$ (s is known beforehand) that is supposed to contain most of the "large" coefficients of x. The goal is then to recover $\hat{x}$ satisfying

$$\|x - \hat{x}\|_p \leq C \min_{\substack{k\text{-sparse } x' \\ \text{supp}(x') \subseteq S}} \|x - x'\|_q \ .$$

We refer to this formulation as the *sparse recovery with partial support knowledge problem* (SRPSK). We focus on the guarantees where $p = q = 1$ or 2 and $C = 1 + \epsilon$, for which we provide lower bounds as well as a method of converting algorithms for "standard" sparse recovery into ones for SRPSK. We also make use of one of the reductions to give an optimal algorithm for SRPSK for the guarantee where $p = q = 2$.

Thesis Supervisor: Piotr Indyk
Title: Professor

Acknowledgments

First, I must thank my advisor, Piotr Indyk, for his incredible patience, guidance and support throughout my years at MIT. He has put up with perhaps not the easiest of students, year after year, has guided and collaborated with me on every piece of research in this thesis, and has given all the space and freedom as well as advice and encouragement I needed to grow and develop as a researcher and an individual. Not least, his ever-jovial and constant presence around the G6 lounge was a regular reminder that all was right with the universe.

Next, I owe a tremendous debt of gratitude for whatever I have accomplished academically to my undergraduate advisor at Dartmouth, Prasad Jayanti. I first became fascinated by theoretical computer science when I took Theory of Computation with him, and he subsequently became stuck for over two years as my advisor, teacher and mentor, spending countless hours with me over research, life and pizza.

Also at Dartmouth, two other professors I will always be grateful to are Amit Chakrabarti, who was a second research mentor and collaborator, and always encouraged and believed in me, and Vladimir Chernov, who kept my love of mathematics alive with his awesome Russianness.

To all of my other co-authors, for both the works in and those leading up to this thesis, I'm grateful to have had the opportunity to work with and learn from them: Alex Andoni, S. Muthukrishnan, Huy L. Nguyen, Huy N. Nguyen, Eric Price, Ronitt Rubinfeld and David Woodruff.

A number of friends have been instrumental in getting me through these past years: my flatmate Minh made sure I ate properly and joined me on many a coffee run, my other flatmate Duong turned me into a fake-news mini-celebrity that foreshadowed my completion of the PhD, Huy welcomed me to MIT and fought side-by-side with me for years until he got married, had a baby and had to graduate in a hurry, and my girlfriend Sara who, especially through my thesis-writing phase, was the conscience keeping me from too much procrastination. I am also thankful for the other close friends throughout my life who always believed in me and on whom I can always count, whether it is to celebrate with or to bail me out of a tight spot. These include, in no particular order, Le Anh, Minh, Tien, Tuan Anh, Ky Ha, Lena, Thu Ha, Cheryl, Henry, Omar and Victoria.

My last, and without a doubt biggest, thanks go to my family. Nothing I say here can possibly be enough, but I will try anyway. My brother Khai, as much as we fought growing up, has always been a most incredibly consistent source of support for me, and the knowledge that he was always watching and looking up to me has motivated me to be the best I could be. My cousins Khiem and Khoa, with whom I also grew up, have always been more brothers than cousins to me, and have been as steady pillars of support as I could possibly hope for. And my parents... From before I could talk until the day I no longer can, my parents have always been and will always be my greatest teachers. They have inspired me with a love of knowledge and reason, of mathematics, science, history, literature, music and sports. They have taught me the foundations of everything I know, and in many things well beyond the foundations. They have given me role models to aspire to be, in academics, work, play, in love, and in life.

To my parents, for providing endless support and inspiration.

Contents

Chapter 1

Introduction

The modern era has brought tremendous technological advances in nearly every area of science and society. With those technologies, we now have access to massive amounts of data never before available: network traffic data, genomic data, medical data, social media data, financial markets data, etc. To tap into this tremendous new resource, however, we need the ability to process this raw data into some useful form. Unfortunately, many existing algorithms are incapable of handling this massive scale of data. This has spurred the development in recent years of the broad area of *massive data algorithmics.*

In the traditional world of algorithms, the requirement for efficiency was polynomial time/space (or any other scarce resource), and the gold standard was linear cost. Today, in increasingly more problem settings, a sublinear cost is the bare minimum.

Streaming and Sketching Algorithms Capturing one type of such settings is the data stream computation model (e.g., [Mut05, Ind07b]). Here, the input data is given to us sequentially, and our goal is to compute or approximate some function or statistic on that data using a sublinear (in both the length of the stream and the number of possible items that can appear in the stream) amount of space; in other words, we can store neither the entire stream nor a counter for each item we see.

Most streaming problems consider statistics that are invariant to the order of the stream. In this case, the underlying data can be thought of as a vector $x \in \mathbb{R}^n$, initialized to the zero vector, to which each item in the stream indicates an update. Specifically, each update is a pair (i, a), which says that a should be added to x_i. The algorithm needs to maintain a small data structure that can be updated very quickly with each new item in the stream, and from which can be computed (an approximation of) the desired function of x.

Perhaps the simplest such data structure is the *linear sketch.* That is, we maintain Φx, for some predefined $m \times n$ matrix Φ. On seeing an update (i, a), we simply add $\Phi(ae_i)$ to the current sketch Φx to obtain the updated sketch (e_i is the indicator vector for i). We therefore get the ability to quickly update our data structure for free, and need only focus on choosing the matrix Φ. It turns out that this seemingly very limiting framework is perhaps the most powerful tool in the toolbox of a data stream algorithmicist.

Sparse Recovery In a second related model, consider a signal $x \in \mathbb{R}^n$ which we are trying to measure (i.e., observe) in compressed form. The traditional approach would be to take its full measurement, then compress the result. For example, a digital camera takes a photo in raw bitmap format, then compresses it to a JPEG before saving it to memory card. This can be very wasteful, as the bulk of the information from the measurements end up discarded. In the new approach, this waste is avoided by combining the measurement and compression steps into one by taking instead a small number of non-adaptive linear measurements of the signal; in other words, a linear sketch. In many settings, there are natural and efficient ways to take such linear measurements (e.g., the above camera example [DDT$^+$08]).

Of course, the signal has to be "compressible" for this setting to make sense. Specifically, we focus on signals that are almost *k-sparse*, that is, have at most k non-zero coefficients. In fact, it suffices that the signal be almost sparse in *some* known basis, as is the case for natural images. Variations of this framework are variously known as *sparse recovery* or *compressed sensing.*

This problem can be loosely viewed as a special case of streaming algorithms, where the "statistic" to estimate is the entire signal itself, and where we are required to use a linear sketch. Variants of this special case have in fact been well-studied in the streaming literature (e.g., [CM05]). In this thesis, we study several questions in these two models that help shed light on our understanding of both the power and the limitations of the linear sketch.

1.1 Efficient Sketches for Earth-Mover Distance (Chapter 2)

For any two multisets A, B of points in $\mathbb{R}^2$, $|A| = |B| = N$, the (planar) *Earth-Mover Distance (EMD)* between A and B is defined as the minimum cost of a perfect matching with edges between A and B, i.e.,

$$EMD(A,B) = \min_{\pi:A\to B} \sum_{a\in A} \|a - \pi(a)\|$$

where π ranges over all one-to-one mappings.

Recently, there has been a significant interest in developing methods for *geometric* representation of EMD. The goal of this line of research is to provide mappings (say, f) that map a set of points A into a vector $f(A)$ in an m-dimensional space, such that the distance $EMD(A,B)$ between any two point sets can be approximated from the vectors $f(A)$ and $f(B)$. To be useful, the space that f maps into must be "simple", e.g., its dimension m must be low, or its distance estimation function should be of simple form. For example, [Cha02, IT03] provide a mapping f that works when the sets A, B are subsets of size N of the discrete square grid $[\Delta]^2$, mapping them into $O(\Delta^2)$ dimensions and guaranteeing that, for some absolute constant $C > 0$,

$$\|f(A) - f(B)\|_1 \leq EMD(A,B) \leq C \log \Delta \cdot \|f(A) - f(B)\|_1 .$$[1]

One important application of geometric representations of EMD is in data stream computation. For this setting, the points of A and B are given one at a time in some arbitrary order, and we need to approximate their EMD at the end of the stream. With geometric representations of A and B, as defined above, we can do this as long as $f(A)$ and $f(B)$ can be efficiently updated over the course of the stream. Specifically, if f is a *linear sketch*, this becomes trivial. In fact, the above mapping of EMD into ℓ_1, combined with ℓ_1-distance-preserving mappings into low dimensions [Ind00], has been used to yield an efficient algorithm for the streaming EMD problem [Ind04]. Specifically, the algorithm provides an $O(\log \Delta)$-approximation in one

[1] In this thesis, all logarithms are base 2 unless otherwise noted.

pass over the data, using $\log^{O(1)}(\Delta N)$ space. Obtaining a better EMD estimation algorithm has been an important open problem in the streaming literature [McG06].

However, representing EMD as vectors in the ℓ_1 space has limitations: it has been shown [NS07] that any such mapping must incur a distortion of at least $\Omega(\sqrt{\log \Delta})$. Thus, in order to obtain more accurate representations, one must consider mappings into spaces other than ℓ_1.

In Chapter 2, we provide a construction of such mappings. Specifically, for any $\epsilon > 0$, we give a (randomized) linear sketch of dimension only $\tilde{O}(\Delta^\epsilon)$, but from which we can efficiently recover an $O(1/\epsilon)$-approximation of EMD. Note that for point sets in $[\Delta]^2$, the standard representation is as characteristic vectors of dimension Δ^2, so our sketch size is significantly *sublinear.*

This is the first sublinear embedding of EMD that yields a constant approximation. It also immediately gives an improved algorithm (in approximation factor) for the streaming EMD problem, as well as an improved algorithm for the approximate nearest neighbor problem under EMD.

The results presented in this chapter are based on joint work with Alexandr Andoni, Piotr Indyk and David Woodruff [ADIW09].

1.2 Lower Bounds for Sparse Recovery (Chapter 3)

The problem of *stable sparse recovery* is defined as follows: devise a matrix $\Phi \in \mathbb{R}^{m \times n}$ (or a distribution over such matrices) and an algorithm $\mathcal{R}$ such that, given any signal $x \in \mathbb{R}^n$, $\mathcal{R}$ will recover from the sketch Φx a vector $\hat{x}$ satisfying

$$\|x - \hat{x}\|_p \leq C \min_{k\text{-sparse } x'} \|x - x'\|_q \tag{1.1}$$

(with probability 3/4 in the randomized case) for some norm parameters p and q and an approximation factor C (in the case where $p \neq q$, possibly dependent on k). Sparse recovery has a tremendous number of applications in areas such as medical and geological imaging [CRT06, Don06, DDT$^+$08], genetic data acquisition and analysis [SAZ10, BGK$^+$10] and data stream algorithms [Mut05, Ind07b].

It is known that there exist matrices Φ and associated recovery algorithms that produce approximations $\hat{x}$ satisfying Equation (1.1) with $p = q = 1$ or $p = q = 2$ (i.e., the "ℓ_1/ℓ_1" and "ℓ_2/ℓ_2" guarantees, respectively), constant C and sketch length

$$m = O(k \log(n/k)) \ . \tag{1.2}$$

In particular, a random Gaussian matrix [CRT06] or a random sparse binary matrix [BGI$^+$08] (building on [CCFC02, CM05]) with $m = O(k \log(n/k))$ rows satisfies the ℓ_1/ℓ_1 guarantee (constant C) with overwhelming probability. A similar bound was later obtained for ℓ_2/ℓ_2 [GLPS10] (building on [CCFC02, CM05, CM06]); specifically, for $C = 1 + \epsilon$, they provide a distribution over matrices Φ with $m = O((k/\epsilon)\log(n/k))$ rows, together with an associated recovery algorithm. In comparison, using a *non-linear* approach, one can obtain a shorter sketch of length $O(k)$: it suffices to store the k coefficients with the largest absolute values, together with their indices.

Surprisingly, it was not known whether the $O(k \log(n/k))$ bound for linear sketching could be improved upon in general, although matching or nearly matching lower bounds were known to hold under certain restrictions (see Chapter 3 for a more detailed overview). This raised hope that the $O(k)$ bound might be achievable even for general vectors x. Such a scheme would have been of major practical interest, since the sketch length determines the compression ratio, and for large n any extra $\log n$ factor worsens that ratio tenfold.

In Chapter 3, we show that, unfortunately, such an improvement is not possible; specifically, that the bound in (1.2) is asymptotically optimal for $C = O(1)$ and $p = q = 1$ or $p = q = 2$, or $C = O(k^{-1/2})$ and $p = 2$, $q = 1$. Thus, our results show that linear compression is inherently more costly than the simple non-linear approach.

The results presented in this chapter are based on joint work with Piotr Indyk, Eric Price and David Woodruff [DIPW10].

1.3 Sparse Recovery with Partial Support Knowledge (Chapter 4)

Although we have shown the "extra" logarithmic factor multiplying k to be necessary, in many applications we may have some additional knowledge about the signal which allows us to bypass this lower bound and achieve a smaller number of measurements.

The challenge of incorporating external knowledge into the sparse recovery process has received a fair amount of attention in recent years [Def10]. Approaches include *model-based compressed sensing* [BCDH10, EB09] (where the sets of large coefficients are known to exhibit some patterns), *Bayesian compressed sensing* [CICB10] (where the signals are generated from a known distribution) and support restriction (see Chapter 4 for an overview).

In Chapter 4, we study the last type of external knowledge. In particular, we consider the scenario where, after the measurements are taken, we are given a set S of size $s \ll n$ (s is known beforehand) that is supposed to contain most of the "large" coefficients of x. The goal is then to recover $\hat{x}$ satisfying

$$\|x - \hat{x}\|_p \leq C \min_{\substack{k\text{-sparse } x' \\ \operatorname{supp}(x') \subseteq S}} \|x - x'\|_q \ . \tag{1.3}$$

We refer to this formulation as the *sparse recovery with partial support knowledge problem* (SRPSK).

There are several scenarios where our formulation could be applicable. For example, for tracking tasks, the object position typically does not change much between frames, so one can limit the search for current position to a small set. The framework can also be useful for exploratory tasks, where there is a collection S of sets, one of which is assumed to contain the support. In that case, setting the probability of failure to $O(\frac{1}{|S|})$ enables exploring all sets in the family and finding the one which yields the best approximation.

We show that SRPSK can be solved, up to an approximation factor of $C = 1 + \epsilon$, using $O((k/\epsilon)\log(s/k))$ measurements, for $p = q = 2$. Moreover, we show that this bound is tight as long as $s = O(\epsilon n / \log(n/\epsilon))$. This completely resolves the asymptotic measurement complexity of the problem except for a very small range of the parameter s. We also give a similar lower bound for the $p = q = 1$ case, as well as a general method to convert certain "standard" sparse recovery algorithms into ones for SRPSK.

From a theoretical perspective, our results provide a smooth tradeoff between the bound of $\Theta(k \log(n/k))$ known for sparse recovery (i.e., $s = n$) and the bound of $\Theta(k)$ known for the *set query problem* [Pri11], where we have full knowledge of the signal support (i.e., $s = k$). To the best of our knowledge, this was the first variant of $(1+\epsilon)$-approximate sparse recovery to have its asymptotic measurement complexity determined. More recently, [PW11] has presented an optimal lower bound of $\Omega((k/\epsilon)\log(n/k))$ for sparse recovery with the ℓ_2/ℓ_2 guarantee, which matches what our lower bound for SRPSK would suggest (if the restriction on s could be lifted).

The results presented in this chapter are based on joint work with Piotr Indyk [DI11].

Chapter 2

Efficient Sketches for Earth-Mover Distance

Background

The Earth-Mover Distance was first introduced in the vision community as a measure of (dis)similarity between images that more accurately reflects human perception than the more traditional ℓ_1-distance, and has since become an important notion in the field [PWR89, CG99, RTG00, RT99]. Variants are also known as the *transportation distance* or *bichromatic matching distance.* Computing the minimum cost bichromatic matching is one of the most fundamental problems in geometric optimization, and there has been an extensive body of work focused on designing efficient algorithms for this problem [Law76, Vai89, AES95, AV99, Cha02, IT03, AV04, Ind07a].

The particular form of the problem we focus on in this thesis is a *geometric representation* of EMD. As mentioned, one application is in the problem of computing EMD over a stream, which has been a standing open problem in the streaming community [McG06]. A second application is in *visual search and recognition.* The embedding of [Cha02, IT03], together with efficient nearest neighbor search methods, have been applied to fast image search in large collections of images [IT03]. Kernel variants of that embedding, such as *pyramid kernels* and *spatial pyramid kernels*, are some of the best known practical methods for image recognition in large data sets [GD05, LSP06].

Main Results

In this chapter we will construct a strongly sublinear-sized linear sketch of point sets in the plane, together with a reconstruction algorithm that yields a $(1+\epsilon)$-approximation of EMD. Specifically, we will prove the following theorem.

Theorem 1. *For any* $0 < \epsilon < 1$*, there is a distribution over linear mappings* $\Phi : \mathbb{R}^{\Delta^2} \to \mathbb{R}^{\Delta^\epsilon r}$*, for* $r = \log^{O(1)} \Delta$*, as well as an estimator function* $\mathcal{E}(\cdot,\cdot)$ *such that for any two multisets* $A, B \subseteq [\Delta]^2$ *of equal size, we have*

$$EMD(A,B) \leq \mathcal{E}(\Phi x(A), \Phi x(B)) = O(1/\epsilon) \cdot EMD(A,B)$$

with probability $2/3$*. The estimator function* $\mathcal{E}$ *can be evaluated in time* $\log^{O(1)} \Delta$*.*

Note that $\mathcal{E}(\cdot,\cdot)$ is *not* a metric distance function. Instead, it involves operations such as median, and as a result it does not satisfy triangle inequality.

