

One-Bit ExpanderSketch for One-Bit Compressed Sensing

Vasileios Nakos*

January 15, 2019

Abstract

Is it possible to obviously construct a set of hyperplanes \mathcal{H} such that you can approximate a unit vector x when you are given the side on which the vector lies with respect to every $\mathbf{h} \in \mathcal{H}$? In the sparse recovery literature, where x is approximately k -sparse, this problem is called one-bit compressed sensing and has received a fair amount of attention the last decade. In this paper we obtain the first scheme that achieves almost optimal measurements and sublinear decoding time for one-bit compressed sensing in the non-uniform case. For a large range of parameters, we improve the state of the art in both the number of measurements and the decoding time.

1 Introduction

Compressed sensing is a signal processing technique for reconstructing an approximately sparse signal, given access to fewer samples than what the Shannon-Nyquist Theorem requires. This technique was initiated the last decade [CT05, Don06] and has received an enormous amount of attention, because of its many applications in fields like machine learning, signal processing, computer vision, genetics etc. In this setting, one obtains m linear measurements of a signal $x \in \mathbb{R}^n$:

$$y = Ax,$$

where $A \in \mathbb{R}^{m \times n}$, and wants to approximately reconstruct x , exploiting prior information about its sparsity. This setting has been examined very carefully under different guarantees and many algorithms have been suggested.

However, in modern acquisition systems measurements need to be quantized: that it means that we have access only to $y = Q(Ax)$ for some $Q : \mathbb{R}^m \rightarrow \mathcal{A}^m$ [BB08]. In other words, Q maps every element of the encoded vector to an element to a finite alphabet \mathcal{A} . The most common paradigm is when $\mathcal{A} = \{-1, 1\}$ and

$$y = \text{sign}(Ax),$$

where the sign function is applied to any element of the vector. In hardware systems such as the analog-to-digital converter (ADC), quantization is the primary bottleneck limiting sample rates [Wal99, LRRB05]. Moreover, as indicated in [LRRB05], the sampling rate has to decrease exponentially in order for the number of bits to be increased linearly. Furthermore, power consumption is

*Harvard University. vasileiosnakos@g.harvard.edu. Supported in part by NSF grant IIS-1447471.

dominated by the quantizer, leading to increased ADC costs. Thus, the one-bit compressed sensing framework provides a way to disburden the quantization bottleneck by reducing the sampling rate, i.e. the total number of measurements [BB08].

Apart from having important applications, the problem of one-bit compressed sensing is also interesting from a theoretical perspective, as it is a natural and fundamental question on high-dimensional geometry. One can think of it in the following way: can we construct a set of hyperplanes \mathcal{H} such that we can approximate the direction a k -sparse vector $x \in \mathbb{R}^n$ given $\text{sign}(\langle x, h \rangle)$, for all $h \in \mathcal{H}$? If we want a uniform guarantee, i.e. being able to approximate the direction of x for every x , this means that every region defined by the hyperplanes and the sphere must have “small” diameter. Othwise, if we want to reconstruct the direction of x with some target probability, then we demand that most regions defined by the sphere and the hyperplane to have small diameter. The latter formulation is very closely related to the problem of random hyperplane tesslations [PV14].

In this work, we focus on designing a scheme for one-bit compressed sensing that enables sub-linear decoding time in the universe size n . Sub-linear decoding time has been extensively in the sparse recovery literature [GSTV07, GLPS12, PS12, GNP⁺13, GLPS17], but overlooked in the literature of one-bit compressed sensing; the only paper that explored sub-linear decoding time is [Nak17].

1.1 Previous Work

The problem of one-bit compressed sensing was introduced in [BB08], and has received a fair amount of attention till then; one can see [LXZL18] for details. Efficient algorithms, which proceed by solving linear or convex programs when the sensing matrix consists of gaussians, appear in [PV13a, PV13b, GNJN13]. Algorithms that are based on iterative hard-thresholding have been suggested in [JDDV13, JLBB13]. Moreover, the paper of Plan and Vershyn [PV14] studies the very relevant problem of random hyperplane tesslations. The authors in [GNJN13, ABK17] give also combinatorial algorithms for support-recovery from one-bit measurements using combinatorial structures called union-free families. Moreover, [Nak17] gives combinatorial algorithms for one-bit compressed sensing that run in sub-linear time, using ideas from the data streams literature and combinatorial group testing.

The work of [BFN⁺16] introduces schemes for one-bit compressed sensing for the scenario where the underlying signal is sparse with respect to an overcomplete dictionary rather than a basis; this scenario is common in practice. Researchers have also tried to reduce the reconstruction error by employing different techniques and under different models. One approach suggested is Sigma-Delta quantization [KSW16, GLP⁺10]. If adaptivity is allowed and, moreover, the measurements take the form of threshold signs, the authors in [BFN⁺17] show that the reconstruction error can be made exponentially small.

1.2 Our Contribution

In this paper, we study the non-uniform case under adversarial noise and give the first result that achieves sublinear decoding time and nearly optimal $\mathcal{O}(\delta^{-2}k + k \log n)$ measurements, where δ is the reconstruction error, k is the sparsity and n is the universe size. For clearness, this scheme allows reconstruction of a fixed $x \in \mathbb{R}^n$ and not of all $x \in \mathbb{R}^n$; we refer to this a non-uniform guarantee.

We compare with two previous schemes, which are the state of the art. The first scheme appears in [PV13b], which achieves $\delta^{-2}k \log(n/k)$ measurements and $\text{poly}(n)$ decoding time, while the other appears in [Nak17] and achieves $\mathcal{O}(\delta^{-2}k + k \log(n/k)(\log k + \log \log n))$ measurements and $\text{poly}(k, \log n)$ decoding time. We mention that the aforementioned two works are incomparable, since they exchange measurements and decoding time. However, generalizing [Nak17] and using the linking/clustering idea of [LNNT16] (which is closely related to list-recoverable codes), we are able to almost get the best of both worlds. Our scheme is strictly better the scheme of [PV13b] when $k \leq n^{1-\gamma}$, for any constant γ ; we note that the exponent of k in our running time is the same as the exponent of n in the running time of the relevant scheme of [PV13b].

