COMPRESSED SENSING WITH SPARSE BINARY MATRICES: INSTANCE OPTIMAL ERROR GUARANTEES IN NEAR-OPTIMAL TIME

M. A. IWEN
DEPARTMENT OF MATHEMATICS
MICHIGAN STATE UNIVERSITY
619 RED CEDAR ROAD
EAST LANSING, MI 48824
EMAIL: MARKIWEN@MATH.MSU.EDU

Abstract. A compressed sensing method consists of a rectangular measurement matrix, $M \in \mathbb{R}^{m \times N}$ with $m \ll N$, together with an associated recovery algorithm, $\mathcal{A} : \mathbb{R}^m \to \mathbb{R}^N$. Compressed sensing methods aim to construct a high quality approximation to any given input vector $\mathbf{x} \in \mathbb{R}^N$ using only $M\mathbf{x} \in \mathbb{R}^m$ as input. In particular, we focus herein on instance optimal nonlinear approximation error bounds for M and $\mathcal{A}$ of the form $\left\|\mathbf{x} - \mathcal{A}\left(M\mathbf{x}\right)\right\|_p \leq \left\|\mathbf{x} - \mathbf{x}_k^{\rm opt}\right\|_p + Ck^{1/p-1/q}\left\|\mathbf{x} - \mathbf{x}_k^{\rm opt}\right\|_q$ for $\mathbf{x} \in \mathbb{R}^N$, where $\mathbf{x}_k^{\rm opt}$ is the best possible k-term approximation to $\mathbf{x}$.

In this paper we develop a compressed sensing method whose associated recovery algorithm, $\mathcal{A}$, runs in $O\left((k \log k) \log N\right)$-time, matching a lower bound up to a $O(\log k)$ factor. This runtime is obtained by using a new class of sparse binary compressed sensing matrices of near optimal size in combination with sublinear-time recovery techniques motivated by sketching algorithms for high-volume data streams. The new class of matrices is constructed by randomly subsampling rows from well-chosen incoherent matrix constructions which already have a sub-linear number of rows. As a consequence, fewer random bits than previously required are needed in order to select the rows utilized by the fast reconstruction algorithms considered herein.

1. Introduction

Noisy group testing problems generally involve designing pooling schemes which use as few expensive tests as possible in order to identify a small number of important elements from a large universe, $\mathcal{U}$, of items (see, e.g., [13]). In this setting each one of the expensive tests in question corresponds to observing the result of an experiment, or calculation, performed on a different subset of $\mathcal{U}$. If each test is sufficiently sensitive to the small number of hidden items in $\mathcal{U}$ that must be identified, one might hope that testing a correspondingly small number of subsets of $\mathcal{U}$ in bulk would still allow the hidden elements to be found. Thus, designing a pooling scheme corresponds to choosing a good collection of subsets of $\mathcal{U}$ to observe so that tests performed on these subsets will always allow one to discover a small number of important elements hidden within $\mathcal{U}$.

Many data mining tasks can be cast in a similar framework – that is – as problems concerned with identifying a small number of interesting items from a tremendously large group without exceeding certain resource constraints (e.g., without using too much memory, communication power, runtime, etc.). Specific examples include the sketching and monitoring of heavy-hitters in high-volume data streams [5, 7], source localization in sensor networks [25], and the design of high throughput sequencing schemes for biological specimen analysis [14]. Note that pooling schemes in many such group testing related applications naturally correspond to binary matrices (i.e., because each row of the binary matrix, $\mathbf{r} \in \{0, 1\}^N$, selects a subset for testing/observation). Furthermore, it is generally better for these binary binary matrices to have a small number of nonzero entries in each

This research was supported in part by NSA grant H98230-13-1-0275. The majority of the work reported on herein was completed while the author was a visiting assistant professor at Duke University.

column (i.e., because this reduces the number of times each item in $\mathcal{U}$ must be tested/observed). Thus, we focus on designing sparse binary measurement matrices herein.

Roughly speaking, one can cast many such applications as a type of compressed sensing [12] problem. The large set containing the small number of important elements we want to identify is modeled as a vector, $\mathbf{x} \in \mathbb{R}^N$. The n^{th} entry in the vector, x_n, is a real number which indicates the "importance" of the n^{th} set element (the larger the magnitude, the more important). Our goal is now to locate $k \ll N$ of the largest magnitude entries of $\mathbf{x}$ (i.e., the important elements). Unfortunately, for reasons that vary with the specific problem at hand (e.g., because only $o(N)$ memory is available in the massive data stream context), we are allowed to store just $m \ll N$ linear measurements of $\mathbf{x}$ which we must compute during a single pass over its entries. The m linear measurement operators are represented as a measurement matrix, $M \in \mathbb{R}^{m \times N}$. Having access to only $M\mathbf{x} \in \mathbb{R}^m$, we seek to identify, and then estimate, the k largest magnitude entries of $\mathbf{x}$. This identification and estimation is performed by a sparse recovery algorithm, $\mathcal{A} : \mathbb{R}^m \to \mathbb{R}^N$, which (implicitly) returns a vector in $\mathbb{R}^N$ having $O(k)$ nonzero entries. We prefer $\mathcal{A}$ to be fast, especially for applications involving massive data sets.