Theorem 1 immediately provides improved algorithms for streaming and nearest neighbor search problems. Consider the aforementioned problem of computing the EMD between the sets A and B of points given in a stream. Note that the linearity of the sketches $\Phi x(A)$ and $\Phi x(B)$ allows them to be maintained under insertions of points to A and B (as well as deletions of points from A and B). Moreover, per [Ind00], the random bits defining a linear mapping Φ can be generated using a pseudo-random generator with bounded space [Nis90] that requires generating and storing only $\Delta^{\epsilon}\log^{O(1)}(\Delta N)$ truly random bits. Finally, our construction guarantees that the entries in the matrix defining Φ are integers in the range $\{-\Delta^{O(1)},\ldots,\Delta^{O(1)}\}$. As a result, for any multiset A of size at most N, each coordinate of $\Phi x(A)$ is in the range $\{-(\Delta N)^{O(1)},\ldots,(\Delta N)^{O(1)}\}$ and can be stored using $O(\log(\Delta N))$ bits. We obtain the following theorem:

Theorem 2. *For any $0<\epsilon<1$, there is a one-pass streaming algorithm that maintains an $O(1/\epsilon)$-approximation of the value of EMD between point-sets from $[\Delta]^2$ given in a stream of length N, using $\Delta^{\epsilon}\log^{O(1)}(\Delta N)$ space.*

Another application of Theorem 1 is to give an improved data structure for the approximate nearest neighbor problem under EMD. Specifically, consider a collection $\mathcal{S}$ of s multisets $A_i \subseteq [\Delta]^2$, each of size at most N. By increasing the dimension of the mapping Φ by a factor of $O(\log s)$ we can ensure that, for any fixed multiset B, one can estimate the distance between B and *all* sets in $\mathcal{S}$ up to a factor of $O(1/\epsilon)$ with probability 2/3. We build a lookup table that, for each value of $\Phi x(B)$, stores the index i that minimizes the value of the estimated distance $\mathcal{E}(\Phi x(A_i),\Phi x(B))$. From the properties of the mapping Φ, we obtain the following theorem:

Theorem 3. *Let $\mathcal{S}$ be a collection of s multisets from $[\Delta]^2$, each of size N. For any $0<\epsilon<1$, there is a data structure that, given a "query" multiset B, reports an $O(1/\epsilon)$-approximate nearest neighbor under EMD of B in $\mathcal{S}$ with probability at least 2/3. The data structure uses $2^{\Delta^{\epsilon}\log^{O(1)}(s\Delta N)}$ space and $(\Delta\log(s\Delta N))^{O(1)}$ query time.*

Thus, we obtain a data structure with very fast query time and space sub-exponential in the dimension Δ^2 of the underlying EMD space. This improves over the result of [AIK09], who obtained an algorithm with a similar space bound while achieving super-constant approximation and query time polynomial in the number of data points s.

Techniques

Our mapping utilizes two components: one old and one new. The first component, introduced in [Ind07a], provides a decomposition of EMD over $[\Delta]^2$ into a convex combination of closely related metrics, called EEMD, defined over $[\Delta^{\epsilon}]^2$. Specifically, consider an extension of EMD to any sets $A, B \subseteq [\Delta]^2$, not necessarily of the same size, defined as:

$$EEMD_{\Delta}(A,B) = \min_{A'\subseteq A, B'\subseteq B, |A'|=|B'|}[EMD(A',B') + \Delta(|A-A'|+|B-B'|)]$$

(we often skip the subscript Δ when it is clear from the context). It is known that the EEMD metric can be induced by a norm $\|\cdot\|_{\mathrm{EEMD}}$, such that for any sets A, B we have

$EEMD(A,B) = \|x(A) - x(B)\|_{\mathrm{EEMD}}$ (see Preliminaries below for the definition), where $x(A) \in \mathbb{R}^{\Delta^2}$ denotes the characteristic vector of A. The decomposition from [Ind07a] can now be stated as follows (after adapting the notation to the setup in this chapter):

Fact 4. *For any $0 < \epsilon < 1$, there exists a distribution over T-tuples of linear mappings $\langle\Phi_1, \ldots, \Phi_T\rangle$, for $\Phi_i : \mathbb{R}^{\Delta^2} \to \mathbb{R}^{\Delta^\epsilon}$, such that for any $x \in \mathbb{R}^{\Delta^2}$, we have*

- $\|x\|_{EEMD} \le \sum_i \|\Phi_i(x)\|_{EEMD}$ *with probability* 1, *and*
- $\mathbb{E}\left[\sum_i \|\Phi_i(x)\|_{EEMD}\right] \le O(1/\epsilon) \cdot \|x\|_{EEMD}$.

It suffices to estimate the sum of the terms $\|\Phi_i(x)\|_{\mathrm{EEMD}}$ in the decomposition. The second component needed for our result (and the main technical development of this chapter) is showing that this sum estimation can be accomplished by using an appropriate linear mapping. In fact, the method works for estimating the sum of norms $\sum_i \|x_i\|_X$ of a vector $x = (x_1, \ldots, x_T) \in X^T$ for *any* normed space $X = (\mathbb{R}^m, \|\cdot\|_X)$. We denote $\|x\|_{1,X} = \sum_{i\in[T]} \|x_i\|_X$. This component is formalized in the following theorem.

Theorem 5 (Linear sketching of a sum of norms). *Fix $n \in \mathbb{N}$, a threshold $M > 0$, and approximation $\gamma > 1$. For $k = (\gamma \log n)^{O(1)}$, there exists a distribution over random linear mappings $\varphi : X^n \to X^k$, and a reconstruction algorithm $\mathcal{A}$, such that for any $x \in X^n$ satisfying $M/\gamma \le \|x\|_{1,X} \le M$, the algorithm $\mathcal{A}$ produces w.h.p. an $O(1)$-approximation to $\|x\|_{1,X}$ from $\varphi(x)$.*

Theorem 5 implies Theorem 1, since we can use the mapping from [Cha02, IT03] to obtain an estimation M of $\|x\|_{1,\mathrm{EEMD}}$ with an approximation factor $\gamma = O(\log\Delta)$. For completeness, we include its proof in Section 2.2.

The main idea behind the construction of the mapping is as follows. First, observe that a natural approach to the sum estimation problem would be to randomly sample a few elements x_i of the vector x. This does not work, however: the mass of the sum could be concentrated in only a single element, and a random sample would likely miss it. An alternative approach, used in the *off-line* algorithm of [Ind07a], is to sample each element x_i with probability approximately proportional to $\|x_i\|_X$, or in the case of EMD, $\|x_i\|_{\mathrm{EEMD}}$. However, it is not clear how this can be done from a small sketch. In particular, for a direct application to the streaming problem, this would require the existence of a streaming algorithm that supports such sampling. [JW09] provides a step towards achieving such an algorithm. However, it applies to the case where one samples just individual coordinates, while we need to sample and retrieve "blocks", [1] in order to then compute the EMD on them directly. Although the two problems are related in principle (having enough samples of block coordinates could provide some information about the norm of the block itself), the tasks seem technically different. Indeed, the sketching and recovery procedure for this sum of norms forms the main technical part of the chapter, even though the final algorithm is quite simple.

Preliminaries

We start by defining the $\|\cdot\|_{\mathrm{EEMD}}$ norm. For any $x \in \mathbb{R}^{\Delta^2}$, let $x^+ = (|x| + x)/2$ be the vector containing only the positive entries in x, and let $x^- = x - x^+$. Then, if A^+ denotes

[1] There are other technical obstacles such as that their algorithm samples with probability proportional to $|x_i|^p$ for $p > 2$, while here we would need the sampling probability to be proportional to the ℓ_1-norm of x_i, i.e., $p = 1$.

the (possibly fractional) multiset for which x^+ is the characteristic vector, and similarly A^- denotes the set with characteristic vector $-x^-$, define $\|x\|_{\text{EEMD}} = EEMD(A^+, A^-)$. Observe that for any sets $A, B \subseteq [\Delta]^2$ we have $EEMD(A, B) = \|x(A) - x(B)\|_{\text{EEMD}}$.

The notation $\chi[E]$ stands for 1 if expression E is true and 0 otherwise.

2.1 Sketching a Sum of Norms

2.1.1 Sketch and Reconstruction Algorithm

We start by giving some intuition behind our construction of the sketching function φ and of the reconstruction algorithm $\mathcal{A}$. The respective algorithms are presented in Figures 1 and 2.

Fix an input $x \in X^n$. We will refer to x_i's as the *elements* of x. As in [IW05] and several further papers, the idea is to partition these elements into exponential levels, depending on their X-norm. Specifically, for a level $j \in \mathbb{N}$, we set the threshold $T_j = M/2^j$ and define the level j to be the set

$$L_j = \left\{ i \in [n] \mid \|x_i\|_X \in (T_j, 2T_j] \right\} .$$

Let $s_j = |L_j|$ be the size of L_j. We will observe that $\|x\|_{1,X}$ is approximated by $\sum_{j \geq 1} T_j s_j$. Furthermore, it is sufficient to consider only levels $j \leq \ell := \log(4n\gamma)$. Henceforth, we will drop the "$j \in [\ell]$" in the summation.

The main challenge is to estimate each s_j for $j \in [\ell]$. We will do so for each j separately. We will subsample the elements from $[n]$ such that, with "good probability", we subsample exactly one element from L_j and no element from $L_{j'}$ for $j' < j$. We refer to this event as E. This "isolation" of an element of i is needed since in order to verify if $i \in L_j$, we need to estimate $\|x_i\|_X$, which requires the recovery of an "approximation" of x_i.

The probability that E holds is in fact roughly proportional to the size of the set L_j, and thus it suffices to just estimate the probability that E holds. To ensure the "rough proportionality" we subsample the elements at a rate for which E holds with a probability that is inversely poly-logarithmic, $\log^{-\Theta(1)} n$. We can then repeat the subsampling experiment $t = (\gamma \log n)^{O(1)}$ times and count the number of experiments wherein the event E holds; this count gives an estimate for s_j (appropriately scaled).

The following core problem remains: for each subsampling experiment $u \in [t]$, we need to actually verify that E holds in this experiment, i.e., whether exactly one element of L_j is subsampled and no element from $L_{j'}$ for $j' < j$. To do so, we hash the subsampled elements, denoted $I_{j,u}$, into a table. Then, E holds roughly when there is one cell that has norm in the right range, which is roughly $(T_j, 2T_j]$, and all the other cells are small. Ideally, if the hash table were huge, then the subsampled elements, $I_{j,u}$, do not collide, in which case the verification procedure is accurate. Since the hash table size is much smaller, of only polylogarithmic size, this verification procedure may fail. Specifically, the verification procedure fails when either the elements from the "lighter" levels $L_{j'}$ for $j' > j$ contribute a lot to one of the cells, or multiple elements from "heavier" levels $L_{j'}$ for $j' < j$ are subsampled *and collide*. If we set the size w of the hash table sufficiently high, we will ensure that neither of these two bad events happens with any significant probability.

The detailed algorithm for the sketch φ is presented in Figure 1. Note that the constructed sketch φ is linear.

Before giving the reconstruction algorithm $\mathcal{A}$, we need the following definition, which describes our procedure of verifying that the event E from the above discussion occurs.

```
For each $j \in [\ell]$, create $t = 4\gamma\ell^2 \log n$ hash tables, denoted $H^{(j,u)}$ for $u \in [t]$, each
  with $w = 640\gamma\ell^2 \log^2 n$ cells, and assign to them independent hash functions
  $h_{j,u} : [n] \to [w]$
For each hash table $H^{(j,u)}$
    Subsample a set $I_{j,u} \subset [n]$ where each $i \in [n]$ is included independently with
      probability $p_j = 2^{-j}/(40\ell)$
    For each $v \in [w]$
        $H_v^{(j,u)} := \sum_{i\in[n]} \chi[i \in I_{j,u}] \cdot \chi[h_{j,u}(i) = v] \cdot x_i$
```

Algorithm 1: Construction of the sketch φ.

```
For each $j \in [\ell]$, let $c_j$ count the number of accepting pairs $(j,u)$ for $u \in [t]$
Return $\alpha = \sum_j T_j \cdot \frac{c_j}{t} \cdot \frac{1}{p_j}$
```

Algorithm 2: Reconstruction algorithm $\mathcal{A}$.

Definition 6. *For $j \in [\ell]$, $u \in [t]$, call the pair (j,u) an* accepting pair *if the following holds:*

- *there is exactly one position $v \in [w]$ such that $\|H_v^{(j,u)}\|_X \in (0.9T_j, 2.1T_j]$, and*
- *for all other $v' \in [w]$, $\|H_{v'}^{(j,u)}\|_X \le 0.9T_j$.*

The resulting reconstruction algorithm is given in Figure 2.

2.1.2 Proof of Correctness

First we observe that the norm $\|x\|_{1,X}$ is approximated by $\sum_{j\in[\ell]} T_j s_j$ up to a factor of 4. Indeed, $\|x\|_{1,X}$ is 2-approximated by the same sum with unrestricted j, i.e., $\sum_{j\ge1} T_j s_j$. Moreover, every element $i \in [n]$ from a higher level $j > \ell$ contributes a norm that is at most

$$\|x_i\|_X \le \frac{M}{2^\ell} = \frac{1}{4n} \cdot \frac{M}{\gamma} \le \frac{1}{4n}\|x\|_{1,X} \ .$$

Thus the elements from the ignored levels contribute at most a quarter of $\|x\|_{1,X}$.

For notational convenience, we therefore assume that for $j \notin [\ell]$, we have $L_j = \varnothing$, i.e., $s_j = 0$. Also, we can assume that $\gamma \le n^c$ for some absolute constant $c > 0$, since, otherwise, the construction with $k = \gamma^{1/c}$ is trivial.

We define $\tilde{s}_j = \frac{c_j}{t} \cdot \frac{1}{p_j}$, which is our estimate of s_j. Then the reconstruction algorithm returns the estimate $\alpha = \sum_j T_j \tilde{s}_j$ of the norm $\|x\|_{1,X}$.

Our main challenge is to prove that $\tilde{s}_j$ is a good estimate of s_j for each $j \in [\ell]$. While we can prove a good upper bound on $\tilde{s}_j$ for all $j \in [\ell]$, we cannot prove a good lower bound on all $\tilde{s}_j$'s. Namely, if s_j is very small, we cannot lower-bound $\tilde{s}_j$ (as we do not have enough subsampling experiments). But in this case, the level j contributes a negligible mass to the norm $\|x\|_{1,X}$, and thus it can simply be ignored.

To formalize the above point, we partition the levels j into two types — important and unimportant levels — depending on both the number s_j of elements in, and the norm range of, each one. Intuitively, the unimportant levels are those which contribute a negligible amount of mass to the norm $\|x\|_{1,X}$.

Definition 7. *Call level j* important *if $s_j \geq \frac{M/\gamma}{T_j} \cdot \frac{1}{8\ell} = \frac{2^j}{8\gamma\ell}$. Call level j* unimportant *if it is not important. Let $\mathcal{J}$ denote the set of important levels.*

The following two lemmas prove, respectively, lower and upper bounds on our estimates $\tilde{s}_j$.

Lemma 8. *For every important level $j \in \mathcal{J}$, with high probability,*

$$\tilde{s}_j \geq s_j/8 \ .$$

Lemma 9. *For every level $j \in [\ell]$, with high probability,*

$$\tilde{s}_j \leq 2\left(s_{j-1} + s_j + s_{j+1} + \frac{2^j}{8\gamma\ell}\right) \ .$$

First, we show how the two lemmas are used to prove Theorem 5.

Proof of Theorem 5. We have already observed that $\sum_j T_j s_j$ approximates $\|x\|_{1,X}$ up to a factor of 4. Thus, by Lemma 9, we have

$$\begin{aligned} \alpha &= \sum_j T_j \tilde{s}_j \\ &\leq O(1) \sum_j T_j \left(s_{j-1} + s_j + s_{j+1} + \frac{2^j}{8\gamma\ell}\right) \\ &\leq O(1) \sum_j T_j s_j + O(\ell) \cdot \frac{M}{8\gamma\ell} \\ &\leq O(1) \cdot \|x\|_{1,X} \ , \end{aligned}$$

where we have used the fact that $\|x\|_{1,X} \geq M/\gamma$.

On the other hand, we can lower bound α by dropping all the unimportant levels j. By Lemma 8, we have

$$\alpha \geq \sum_{j \in \mathcal{J}} T_j \tilde{s}_j \geq \Omega(1) \sum_{j \in \mathcal{J}} T_j s_j \ .$$

The contribution of the unimportant levels is, by the definition of importance,

$$\sum_{j \notin \mathcal{J}} T_j s_j < \ell \cdot \frac{M/\gamma}{8\ell} \leq \frac{1}{8} \|x\|_{1,X} \ .$$

Thus, we conclude

$$\sum_{j \in \mathcal{J}} T_j s_j = \sum_j T_j s_j - \sum_{j \notin \mathcal{J}} T_j s_j \geq \frac{1}{4}\|x\|_{1,X} - \frac{1}{8}\|x\|_{1,X} = \Omega(1) \cdot \|x\|_{1,X} \ ,$$

which completes the proof of Theorem 5. □

2.1.3 Proofs of Lemmas 8 and 9

As mentioned before, at a given level j, we are trying to estimate the size s_j of the set L_j. We do so by subsampling the elements t times, each at a rate of roughly $1/s_j$, and counting the number of times the subsampling produces exactly one element from L_j. The hope is that the pair (j, u) is accepting iff the event E holds, that is, the subsample $I_{j,u}$ contains exactly one element from L_j and none from $L_{j'}$ for $j' < j - 1$. The main difficulty turns out to be bounding the contribution of the elements from the sets $L_{j'}$ for $j' \geq j + 2$: the sets $L_{j'}$ may be much larger than L_j and thus a fraction of them is likely to be present in the subsample. Fortunately, the elements from these sets $L_{j'}$ are small in norm and thus are distributed nearly uniformly in the hash table $H^{(j,u)}$.