We note that [PV13b] discusses also uniform guarantees for the one-bit compressed sensing problem. Our result is non-uniform and thus incomparable with some of the results in that paper; the relevant parts from [PV13b] are Theorem 1.1 and subsection 3.1. It is important to note that the guarantee of our algorithm **cannot** be achieved in the uniform setting, even when linear measurement are allowed [CDD09] (i.e. we do not have access only to the sign of the measurement), thus a comparison is meaningful (and fair) only with a non-uniform algorithm.

1.3 Preliminaries and Notation

For a vector $x \in \mathbb{R}^n$ we define $H(x, k) = \{i \in [n] : |x_i|^2 \geq \frac{1}{k} \|x_{-k}\|_2^2\}$. If $i \in H(x, k)$, we will say that i is a $1/k$ -heavy hitter of x . For a set S we define x_S to be the vector that occurs after zeroing out every $i \in [n] \setminus S$. We define $\text{head}(k)$ to be the largest k in magnitude coordinates of x , breaking ties arbitrarily, and we define $x_{-k} = x_{[n] \setminus \text{head}(k)}$, which we will also refer to as the tail of x . Let $\mathcal{S}^{n-1} = \{x \in \mathbb{R}^n : \|x\|_2 = 1\}$. For a number θ we set $\text{sign}(\theta) = 1$ if $\theta \geq 0$, and -1 otherwise. For a vector $v = (v_1, v_2, \dots, v_n)$ we set $\text{sign}(v) = (\text{sign}(v_1), \text{sign}(v_2), \dots, \text{sign}(v_n))$. We also denote $\mathcal{P}([n])$ to be the powerset of $[n]$.

Definition 1 (Vertex Expander). *Let $\Gamma : [N] \times [D] \rightarrow [M]$ be a bipartite graph with N left vertices, M right vertices and left degree D . Then, the graph G will be called a (k, ζ) vertex expander if for all sets $S \subseteq [N], |S| \leq k$ it holds that $|\Gamma(S)| \geq (1 - \zeta)|S|D$.*

1.4 Main Result

The main result of our paper is the following.

Theorem 1. *There exists a distribution $\mathcal{D} \in \mathbb{R}^{m \times n}$, a procedure $\text{DEC} : \{-1, +1\}^m \rightarrow \mathbb{R}^n$ and absolute constants $C_1, C_2 > 1$ such that*

$$\forall x \in \mathcal{S}^{n-1} : \mathbb{P}_{\Phi \sim \mathcal{D}}[\hat{x} = \text{DEC}(\text{sign}(\Phi x)) : \|x - \hat{x}\|_2^2 > 2\|x_{-k}\|_2^2 + \delta] \leq e^{-C_1 \delta^{-1} k} + n^{-C_2},$$

and $\|\hat{x}\|_0 = O(k)$.

The number of rows of Φ is $m = \mathcal{O}(k \log n + \delta^{-2} k)$, and the running time of Dec is $\text{poly}(k, \log n)$.

It should clear that since \hat{x} is $O(k)$ -sparse, we do not need to output an n -dimensional vector, but only the positions where the vector is non-zero.

1.5 Overview of our Approach

The one-bit compressed sensing framework has a neat geometrical representation: one can think of every measurement $\text{sign}(\langle \Phi_j, x \rangle)$ indicating on which side of the hyperplane Φ_j the vector x lies. One of the results of [PV13b] shows that this is possible with $\mathcal{O}(\delta^{-2}k \log(n/k))$ random hyperplanes when random post-measurement noise v is added, i.e. $y = \text{sign}(\Phi x + v)$; the paper gives also other, very interesting results, but we will not focus on them in this work. To achieve sublinear decoding time we do not pick the hyperplanes (measurements) at random, but we construct a structured matrix that allows us to find all $1/k$ -heavy hitters of x . This approach also has been followed in one of the schemes of [Nak17]. There the author implemented the dyadic trick [CH08] in the one-bit model, showing that it is possible to recover the heavy hitters of x from one-bit measurements, using $O(k \log(n/k)(\log k + \log \log n))$ measurements. Our results is an extension and generalization of that paper, along with the linking and clustering technique of [LNNT16].

In the core of our scheme, lies the design of a randomized scheme which is analogous to the “partition heavy hitters” data structure of [LNNT16]; we call this scheme ONE-BIT PARTITIONPOINTQUERY. More concretely, the question is the following: given a partition \mathcal{P} of the universe $[n]$, is it possible to decide if a given set $S \in \mathcal{P}$ is heavy, when we are given access only to one-bit measurements? We answer this question in the affirmative and then combine this routine with the graph clustering technique of [LNNT16]. We thus show that, similarly to that paper, it is possible to reduce the problem of finding the heavy coordinates in the one-bit framework to the same clustering problem.

1.6 Toolkit

Lemma 1 (Chernoff Bound). *Let X_1, \dots, X_r be Bernoulli random variables with $\mathbb{E}[X_i] = p$. There exists an absolute constant c_{ch} such that*

$$\mathbb{P} \left[\left| \sum_i X_i - pr \right| > \epsilon pr \right] \leq e^{-c_{ch} \epsilon^{-2} pr}$$

Lemma 2 (Bernstein’s Inequality). *There exists an absolute constant c_B such that for independent random variables X_1, \dots, X_r , with $|X_i| \leq K$ we have that*

$$\forall \lambda > 0, \mathbb{P} \left[\left| \sum_i X_i - \mathbb{E} \sum_i X_i \right| > \lambda \right] \leq e^{-C_B \lambda / \sigma^2} + e^{-C_B \lambda / K},$$

where $\sigma^2 = \sum_i \mathbb{E}(X_i - \mathbb{E}X_i)^2$.

Theorem 2 (Fixed Signal, Random Noise Before Quantization [PV13b]). *Let $x \in \mathbb{R}^N$ and $G \in \mathbb{R}^{m \times N}$, each entry of which is a standard gaussian. If $y = \text{sign}(Gx + v)$, where $v \sim \mathcal{N}(0, \sigma^2 I)$, then the following program*

$$\hat{x} = \text{argmax} \langle y, Gx \rangle, \text{ s.t. } \|z\|_1 \leq \sqrt{k}$$

returns a vector \hat{x} such that $\|x - \hat{x}\|_2^2 \leq \delta$, as long as

$$m = \Omega(\delta^{-2}(\sigma^2 + 1)k \log(N/k)).$$

2 Main Algorithm

Our algorithm proceeds by finding a set S of size $\mathcal{O}(k)$ containing all coordinates $i \in H(x, k)$ and then runs the algorithm of [PV13b], by restricting on columns indexed by S . The scheme that is used to find the desired set S is guaranteed by the following Theorem.

Theorem 3. *There exists a randomized construction of a matrix $\Phi \in \mathbb{R}^{m' \times n}$, a decoding procedure $\text{OneBitHeavyHitters} : \{-1, 1\}^{m'} \rightarrow \mathcal{P}([n])$ and an absolute constant c , such that $S = \text{OneBitHeavyHitters}(\text{sign}(\Phi x))$ satisfies the following, with probability $1 - \frac{1}{n^{c_1}}$. a) $|S| \leq ck$, and b) $\forall i \in H(x, k), i \in S$. Moreover, the number of rows of Φ equals $m' = \mathcal{O}(k \log n)$ and the running time of $\text{OneBitHeavyHitters}$ is $\mathcal{O}(k \cdot \text{poly}(\log n))$.*

Given the above theorem we show how to prove the Theorem 1.

Proof.

We vertically concatenate the matrix Φ from Theorem 3 and the matrix G guaranteed by Theorem 2. Then, we run the algorithm $\text{OneBitHeavyHitters}(\text{sign}(\Phi x))$ to obtain a set S . Then we run the following algorithm:

$$\hat{x} = \text{argmax} \langle y, G_S z \rangle, \text{ s.t. } \|z\|_1 \leq \sqrt{k}.$$

Last, we output \hat{x} . Since $Gx = G_S x_S + G_{[n] \setminus S} x_{[n] \setminus S}$, and $G_{[n] \setminus S} x_{[n] \setminus S} \sim \mathcal{N}(0, \|x_{[n] \setminus S}\|_2^2 I)$ and $\|x_{[n] \setminus S}\|_2 \leq 1$, by combining the guarantees of theorems 2 and 3 we have that

$$\|x - \hat{x}\|_2^2 = \|x_S - \hat{x}_S\|_2^2 + \|x_{[n] \setminus S}\|_2^2 \leq \delta + 2\|x_{-k}\|_2^2,$$

because $\|x_{[n] \setminus S}\|_2^2 \leq \|x_{-k}\|_2^2 + \sum_{i \in \text{head}(k) \setminus H(x, k)} x_i^2 \leq \|x_{-k}\|_2^2 + k \frac{1}{k} \|x_{-k}\|_2^2 = 2\|x_{-k}\|_2^2$. ■

Remark: From the discussion in this subsection, it should be clear than any algorithm that runs in linear time in n and has the same guarantees as as Theorem 2 immediately implies, by our reduction, an algorithm that achieves $\mathcal{O}(k \text{poly}(\log n))$ time. Thus, any subsequent improvement of that type over [PV13b] gives an improvement of our main result in a black-box way.

2.1 Reduction to small Sparsity

The following trick is also used in [LNNT16]. If $k = \Omega(\log n)$, we can hash every coordinate to $\Theta(k/\log n)$ buckets and show that it suffices to find the $\frac{1}{\log n}$ -heavy hitters in every bucket separately. Here, we give a proof for completeness. First, we state the following lemma, which is proven in section 2.4.

Theorem 4. *Let C', C_0 be absolute constants and suppose that $k \leq C' \log n$. Then there exists a randomized construction of a matrix $\Phi \in \mathbb{R}^{m'' \times n}$ with $m'' = \mathcal{O}(k \log n)$ rows, such that given $y = \text{sign}(\Phi x)$, we can find, with probability $1 - n^{-C_0}$ and in time $\mathcal{O}(\text{poly}(\log n))$, a set S of size $\mathcal{O}(k)$ containing every $i \in H(x, k)$.*

Given this lemma, we show how to prove Theorem 2. This lemma is also present in [LNNT16], but, for completeness, we prove it again here.

Proof. If $k < C' \log n$, we run the algorithm guaranteed by the previous lemma. Otherwise, we pick a hash function $g : [n] \rightarrow [C''k/\log n]$ and for $j \in [C''k/\log n]$ we obtain set S_j using lemma. We then output the union of all these sets. Define $z = C''k/\log n$. We argue correctness.

For $j \in [C''k/\log n]$ we use the Chernoff Bound to obtain that

$$\mathbb{P} [|g^{-1}(j) \cap H(x, k)| \geq \log n] \leq e^{-C''' \log n}.$$

We will now invoke Bernstein's inequality for the random variables $\{X_i = \mathbf{1}_{g(i)=j}\}_{i \in [n] \setminus H(x, k)}$; for these variables we have $K < \frac{1}{k} \|x_{-k}\|_2^2$ and

$$\begin{aligned} \sigma^2 &< \sum_{i \in [n] \setminus H(x, k)} x_i^4 (z^{-1} - z^{-2}) \leq \frac{k}{z} \|x_{-k}\|_2^4 \sum_{i \in [n] \setminus H(x, k)} x_i^2 = \frac{k}{z} \|x_{-k}\|_2^4 \\ \mathbb{P} \left[\left| \sum_{i \in g^{-1}(j) \setminus H(x, k)} x_i^2 \geq \frac{\log k}{k} \|x_{-k}\|_2^2 \right. \right] &\leq e^{-C''' \log n}. \end{aligned}$$

By a union-bound over all $2z = 2C''k/\log n$ events, 2 for every buckets $j \in [z]$, we get the proof of the lemma. ■

Our paper now is devoted to proving Theorem 4.