To formalize this intuition, we will prove the Noise Lemma (Lemma 10) that quantifies the "noise" (norm mass) contributed by the elements from the sets $L_{j'}$, for $j' \geq j + 2$, in a hash table $H^{(j,u)}$. This will be used to prove both Lemma 8 and Lemma 9.

The Noise Lemma has two parts. The first part gives a strong bound on the noise in a given cell of the hash table $H^{(j,u)}$, but the probability guarantee is for a given cell only. The second part gives a somewhat weaker bound on the noise, but holds for *all* the cells of $H^{(j,u)}$ simultaneously.

To simplify notation, denote by $L_{\geq j}$ the union $\bigcup_{j' \geq j} L_{j'}$, and similarly, $L_{\leq j}$.

Lemma 10 (Noise Lemma). *Fix some $j \in [l]$ and $u \in [t]$. Consider some cell v of the hash table $H^{(j,u)}$. Then*

$$\sum_{i \in L_{\geq j+2}} \chi[i \in I_{j,u}] \cdot \chi[h_{j,u}(i) = v] \cdot \|x_i\|_X \leq 0.1T_j \tag{2.1}$$

with probability at least $1 - \frac{1}{2w}$.

Furthermore, with probability at least $1 - \frac{\log^2 n}{w}$, we have

$$\max_{v' \in [w]} \sum_{i \in L_{\geq j+2}} \chi[i \in I_{j,u}] \cdot \chi[h_{j,u}(i) = v'] \cdot \|x_i\|_X \leq 0.6T_j \ . \tag{2.2}$$

Proof. We begin by proving equation (2.1). Consider some level $j' \geq j + 2$. Level j' contains $s_{j'} \leq 2^{j'}$ elements, each subsampled with probability p_j and hashed to v with probability $1/w$. Thus, we can write

$$\begin{aligned} \mathbb{E}\left[\sum_{i \in L_{j'}} \chi[i \in I_{j,u}] \cdot \chi[h_{j,u}(i) = v] \cdot \|x_i\|_X\right] &= \sum_{i \in L_{j'}} \frac{p_j}{w} \cdot \|x_i\|_X \\ &\leq 2^{j'} \cdot \frac{2^{-j}}{40\ell w} \cdot 2T_{j'} = \frac{T_j}{20\ell w} \ . \end{aligned} \tag{2.3}$$

Then, denoting by LHS the left-hand side of Equation (2.1), we have

$$\mathbb{E}[\text{LHS}] \leq \sum_{j' \geq j+2} \frac{T_j}{20\ell w} \leq \frac{T_j}{20w} \ .$$

Using the Markov bound, we can thus conclude that $\mathbb{P}[\text{LHS} \geq 0.1T_j] \leq \frac{1}{2w}$, which proves the first part of the Noise Lemma.

We now prove the second part of the Noise Lemma, Equation (2.2). Note that we cannot hope to prove that *all* cells will have noise at most $0.1T_j$, because even just one element from a set L_{j+2} can contribute as much as $T_j/2$. To prove this part, we partition the elements in $L_{\geq j+2}$ into two types: *heavy* elements (of mass close to T_j) and *light* elements (of mass much smaller than T_j). For heavy elements, we will prove that we subsample only a few of them, and thus they are unlikely to collide in the hash table. The light elements as so light that they can be upper-bounded using a concentration bound.

Specifically, we define the following sets of light and heavy elements, respectively:

$$\begin{aligned} L_l &:= \bigcup_{j' \geq j+\log\log n+1} L_{j'} \\ L_h &:= L_{\geq j+2} \setminus L_l = \bigcup_{j+2 \leq j' < j+\log\log n+1} L_{j'} \ . \end{aligned}$$

We first show that the light elements do not contribute more than $0.1T_j$ to *any* cell w.h.p. Namely, we will bound the noise in a cell $v' \in [w]$ using a Hoeffding bound, and then use a union bound over all v'. We use the following variant of Hoeffding's inequality, which can be deduced from [Hoe63].

Lemma 11 (Hoeffding). *Let $Z_1, \ldots, Z_n$ be n independent random variables such that for every i, $Z_i \in [0, D]$, for $D > 0$, and $\mathbb{E}[\sum_i Z_i] = \mu$. Then, for any $a > 0$, we have that*

$$\mathbb{P}\left[\sum_i Z_i > a\right] \leq e^{-(a-2\mu)/D} \ .$$

We use the lemma for variables $Z_i = \chi[i \in I_{j,u}] \cdot \chi[h_{j,u}(i) = v'] \cdot \|x_i\|_X$, where $i \in L_l$. To get a bound D on each Z_i, observe that, for $i \in L_l$, we have

$$\|x_i\|_X \leq T_{j+\log\log n} = T_j/2^{\log\log n} = T_j/\log n \ .$$

We also have an upper bound on the expected sum of

$$\mu = \mathbb{E}[\sum_{j \in L_l} Z_i] \leq T_j/(20w)$$

by summing up the bound in Equation (2.3) over all ℓ levels. Thus, applying Lemma 11, we obtain

$$\begin{aligned} \mathbb{P}\left[\sum_{i \in L_l} \chi[i \in I_{j,u}] \cdot \chi[h_{j,u}(i) = v'] \cdot \|x_i\|_X > (0.1)T_j\right] &\leq e^{-(0.1-1/(10w))T_j/(T_j/\log n)} \\ &< e^{-0.05\log n} = n^{-\Omega(1)} \ . \end{aligned}$$

Taking the union bound over all cells, we obtain the same bound on all cells $v' \in [w]$.

We now analyze the behavior of the heavy elements, i.e., elements from the set L_h. We can bound the expected number of subsampled heavy elements as follows:

$$\mathbb{E}\left[\sum_{i \in L_h} \chi[i \in I_{j,u}]\right] \leq \left(\sum_{j'=j+2}^{j+\log\log n} 2^{j'}\right) p_j < 2^{j+\log\log n+1} \cdot \frac{2^{-j}}{40\ell} = \frac{\log n}{20\ell} \leq O(1) \ .$$

Applying the Hoeffding bound from above, we obtain

$$\mathbb{P}\left[\sum_{i\in L_h} \chi[i \in I_{j,u}] > \log n\right] \leq e^{-\Omega(\log n)} = n^{-\Omega(1)} \ .$$

Thus, w.h.p., no more than $\log n$ heavy elements are subsampled. Given this, we can further bound the probability that any two of them hash into the same cell via a union bound by

$$\binom{\log n}{2}/w \leq \frac{\log^2 n}{2w} \ .$$

To conclude, with probability at least

$$1 - \frac{\log^2 n}{2w} - n^{-\Omega(1)} > 1 - \frac{\log^2 n}{w} \ ,$$

no cell receives a contribution of more than $0.1T_j$ from the light elements or more than $T_j/2$ from the heavy elements, for a total of $0.6T_j$, as required. □

We are now ready to prove Lemmas 8 and 9.

Proof of Lemma 8. Fix an important $j \in \mathcal{J}$. For each hash table $H^{(j,u)}$, for $u \in [t]$, let $A_{j,u}$ denote the event that (j,u) is an accepting pair. Define the following two events:

$E1$: exactly one element of L_j is subsampled in $I_{j,u}$, and

$E2$: no element from $L_{j'}$ is subsampled in $I_{j,u}$, for all $j' < j$ and $j' = j+1$.

We will prove the following claim.

Claim 12. *For fixed $u \in [t]$, if $E1$ and $E2$ hold, then $A_{j,u}$ occurs with probability at least 1/2. Moreover, $E1$ and $E2$ occur simultaneously with probability at least $\frac{1}{2}s_j p_j$.*

Proof of Claim 12. To prove the first part, assume $E1$ and $E2$ hold. Let i^* be the the element in $I_{j,u} \cap L_j$ (guaranteed to be unique by $E1$), and let v^* be the cell that contains element i^*. First, we note that, using the triangle inequality in X and the Noise Lemma 10, we have

$$\|H_{v^*}^{(j,u)}\|_X \geq \|x_{i^*}\|_X - \sum_{i\in I_{j,u}\setminus\{i^*\}} \chi[h_{j,u}(i) = v^*] \cdot \|x_i\|_X > T_j - 0.1T_j = 0.9T_j \ ,$$

and

$$\|H_{v^*}^{(j,u)}\|_X \leq \|x_{i^*}\|_X + \sum_{i\in I_{j,u}\setminus\{i^*\}} \chi[h_{j,u}(i) = v^*] \cdot \|x_i\|_X \leq 2.1T_j \ ,$$

with probability at least $1 - 1/(2w) \geq 3/4$. Furthermore, for every other cell $v \neq v^*$, we have that, similarly to the above:

$$\max_{v\neq v^*} \|H_v^{(j,u)}\|_X \leq \max_{v\neq v^*} \sum_{i\in I_{j,u}} \chi[h_{j,u}(i) = v] \cdot \|x_i\|_X \leq 0.6T_j$$

with probability at least $1 - \log^2 n/w \geq 3/4$. Thus, all three hold at the same time with probability at least 1/2, in which case $A_{j,u}$ occurs.

Next we show that $E1$ and $E2$ hold with probability at least $\frac{1}{2}s_j p_j$. We have

$$\mathbb{P}[E1] = s_j p_j (1-p_j)^{s_j-1} \geq s_j p_j (1 - s_j p_j) \geq \tfrac{2}{3} s_j p_j \ ,$$

where we use the fact that $s_j \leq 2^j = \frac{1}{40\ell p_j}$. To estimate $\mathbb{P}[E2]$, we first consider all $j' < j$. We can bound the probability that anything from $\bigcup_{j'<j} L_{j'}$ is subsampled by

$$\mathbb{P}\left[\bigcup_{j'<j} L_{j'} \cap I_{j,u} \neq \varnothing\right] \leq \sum_{j'<j} s_{j'} p_j \leq \sum_{j'<j} 2^{j'} p_j < 2^j p_j = \frac{1}{40\ell} \ .$$

Similarly, we have

$$\mathbb{P}[L_{j+1} \cap I_{j,u} \neq \varnothing] \leq s_{j+1} p_j \leq \frac{1}{20\ell} \ .$$

Thus we obtain $\mathbb{P}[E2] \geq 1 - \frac{1}{10\ell}$.

Note that $E1$ and $E2$ are indendent events since they concern different levels. We can therefore conclude that

$$\mathbb{P}[E1 \wedge E2] = \mathbb{P}[E1] \cdot \mathbb{P}[E2] \geq \tfrac{2}{3} s_j p_j \left(1 - \frac{1}{10\ell}\right) \geq \tfrac{1}{2} s_j p_j \ ,$$

which finishes the proof of Claim 12. □

We now complete the proof of Lemma 8. We can lower bound the probability of $A_{j,u}$ as follows:

$$\mathbb{P}[A_{j,u}] \geq \mathbb{P}[A_{j,u} \wedge E1 \wedge E2] = \mathbb{P}[A_{j,u} \mid E1 \wedge E2] \cdot \mathbb{P}[E1 \wedge E2] \geq \tfrac{1}{4} s_j p_j \ .$$

Now, we can finally analyze the estimate $\tilde{s}_j$ of the size of the set L_j. Since $\tilde{s}_j = \frac{c_j}{t} \cdot \frac{1}{p_j}$, we will lower bound c_j. Observe that

$$\mathbb{E}[c_j] = t\mathbb{P}[A_{j,u}] \geq \frac{t}{4} s_j p_j \geq \frac{t}{4} \cdot \frac{2^j}{8\gamma\ell} \cdot \frac{2^{-j}}{40\ell} \geq \Omega(\log n) \ .$$

A standard application of the Chernoff bound suffices to conclude that $c_j \geq \frac{t}{8} s_j p_j$, w.h.p., and thus

$$\tilde{s}_j = \tfrac{c_j}{t} \cdot \tfrac{1}{p_j} \geq \tfrac{1}{8} s_j p_j \cdot \frac{1}{p_j} = \tfrac{1}{8} s_j \ ,$$

also w.h.p. This concludes the proof of Lemma 8. □

We now prove Lemma 9 which upper-bounds the estimate $\tilde{s}_j$.

Proof of Lemma 9. First, fix any important j, and consider any particular hash table $H^{(j,u)}$. As before, let $A_{j,u}$ denote the event that (j,u) is an accepting pair, and define the following new event:

E3: at least one element of $L_{j-1} \cup L_j \cup L_{j+1}$ is subsampled.

Claim 13. *If E3 does not occur, $A_{j,u}$ holds with probability at most $p_j\left(\frac{2^j}{8\gamma\ell}\right)$. Moreover, E3 holds with probability at most $p_j(s_{j-1} + s_j + s_{j+1})$.*

Proof. For the first part, we prove that, with probability at least $1 - p_j\left(\frac{2^j}{8\gamma\ell}\right)$, no cell of $H^{(j,u)}$ can have a norm that is in the accepting range of $(0.9T_j, 2.1T_j]$. A cell v of $H^{(j,u)}$ may have a norm in the accepting range only when one of the following occurs: (1) more than one element from $L_{\le j-2}$ falls into v, or (2) the noise in v from elements in $L_{\ge j+2}$ exceeds $0.6T_j$. In particular, if neither (1) nor (2) hold, then either v contains no element from $L_{\ge j+2}$, in which case $\|H_v^{(j,u)}\|_X \le 0.6T_j \le 0.9T_j$, or v contains exactly one element from $L_{\ge j+2}$, in which case $\|H_v^{(j,u)}\|_X > 4T_j - 0.6T_j > 2.1T_j$.

Now, the probability that (2) holds for *any* cell v is at most $\frac{\log^2 n}{w}$ by the Noise Lemma 10. It remains to bound the probability of (1). We note that expected number of subsampled elements from $L_{\le j-2}$ is upper bounded by $2^j p_j \le O(1)$. Thus, with high probability, at most $\log n$ of the elements in $L_{\le j-2}$ appear in $I_{j,u}$. Furthermore, these $\log n$ elements collide with probability at most $\frac{\log^2 n}{2w}$. It follows that the probability that (1) holds for *any* cell v is at most $\frac{\log^2 n}{w}$.

Thus, we have that

$$\mathbb{P}[A_{j,u} \mid \overline{E3}] \le 2 \cdot \frac{\log^2 n}{w} \le p_j\left(\frac{2^j}{8\gamma\ell}\right) = \frac{1}{320\gamma\ell^2} \ .$$

For the second part, we need to bound $\mathbb{P}[E3]$. But this follows from a simple union bound over all elements in $L_{j-1} \cup L_j \cup L_{j+1}$. □

We can now finish the proof of the lemma. From the above claim, we obtain the following bound on the probability of an accepting pair:

$$\mathbb{P}[A_{j,u}] \le \mathbb{P}[A_{j,u} \mid \overline{E3}] + \mathbb{P}[E3] \le p_j\left(s_{j-1} + s_j + s_{j+1} + \frac{2^j}{8\gamma\ell}\right) \ ,$$

We can now upper bound the estimate $\tilde{s}_j$:

$$\mathbb{E}[\tilde{s}_j] = \frac{\sum_u \mathbb{P}[A_{j,u}]}{t} \cdot \frac{1}{p_j} \le \left(s_{j-1} + s_j + s_{j+1} + \frac{2^j}{8\gamma\ell}\right) \ .$$

Again, by the Chernoff bound, $\tilde{s}_j \le 2\left(s_{j-1} + s_j + s_{j+1} + \frac{2^j}{8\gamma\ell}\right)$ w.h.p. This completes the proof of Lemma 9. □

2.2 Sketching EMD

We now prove our main Theorem 1. As mentioned in the introduction, its main ingredient is Theorem 5.

Proof of Theorem 1. The sketch Φ consists of two parts. The first part is just a linear map f of planar EMD into ℓ_1 as in [Cha02, IT03], that approximates the EMD distance up to $\gamma = O(\log \Delta)$ approximation.

The second part is a collection of $O(\log \Delta)$ sketches ν_i. Each ν_i is a composition of two linear maps: the map $\Phi^{(i)} = \langle \Phi_1^{(i)}, \ldots, \Phi_T^{(i)} \rangle$ obtained from an application of Fact 4 and a sketch φ_i obtained from an application of the Theorem 5. Specifically, for $i \le \log \Delta$, the sketch φ_i is given by the Theorem 5 for $M = 2^i$, $n = T$, and γ as defined above. The final sketch is then the following linear map:

$$\Phi = \langle f, \varphi_1 \circ \Phi^{(1)}, \dots, \varphi_{\log \Delta} \circ \Phi^{(\log \Delta)} \rangle \ .$$

The reconstruction algorithm $\mathcal{A}$ works in a straightforward manner. Given sketches $\Phi x(A)$ and $\Phi x(B)$, compute first a γ-approximation to $EMD(A, B)$ using the map f. Then, use the corresponding map $\nu_i = \varphi_i \circ \Phi^{(i)}$ to compute the estimate $\sum_j \|\Phi_j^{(i)}(x(A) - x(B))\|_{\mathrm{EEMD}}$. This estimate is an $O(1/\epsilon)$ approximation to $EMD(A, B)$ by Fact 4, completing our proof. □

Chapter 3

Lower Bounds for Sparse Recovery

Background

There have been a number of earlier works that have, directly or indirectly, shown lower bounds for various models of sparse recovery and certain classes of matrices and algorithms. For example, one of the most well-known recovery algorithms used in compressed sensing is ℓ_1-minimization, where a signal $x \in \mathbb{R}^n$ measured by matrix Φ is reconstructed as

$$\hat{x} := \operatorname*{arg\,min}_{x' \colon \Phi x' = \Phi x} \|x'\|_1 \ .$$

Kashin and Temlyakov [KT07] (building on prior work on Gelfand width [GG91, Glu84, Kas98], see also [Don06]) gave a characterization of matrices Φ for which the above recovery algorithm yields the ℓ_2/ℓ_1 guarantee, i.e.,

$$\|x - \hat{x}\|_2 \leq Ck^{-1/2} \min_{k\text{-sparse } x'} \|x - x'\|_1$$

for some constant C, from which it can be shown that such an Φ must have $m = \Omega(k \log(n/k))$ rows.