2.2 One-BitPartitionPointQuery

In this section we prove the following Theorem, which is the main building block of our algorithm.

Theorem 5. *Let $x \in \mathbb{R}^n$ and a partition $\mathcal{P} = \{P_1, P_2, \dots, P_T\}$ of $[n]$. There exists an oblivious randomized construction of a matrix $Z \in \mathbb{R}^{m \times n}$ along with a procedure `ONE-BITPARTITIONPOINTQUERY` : $[T] \rightarrow \{0, 1\}$, where $m = \mathcal{O}(k \log(1/\delta))$, such that given $y = \text{sign}(Zx)$ the following holds for $j^* \in [T]$.*

1. *If P_{j^*} contains a coordinate $i \in H(x, k)$, then `ONE-BITPARTITIONPOINTQUERY`(j^*) = 1 with probability $1 - \delta$.*
2. *If there exist at least ck indices such that $\|x_{P_j}\|_2 \geq \|x_{P_{j^*}}\|_2$, then `ONE-BITPARTITIONPOINTQUERY`(j^*) = 0 with probability $1 - \delta$.*

Moreover, The running time of is $\mathcal{O}(\log(1/\delta))$.

We describe the construction of the the matrix Z . We are going to describe the matrix as a set of linear measurements on the vector x . For $i \in [n], j \in [T], B \in [C_B k], \ell \in [3], r \in [C_3 \log(1/\delta)]$ we pick the following random variables:

1. fully independent hash functions $h_{r, \ell} : [T] \rightarrow [C_B k]$.
2. random signs $\sigma_{j, B, \ell, r}$. Intuitively, one can think of this random variable as the sign assigned to set P_j in bucket B of sub-iteration ℓ of iteration r .
3. normal random variables $g_{i, r}$. One can think of this random variable as the gaussian associated with i in iteration r .

Then, for every $B \in [C_B k], \ell \in [3], r \in [C_B \log(1/\delta)]$ we perform linear measurements

$$z_{B,\ell,r} = \sum_{j \in h_{r,\ell}^{-1}(B)} \sigma_{j,B,\ell,r} \sum_{i \in P_j} g_{i,r} x_i,$$

as well as measurements $-z_{B,\ell,r}$ (the reason why we need this will become clear later).

Of course we have access only to the sign of the measurement: $y_{B,\ell,r} = \text{sign}(z_{B,\ell,r})$. We slightly abuse notation here, as y is described as a 3-dimensional vector; it is straightforward to see how this vector can be mapped to a 1-dimensional vector.

We will make use of the following lemmata. The value C_B is a large enough constant, chosen in order for the analysis to work out. Before proceeding with the lemmas, we pick constants C_u, C_d such that

1. $\mathbb{P}_{Y \sim \mathcal{N}(0,1)}[|Y| < C_u] = \frac{19}{20}$.
2. $\mathbb{P}_{Y \sim \mathcal{N}(0,1)}[|Y| > C_d] = \frac{19}{20}$.

Lemma 3. Fix $i^* \in H(x, k)$, j^* such that $i^* \in P_{j^*}$, as well as $r \in [C_3 \log(1/\delta)]$. We also set $B_\ell = h_{r,\ell}(j^*)$. Then, with probability at least $\frac{3}{5}$ we have that for all $\ell \in [3]$ either

$$y_{B_\ell,\ell,r} = \sigma_{j^*,B_\ell,\ell,r} \text{ or } y_{B_\ell,\ell,r} = -\sigma_{j^*,B_\ell,\ell,r}.$$

Proof.

For the need of the proof we define $\mathcal{B}_\ell^{-1} = h_{r,\ell}^{-1}(h_{r,\ell}(j^*))$. First, observe that for all $\ell \in [3]$ that the random variable

$$Y_\ell = \sum_{j \in \mathcal{B}_\ell^{-1} \setminus \{j^*\}} \sigma_{j,B_\ell,\ell,r} \sum_{i \in P_j} g_{i,r} x_i$$

is distributed as

$$\sqrt{\left(\sum_{j \in \mathcal{B}_\ell^{-1} \setminus \{j^*\}} \sum_{i \in P_j} x_i^2 \right)} \cdot \mathcal{N}(0, 1).$$

Observe that with probability at least $\frac{19}{20}$, $|Y_\ell|$ will be at most

$$C_u \sqrt{\sum_{j \in \mathcal{B}_\ell^{-1} \setminus \{j^*\}} \sum_{i \in P_j} x_i^2}.$$

Define

$$Z_\ell = \sum_{j \in \mathcal{B}_\ell^{-1} \setminus \{j^*\}} \sum_{i \in P_j} x_i^2.$$

Consider now the set \mathcal{P}_{bad} of P_j , $j \in [T] \setminus \{j^*\}$ for which there exists $i \in H(x, k)$ such that $i \in P_j$. Since there are at most $2k$ elements in \mathcal{P}_{bad} , with probability at least $1 - \frac{2}{C_B}$ it holds that $\mathcal{B}_\ell^{-1} \cap \mathcal{P}_{\text{bad}} = \emptyset$. Let this event be \mathcal{W} . It is a standard calculation that $\mathbb{E}[Z_\ell | \mathcal{W}] \leq \frac{1}{C_B k} \|x_{-k}\|_2^2$.

Invoking Markov's inequality one gets that Z_l is at most $\frac{20}{C_B k} \|x_{-k}\|_2^2$ with probability at least $\frac{19}{20}$. Putting everything together, this gives that

$$|Y_\ell| > C_u \sqrt{\frac{20}{C_B k}} \|x_{-k}\|_2$$

with probability $\frac{1}{20}$. The probability that there exist $l \in [3]$ such that $|Y_\ell| > C_u \sqrt{\frac{20}{C_B k}} \|x_{-k}\|_2$ is at most $\frac{3}{20}$. We now observe that the

$$\left| \sum_{i \in P_{j^*}} g_{i,r} x_i \right| \geq C_d \|x_{P_{j^*}}\|_2 \geq C_d \frac{1}{\sqrt{k}} \|x_{-k}\|_2$$

with probability at least $\frac{19}{20}$. The above discussion implies that with probability at least $\frac{15}{20}$ the quantity $|\sum_{i \in P_{j^*}} g_{i,r} x_i|$ is larger than $|Y_\ell|$, for all $l \in [3]$, if $C_d/\sqrt{k} > C_u \sqrt{\frac{20}{C_B k}}$. This means that, with probability at least $\frac{3}{4}$, the sign of $z_{B_\ell, \ell, r}$ will be determined by the sign of $\sigma_{j^*, B_\ell, \ell, r} \sum_{i \in P_{j^*}} g_{i,r} x_i$ for all $l \in [3]$. This implies that if $\sum_{i \in P_{j^*}} g_{i,r} x_i > 0$, we will get that $y_{B_\ell, \ell, r} = \sigma_{j^*, B_\ell, \ell, r}$. On the other hand, if $\sum_{i \in P_{j^*}} g_{i,r} x_i < 0$ then $y_{B_\ell, \ell, r} = -\sigma_{j^*, B_\ell, \ell, r}$. This gives the proof of the lemma. \blacksquare

Lemma 4. *Let j^* such that $\|x_{P_{j^*}}\|_2 > 0$. We also define $B_\ell = h_{r, \ell}(j^*)$. Assume that there exist at least ck indices j such that $\|x_{P_j}\|_2 \geq \|x_{P_{j^*}}\|_2$, for some absolute constant c . Then, with probability $\frac{3}{5}$, there exists indices $\ell_1, \ell_2 \in [3]$ such that*

$$y_{B_{\ell_1}, \ell_1, r} = \sigma_{j^*, B_{\ell_1}, \ell_1, r} \text{ and } y_{B_{\ell_2}, \ell_2, r} = -\sigma_{j^*, B_{\ell_2}, \ell_2, r}.$$

Proof. For the need of the proof we also define $\mathcal{B}_\ell^{-1} = h_{r, \ell}^{-1}(h_{r, \ell}(j^*))$. Fix $\ell \in [3]$. Let $\mathcal{P}_{\text{good}}$ be the set of indices $j \in [T]$ such that $\|x_{P_j}\|_2 \geq \|x_{P_{j^*}}\|_2$. Let the random variable Z_ℓ be defined as

$$Z_\ell = |\{j \in \mathcal{P}_{\text{good}} \setminus \{j^*\} : j \in \mathcal{B}_\ell^{-1}\}|.$$

Observe now that $\mathbb{E}[Z_\ell] = \frac{ck}{C_B k} = \frac{c}{C_B}$ and moreover Z_ℓ is a sum of independent Bernoulli random variables with mean $\frac{1}{C_B k}$, hence a standard concentration bound gives that, for c large enough, Z_ℓ will be larger than $4C_d^2 C_u^2$ with probability $\frac{19}{20}$. This implies that

$$\sum_{j \in \mathcal{B}_\ell^{-1} \setminus \{j^*\}} \|x_{P_j}\|_2^2 \geq 4C_d^2 C_u^2 \|x_{P_{j^*}}\|_2^2.$$

for all $\ell \in [3]$. This implies that, for any $\lambda \in \mathbb{R}$,

$$\begin{aligned} \mathbb{P} \left[\left| \sum_{j \in \mathcal{B}_\ell^{-1} \setminus \{j^*\}} \sigma_{j, B_\ell} \sum_{i \in P_j} g_{i,r} x_i \right| \geq \lambda \right] &\geq \\ \mathbb{P} \left[2C_d C_u \|x_{P_{j^*}}\|_2 \cdot |\mathcal{N}(0, 1)| \geq \lambda \right] &. \end{aligned}$$

The above implies that

$$\mathbb{P} \left[\left| \sum_{j \in B_\ell \setminus \{j^*\}} \sigma_{j, B_\ell, \ell, r} \sum_{i \in P_j} g_{i, r} x_i \right| \geq 2C_u \|x_{P_{j^*}}\|_2 \right] \geq \frac{19}{20}$$

and moreover

$$\mathbb{P} \left[\left| \sum_{i \in P_{j^*}} g_{i, r} x_i \right| \leq C_u \|x_{P_{j^*}}\|_2 \right] \geq \frac{19}{20},$$

which implies that with probability $\frac{17}{20}$ we have that

$$\left| \sum_{j \in B_\ell^{-1} \setminus \{j^*\}} \sigma_{j, B_\ell^{-1}, \ell, r} \sum_{i \in P_j} g_{i, r} x_i \right| \leq 2 \left| \sum_{i \in P_{j^*}} g_{i, r} x_i \right|.$$

Observe now that $y_{B_\ell, \ell, r}$ is the same as the sign of $\sum_{j \in B_\ell \setminus \{j^*\}} \sigma_{j, B_\ell^{-1}, \ell, r} \sum_{i \in P_j} g_{i, r} x_i$, which, because of the random signs, means that

$$\mathbb{P} [y_{B_\ell, \ell, r} = 1] = \frac{1}{2}.$$

Moreover, we get that $y_{B_\ell, \ell, r}$ and $\sigma_{j^*, B_\ell, \ell, r}$ are independent. Conditioned on the previous events, the probability that either

$$y_{B_\ell, \ell, r} = \sigma_{j^*, B_\ell, \ell, r}$$

for all $\ell \in [3]$, or

$$y_{B_\ell, \ell, r} = -\sigma_{j^*, B_\ell, \ell, r}$$

for all $\ell \in [3]$, is $\frac{2}{8}$. This gives the proof of the claim since $\frac{3}{20} + \frac{2}{8} \leq \frac{8}{20} = \frac{2}{5}$. ■

We are now ready to proceed with the proof of Theorem 5.