The results on Gelfand width can be also used to obtain lower bounds for *general* recovery algorithms (for the deterministic recovery case), as long as the sparsity parameter k is larger than some constant. This was explicitly stated in [FPRU10], see also [Don06].

On the other hand, instead of assuming a specific recovery algorithm, Wainwright [Wai07] assumes a specific (randomized) measurement matrix. More specifically, the author assumes a k-sparse binary signal $x \in \{0, \alpha\}^n$, for some $\alpha > 0$, to which is added i.i.d. standard Gaussian noise in each component. The author then shows that with a random Gaussian matrix Φ, with each entry also drawn i.i.d. from the standard Gaussian, we cannot hope to recover x from Φx with any sub-constant probability of error unless Φ has $m = \Omega(\frac{1}{\alpha^2} \log \frac{n}{k})$ rows. The author also shows that for $\alpha = \sqrt{1/k}$, this is tight, i.e., that $m = \Theta(k \log(n/k))$ is both necessary and sufficient. Although this is only a lower bound for a specific (random) matrix, it is a fairly powerful one and provides evidence that the often observed upper bound of $O(k \log(n/k))$ is likely tight.

More recently, Dai and Milenkovic [DM09], extending on [EG88] and [FR99], showed an upper bound on superimposed codes that translates to a lower bound on the number of rows in a compressed sensing matrix that deals only with k-sparse signals but can tolerate measurement noise. Specifically, if we assume a k-sparse signal $x \in ([-t, t] \cap \mathbb{Z})^n$, and that arbitrary noise $\nu \in \mathbb{R}^n$ with $||\nu||_1 < d$ is added to the measurement vector Φx, then if exact recovery is still possible, Φ must have had $m \geq Ck \log n / \log k$ rows, for some constant $C = C(t, d)$ and sufficiently large n and k.[1]

Main Results

We address two types of recovery schemes:

- A *deterministic* one, which involves a fixed matrix Φ and a recovery algorithm which works for all signals x. The aforementioned results of [CRT06] and others are examples of such schemes.

- A *randomized* one, where the matrix Φ is chosen at random from some distribution, and for each signal x (the choice of Φ is independent of x) the recovery procedure is correct with constant probability (say, 3/4). Some of the early schemes proposed in the data stream literature (e.g., [CCFC02, CM05]) belong to this category.

Our main result is the following lower bound even in the stronger randomized case.

Theorem 14. *For any randomized sparse recovery algorithm $(\Phi, \mathcal{R})$, with the ℓ_1/ℓ_1, ℓ_2/ℓ_2 or ℓ_2/ℓ_1 guarantee and approximation factor $C = O(1)$, Φ must have $m = \Omega(k \log(n/k))$ rows.*

By the aforementioned results of [CRT06, BGI$^+$08, GLPS10] this bound is tight.

Techniques

For simplicity of exposition we first restrict ourselves to the ℓ_1/ℓ_1 case. Mostly technical modifications yield the result for the ℓ_2/ℓ_2 and ℓ_2/ℓ_1 cases.

At a high level, our approach is simple and natural, and utilizes the packing approach: we show that any two "sufficiently" different vectors x and x' are mapped to images Φx and $\Phi x'$ that are "sufficiently" different themselves, which requires that the image space is "sufficiently" high-dimensional. However, the actual arguments are somewhat subtle.

Consider first the (simpler) deterministic case. We focus on signals $x = y + z$, where y can be thought of as the "head" of the signal and z as the "tail". The "head" vectors y come from a set Y that is a binary error-correcting code, with a minimum distance $\Omega(k)$, where each code word has Hamming weight k. On the other hand, the "tail" vectors z come from an ℓ_1 ball (denote B) with a radius that is a small fraction of k. It can be seen that for any two elements $y, y' \in Y$, the balls $y + B$ and $y' + B$, as well as their images under Φ, must be disjoint. At the same time, since all vectors x live in a "large" ℓ_1 ball B' of radius $O(k)$, all images Φx must live in a set $\Phi B'$. The key observation is that the set $\Phi B'$ is a scaled version of $\Phi(y + B)$ and therefore the ratios of their volumes can be bounded by the scaling factor to the power of the dimension m. Since the number of elements of Y is large, this gives a lower bound on m.

Unfortunately, the aforementioned approach does not seem to extend to the randomized case. A natural approach would be to use Yao's principle, and focus on showing a lower bound for a scenario where the matrix Φ is fixed while the vectors $x = y + z$ are "random".

[1] Here Φ is assumed to have its columns normalized to have ℓ_1-norm 1. This is natural since otherwise we could simply scale Φ up to make the image points Φx arbitrarily far apart, effectively nullifying the noise.

However, this approach fails, in a very strong sense. Specifically, we are able to show that there is a distribution over matrices Φ with *only* $O(k)$ rows such that for a fixed $y \in Y$ and z chosen uniformly at random from the small ball B, we can recover y from $\Phi(y+z)$ with high probability. In a nutshell, the reason is that a random vector from B has an ℓ_2 norm that is much smaller than the ℓ_2 norm of elements of Y (even though the ℓ_1 norms are comparable). This means that the vector x is "almost" k-sparse in the ℓ_2 norm, which enables us to achieve the $O(k)$ measurement bound.

Instead, we employ an entirely different approach via *communication complexity* [KN97]. We start by considering a "discrete" scenario where both the matrix Φ and the vectors x have entries restricted to the polynomial range $\{-n^c, \dots, n^c\}$ for some $c = O(1)$. In other words, we assume that the matrix and vector entries can be represented using $O(\log n)$ bits. In this setting we show the following: there is a method for encoding a sequence of $d = O(k \log(n/k) \log n)$ bits into a vector x, so that any sparse recovery algorithm can recover that sequence given Φx. Since each entry of Φx conveys only $O(\log n)$ bits, it follows that the number m of rows of Φ must be $\Omega(k \log(n/k))$.

The encoding is performed by taking

$$x = \sum_{j=1}^{\log n} D^j y_j \ ,$$

where $D = O(1)$ and the y_j's are chosen from the error-correcting code Y defined as in the deterministic case. The intuition behind this approach is that a good ℓ_1/ℓ_1 approximation to x reveals most of the bits of $y_{\log n}$. This enables us to identify $y_{\log n}$ exactly using error correction. We could then compute $\Phi x - \Phi y_{\log n} = \Phi(\sum_{j=1}^{\log n - 1} D^j y_j)$, and identify $y_{\log n-1}, \dots, y_1$ in a recursive manner. The only obstacle to completing this argument is that we would need the recovery algorithm to work for *all* y_i, which would require lower probability of algorithm failure (roughly $1/\log n$). To overcome this problem, we replace the encoding argument by a reduction from a related communication complexity problem called *Augmented Indexing.* This problem has been used in the data stream literature [CW09, KNW10] to prove lower bounds for linear algebraic and norm estimation problems. Since the problem has communication complexity of $\Omega(d)$, the conclusion follows.

We apply the argument to arbitrary matrices Φ by representing them as a sum $\Phi' + \Phi''$, where Φ' has $O(\log n)$ bits of precision and Φ'' has "small" entries. We then show that $\Phi' x = \Phi(x + \sigma)$ for some σ with $\|\sigma\|_1 < n^{-\Omega(1)} \|x\|_1$. In the communication game, this means we can transmit $\Phi' x$ and recover $y_{\log n}$ from $\Phi'(\sum_{j=1}^{\log n} D^j y_j) = \Phi(\sum_{j=1}^{\log n} D^j y_j + \sigma)$.

One catch is that σ depends on Φ. The recovery algorithm is guaranteed to work with probability $3/4$ for any x, but the choice of Φ must be independent of x. There is no guarantee about recovery of $x + \sigma$ when σ depends on Φ (even if σ is tiny). To deal with this, we choose a u uniformly from the ℓ_1 ball of radius k. We can set $\|\sigma\|_1 \ll k/n$, so $x + u$ and $x + u + \sigma$ are distributions with $o(1)$ statistical distance. Hence recovery from $\Phi(x+u+\sigma)$ matches recovery from $\Phi(x+u)$ with probability $1 - o(1)$, and $\|u\|_1$ is small enough that successful recovery from $\Phi(x+u)$ identifies $y_{\log n}$. Hence we can recover $y_{\log n}$ from $\Phi(x+u+\sigma) = \Phi' x + \Phi u$ with probability $3/4 - o(1) > 2/3$, which means that the Augmented Indexing reduction applies to arbitrary matrices as well.

Preliminaries

In this chapter we consider the following types of recovery guarantees: the ℓ_p/ℓ_p guarantee, for $p = 1$ or 2, is that the recovered $\hat{x}$ satisfies

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

for constant C, and the ℓ_2/ℓ_1 guarantee is that $\hat{x}$ satisfies

$$\|x - \hat{x}\|_2 \leq C k^{-1/2} \min_{k\text{-sparse } x'} \|x - x'\|_1 \ ,$$

also for constant C.

We will denote a sparse recovery algorithm by a pair $(\Phi, \mathcal{R})$, where Φ is an $m \times n$ measurement matrix (or, in the randomized case, a random variable with a distribution over such matrices) and $\mathcal{R}$ is a recovery algorithm that, for any signal $x \in \mathbb{R}^n$, maps Φx (called the *sketch* of x) to some $\hat{x}$ satisfying one of the above recovery guarantees (in the randomized case, with probability at least 3/4).

We use $B_p^n(r)$ to denote the ℓ_p ball of radius r in $\mathbb{R}^n$; we skip the superscript n if it is clear from the context.

3.1 Deterministic Lower Bound

We begin by proving a lower bound on m for any C-approximate deterministic recovery algorithm. First we use a discrete volume bound (Lemma 15) to find a large set Y of points that are at least k apart from each other. Then we use another volume bound (Lemma 16) on the images of small ℓ_1 balls around each point in Y. The idea is that if m is too small, some two images overlap. But the recovery algorithm, applied to a point in the collision, must yield an answer close to two points in Y. This is impossible, so m must be large.

Lemma 15. *(Gilbert-Varshamov) For any $q, k \in \mathbb{N}$, $\epsilon > 0$ and $\epsilon < 1 - 1/q$, there exists a set $Y \subseteq \{0,1\}^{qk}$ of binary vectors with exactly k ones each, such that vectors in Y have minimum Hamming distance $2\epsilon k$ and*

$$\log |Y| > (1 - H_q(\epsilon)) k \log q \ ,$$

where H_q is the q-ary entropy function $H_q(x) = -x \log_q \frac{x}{q-1} - (1-x)\log_q(1-x)$.

Proof. We will construct a code book X of block length k, alphabet q, and minimum Hamming distance ϵk. Replacing each character i with the q-long standard basis vector e_i will create a binary qk-dimensional code book Y with minimum Hamming distance $2\epsilon k$ of the same size as X, where each element of Y has exactly k ones.

The Gilbert-Varshamov bound, based on volumes of Hamming balls, states that a code book of size L exists for some

$$L \geq \frac{q^k}{\sum_{i=0}^{\epsilon k - 1} \binom{k}{i} (q-1)^i} \ .$$

We claim (analogous to [vL98], p. 21, proven below) that for $\epsilon < 1 - 1/q$,

$$\sum_{i=0}^{\epsilon k} \binom{k}{i} (q-1)^i < q^{H_q(\epsilon)k} \ .$$

It would follow that $\log L > (1 - H_q(\epsilon))k \log q$, as desired.

To prove the claim, note that

$$q^{-H_q(\epsilon)} = \left(\frac{\epsilon}{(q-1)(1-\epsilon)}\right)^{\epsilon} (1-\epsilon) < (1-\epsilon) \ .$$

Then

$$\begin{aligned} 1 &= (\epsilon + (1-\epsilon))^k \\ &> \sum_{i=0}^{\epsilon k} \binom{k}{i} \epsilon^i (1-\epsilon)^{k-i} \\ &= \sum_{i=0}^{\epsilon k} \binom{k}{i} (q-1)^i \left(\frac{\epsilon}{(q-1)(1-\epsilon)}\right)^i (1-\epsilon)^k \\ &> \sum_{i=0}^{\epsilon k} \binom{k}{i} (q-1)^i \left(\frac{\epsilon}{(q-1)(1-\epsilon)}\right)^{\epsilon k} (1-\epsilon)^k \\ &= q^{-H_q(\epsilon)k} \sum_{i=0}^{\epsilon k} \binom{k}{i} (q-1)^i \ . \end{aligned}$$

□

Lemma 16. *Take an $m \times n$ real matrix Φ, positive reals ϵ, p, λ, and $Y \subset B_p^n(\lambda)$. If $|Y| > (1 + 1/\epsilon)^m$, then there exist $z, \overline{z} \in B_p^n(\epsilon\lambda)$ and $y, \overline{y} \in Y$ with $y \neq \overline{y}$ and $\Phi(y + z) = \Phi(\overline{y} + \overline{z})$.*

Proof. If the statement is false, then the images of all $|Y|$ balls $y + B_p^n(\epsilon\lambda)$, for $y \in Y$, are disjoint. However, those balls all lie within $B_p^n((1+\epsilon)\lambda)$ since $Y \subseteq B_p^n(\lambda)$. A volume argument gives the result, as follows.

Let $P = \Phi B_p^n(1)$ be the image of the n-dimensional ball of radius 1 in m-dimensional space. This is a polytope with some volume V. The image of $B_p^n(\epsilon\lambda)$ is a linearly scaled P with volume $(\epsilon\lambda)^m V$, and the volume of the image of $B_p^n((1+\epsilon)\lambda)$ is again similar with volume $((1+\epsilon)\lambda)^m V$. If the images of the small balls $y + B_p^n(\epsilon\lambda)$ are all disjoint and lie inside the image of the big ball $B_p^n((1+\epsilon)\lambda)$, we have

$$|Y| \, (\epsilon\lambda)^m V \leq ((1+\epsilon)\lambda)^m V \ ,$$

or $|Y| \leq (1 + 1/\epsilon)^m$. If Y has more elements than this, the images of some two balls $y + B_p^n(\epsilon\lambda)$ and $\overline{y} + B_p^n(\epsilon\lambda)$ must intersect, implying the lemma. □

Theorem 17. *For any deterministic sparse recovery algorithm $(\Phi, \mathcal{R})$ with the ℓ_1/ℓ_1 guarantee and approximation factor C, Φ must have*

$$m \geq \frac{1 - H_{\lfloor n/k \rfloor}(1/2)}{\log(4 + 2C)} k \log \left\lfloor \frac{n}{k} \right\rfloor$$

rows.

Proof. Let Y be a maximal set of k-sparse n-dimensional binary vectors with minimum Hamming distance k, and let $\epsilon = 1/(3+2C)$. By Lemma 15 with $q = \lfloor n/k \rfloor$ we have $\log |Y| > (1 - H_{\lfloor n/k \rfloor}(1/2))k \log \lfloor n/k \rfloor$.

Suppose that the theorem is not true; then

$$m < \frac{\log |Y|}{\log(4+2C)} = \frac{\log |Y|}{\log(1+1/\epsilon)} ,$$

or $|Y| > (1+\frac{1}{\epsilon})^m$. Hence Lemma 16 gives us some $y, \overline{y} \in Y$ and $z, \overline{z} \in B_1(\epsilon k)$ with $\Phi(y+z) = \Phi(\overline{y}+\overline{z})$.

Let $\hat{x}$ be the result of running the recovery algorithm on $\Phi(y+z)$. By the definition of a deterministic recovery algorithm, we have

$$\begin{aligned}
&\|y+z-\hat{x}\|_1 \le C \min_{k\text{-sparse } x'} \|y+z-x'\|_1 \\
&\Rightarrow \|y-\hat{x}\|_1 - \|z\|_1 \le C\|z\|_1 \\
&\Rightarrow \|y-\hat{x}\|_1 \le (1+C)\|z\|_1 \le (1+C)\epsilon k = \tfrac{1+C}{3+2C}\,k,
\end{aligned}$$

and similarly $\|\overline{y}-\hat{x}\|_1 \le \frac{1+C}{3+2C}\,k$, so

$$\|y-\overline{y}\|_1 \le \|y-\hat{x}\|_1 + \|\overline{y}-\hat{x}\|_1 = \frac{2+2C}{3+2C}k < k .$$

But this contradicts the definition of Y, so m must be large enough for the guarantee to hold. □

Corollary 18. *For $C = O(1)$, $m = \Omega(k \log(n/k))$.*

3.2 Randomized Upper Bound for Uniform Noise

The standard way to prove a randomized lower bound is to find a distribution of hard inputs, and to show that any deterministic algorithm is likely to fail on that distribution. In our context, we would like to define a "head" random variable y from a distribution $\mathcal{Y}$ and a "tail" random variable z from a distribution $\mathcal{Z}$, such that any algorithm given the sketch of $y+z$ must recover an incorrect y with non-negligible probability.

Using our deterministic bound as inspiration, we could take $\mathcal{Y}$ to be uniform over a set of k-sparse binary vectors of minimum Hamming distance k and $\mathcal{Z}$ to be uniform over the ball $B_1(\epsilon k)$ for some constant $\epsilon > 0$. Unfortunately, as the following theorem shows, one can actually perform a recovery of such vectors using only $O(k)$ measurements; this is because $\|z\|_2$ is very small (namely, $\tilde{O}(k/\sqrt{n})$) with high probability.