Proof.

We iterate over all $r \in [C_3 \log(1/\delta)]$ and count the number of “good” repetitions: a repetition r is good if for all $\ell \in [3]$, $y_{h_r, \ell(j^*), \ell, r} = \sigma_{j, h_r, \ell(j^*), \ell, r}$ or $y_{h_r, \ell(j^*), \ell, r} = -\sigma_{j, h_r, \ell(j^*), \ell, r}$. We also check if there exists $l \in [3]$ such that $y_{h_r, l(j^*), l, r} = 0$ by checking the values of $y_{h_r, l(j^*), l, r} = 0$ and $-y_{h_r, l(j^*), l, r} = 0$. If there exists no such l and the number of good repetitions is at least $\lceil \frac{1}{2} C_3 \log(T/\delta) \rceil + 1$ we output 1, otherwise we output 0.

We proceed with the analysis. First of all, if there exists an $\ell \in [3]$ that satisfies $y_{h_r, \ell(j^*), \ell, r} = 0$, this would mean that $\|x_{P_{j^*}}\| = 0$. Let us assume that this is not the case, otherwise we can ignore j^* . If $i^* \in H(x, k)$ belongs to P_{j^*} , for some j^* , using Lemma 3 the expected number of good iterations equals $(3/5)C_3 \log(1/\delta)$, and by a Chernoff Bound we get that at least $(2.6/3) \cdot (3/5)C_3 \log(1/\delta) = (2.6/5)C_3 \log(1/\delta)$ repetitions will be good with probability

$$1 - e^{-\Omega(\log(|T|/\delta))} \geq 1 - \delta,$$

for large enough C_3 . In the same way, using Lemma 4 we can bound by δ the probability that a set P_{j^*} , for which there exist at least ck set P_j with $\|x_{P_j}\|_2 \geq \|x_{P_{j^*}}\|_2$, has more than $\lceil \frac{1}{2}C_3 \log(T/\delta) \rceil - 1$ good repetitions. This concludes the proof of the lemma. \blacksquare

The following lemma is immediate by taking $\delta = T^{-C_0-1}$ and taking a union-bound over all $j \in [T]$.

Lemma 5 (ONE-BITPARTITIONCOUNTSKETCH). *Let $x \in \mathbb{R}^n$ and a partition $\mathcal{P} = \{P_1, P_2, \dots, P_T\}$ of $[n]$. There exists a randomized construction of a matrix $Z \in \mathbb{R}^{m \times n}$, such that given $y = \text{sign}(Zx)$, we can find in time $\mathcal{O}(k \log T)$ a set S of size $\mathcal{O}(k)$ that satisfies contains every $j \in [T]$ for which there exists $i \in H_k(x) \cap P_j$. Moreover, the failure probability is T^{-C_0} .*

2.3 One-Bit b -tree

We now describe the scheme of ONE-BIT b -tree. The b -tree is a folklore data structure in streaming algorithms, first appearing in [CH08] in the case of vectors with positive coordinates. The version of the b -tree we are using here is more closely related in [LNNT16]. We remind the reader that the aforementioned papers treated the case where we have access to Φx and not only to $\text{sign}(\Phi x)$. Here, we describe it a sensing matrix associated with a decoding procedure, rather than a data structure. Given the b -tree, we can find elements $i \in H(x, k)$ and get an analog of Theorem 1; however, this would only give $1/\text{poly}(\log n)$ failure probability. Getting $1/\text{poly}(n)$ failure probability requires using the EXPANDERSKETCH algorithm of [LNNT16]. In fact, we can use the ONE-BIT b -tree to speed up the ONE-BIT EXPANDERSKETCH decoding procedure, but since our overall scheme already has a polynomial dependence on k in the running time due to the application of Theorem, this will not give us any crucial improvement. However, we believe that it might of independent interest in the sparse recovery community.

The following lemma holds.

Lemma 6. *Let $k, b < n$ be integers. There exists a randomized construction of a matrix $A \in \mathbb{R}^{M \times n}$ such that given $y = \text{sign}(Ax)$ we can find a set S of size $\mathcal{O}(k)$ such that $\forall i \in H(x, k), i \in S$. The total number of measurements equals*

$$M = \mathcal{O}\left(k \frac{\log(n/k)}{\log b} (\log(k/\delta) + \log \log(n/k) - \log \log b)\right)$$

the decoding time is

$$\mathcal{O}\left(bk \frac{\log(n/k)}{\log b} (\log(k/\delta) + \log \log(n/k) - \log \log b)\right)$$

and the failure probability is δ .

Proof.

Let R be the smallest integer such that $kb^R \geq n$; this means that $R = \lceil \log(n/k)/\log b \rceil$. For $r = 0, \dots, R$ we use the ONE-BIT PARTITIONCOUNTSKETCH scheme guaranteed by Lemma, with $\delta = \delta/(bkR)$ and partition $\mathcal{P}_r = \{\{1, \dots, \lceil \frac{n}{kb^r} \rceil\}, \{\lceil \frac{n}{kb^r} \rceil + 1, \dots, 2 \lceil \frac{n}{kb^2} \rceil\}, \dots\}$, of size $T_r = \Theta(kb^r)$.

The total number of measurements equals

$$\mathcal{O}(Rk \log(bkR/\delta)) = \mathcal{O}\left(k \frac{\log(n/k)}{\log b} (\log(k/\delta) + \log \log(n/k) - \log \log b)\right).$$

We can think of the partitions T_1, T_2, \dots, T_R as the levels of a b -ary tree; for every set if $T \in T_r$, there are b sets $T' \in T_{r+1}$ which are neighbours of T . The decoding algorithms starts at quering the ONE-BIT PARTITIONCOUNTSKETCH for $r = 0$ to obtain a set S_0 . Then, for every $i \in [1, r]$, it computes all the neighbours of S_{r-1} , where the for a total of $\mathcal{O}(b|S|)$ sets. Then using ONE-BITPARTITIONPOINTQUERY we query every new partition, to obtain a set S_r of size $\mathcal{O}(k)$. The output of the algorithm is the set S_R . The running time then is computed as

$$\mathcal{O}(bRk \log(bkR/\delta)) = \mathcal{O}(bk \frac{\log(n/k)}{\log b} (\log(k/\delta) + \log \log(n/k) - \log \log b))$$

■

From the above lemma, we get the following result, by carefully instatianting the parameter b .

Lemma 7. *There exists a b such that the ONE-BIT b -tree uses $\mathcal{O}(\gamma^{-1}k \log(n/\delta))$ measurements and runs in time $\mathcal{O}(\gamma^{-1}(k \log(n/\delta))^{2+\gamma})$, for any arbitrarily constant γ .*

Proof.

We set $b = (k \log(n/\delta))^\gamma$ and observe that the number measurements is at most

$$\mathcal{O} \left(k \frac{\log n}{\gamma(\log(k/\delta) + \log \log n)} (\log(k/\delta) + \log \log n) \right) = \mathcal{O} \left(\frac{1}{\gamma} k \log(n/\delta) \right),$$

while the decoding time becomes

$$\mathcal{O} \left((k \log(n/\delta))^\gamma k \frac{\log n}{\gamma(\log(k/\delta) + \log \log n)} (\log(k/\delta) + \log \log n) \right) = \mathcal{O} \left(\frac{1}{\gamma} (k \log(n/\delta))^{2+\gamma} \right)$$

■

2.4 One-Bit ExpanderSketch

In this subsection we prove Theorem 4. Given the results about ONE-BITPARTITIONCOUNTSKETCH we developed in the previous sections, the proof of the theorem is almost identical to [LNNT16] with a very simple modification. For completeness, we go again over their construction. We remind the reader that in our case $k = \mathcal{O}(\log n)$.

Construction of the Sensing Matrix: We first pick a code $\text{enc} : \{0, 1\}^{\log n} \rightarrow \{0, 1\}^{\mathcal{O}(\log n)}$, which corrects a constant fraction of errors with linear-time decoding; such a code is guaranteed by [Spi96]. We then partition $\text{enc}(i)$ into $s = \Theta(\log n / \log \log n)$ continuous substrings of length $t = \Theta(\log \log n)$. We denote by $\text{enc}(i)_j$ the j -th bitstring of length t in $\text{enc}(i)$.

We define s hash functions $h_1, h_2, \dots, h_s : [n] \rightarrow [\text{poly}(\log n)]$. Let also F be an arbitrary d -regular connected expander on the vertex set $[s]$ for some $d = \mathcal{O}(1)$. For $j \in [s]$, we define $\Gamma_j \subset [s]$ as the set of neighbours of j . Then, for every $j \in [n]$ we define the bit-strings

$$m_{i,j} = h_j(i) \circ \text{enc}(i)_j \circ h_{\Gamma_1(j)}(i) \dots \circ h_{\Gamma_d(j)}(i),$$

and the following partitions $\mathcal{P}^{(j)}$ containg set $P_{m_{i,j}}^{(j)}$, where $m_{i,j}$ is a string of $\Theta(t)$ bits, such that:

$$\forall i \in [n], i \in P_{m_{i,j}}^{(j)}$$

Then for every partition $\mathcal{P}^{(j)}$ we pick a random matrix $\Phi^{(j)}$ using Lemma 5 with sparsity k , as well as a random matrix $Z^{(j)}$ using Lemma 5 with sparsity k and failure probability $\frac{1}{\text{poly}(\log n)}$. Each of these matrices has $\mathcal{O}(k \log(2^{\mathcal{O}(t)})) = \mathcal{O}(kt) = \mathcal{O}(k \log \log n)$ rows. The total number of rows is $\mathcal{O}(sk \log \log n) = \mathcal{O}(k \log n)$. Then our sensing matrix is the vertical concatenation of $\Phi^{(1)}, Z^{(1)}, \dots, \Phi^{(s)}, Z^{(s)}$.

Decoding Algorithm: For every $j \in [s]$ we run the decoding algorithm of Lemma 5 on matrix $\Phi^{(j)}$ to obtain a list L_j of size $\mathcal{O}(k)$ such that every “heavy” set of $\mathcal{P}^{(j)}$ is included. The running time in total is $m \cdot k \cdot \text{poly}(\log n) = \text{poly}(\log n)$. For every $j \in [s]$, we now have that:

- With probability $1/\text{poly}(\log n)$, h_j perfectly hashes every $P_{m_{i,j}}^{(j)}$ for every $i \in H(x, k)$.
- With probability $1/\text{poly}(\log n)$, for every $i^* \in H(x, k)$, $\|x_{P_{m_{i,j}}^{(j)}}\|_2 \geq \frac{9}{10} \|x_{-k}\|_2$.
- With probability $1/\text{poly}(\log n)$, the decoding procedure on $\Phi^{(j)}$ succeeds. This follows by taking a union bound over the events of the previous two bullets and the failure probability guarantee of Lemma 5 in our instance.

We call by “name” of $P_{m_{i,j}}^{(j)}$ the $\mathcal{O}(\log \log n)$ -length substring of bits of $m_{i,j}$, which correspond to the bits of $h_j(i)$. We then filter out vertices in layer j , by keeping only those that have unique names. Our next step is to point-query every set $z \in L_j$ using the matrices $Z^{(j)}$ and Theorem 5 and keep the largest $\mathcal{O}(k)$ coordinates; this is the difference with [LNNT16], since we can implement only one-bit point query. Now we let G be the graph created by including the at most $(d/2) \sum_{j=1}^s L_j$ edges suggested by the z ’s across all L_j , where we only include an edge if both endpoints suggest it. Now the algorithm and analysis proceeds exactly as [LNNT16].