Theorem 19. *Let $Y \subseteq \mathbb{R}^n$ be a set of signals with the property that for every distinct $y, \overline{y} \in Y$, $\|y-\overline{y}\|_2 \ge \alpha$, for some parameter $\alpha > 0$. Consider "noisy signals" $x = y+z$, where $y \in Y$ and z is a "noise vector" chosen uniformly at random from $B_1(\beta)$, for another parameter $\beta > 0$. Then using an $m \times n$ Gaussian measurement matrix $\Phi = (1/\sqrt{m})(g_{ij})$, where g_{ij}'s are i.i.d. standard Gaussians, we can recover any $y \in Y$ from $\Phi(y+z)$ with probability $1-1/n$ (where the probability is over both Φ and z), as long as*

$$\beta \le O\left(\frac{\alpha m^{1/2} n^{1/2-1/m}}{|Y|^{1/m} \log^{3/2} n}\right) .$$

To prove the theorem we will need the following two lemmas.

Lemma 20. *For any $\delta > 0$, $y, \overline{y} \in Y$, $y \neq \overline{y}$, and $z \in \mathbb{R}^n$, each of the following holds with probability at least $1 - \delta$:*

- $\|\Phi(y-\overline{y})\|_2 \geq \frac{\delta^{1/m}}{3}\|y-\overline{y}\|_2$, *and*
- $\|\Phi z\|_2 \leq (\sqrt{(8/m)\log(1/\delta)} + 1)\|z\|_2$.

Proof. By standard arguments (see, e.g., [IN07]), for any $D > 0$ we have

$$\mathbb{P}\left[\|\Phi(y-\overline{y})\|_2 \leq \frac{\|y-\overline{y}\|_2}{D}\right] \leq \left(\frac{3}{D}\right)^m$$

and

$$\mathbb{P}[\|\Phi z\|_2 \geq D\|z\|_2] \leq e^{-m(D-1)^2/8} \ .$$

Setting both right-hand sides to δ yields the lemma. □

Lemma 21. *A random vector z chosen uniformly from $B_1(\beta)$ satisfies, for any $\gamma > 1$,*

$$\mathbb{P}[\|z\|_2 > \gamma\beta\log n/\sqrt{n}] < 1/n^{\gamma-1} \ .$$

Proof. Consider the distribution of a single coordinate of z, say, z_1. The probability density of $|z_1|$ taking value $t \in [0, s]$ is proportional to the $(n-1)$-dimensional volume of $B_1^{(n-1)}(s-t)$, which in turn is proportional to $(s-t)^{n-1}$. Normalizing to ensure the probability integrates to 1, we derive this probability as

$$p(|z_1| = t) = \frac{n}{s^n}(s-t)^{n-1} \ .$$

It follows that, for any $D \in [0, s]$,

$$\mathbb{P}[|z_1| > D] = \int_D^s \frac{n}{s^n}(s-t)^{n-1}\,dt = (1 - D/s)^n \ .$$

In particular, for any $\alpha > 1$,

$$\mathbb{P}[|z_1| > \alpha s\log n/n] = (1 - \alpha\log n/n)^n < e^{-\alpha\log n} = 1/n^\alpha \ .$$

Now, by symmetry this holds for every other coordinate z_i of z as well, so by the union bound

$$\mathbb{P}[\|z\|_\infty > \alpha s\log n/n] < 1/n^{\alpha-1} \ .$$

Since $\|z\|_2 \leq \sqrt{n}\|z\|_\infty$ for any vector z, the lemma follows. □

Proof of Theorem 19. Lemma 20 says that Φ cannot bring faraway signals too close together, and cannot blow up a small noise vector too much. Now, we already assumed the signals to be far apart, and Lemma 21 tells us that the noise is indeed small (in ℓ_2 distance). The result is that in the image space, the noise is not enough to confuse different signals. Quantitatively, applying the second part of Lemma 20 with $\delta = 1/n^2$, and Lemma 21 with $\gamma = 3$, gives us

$$\|\Phi z\|_2 \leq O\left(\frac{\log^{1/2} n}{m^{1/2}}\right)\|z\|_2 \leq O\left(\frac{\beta\log^{3/2} n}{(mn)^{1/2}}\right) \tag{3.1}$$

with probability at least $1-2/n^2$. On the other hand, given signal $y \in Y$, we know that every other signal $\overline{y} \in Y$ satisfies $\|y-\overline{y}\|_2 \geq \alpha$, so by the first part of Lemma 20 with $\delta = 1/(2n|Y|)$, together with a union bound over every $\overline{y} \in Y$,

$$\|\Phi(y-\overline{y})\|_2 \geq \frac{\|y-\overline{y}\|_2}{3(2n|Y|)^{1/m}} \geq \frac{\alpha}{3(2n|Y|)^{1/m}} \tag{3.2}$$

holds for every $\overline{y} \in Y$, $\overline{y} \neq y$, simultaneously with probability $1-1/(2n)$.

Finally, observe that as long as $\|\Phi z\|_2 < \|\Phi(y-\overline{y})\|_2/2$ for every competing signal $\overline{y} \in Y$, we are guaranteed that

$$\begin{aligned} \|\Phi(y+z) - \Phi y\|_2 &= \|\Phi z\|_2 \\ &< \|\Phi(y-\overline{y})\|_2 - \|\Phi z\|_2 \\ &\leq \|\Phi(y+z) - \Phi\overline{y}\|_2 \end{aligned}$$

for every $\overline{y} \neq y$, so we can recover y by simply returning the signal whose image is closest to our measurement point $\Phi(y+z)$ in ℓ_2 distance. To achieve this, we can chain Equations (3.1) and (3.2) together (with a factor of 2), to see that

$$\beta \leq O\left(\frac{\alpha m^{1/2} n^{1/2-1/m}}{|Y|^{1/m}\log^{3/2} n}\right)$$

suffices. Our total probability of failure is at most $2/n^2 + 1/(2n) < 1/n$. □

The main consequence of this theorem is that for the setup we used in Section 3.1 to prove a deterministic lower bound of $\Omega(k\log(n/k))$, if we simply draw the noise uniformly randomly from the same ℓ_1 ball (in fact, even one with a much larger radius, namely, polynomial in n), this "hard distribution" can be defeated with just $O(k)$ measurements:

Corollary 22. *If Y is a set of binary k-sparse vectors, as in Section 3.1, and noise z is drawn uniformly at random from $B_1(\beta)$, then for any $\epsilon > 0$, $m = O(k/\epsilon)$ measurements suffice to recover any signal in Y with probability $1-1/n$, as long as*

$$\beta \leq O\left(\frac{k^{3/2+\epsilon} n^{1/2-\epsilon}}{\log^{3/2} n}\right) .$$

Proof. The parameters in this case are $\alpha = k$ and $|Y| \leq \binom{n}{k} \leq (ne/k)^k$, so by Theorem 19, it suffices to have

$$\beta \leq O\left(\frac{k^{3/2+k/m} n^{1/2-(k+1)/m}}{\log^{3/2} n}\right) .$$

Choosing $m = (k+1)/\epsilon$ yields the corollary. □

3.3 Randomized Lower Bound

Although it is possible to partially circumvent this obstacle by focusing our noise distribution on "high" ℓ_2 norm, sparse vectors, we are able to obtain stronger results via a reduction from a communication game and the corresponding lower bound.

The communication game will show that a message Φx must have a large number of bits. To show that this implies a lower bound on the number of rows of Φ, we will need Φ to be discrete. Hence we first show that discretizing Φ does not change its recovery characteristics by much.

3.3.1 Discretizing Matrices

Before we discretize by rounding, we need to ensure that the matrix is well conditioned. We argue that without loss of generality, the rows of Φ are orthonormal.

We can multiply Φ on the left by any invertible matrix to get another measurement matrix with the same recovery characteristics. Consider the singular value decomposition $\Phi = U\Sigma V^*$, where U and V are orthonormal and Σ is 0 off the diagonal. We can eliminate U and make the entries of Σ be either 0 or 1. The result is a matrix consisting of m orthonormal rows. For such matrices, we prove the following:

Lemma 23. *Consider any $m \times n$ matrix Φ with orthonormal rows. Let Φ' be the result of rounding Φ to b bits per entry. Then for any $v \in \mathbb{R}^n$, there exists an $\sigma \in \mathbb{R}^n$ with $\Phi' v = \Phi(v - \sigma)$ and $\|\sigma\|_p < n^2 2^{-b} \|v\|_p$, for $p = 1$ and 2.*

Proof. Let $\Phi'' = \Phi - \Phi'$ be the roundoff error when discretizing Φ to b bits, so each entry of Φ'' is less than 2^{-b}. Then for $\sigma = \Phi^T \Phi'' v$, we have $\Phi\sigma = \Phi'' v$ and

$$\|\sigma\|_p = \left\|\Phi^T \Phi'' v\right\|_p \leq n \left\|\Phi'' v\right\|_p \leq mn2^{-b} \|v\|_p \leq n^2 2^{-b} \|v\|_p \ .$$

□

3.3.2 Communication Complexity

We use a few definitions and results from two-party communication complexity. For further background see the book by Kushilevitz and Nisan [KN97]. Consider the following communication game. There are two parties, Alice and Bob. Alice is given a string $a \in \{0,1\}^d$. Bob is given an index $i \in [d]$, together with $a_{i+1}, a_{i+2}, \ldots, a_d$. The parties also share an arbitrarily long common random string ρ. Alice sends a single message $M(a, \rho)$ to Bob, who must output a_i with probability at least $2/3$, where the probability is taken over ρ. We refer to this problem as Augmented Indexing. The communication cost of Augmented Indexing is the minimum, over all correct protocols, of the length of the message $M(a, \rho)$ on the worst-case choice of ρ and a.

The next theorem is well-known and follows from Lemma 13 of [MNSW98] (see also Lemma 2 of [BJKK04]).

Theorem 24. *The communication cost of Augmented Indexing is $\Omega(d)$.*

Proof. First, consider the private-coin version of the problem, in which both parties can toss coins, but do not share a random string ρ (i.e., there is no public coin). Consider any correct protocol for this problem. We can assume the probability of error of the protocol is an arbitrarily small positive constant by increasing the length of Alice's message by a constant factor (e.g., by independent repetition and a majority vote). Applying Lemma 13 of [MNSW98] (with, in their notation, $t = 1$ and $a = c'd$ for a sufficiently small constant $c' > 0$), the communication cost of such a protocol must be $\Omega(d)$. Indeed, otherwise there would be a protocol in which Bob could output a_i with probability greater than 1/2 without any interaction with Alice, contradicting the fact that Bob has no information about a_i. The theorem then follows from Newman's theorem (see, e.g., Theorem 2.4 of [KNR99]), which states that the communication cost of the best public coin protocol is at least that of the private coin protocol minus $O(\log d)$. □

3.3.3 Lower Bound Theorem for ℓ_1/ℓ_1

We can now prove the chapter's main result. We will first prove the theorem under the ℓ_1/ℓ_1 guarantee, then point out the modifications necessary to obtain the ℓ_2/ℓ_2 and ℓ_2/ℓ_1 cases.

Proof of Theorem 14 under ℓ_1/ℓ_1 guarantee. We shall assume, without loss of generality, that n and k are powers of 2 and that the rows of Φ are orthonormal. The proof for the general case follows with minor modifications.

Let $(\Phi, \mathcal{R})$ be such a recovery algorithm. We will show how to solve Augmented Indexing on instances of size $d = \Omega(k \log(n/k) \log n)$ with communication cost $O(m \log n)$. The theorem will then follow by Theorem 24.

Let Y be a maximal set of k-sparse n-dimensional binary vectors with minimum Hamming distance k. From Lemma 15 we have $\log |Y| = \Omega(k \log(n/k))$. Let $d = \lfloor \log |Y| \rfloor \log n$, and define $D = 2C + 3$.

Alice is given a string $a \in \{0,1\}^d$, and Bob is given $i \in [d]$ together with $a_{i+1}, a_{i+2}, \ldots, a_d$, as in the setup for Augmented Indexing. Alice splits her string a into $\log n$ contiguous chunks $a^1, a^2, \ldots, a^{\log n}$, each containing $\lfloor \log |Y| \rfloor$ bits. She uses a^j as an index into Y to choose y_j. Alice defines

$$x = D^1 y_1 + D^2 y_2 + \cdots + D^{\log n} y_{\log n} \ .$$

Alice and Bob use the common randomness ρ to agree upon a random matrix Φ with orthonormal rows. Both Alice and Bob round Φ to form Φ' with $b = \lceil (4 + 2 \log D) \log n \rceil = O(\log n)$ bits per entry. Alice computes $\Phi' x$ and transmits it to Bob.

From Bob's input i, he can compute the value $j = j(i)$ for which the bit a_i occurs in a^j. Bob's input also contains $a_{i+1}, \ldots, a_n$, from which he can reconstruct $y_{j+1}, \ldots, y_{\log n}$, and in particular can compute

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

Set $w = x - z = \sum_{i=1}^{j} D^i y_i$. Bob then computes $\Phi' z$, and using $\Phi' x$ and linearity, $\Phi' w$. Observe that

$$\|w\|_1 \leq \sum_{i=1}^{j} k D^i < k \frac{D^{j+1}}{D-1} < k D^{2 \log n} \ . \tag{3.3}$$

So from Lemma 23, there exists some σ with $\Phi' w = \Phi(w - \sigma)$ and

$$\|\sigma\|_1 < n^2 2^{-4 \log n - 2 \log D \log n} \|w\|_1 < k/n^2 \ . \tag{3.4}$$

Bob chooses another vector u uniformly from $B_1^n(k)$, the ℓ_1 ball of radius k, and computes $\Phi(w - \sigma - u) = \Phi' w - \Phi u$. He runs the recovery algorithm $\mathcal{R}$ on Φ and $\Phi(w - \sigma - u)$, obtaining $\hat{w}$. We have that u is independent of w and σ, and that $\|u\|_1 \leq k(1 - 1/n^2) \leq k - \|\sigma\|_1$ with probability

$$\frac{\mathrm{Vol}(B_1(k(1 - 1/n^2)))}{\mathrm{Vol}(B_1(k))} = (1 - 1/n^2)^n > 1 - 1/n \ .$$

But

$$\{w - u \mid \|u\|_1 \leq k - \|\sigma\|_1\} \subseteq \{w - \sigma - u \mid \|u\|_1 \leq k\} \ ,$$

so the ranges of the random variables $w-\sigma-u$ and $w-u$ overlap in at least a $1-1/n$ fraction of their volumes. Therefore, $w - \sigma - u$ and $w - u$ have statistical distance at most $1/n$. The distribution of $w - u$ is independent of Φ, so running the recovery algorithm on $\Phi(w - u)$ would work with probability at least $3/4$. Hence, with probability at least $3/4 - 1/n \geq 2/3$ (for n large enough), $\hat{w}$ satisfies the recovery criterion for $w - u$, meaning

$$\|w - u - \hat{w}\|_1 \leq C \min_{k\text{-sparse } w'} \|w - u - w'\|_1 \ . \tag{3.5}$$

Now,

$$\begin{aligned}
\left\|D^j y_j - \hat{w}\right\|_1 &\leq \left\|w - u - D^j y_j\right\|_1 + \|w - u - \hat{w}\|_1 \\
&\leq (1 + C)\left\|w - u - D^j y_j\right\|_1 \\
&\leq (1 + C)(\|u\|_1 + \sum_{i=1}^{j-1} \left\|D^i y_i\right\|_1) \\
&\leq (1 + C)k \sum_{i=0}^{j-1} D^i \\
&< k \cdot \frac{(1 + C)D^j}{D - 1} \\
&= kD^j/2 \ .
\end{aligned} \tag{3.6}$$

And since the minimum Hamming distance in Y is k, this means $\left\|D^j y_j - \hat{w}\right\|_1 < \left\|D^j y' - \hat{w}\right\|_1$ for all $y' \in Y, y' \neq y_j$. So Bob can correctly identify y_j with probability at least $2/3$. From y_j he can recover a^j, and hence the bit a_i that occurs in a^j.

Hence, Bob solves Augmented Indexing with probability at least $2/3$ given the message $\Phi' x$. The entries in Φ' and x are polynomially bounded integers (up to scaling of Φ'), and so each entry of $\Phi' x$ takes $O(\log n)$ bits to describe. Hence, the communication cost of this protocol is $O(m \log n)$. By Theorem 24, $m \log n = \Omega(k \log(n/k) \log n)$, or $m = \Omega(k \log(n/k))$. □

3.3.4 Modifications for ℓ_2/ℓ_2 and ℓ_2/ℓ_1

We make the straightforward changes to obtain the ℓ_2/ℓ_2 result:

Proof of Theorem 14 under ℓ_2/ℓ_2 guarantee. The protocol for Alice and Bob remains identical except in two places. First, Bob chooses u from $B_2(\sqrt{k})$, the ℓ_2 ball of radius $\sqrt{k}$, instead of $B_1(k)$. Second, in the final step, having recovered $\hat{w}$, he selects the code word y_j such that $D^j y_j$ is closest to $\hat{w}$ in ℓ_2-distance instead of ℓ_1.

In the analysis, the observation of (3.3) simply becomes $\|w\|_2 < \sqrt{k} D^{2 \log n}$, and similarly, (3.4) becomes $\|\sigma\|_2 < \sqrt{k}/n^2$. The statistical distance between $w - \sigma - u$ and $w - u$ remains $1/n$, so that the $\hat{w}$ that Bob recovers satisfies the recovery criterion for $w - u$ with probability at least $2/3$. However, this time the ℓ_1-norms in (3.5) are replaced by ℓ_2-norms. Finally, the inequality (3.6) becomes $\left\|D^j y_j - \hat{w}\right\|_2 < \sqrt{k} D^j/2$, so that, with code words in Y being a minimum of $\sqrt{k}$ in ℓ_2-distance apart, Bob recovers the correct one to get his bit, completing the protocol. □

The ℓ_2/ℓ_1 guarantee requires only slightly trickier modifications:

Proof of Theorem 14 under ℓ_2/ℓ_1 guarantee. This time we need both an ℓ_1 bound and an ℓ_2 bound on u, so Bob will simply choose u uniformly from the intersection of the two balls $B_1(k)$ and $B_2(\sqrt{k})$. Now, we will have $\|u\|_1 \leq k(1-1/n^2) \leq k - \|\sigma\|_1$ and, simultaneously, $\|u\|_2 \leq \sqrt{k}(1-1/n^2) \leq \sqrt{k} - \|\sigma\|_2$, with probability

$$\frac{\mathrm{Vol}(B_1(k(1-1/n^2)) \cap B_2(\sqrt{k}(1-1/n^2)))}{\mathrm{Vol}(B_1(k) \cap B_2(\sqrt{k}))} = (1-1/n^2)^n > 1-1/n \ ,$$

since the two intersections are still just scaled versions of each other, with a scaling factor of $(1-1/n^2)$. As before,

$$\{w - u \mid \|u\|_1 \leq k - \|\sigma\|_1 \text{ and } \|u\|_2 \leq \sqrt{k} - \|\sigma\|_2\} \subseteq \{w - \sigma - u \mid \|u\|_1 \leq k \text{ and } \|u\|_2 \leq \sqrt{k}\} \ ,$$

so $w - \sigma - u$ and $w - u$ still have statistical distance only $1/n$. Note that we have used the fact Lemma 23 gives us the *same* σ satisfying $\|\sigma\|_1 < k/n^2$ and $\|\sigma\|_2 < \sqrt{k}/n^2$.

It follows this time that Bob, running $\mathcal{R}$ on $\Phi(w - \sigma - u)$, obtains with probability 2/3 a $\hat{w}$ satisfying the ℓ_2/ℓ_1 guarantee:

$$\|w - u - \hat{w}\|_2 \leq \frac{C}{\sqrt{k}} \min_{k\text{-sparse } w'} \|w - u - w'\|_1 \ .$$

Following (3.6), but making use of both bounds on u, we have

$$\begin{aligned}
\left\|D^j y_j - \hat{w}\right\|_2 &\leq \left\|w - u - D^j y_j\right\|_2 + \|w - u - \hat{w}\|_2 \\
&\leq \left(\|u\|_2 + \sqrt{k}\sum_{i=1}^{j-1} D^i\right) + \left(\frac{C}{\sqrt{k}}\|u\|_1 + C\sqrt{k}\sum_{i=1}^{j-1} D^i\right) \\
&\leq (1+C)\sqrt{k}\sum_{i=0}^{j-1} D^i \\
&< \sqrt{k}D^j/2 \ .
\end{aligned}$$

Finally, selecting the code word similarly to the ℓ_2/ℓ_2 case completes the protocol. □

Chapter 4

Sparse Recovery with Partial Support Knowledge

Background

Several formulations utilizing partial support information to improve sparse recovery have been studied in the literature. In [LV10a], the authors study exact reconstruction when given a set S very close to the true support of x. Specifically, their recovery guarantee is to return the approximation $\hat{x}$ which is sparsest outside S, which they do by solving the corresponding ℓ_1-minimization problem along the lines of [CRT06, Don06]. In [Jac10], the ideas of [LV10a] are extended to give the following ℓ_2/ℓ_1-type guarantee:

$$\|x - \hat{x}\|_2 \leq C\|x_{\bar{S}} - x_{(k)}\|_1 \ .^1$$

Intuitively, we are being told that the signal is mostly supported by the set S, while it is k-sparse *outside* of S. In contrast, in our formulation we target signals that are sparse *within* the set S.

In [FMSY12], the authors give the complicated guarantee:

$$\|x - \hat{x}\|_2 \leq C(\omega\|x - x_{(k)}\|_1 + (1-\omega)\|(x - x_{(k)})_{\bar{S}}\|_1) \ ,$$

where $0 \leq \omega \leq 1$ is a parameter. Roughly, the smaller ω is the less we penalize the signal's "tail" (the components not among the k largest), as long as it lies mostly inside S.

In [Pri11], the author considers the *set-query problem*, where essentially full knowledge of the signal support is assumed. The reconstruction guarantee there can be written as

$$\|x - \hat{x}\|_2 \leq (1+\epsilon) \min_{\text{supp}(x')\subseteq S} \|x - x'\|_2 \ .$$

This setting is a special case of SRPSK, namely, where $k = s$.

Several other works address variants of sparse recovery with partial knowledge of signal support in some form [KXAH09, vBMP07, LV10b], but these had significantly different model assumptions and/or recovery objectives from ours.

[1] See Preliminaries for the "x_S" and "$x_{(k)}$" notation.

Main Results

In this chapter we will focus only on randomized measurement matrices and the ℓ_1/ℓ_1 and ℓ_2/ℓ_2 guarantees. We first prove the following lower bound on the number of measurements for any $(1+\epsilon)$-approximate solution to SRPSK with either guarantee.

Theorem 25. *Any $(1+\epsilon)$-approximate solution to SRPSK with the ℓ_1/ℓ_1 or the ℓ_2/ℓ_2 guarantee requires, for $s = O(\epsilon n/\log(n/\epsilon))$, at least $\Omega\left((k/\epsilon)\log(s/k)\right)$ measurements.*

We then give an algorithm that matches Theorem 25 in the ℓ_2/ℓ_2 case.

Theorem 26. *There exists an $(1+\epsilon)$-approximate solution to SRPSK with the ℓ_2/ℓ_2 guarantee, where the measurement matrix Φ has $m = O((k/\epsilon)\log(s/k))$ rows. Moreover, Φ has, in expectation, $O(\log^2 k \log(s/k))$ non-zeros per column, and the recovery algorithm $\mathcal{R}$ runs in $O(s\log^2 k + (k/\epsilon)\log^{O(1)} s)$ time.*

Techniques

Consider the upper bound first. The general approach of our algorithm is to reduce SRPSK to the *noisy sparse recovery problem* (NSR). The latter is a generalization of sparse recovery where the recovery algorithm is given $\Phi x + \nu$, where ν is the *measurement noise.* The reduction proceeds by representing Φx as $\Phi x_S + \Phi x_{\bar{S}}$, and interpreting the second term as noise. Since the vector x_S has dimension s, not n, we can use Φ with only $O(k\log(s/k))$ rows. This yields the desired measurement bound.

To make this work, however, we need to ensure that for any fixed S, the sub-matrix Φ_S of Φ (containing the columns with indices in S) is a valid sparse recovery matrix for s-dimensional vectors. This would be almost immediate if (as often happens, e.g. [CRT06]) each column of Φ was an i.i.d. random variable chosen from some distribution: we could simply sample the n columns of Φ from the distribution parametrized by k and s. Unfortunately, the algorithm of [GLPS10] (which has the best known dependence on ϵ) does not have this independence property; in fact, the columns are highly dependent on each other. However, we show that it is possible to modify it so that the independence property holds.

Our lower bound argument mimics the approach of Chapter 3. Specifically, fix s and let $\alpha = n/s$. We show how to encode α code words $y_1, \ldots, y_\alpha$, from some code Y containing $2^{\Theta(k\log(s/k))}$ code words, into a vector x, such that a $(1+\epsilon)$-approximate algorithm for SRPSK can iteratively decode all y_i's, starting from y_α and ending with y_1. This implies that one can "pack" $\Omega(\alpha k\log(s/k))$ bits into Φx. Then by showing that each coordinate of Φx yields only $O(\epsilon\alpha)$ bits of information, it follows that Φx has to have $\Omega((k/\epsilon)\log(s/k))$ coordinates.

The caveat is that the argument of Chapter 3 applied to the case of when ϵ is a constant bounded away from 0 (i.e., $\epsilon = \Omega(1)$). For $\epsilon = o(1)$, use of the triangle inequality prevents us from strengthening the lower bound by the desired factor of $1/\epsilon$. We show that the formulation of SRPSK avoids this problem. Intuitively, this is because we can choose the y_i's to have disjoint supports, and SRPSK enables us to restrict sparse approximation to a particular subset of coordinates. For technical reasons we need to put a restriction on how big s can be for the full argument to go through.

Preliminaries

For positive integer n, let $[n] = \{1, 2, \ldots, n\}$. For positive integer $s \leq n$, let $\binom{[n]}{s}$ denote the collection of subsets of $[n]$ of cardinality s.

Consider $v \in \mathbb{R}^n$, positive integers $k \leq s$, and set $S \in \binom{[n]}{s}$. Denote by $v_{(k)} \in \mathbb{R}^n$ the vector comprising the k largest components of v, breaking ties by some canonical ordering (say, leftmost-first), and 0 elsewhere. Denote by $v_S \in \mathbb{R}^n$ the vector comprising of components of v indexed by S, with 0 elsewhere, and denote by $v_{S,k} \in \mathbb{R}^n$ the vector comprising the k largest components of v among those indexed by S, with 0 elsewhere.

Let $\Pi_S \in \mathbb{R}^{s \times n}$ denote the projection matrix that keeps only components indexed by S (the dimension n will be clear from context). In particular, $\Pi_S v \in \mathbb{R}^s$ consists of components of v indexed by S, and for any matrix $\Phi \in \mathbb{R}^{m \times n}$, $\Phi \Pi_S^T \in \mathbb{R}^{m \times s}$ consists of the columns of Φ indexed by S.

Define the *ℓ_p/ℓ_p sparse recovery with partial support knowledge problem* (denoted SRPSK$_p$) to be the following:

Given parameters (n, s, k, ϵ), where $1 \leq k \leq s \leq n$ and $0 < \epsilon < 1$, design an algorithm $\mathcal{R}$ and a distribution over matrices $\Phi \in \mathbb{R}^{m \times n}$, where $m = m(n, s, k, \epsilon)$, such that for any $x \in \mathbb{R}^n$, given Φx and a specified set $S \in \binom{[n]}{s}$, $\mathcal{R}$ recovers (with knowledge of Φ) a vector $\hat{x} \in \mathbb{R}^n$ such that, with probability 3/4, $\operatorname{supp}(\hat{x}) \subseteq S$ and

$$\|x - \hat{x}\|_p^p \leq (1 + \epsilon)\|x - x_{S,k}\|_p^p \ .$$

Define the *ℓ_p/ℓ_p noisy sparse recovery problem* (NSR$_p$) to be the following:

Given parameters (n, k, ϵ), where $1 \leq k \leq n$ and $0 < \epsilon < 1$, design an algorithm $\mathcal{R}$ and a distribution over matrices $\Phi \in \mathbb{R}^{m \times n}$, where $m = m(n, k, \epsilon)$, such that for any $x \in \mathbb{R}^n$ and $\nu \in \mathbb{R}^m$, $\mathcal{R}$ recovers from $\Phi x + \nu$ (with knowledge of Φ) a vector $\hat{x} \in \mathbb{R}^n$ such that, with probability 3/4,

$$\|x - \hat{x}\|_p^p \leq (1 + \epsilon)\|x - x_{(k)}\|_p^p + \epsilon\|\nu\|_p^p \ .$$

The distribution of Φ must be "normalized" so that for any $v \in \mathbb{R}^n$, $\mathbb{E}[\|\Phi v\|_p] \leq \|v\|_p$.

For all four problems, we will denote a solution by a pair $(\Phi, \mathcal{R})$, where Φ is the measurement matrix and $\mathcal{R}$ is the recovery algorithm. For the recovery algorithms of SRPSK$_1$ and SRPSK$_2$, we will also sometimes indicate the parameter S by a subscript, i.e., $\mathcal{R}_S$.

4.1 Lower Bounds

Recall the Augmented Indexing problem from communication complexity, defined in Chapter 3. Making use of Theorem 24 and Lemma 23 from Chapter 3 to lower bound the communication complexity of Augmented Indexing and to discretize any measurement matrix, respectively, we will now prove our lower bounds for SRPSK$_1$ and SRPSK$_2$.

4.1.1 Lower Bound for ℓ_1/ℓ_1

Proof of Theorem 25 under ℓ_1/ℓ_1. For $\alpha = n/s$, divide $[n]$ into α equal-sized disjoint blocks, S_i for $i = 1, \ldots, \alpha$. For each block S_i, we will choose a binary error-correcting code $Y_i \subseteq \{0, 1\}^n$ with minimum Hamming distance k, where all the code words have Hamming weight exactly k and support contained in S_i. Since $|S_i| = s = n/\alpha$, we know each Y_i can be chosen big enough that

$$\log |Y_i| = \Theta(k \log(n/(\alpha k))) \ .$$

Now, we will use any solution to SRPSK$_1$ to design a protocol for Augmented Indexing with instance size

$$d = \Theta(\alpha k \log(n/(\alpha k))) \ .$$

The protocol is as follows:

Alice divides her input a into α equal-sized blocks, $a^1, \ldots, a^\alpha$, each of size

$$d/\alpha = \Theta(k \log(n/(\alpha k))) \ .$$

Interpreting each block a^i as a binary number, she uses it to index into Y_i (notice that Y_i has sufficiently many code words for each a^i to index a different one), specifying a code word $y_i \in Y_i$. She then computes

$$x = D^1 y_1 + D^2 y_2 + \cdots + D^\alpha y_\alpha$$

for some fixed D dependent on ϵ. Then, using shared randomness, and following the hypothetical protocol, Alice and Bob agree on a matrix Φ (without loss of generality, with orthonormal rows), which they both round to Φ' so that each entry has b bits. Alice computes $\Phi' x$ and sends it to Bob.

Bob, knowing his input i, can compute the $j = j(i)$ for which block a^j of Alice's input contains i, and hence knows the set S_j supporting block a^j. Moreover, he knows $a^{j'}$, and thereby $y_{j'}$, for every $j' > j$, so he can compute

$$z = D^{j+1} y_{j+1} + \cdots + D^\alpha y_\alpha \ .$$

Let $w = x - z = \sum_{i=1}^{j} D^i y_i$. From Alice's message, using linearity, he can then compute $\Phi' w$. Now, similarly to the proof of Theorem 14, by Lemma 23, there must exist some $\sigma \in \mathbb{R}^n$ with $\Phi' w = \Phi(w - \sigma)$ and

$$\|\sigma\|_1 < n^2 2^{-b} \|w\|_1 < n^2 2^{-b} k \tfrac{D^{j+1}}{D-1} \ .$$

Bob chooses another vector u uniformly from $B_1^n(n^4 2^{-b} k \frac{D^{j+1}}{D-1})$ and computes $\Phi(w - \sigma - u) = \Phi' w - \Phi u$. He runs the recovery algorithm $\mathcal{R}$ on $\Phi(w - \sigma - u)$, with target support set S_j, obtaining $\hat{w}$. Now, u is independent of σ, so

$$\|u\|_1 \leq n^4 2^{-b} k \tfrac{D^{j+1}}{D-1} - \|\sigma\|_1$$

with probability

$$\frac{\mathrm{Vol}(B_1(n^4 2^{-b} k \frac{D^{j+1}}{D-1} - \|\sigma\|_1))}{\mathrm{Vol}(B_1(n^4 2^{-b} k \frac{D^{j+1}}{D-1}))} \geq (1 - 1/n^2)^n > 1 - 1/n \ .$$

Moreover,

$$\{w - u \mid \|u\|_1 \leq n^4 2^{-b} k \tfrac{D^{j+1}}{D-1} - \|\sigma\|_1\} \subseteq \{w - u - \sigma \mid \|u\|_1 \leq n^4 2^{-b} k \tfrac{D^{j+1}}{D-1}\} \ ,$$

so $w - u$ and $w - \sigma - u$ have statistical distance at most $1/n$. It follows that with probability $2/3$, $\hat{w}$ satisfies the recovery criterion for $w - u$, namely, that $\mathrm{supp}(\hat{w}) \subseteq S_j$ and

$$\begin{aligned}
\|w - u - \hat{w}\|_1 &\leq (1+\epsilon) \left\|w - u - (w-u)_{S_j,k}\right\|_1 \\
&\leq (1+\epsilon) \left\|w - u - D^j y_j\right\|_1 \\
&\leq (1+\epsilon)(k\tfrac{D^j - D}{D-1} + \|u\|_1) \ .
\end{aligned} \tag{4.1}$$

Bob then finds the code word in Y_j that is closest in ℓ_1-distance to $\hat{w}/D^j$ (which he hopes is y_j) and, looking at the index of that code word within Y_j (which he hopes is a^j), he returns the bit corresponding to his index i.

Now, suppose that Bob was wrong. This means he obtained a $\hat{w}$ that, appropriately scaled, was closer or equidistant to another code word in Y_j than y_j, implying that $\|D^j y_j - \hat{w}\|_1 \geq kD^j/2$. Since $\mathrm{supp}(\hat{w}) \subseteq S_j$ and all the y_j's have disjoint support, we can write

$$\begin{aligned}
\|w - u - \hat{w}\|_1 &\geq \|w - \hat{w}\|_1 - \|u\|_1 \\
&= k \sum_{i=1}^{j-1} D^i + \left\|D^j y_j - \hat{w}\right\|_1 - \|u\|_1 \\
&\geq k\left(\tfrac{D^j - D}{D-1} + D^j/2\right) - \|u\|_1 \ .
\end{aligned} \tag{4.2}$$

We will show that for appropriate choices of D and b, (4.1) and (4.2) contradict each other, implying that Bob must have correctly extracted his bit and solved Augmented Indexing. To this end, it suffices to prove the following inequality:

$$\|u\|_1 < \frac{kD^j}{3}\left(\frac{1}{2} - \frac{\epsilon}{D-1}\right) \ , \tag{4.3}$$

where we assumed $\epsilon < 1$ to simplify things.