References

- [ABK17] Jayadev Acharya, Arnab Bhattacharyya, and Pritish Kamath. Improved bounds for universal one-bit compressive sensing. *arXiv preprint arXiv:1705.00763*, 2017.
- [BB08] Petros T Boufounos and Richard G Baraniuk. 1-bit compressive sensing. In *Information Sciences and Systems, 2008. CISS 2008. 42nd Annual Conference on*, pages 16–21. IEEE, 2008.
- [BFN⁺16] Rich Baraniuk, Simon Foucart, Deanna Needell, Yaniv Plan, and Mary Wootters. One-bit compressive sensing of dictionary-sparse signals. *arXiv preprint arXiv:1606.07531*, 2016.
- [BFN⁺17] Richard G Baraniuk, Simon Foucart, Deanna Needell, Yaniv Plan, and Mary Wootters. Exponential decay of reconstruction error from binary measurements of sparse signals. *IEEE Transactions on Information Theory*, 63(6):3368–3385, 2017.
- [CDD09] Albert Cohen, Wolfgang Dahmen, and Ronald DeVore. Compressed sensing and best k -term approximation. *Journal of the American mathematical society*, 22(1):211–231, 2009.

- [CH08] Graham Cormode and Marios Hadjieleftheriou. Finding frequent items in data streams. *Proceedings of the VLDB Endowment*, 1(2):1530–1541, 2008.
- [CT05] Emmanuel J Candes and Terence Tao. Decoding by linear programming. *IEEE transactions on information theory*, 51(12):4203–4215, 2005.
- [Don06] David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.
- [GLP⁺10] C Sinan Güntürk, Mark Lammers, Alex Powell, Rayan Saab, and Özgür Yilmaz. Sigma delta quantization for compressed sensing. In *Information Sciences and Systems (CISS), 2010 44th Annual Conference on*, pages 1–6. IEEE, 2010.
- [GLPS12] Anna C Gilbert, Yi Li, Ely Porat, and Martin J Strauss. Approximate sparse recovery: optimizing time and measurements. *SIAM Journal on Computing*, 41(2):436–453, 2012.
- [GLPS17] Anna C Gilbert, Yi Li, Ely Porat, and Martin J Strauss. For-all sparse recovery in near-optimal time. *ACM Transactions on Algorithms (TALG)*, 13(3):32, 2017.
- [GNJN13] Sivakant Gopi, Praneeth Netrapalli, Prateek Jain, and Aditya Nori. One-bit compressed sensing: Provable support and vector recovery. In *International Conference on Machine Learning*, pages 154–162, 2013.
- [GNP⁺13] Anna C Gilbert, Hung Q Ngo, Ely Porat, Atri Rudra, and Martin J Strauss. l_2/l_2 -foreach sparse recovery with low risk. In *International Colloquium on Automata, Languages, and Programming*, pages 461–472. Springer, 2013.
- [GSTV07] Anna C Gilbert, Martin J Strauss, Joel A Tropp, and Roman Vershynin. One sketch for all: fast algorithms for compressed sensing. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 237–246. ACM, 2007.
- [JDDV13] Laurent Jacques, Kévin Degraux, and Christophe De Vleeschouwer. Quantized iterative hard thresholding: Bridging 1-bit and high-resolution quantized compressed sensing. *arXiv preprint arXiv:1305.1786*, 2013.
- [JLBB13] Laurent Jacques, Jason N Laska, Petros T Boufounos, and Richard G Baraniuk. Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors. *IEEE Transactions on Information Theory*, 59(4):2082–2102, 2013.
- [KSW16] Karin Knudson, Rayan Saab, and Rachel Ward. One-bit compressive sensing with norm estimation. *IEEE Transactions on Information Theory*, 62(5):2748–2758, 2016.
- [LNNT16] Kasper Green Larsen, Jelani Nelson, Huy L Nguyễn, and Mikkel Thorup. Heavy hitters via cluster-preserving clustering. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 61–70. IEEE, 2016.
- [LRRB05] Bin Le, Thomas W Rondeau, Jeffrey H Reed, and Charles W Bostian. Analog-to-digital converters. *IEEE Signal Processing Magazine*, 22(6):69–77, 2005.
- [LXZL18] Zhilin Li, Wenbo Xu, Xiaobo Zhang, and Jiaru Lin. A survey on one-bit compressed sensing: theory and applications. *Frontiers of Computer Science*, pages 1–14, 2018.

- [Nak17] Vasileios Nakos. On fast decoding of high-dimensional signals from one-bit measurements. In *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, pages 61:1–61:14, 2017.
- [PS12] Ely Porat and Martin J Strauss. Sublinear time, measurement-optimal, sparse recovery for all. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 1215–1227. Society for Industrial and Applied Mathematics, 2012.
- [PV13a] Yaniv Plan and Roman Vershynin. One-bit compressed sensing by linear programming. *Communications on Pure and Applied Mathematics*, 66(8):1275–1297, 2013.
- [PV13b] Yaniv Plan and Roman Vershynin. Robust 1-bit compressed sensing and sparse logistic regression: A convex programming approach. *IEEE Transactions on Information Theory*, 59(1):482–494, 2013.
- [PV14] Yaniv Plan and Roman Vershynin. Dimension reduction by random hyperplane tessellations. *Discrete & Computational Geometry*, 51(2):438–461, 2014.
- [Spi96] Daniel A Spielman. Linear-time encodable and decodable error-correcting codes. *IEEE Transactions on Information Theory*, 42(6):1723–1731, 1996.
- [Wal99] Robert H Walden. Analog-to-digital converter survey and analysis. *IEEE Journal on selected areas in communications*, 17(4):539–550, 1999.