Let us fix $D = 1 + 4\epsilon$. (4.3) becomes

$$\|u\|_1 < \tfrac{k}{3}(1+4\epsilon)^j\left(\frac{1}{2} - \frac{1}{4}\right) = k(1+4\epsilon)^j/12 \ .$$

By our choice of u, we know that

$$\|u\|_1 < n^4 2^{-b} k \tfrac{D^{j+1}}{D-1} = n^4 2^{-b} k (1+4\epsilon)^{j+1}/(4\epsilon) \ ,$$

so we need only choose b large enough that $2^b \geq 3(1+4\epsilon)n^4/\epsilon$, i.e., $b = \Theta(\log(n/\epsilon))$ suffices. Recall that b is the number of bits per component of Φ', and each component of x can require up to $\alpha \log D = O(\epsilon\alpha)$ bits, so the message $\Phi' x$ which Alice sends to Bob contains at most $O(m(b + \epsilon\alpha)) = O(m(\log(n/\epsilon) + \epsilon\alpha))$ bits, with which they solve Augmented Indexing with problem size $d = \Theta(\alpha k \log(n/(\alpha k)))$. It follows from Theorem 24 that

$$m = \Omega\left(\frac{\alpha k \log(n/(\alpha k))}{\log(n/\epsilon) + \epsilon\alpha}\right) \ .$$

Finally, as long as $\epsilon\alpha = \Omega(\log(n/\epsilon))$, or equivalently, $s = n/\alpha = O(\epsilon n/\log(n/\epsilon))$, this simplifies to

$$m = \Omega((k/\epsilon)\log(s/k)) \ .$$

□

4.1.2 Lower Bound for ℓ_2/ℓ_2

Proof of Theorem 25 under ℓ_2/ℓ_2. We will only make the modifications necessary to adapt the proof of the ℓ_1/ℓ_1 case.

Alice and Bob will follow an almost identical protocol to solve Augmented Indexing (with the same instance size). The only differences are that, first, he selects u from an ℓ_2-ball (of the same radius) instead of an ℓ_1-ball, and second, having recovered $\hat{w}$, he picks the code word that is closes in ℓ_2-distance to $\hat{w}/D^j$ instead of in ℓ_1-distance.

In the analysis, we again apply Lemma 23 as before, so that there must exist some $\sigma \in \mathbb{R}^n$ with $\Phi' w = \Phi(w - \sigma)$ and

$$\|\sigma\|_2 \leq \|\sigma\|_1 < n^2 2^{-b} k \tfrac{D^{j+1}}{D-1} \ . \tag{4.4}$$

The new choice of u still ensures that $w - u$ and $w - \sigma - u$ have statistical distance at most $1/n$. Thus, with probability $2/3$, $\hat{w}$ satisfies the ℓ_2/ℓ_2 guarantee for $w - u$, namely, that $\mathrm{supp}(\hat{w}) \subseteq S_j$ and

$$\begin{aligned} \|w - u - \hat{w}\|_2 &< (1+\epsilon) \|w - u - (w-u)_{S,k}\|_2 \\ &\leq (1+\epsilon) \left\|w - u - D^j y_j\right\|_2 \\ &\leq (1+\epsilon) \left(\left(k \tfrac{D^{2j} - D^2}{D^2 - 1} \right)^{1/2} + \|u\|_2 \right) \ . \end{aligned} \tag{4.5}$$

Now, if Bob fails to recover y_j, it must be that $\|D^j y_j - \hat{w}\|_2 \geq \sqrt{k} D^j / 2$ (since the code words have minimum ℓ_2-distance $\sqrt{k}$). Since $\mathrm{supp}(\hat{w}) \subseteq S^j$ and the y_j's have disjoint support, we can write

$$\begin{aligned} \|w - u - \hat{w}\|_2 &\geq \|w - \hat{w}\|_2 - \|u\|_2 \\ &= \left(\sum_{i=1}^{j-1} \left\|D^i y_i\right\|_2^2 + \left\|D^j y_j - \hat{w}\right\|_2^2 \right)^{1/2} - \|u\|_2 \\ &\geq \sqrt{k} \left(\tfrac{D^{2j} - D^2}{D^2 - 1} + D^{2j}/4 \right)^{1/2} - \|u\|_2 \ . \end{aligned} \tag{4.6}$$

We will show, as before, that for appropriate choices of D and b, (4.5) and (4.6) contradict each other, implying that Bob must have correctly extracted his bit and solved Augmented Indexing. To this end, it suffices to prove, after a little algebra and assuming $\epsilon < 1$,

$$3k^{-1/2} \|u\|_2 \leq \left(\tfrac{D^{2j} - D^2}{D^2 - 1} + D^{2j}/4 \right)^{1/2} - \left((1 + 3\epsilon) \tfrac{D^{2j} - D^2}{D^2 - 1} \right)^{1/2} \ . \tag{4.7}$$

Now, choose $D = (1 + 24\epsilon)^{1/2}$. Plugging this into (4.7), it remains to prove

$$3k^{-1/2} \|u\|_2 < \left(\tfrac{(1+24\epsilon)^j - (1+24\epsilon)}{24\epsilon} + \tfrac{(1+24\epsilon)^j}{4} \right)^{1/2} - \left(\tfrac{(1+24\epsilon)^j - (1+24\epsilon)}{24\epsilon} + \tfrac{(1+24\epsilon)^j}{8} \right)^{1/2} \ . \tag{4.8}$$

But, since the square root function $\sqrt{\cdot}$ is concave and has derivative $\frac{1}{2\sqrt{\cdot}}$, we know that, as long as $\epsilon \leq 1/6$, the right-hand side of (4.8) is at least

$$\frac{(1+24\epsilon)^j}{8} \cdot \frac{1}{2\left((1+24\epsilon)^j/(12\epsilon)\right)^{1/2}} = \frac{(1+24\epsilon)^{j/2}}{8/\sqrt{3\epsilon}} \ .$$

Applying the ℓ_2 bound on u, it remains to show

$$3n^4 2^{-b}\sqrt{k}\frac{(1+24\epsilon)^{(j+1)/2}}{(1+24\epsilon)^{1/2}-1} < \frac{(1+24\epsilon)^{j/2}}{8/\sqrt{3\epsilon}} \ ,$$

which reduces to $b > \Theta(\log(n/\epsilon))$ as in the SRPSK$_1$ case. The rest of the proof remains unchanged. □

4.2 Upper Bounds

4.2.1 Reductions to Noisy Sparse Recovery

First we prove a general black box reduction from SRPSK$_1$ to NSR$_1$ that works if the solution to NSR$_1$ has certain additional properties:

Lemma 27. *Suppose we have a solution to NSR$_1$ with parameters (n, k, ϵ), where the $m \times n$ measurement matrix Ψ has $m = m(n, k, \epsilon)$ rows. Suppose in addition that the columns of Ψ are generated i.i.d. from some distribution. Then there exists a solution $(\Phi, \mathcal{R})$ to SRPSK$_1$ with parameters (n, s, k, ϵ) that uses $O(m(s, k, \Theta(\epsilon)))$ measurements. Moreover, if Ψ has, in expectation, $h(n, k, \epsilon)$ non-zeros per column, and the NSR$_1$ recovery time is $t(n, k, \epsilon)$, then Φ has, in expectation, $O(h(s, k, \Theta(\epsilon)))$ non-zeros, and $\mathcal{R}$ runs in $O(t(s, k, \Theta(\epsilon)))$ time.*

Proof. We construct our solution $(\Phi, \mathcal{R})$ to SRPSK$_1$ as follows:

1. Let $\delta > 0$ be a constant to be specified later. Consider an instantiation of the solution to NSR$_1$ with parameters $(s, k, \delta\epsilon)$, so that its measurement matrix Ψ is $m \times s$, where $m = m(s, k, \delta\epsilon)$. Generate the n columns of our $m \times n$ measurement matrix Φ i.i.d. from the same distribution used to generated the i.i.d. columns of Ψ (note that the number of rows m is the same for both Φ and Ψ).

2. Given $S \subseteq [n]$, $|S| = s$, let $\mathcal{R}'_S$ denote the recovery algorithm for NSR$_1$ corresponding to the parameters $(s, k, \delta\epsilon)$ and given the matrix $\Phi\Pi_S^T$ (recall that a recovery algorithm for NSR$_1$ is allowed to behave differently given different instances of the measurement matrix). Define our recovery procedure $\mathcal{R}_S$ by $\mathcal{R}_S(\cdot) = \Pi_S^T(\mathcal{R}'_S(\cdot))$; in words, we run $\mathcal{R}'_S$ on our m-dimensional measurement vector to obtain an s-dimensional vector, which we embed into an n-dimensional vector at positions corresponding to S, filling the rest with zeros.

Note that the number of non-zeros per column of Φ and the running time of $\mathcal{R}$ follow immediately.

Observe that, thanks to the independence of the columns of Φ, the submatrix comprised of any s of them (namely, $\Phi\Pi_S^T$) is a valid $m \times s$ measurement matrix. Thus we have the guarantee that for any signal $x' \in \mathbb{R}^s$ and noise vector $\nu \in \mathbb{R}^m$, $\mathcal{R}'_S$ recovers from $\Phi\Pi_S^T x' + \nu$ a vector $\hat{x}' \in \mathbb{R}^s$ satisfying, with probability 3/4,

$$\|x' - \hat{x}'\|_1 \leq (1+\epsilon)\|x' - x'_{(k)}\|_1 + \delta\epsilon\|\nu\|_1 \ .$$

Now, let $x \in \mathbb{R}^n$ be our signal for SRPSK$_1$. We interpret $\Pi_S x \in \mathbb{R}^s$ to be the sparse signal and $\Phi x_{\bar{S}} \in \mathbb{R}^m$ to be the noise, so that running $\mathcal{R}'_S$ on $\Phi\Pi_S^T(\Pi_S x) + \Phi x_{\bar{S}}$ returns $\hat{x}' \in \mathbb{R}^s$

satisfying, with probability 3/4,

$$\begin{aligned}\|\Pi_S x - \hat{x}'\|_1 &\leq (1+\epsilon)\|\Pi_S x - (\Pi_S x)_{(k)}\|_1 + \delta\epsilon\|\Phi x_{\bar{S}}\|_1 \\ &= (1+\epsilon)\|x_S - x_{S,k}\|_1 + \delta\epsilon\|\Phi x_{\bar{S}}\|_1 \ .\end{aligned}$$

Finally, consider the $\hat{x} \in \mathbb{R}^n$ recovered by $\mathcal{R}_S$ in our procedure for SRPSK$_1$ when run on

$$\Phi x = \Phi x_S + \Phi x_{\bar{S}} = \Phi \Pi_S^T(\Pi_S x) + \Phi x_{\bar{S}} \ .$$

We have $\hat{x} = \Pi_S^T \hat{x}'$, or, equivalently, $\Pi_S \hat{x} = \hat{x}'$, so

$$\begin{aligned}\|x - \hat{x}\|_1 &= \|x_{\bar{S}}\|_1 + \|x_S - \hat{x}\|_1 = \|x_{\bar{S}}\|_1 + \|\Pi_S x - \hat{x}'\|_1 \\ &\leq \|x_{\bar{S}}\|_1 + (1+\epsilon)\|x_S - x_{S,k}\|_1 + \delta\epsilon\|\Phi x_{\bar{S}}\|_1 \\ &= \|x_{\bar{S}}\|_1 + (1+\epsilon)(\|x - x_{S,k}\|_1 - \|x_{\bar{S}}\|_1) + \delta\epsilon\|\Phi x_{\bar{S}}\|_1 \\ &= (1+\epsilon)\|x - x_{S,k}\|_1 - \epsilon\|x_{\bar{S}}\|_1 + \delta\epsilon\|\Phi x_{\bar{S}}\|_1 \ .\end{aligned}$$

Thus, if we can ensure that $\|\Phi x_{\bar{S}}\|_1 \leq (1/\delta)\|x_{\bar{S}}\|_1$, we would obtain the desired guarantee for SRPSK$_1$ of

$$\|x - \hat{x}\|_1 \leq (1+\epsilon)\|x - x_{S,k}\|_1 \ .$$

Now, we claim that $\mathbb{E}[\|\Phi x_{\bar{S}}\|_1] \leq \|x_{\bar{S}}\|_1$, so that by the Markov bound

$$\mathbb{P}\left[\|\Phi x_{\bar{S}}\|_1 > (1/\delta)\|x_{\bar{S}}\|_1\right] \leq \delta \ .$$

Choosing, say, $\delta = 1/12$ would give us an overall success probability of at least 2/3, which can be amplified by independent repetitions and taking a componentwise median in the standard way.

All that remains is then to prove the above claim. By definition of NSR$_1$, we know that for every $v \in \mathbb{R}^s$ and an $m \times s$ matrix Ψ generated according to the specified distribution, we have $\mathbb{E}[\|\Psi v\|_1] \leq \|v\|_1$. But since the columns are by assumption generated i.i.d., combining any s such columns together results in a matrix with this normalization property. In particular, for our $m \times n$ matrix Φ, we can split Φ into groups of s consecutive columns, each of which will have the normalization property. Namely, if c_i denotes the i^{th} column of Φ, we have, for any $u \in \mathbb{R}^n$, and in particular for $u = x_{\bar{S}}$ (for simplicity, assume s divides n),

$$\Phi u = (c_1 u_1 + \cdots + c_s u_s) + \cdots + (c_{n-s+1} u_{n-s+1} + \cdots + c_n u_n) \ ,$$

where

$$\mathbb{E}[\|c_1 u_1 + \cdots + c_s u_s\|_1] \leq \sum_{i=1}^{s} |u_i| \ ,$$

and similarly for the other s-term groups. As a result,

$$\begin{aligned}\mathbb{E}[\|\Phi u\|_1] &\leq \mathbb{E}[\|c_1 u_1 + \cdots + c_s u_s\|_1 + \cdots + \|c_{n-s+1} u_{n-s+1} + \cdots + c_n u_n\|_1] \\ &\leq \sum_{i=1}^{n} |u_i| = \|u\|_1 \ ,\end{aligned}$$

as required. □

Mostly straightforward modification of the above proof yields the ℓ_2/ℓ_2 version, which for technical reasons we will see requires an additional assumption:

Lemma 28. *Suppose we have a solution to NSR$_2$ with parameters (n, k, ϵ), where the $m \times n$ measurement matrix Ψ has $m = m(n, k, \epsilon)$ rows. Suppose in addition that the columns of Ψ are generated i.i.d. from some distribution with zero mean. Then there exists a solution $(\Phi, \mathcal{R})$ to SRPSK$_2$ with parameters (n, s, k, ϵ) that uses $O(m(s, k, \Theta(\epsilon)))$ measurements. Moreover, if Ψ has, in expectation, $h(n, k, \epsilon)$ non-zeros per column, and the NSR$_2$ recovery time is $t(n, k, \epsilon)$, then Φ has, in expectation, $O(h(s, k, \Theta(\epsilon)))$ non-zeros, and $\mathcal{R}$ runs in $O(t(s, k, \Theta(\epsilon)))$ time.*
[2]

Proof. Following the proof of Lemma 27, we can simply replace $\|\cdot\|_1$ with $\|\cdot\|_2^2$ to arrive at the sufficient claim that for $m \times n$ matrix Φ with columns generated i.i.d. according to the specified distribution, and any $u \in \mathbb{R}^n$, $\mathbb{E}[\|\Phi u\|_2^2] \leq \|u\|_2^2$. Unfortunately, with ℓ_2 we cannot avoid square roots and make use of triangle inequality at the same time, so the trick of grouping columns of Φ that worked for the ℓ_1 case does not work here. Instead, consider the u that maximizes $f(u) := \mathbb{E}[\|\Phi u\|_2^2]/\|u\|_2^2$. Since $f(u)$ is clearly invariant under scaling, this is equivalent to maximizing $\mathbb{E}[\|\Phi u\|_2^2]$ subject to $\|u\|_2 = 1$. Expanding this, and expressing the elements of Φ as (ϕ_{ij}), we have

$$\mathbb{E}[\|\Phi u\|_2^2] = \mathbb{E}\left[\left(\sum_j \phi_{1j} u_j\right)^2 + \cdots + \left(\sum_j \phi_{nj} u_j\right)^2\right] . \tag{4.9}$$

Taking the first term and expanding further, we get

$$\mathbb{E}\left[\left(\sum_j \phi_{1j} u_j\right)^2\right] = \mathbb{E}\left[\sum_j \phi_{1j}^2 u_j^2 + 2\sum_{j<j'} \phi_{1j}\phi_{1j'} u_j u_{j'}\right] . \tag{4.10}$$

But observe that for $j < j'$, ϕ_{1j} and $\phi_{1j'}$ are independent and have zero mean, so $\mathbb{E}[\phi_{1j}\phi_{1j'}] = 0$. Equation (4.10) therefore simplifies to

$$\mathbb{E}\left[\left(\sum_j \phi_{1j} u_j\right)^2\right] = \mathbb{E}\left[\sum_j \phi_{1j}^2 u_j^2\right] = \sum_j u_j^2 \mathbb{E}[\phi_{1j}^2] .$$

Returning to Equation (4.9), and simplifying all its terms similarly, we get

$$\mathbb{E}[\|\Phi u\|_2^2] = \sum_j u_j^2 \sum_i \mathbb{E}[\phi_{ij}^2] .$$

But since the columns of Φ are i.i.d., for each j, $\sum_i \mathbb{E}[\phi_{ij}^2]$ must be the same value. It follows that we are simply maximizing $\|u\|_2^2$, so any u with $\|u\|_2 = 1$ is a maximizer. Thus, if we choose u consisting of s 1's followed by $n - s$ 0's, to prove the lemma it suffices to show that for the submatrix $\Phi\Pi_{[s]}^T$ consisting of the first s columns of Φ,

$$\mathbb{E}[\|\Phi\Pi_{[s]}^T(\Pi_{[s]}u)\|_2^2] \leq \|\Pi_{[s]}u\|_2^2 .$$

But we know this is true by definition of NSR$_2$, completing our proof. □

[2] Note that this recovery time is based on the assumption that the solution to NSR generated the columns of its measurement matrix i.i.d. In our application of this reduction (Lemmas 29 and 30), we will need to modify the NSR solution to enforce this requirement, which will increase its recovery time.

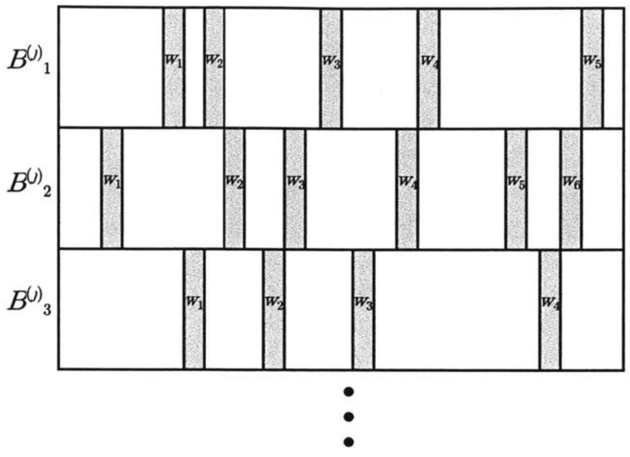

Figure 4.1: Example of an i.i.d. submatrix in $D^{(j)}$ consisting of kc^j blocks. Each grey rectangle represents a code word, and white space represents zeros.

4.2.2 Optimal Algorithm for ℓ_2/ℓ_2

By a modification of the algorithm of [GLPS10], we prove the following result:

Lemma 29. *There exists a solution $(\Phi, \mathcal{R})$ to SRPSK$_2$ with parameters (n, s, k, ϵ) that uses $m = O((k/\epsilon)\log(s/k))$ measurements.*

Proof. To apply a NSR$_2$ solution to SRPSK$_2$ using Lemma 28, we need the columns of the measurement matrix to be generated independently. However, this requirement does not hold with the algorithm in [GLPS10] as is. Therefore, we show how to modify it to satisfy this requirement without changing its recovery properties and asymptotic number of measurements. For simplicity, we will ignore pseudo-randomness considerations, and replace all k-wise independence by full independence in the construction of [GLPS10].

We begin by describing the measurement matrix Φ of [GLPS10] (also denoted by Φ in that paper). At the highest level, Φ is formed by vertically stacking matrices $\Phi^{(j)}$, for $j = 1, \ldots, \log k$. Each $\Phi^{(j)}$ is formed by vertically stacking two matrices, $E^{(j)}$ and $D^{(j)}$. It will suffice for our purposes if the columns of each $E^{(j)}$ and each $D^{(j)}$ are independent.

Consider, first, $E^{(j)}$, which consists of several i.i.d. submatrices, again stacked vertically, in each of which every entry is set i.i.d. (to 1, -1 or 0). Thus, every entry, and therefore every column, of $E^{(j)}$ is already independent without modification.

Next, consider $D^{(j)}$, which consists of several similarly stacked i.i.d. submatrices. For some fixed $c < 1$, each of these submatrices consists of kc^j i.i.d. "blocks" $B_1^{(j)}, B_2^{(j)}, \ldots, B_{kc^j}^{(j)}$, which will be the smallest unit of vertically stacked submatrices we need to consider (see Fig. 4.1). Within each block $B_i^{(j)}$, each column is independently chosen to be non-zero with some probability, and the i^{th} non-zero column is equal to the i^{th} code word w_i from some error-correcting code C. The code C has a constant rate and constant fractional distance. Therefore, each block has $O(\log h)$ rows (and C needs to have $O(h)$ code words), where h is the expected number of non-zero columns per block.

The problem with the construction of $D^{(j)}$ (from our perspective) is that each column chosen to be non-zero is not independently chosen, but instead is determined as a code word that depends on how many non-zero columns are to its left. In order to overcome this obstacle, we observe that the algorithm of [GLPS10] only requires that the code words of the consecutive non-zero columns are *distinct*, not *consecutive.* Thus, we use as our code C' with the same rate and error-correction, but with $O(h^3)$ code words instead of $O(h)$; for each column chosen to be non-zero, we set it to a code word chosen uniformly at random from C', *with replacement.* In terms of Fig. 4.1, each grey rectangle, instead of being the code word from C specified in the figure, is instead a random code word from a larger code C'. Note that each block still has $O(\log h)$ rows as before.

A block is *good* if all code words corresponding to it are distinct. Observe that for any given block, the probability it is not good is at most $O(1/h)$. If there are fewer than $O(h)$ blocks in all of $D^{(j)}$, we could take a union bound over all of them to show that all blocks are good with constant probability. Unfortunately, for $j = 1$, we have $h = O(n/k)$ while the number of blocks is $\Omega(k)$. The latter value could be much larger than h.

Instead, we will simply double the number of blocks. Even though we cannot guarantee that all blocks are good, we know that most of them will be, since each one is with probability $1 - O(1/h)$. Specifically, by the Chernoff bound, at least half of them will be with high probability (namely, $1 - e^{-\Omega(k)}$). We can use only those good blocks during recovery and still have sufficiently many of them to work with.

The result is a solution to NSR$_2$ still with $O((k/\epsilon)\log(n/k))$ rows (roughly 6 times the solution of [GLPS10]), but where each column of the measurement matrix is independent, as required by Lemma 28. The last component we need is that the column distribution has zero mean, but this is guaranteed by the random sign flip applied to every non-zero element. A direct application of the lemma gives us the theorem. □

Lemma 30. *The matrix Φ of Lemma 29 has, in expectation, $O(\log^2 k \log(s/k))$ non-zeros per column, and $\mathcal{R}$ runs in $O(s \log^2 k + (k/\epsilon)\log^{O(1)} s)$ time.*

Proof. It suffices to show that the modifications we made to [GLPS10] do not change the asymptotic expected number of non-zeros in each column and does not increase the recovery time by more than an additive term of $O(n \log^2 k)$. Lemma 28 then gives us this lemma (by replacing n with s in both quantities).

Consider, first, the number of non-zeros. In both the unmodified and the modified matrices, this is dominated by the number of non-zeros in the (mostly dense) code words in the D^j's. But in the modified D^j, we do not change the asymptotic length of each code word, while only doubling, in expectation, the number of code words (in each column as well as overall). Thus the expected number of non-zeros per column of Φ remains $O(\log^2 k \log(n/k))$ as claimed.

Next, consider the running time. The first of our modifications, namely, increasing the number of code words from $O(h)$ to $O(h^3)$, and hence their lengths by a constant factor, does not change the asymptotic running time since we can use the same encoding and decoding functions (it suffices that these be polynomial time, while they are in fact polylogarithmic time). The second of our modifications, namely, doubling the number of blocks, involves a little additional work to identify the good blocks at recovery time. Observe that, for each block, we can detect any collision in time linear in the number of code words. In $D^{(j)}$ there are $O(jkc^j)$ blocks each containing $O(n/(kc^j))$ code words, so the time to process $D^{(j)}$ is $O(jn)$. Thus, overall, for $j = 1, \ldots, \log k$, it takes $O(n \log^2 k)$ time to identify all good blocks. After that, we need only work with the same number of blocks as there had been in the unmodified matrix, so the overall running time is $O(n \log^2 k + (k/\epsilon) \log^{O(1)} n)$ as required. □

Lemmas 29 and 30 together give us Theorem 26.

Bibliography

[ADIW09] A. Andoni, K. Do Ba, P. Indyk, and D. Woodruff. Efficient sketches for earth-mover distance, with applications. *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, 2009.

[AES95] P. K. Agarwal, A. Efrat, and M. Sharir. Vertical decomposition of shallow levels in 3-dimensional arrangements and its applications. *Proceedings of the ACM Symposium on Computational Geometry (SoCG)*, 1995.

[AIK09] A. Andoni, P. Indyk, and R. Krauthgamer. Overcoming the ℓ_1 non-embeddability barrier: Algorithms for product metrics. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2009.

[AV99] P. K. Agarwal and K. Varadarajan. Approximation algorithms for bipartite and non-bipartite matching in the plane. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 1999.

[AV04] P. Agarwal and K. Varadarajan. A near-linear constant factor approximation for euclidean matching? *Proceedings of the ACM Symposium on Computational Geometry (SoCG)*, 2004.

[BCDH10] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde. Model-based compressive sensing. *IEEE Transactions on Information Theory*, 56(4):1982–2001, 2010.

[BGI$^+$08] R. Berinde, A. Gilbert, P. Indyk, H. Karloff, and M. Strauss. Combining geometry and combinatorics: a unified approach to sparse signal recovery. *Proceedings of the Allerton Conference on Communication, Control, and Computing*, 2008.

[BGK$^+$10] A. Bruex, A. Gilbert, R. Kainkaryam, J. Schiefelbein, and P. Woolf. Poolmc: Smart pooling of mRNA samples in microarray experiments. *BMC Bioinformatics*, 11:299, 2010.

[BJKK04] Z. Bar-Yossef, T. S. Jayram, R. Krauthgamer, and R. Kumar. Approximating edit distance efficiently. In *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, 2004.

[CCFC02] M. Charikar, K. Chen, and M. Farach-Colton. Finding frequent items in data streams. *Proceedings of the International Colloquium on Automata, Languages and Programming (ICALP)*, 2002.

[CG99] S. Cohen and L. Guibas. The Earth Mover's Distance under transformation sets. *Proceedings of the International Conference on Computer Vision (ICCV)*, 1999.

[Cha02] M. Charikar. Similarity estimation techniques from rounding. *Proceedings of the ACM Symposium on Theory of Computing (STOC)*, 2002.

[CICB10] V. Cevher, P. Indyk, L. Carin, and R.G Baraniuk. Sparse signal recovery and acquisition with graphical models. *Signal Processing Magazine*, pages 92–103, 2010.

[CM05] G. Cormode and S. Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithms*, 55(1):58–75, 2005.

[CM06] G. Cormode and S. Muthukrishnan. Combinatorial algorithms for compressed sensing. *Proceedings of the Annual Conference on Information Sciences and Systems (CISS)*, 2006.

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

[CW09] K. L. Clarkson and D. P. Woodruff. Numerical linear algebra in the streaming model. *Proceedings of the ACM Symposium on Theory of Computing (STOC)*, 2009.

[DDT$^+$08] M. Duarte, M. Davenport, D. Takhar, J. Laska, T. Sun, K. Kelly, and R. Baraniuk. Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine*, pages 83–91, March 2008.

[Def10] Defense Sciences Office. Knowledge enhanced compressive measurement. *Broad Agency Announcement*, DARPA-BAA-10-38, 2010.

[DI11] K. Do Ba and P. Indyk. Sparse recovery with partial support knowledge. *Proceedings of the International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, 2011.

[DIPW10] K. Do Ba, P. Indyk, E. Price, and D. Woodruff. Lower bounds for sparse recovery. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2010.

[DM09] W. Dai and O. Milenkovic. Weighted superimposed codes and constrained integer compressed sensing. *IEEE Transactions on Information Theory*, 55(5):2215–2229, 2009.

[Don06] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[EB09] Y.C. Eldar and H. Bolcskei. Block-sparsity: Coherence and efficient recovery. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2009.

[EG88] T. Ericson and L. Györfi. Superimposed codes in $\mathbb{R}^n$. *IEEE Transactions on Information Theory*, 34(4):877–880, 1988.

[FMSY12] M. P. Friedlander, H. Mansour, R. Saab, and Ö. Yilmaz. Recovering compressively sampled signals using partial support information. *IEEE Transactions on Information Theory*, 58(2):1122–1134, 2012.

[FPRU10] S. Foucart, A. Pajor, H. Rauhut, and T. Ullrich. The Gelfand widths of ℓ_p-balls for $0 < p \leq 1$. *Journal of Complexity*, 26(6), 2010.

[FR99] Z. Füredi and M. Ruszinkó. An improved upper bound of the rate of euclidean superimposed codes. *IEEE Transactions on Information Theory*, 45(2):799–802, 1999.

[GD05] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2005.

[GG91] A. Gersho and R.M. Gray. *Vector Quantization and Data Compression.* Kluwer, 1991.

[GLPS10] A. Gilbert, Y. Li, E. Porat, and M. Strauss. Approximate sparse recovery: optimizing time and measurements. *Proceedings of the ACM Symposium on Theory of Computing (STOC)*, 2010.

[Glu84] E. D. Gluskin. Norms of random matrices and widths of finite-dimensional sets. *Mathematics of the USSR, Sbornik*, 48:173–182, 1984.

[Hoe63] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[IN07] P. Indyk and A. Naor. Nearest neighbor preserving embeddings. *ACM Transactions on Algorithms*, 3(3), 2007.

[Ind00] P. Indyk. Dimensionality reduction techniques for proximity problems. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2000.

[Ind04] P. Indyk. Algorithms for dynamic geometric problems over data streams. *Proceedings of the ACM Symposium on Theory of Computing (STOC)*, 2004.

[Ind07a] P. Indyk. A near linear time constant factor approximation for euclidean bichromatic matching (cost). In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2007.

[Ind07b] P. Indyk. Sketching, streaming and sublinear-space algorithms. *Graduate course notes, available at* `http://stellar.mit.edu/S/course/6/fa07/6.895/`, 2007.

[IT03] P. Indyk and N. Thaper. Fast color image retrieval via embeddings. *Workshop on Statistical and Computational Theories of Vision (at ICCV)*, 2003.

[IW05] P. Indyk and D. Woodruff. Optimal approximations of the frequency moments of data streams. *Proceedings of the ACM Symposium on Theory of Computing (STOC)*, 2005.

[Jac10] L. Jacques. A short note on compressed sensing with partially known signal support. *Signal Processing*, 90(12):3308–3312, 2010.

[JW09] T.S. Jayram and D. Woodruff. The data stream space complexity of cascaded norms. *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, 2009.

[Kas98] S. Kaski. Dimensionality reduction by random mapping: Fast similarity computation for clustering. *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 1998.

[KN97] E. Kushilevitz and N. Nisan. *Communication Complexity.* Cambridge University Press, 1997.

[KNR99] I. Kremer, N. Nisan, and D. Ron. On randomized one-round communication complexity. *Computational Complexity*, 8(1):21–49, 1999.

[KNW10] D. Kane, J. Nelson, and D. Woodruff. On the exact space complexity of sketching and streaming small norms. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2010.

[KT07] B. S. Kashin and V. N. Temlyakov. A remark on compressed sensing. *Preprint*, 2007.

[KXAH09] M. A. Khajehnejad, W. Xu, A. S. Avestimehr, and B. Hassibi. Weighted ℓ_1 minimization for sparse recovery with prior information. *Proceedings of the International Symposium on Information Theory (ISIT)*, 2009.

[Law76] E. Lawler. *Combinatorial Optimization: Networks and Matroids.* Holt, Rinehart and Winston, 1976.

[LSP06] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[LV10a] W. Lu and N. Vaswami. Modified-cs: Modifying compressive sensing for problems with partially known support. *IEEE Transactions on Signal Processing*, 58(9), 2010.

[LV10b] W. Lu and N. Vaswani. Modified Basis Pursuit Denoising (Modified-BPDN) for noisy compressive sensing with partially known support. *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2010.

[McG06] A. McGregor. Open problems in data streams and related topics. *IITK Workshop on Algorithms For Data Streams*, 2006. Available at `http://www.cse.iitk.ac.in/users/sganguly/workshop.html`.

[MNSW98] P. B. Miltersen, N. Nisan, S. Safra, and A. Wigderson. On data structures and asymmetric communication complexity. *Journal of Computer and System Sciences*, 57(1):37–49, 1998.

[Mut05] S. Muthukrishnan. Data streams: Algorithms and applications. *Foundations and Trends in Theoretical Computer Science*, 2005.

[Nis90] N. Nisan. Pseudorandom generators for space-bounded computation. *Proceedings of the ACM Symposium on Theory of Computing (STOC)*, 1990.

[NS07] A. Naor and G. Schechtman. Planar earthmover is not in l_1. *SIAM Journal on Computing*, 37(3):804–826, 2007.

[Pri11] E. Price. Efficient sketches for the set query problem. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2011.

[PW11] E. Price and D. Woodruff. $(1+\epsilon)$-approximate sparse recovery. *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, 2011.

[PWR89] S. Peleg, M. Werman, and H. Rom. A unified approach to the change of resolution: Space and gray-level. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):739–742, 1989.

[RT99] M. A. Ruzon and C. Tomasi. Corner detection in textured color images. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1999.

[RTG00] Y. Rubner, C. Tomasi, and L. J. Guibas. The Earth Mover's Distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[SAZ10] N. Shental, A. Amir, and Or Zuk. Identification of rare alleles and their carriers using compressed se(que)nsing. *Nucleic Acids Research*, 38(19):1–22, 2010.

[Vai89] P. Vaidya. Geometry helps in matching. *SIAM Journal on Computing*, 18:1201–1225, 1989.

[vBMP07] R. von Borries, C. J. Miosso, and C. Potes. Compressed sensing using prior information. *Proceedings of the IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMPSAP)*, 2007.

[vL98] J. H. van Lint. *Introduction to Coding Theory.* Springer, 1998.

[Wai07] M. Wainwright. Information-theoretic bounds on sparsity recovery in the high-dimensional and noisy setting. *Proceedings of the International Symposium on Information Theory (ISIT)*, 2007.