In this paper we consider the design of sparse matrices $M \in \{0,1\}^{m \times N}$, with $m \ll N$, together with associated nonlinear functions, $\mathcal{A} : \mathbb{R}^m \to \mathbb{R}^N$, which have the property that $\mathcal{A}(M\mathbf{x}) \approx \mathbf{x}$ for all vectors $\mathbf{x} \in \mathbb{R}^N$ that are well approximated by their best k-term approximation,

$$\mathbf{x}_k^{\text{opt}} := \underset{\mathbf{y} \in \mathbb{R}^N, \|\mathbf{y}\|_0 \leq k}{\arg\min} \|\mathbf{x} - \mathbf{y}\|_2.^{1} \tag{1}$$

More specifically, we will focus on designing $(M, \mathcal{A})$-pairs which achieve error guarantees of the form

$$\left\|\mathbf{x} - \mathcal{A}\left(M\mathbf{x}\right)\right\|_p \leq \min_{\mathbf{y} \in \mathbb{R}^N, \|\mathbf{y}\|_0 \leq k} \left\|\mathbf{x} - \mathbf{y}\right\|_p + C_{p,q} \cdot k^{1/p - 1/q} \left\|\mathbf{x} - \mathbf{y}\right\|_q \tag{2}$$

for constants $1 \leq q \leq p \leq 2$, and $C_{p,q} \in \mathbb{R}^+$ (e.g., see [6, 15]). We will refer to such an error guarantee as an "ℓ_p, ℓ_q" error guarantee below.

Over the past several years this type of design problem has achieved a considerable amount of attention under the moniker of "compressed sensing" (e.g., see [15, 12, 4, 10, 3, 2, 20], and references therein). Most compressed sensing papers – this one included – generate their measurement matrices, M, randomly. This leads to two different probabilistic models in which the aforementioned "ℓ_p, ℓ_q" error guarantees may hold. In the first model, a single randomly generated measurement matrix, M, is shown to satisfy (2) *for all* $\mathbf{x} \in \mathbb{R}^N$ with high probability. We will refer to this as the "for all" model. In the second model, a randomly generated measurement matrix is shown to satisfy (2) *for each given* $\mathbf{x} \in \mathbb{R}^N$ with high probability (assuming that M is generated independently of $\mathbf{x}$). We will refer to this second model as the "for each" model. All results proven herein are proven in the second, "for each", model.

1.1. Results and Related Work.

Any sparse recovery algorithm, $\mathcal{A}$, that achieves either an "ℓ_1, ℓ_1", "ℓ_2, ℓ_1", or "ℓ_2, ℓ_2" error guarantee in the "for each" model must use an associated measurement matrix, M, having at least $m \geq Ck\log(N/k))$ rows [11, 23].[2] Note that this implies an $\Omega(k \log(N/k))$ lower runtime complexity bound for the recovery algorithm, $\mathcal{A}$. It remains an open problem to prove (or disprove) the existence of a $O(k \log N)$-time recovery algorithm achieving any such error guarantee. In this paper we present a compressed sensing matrix/recovery algorithm pair, $(M, \mathcal{A})$, with an "ℓ_2, ℓ_1" guarantee, where $\mathcal{A}$ runs in $O((k \log k) \log N)$-time − a single $O(\log k)$-factor from the known lower bound. We also present two other compressed sensing results

[1] Here $\|\mathbf{y}\|_0$ denotes the number of nonzero entries in $\mathbf{y} \in \mathbb{R}^N$, while $\|\mathbf{y}\|_p$ denotes the standard ℓ_p-norm for all $p \geq 1$, i.e., $\|\mathbf{y}\|_p = \left(\sum_{n=0}^{N-1} |y_n|^p\right)^{1/p}$ for all $\mathbf{y} \in \mathbb{R}^N$.

[2] C will always represent an absolute constant.

Paper	**Measurements**, m	**Runtime of** $\mathcal{A}$	**Error Guarantee**
[17]	$k \log^{\geq 2} N$	$k^2 \log^{\geq 2} N$	ℓ_2, ℓ_1 ✓
[22]	$k \log N$	$N^{<1}$	ℓ_1, ℓ_1 ✓
[9]	$k \log^3 N$	$k \log^3 N$	ℓ_2, ℓ_2
[16]	$k \log N$	$k \log^{\geq 2} N$	ℓ_2, ℓ_2
Herein*	$k \log^2 N$	$k \log^2 N$	ℓ_2, ℓ_1
Herein	$(k \log k) \log N$	$(k \log k) \log N$	ℓ_2, ℓ_1

✓Error guarantees hold in the "for all" model.
*Requires only $O(\log^2 k)$ random bits.

Figure 1. Summary of previous sub-linear time results, and the results obtained herein.

which can be obtained using the same methods: one which uses an optimal number (up to constant factors) of randomly selected rows from an incoherent binary matrix as measurements, and another $O(k \log^2 N)$-time recovery result which requires fewer random bits[3] than previous algorithms (i.e., less randomness).

Previous work involving the development of compressed sensing methods having both sub-linear time reconstruction algorithms, and the type of "ℓ_p, ℓ_q" error guarantees considered herein, began with [9]. In [9] Cormode et al. built on streaming algorithm techniques with weaker error guarantees (e.g., see [5, 7, 8]) in order to develop $O(k \log^3 N)$-time recovery algorithms, $\mathcal{A}$, with associated "ℓ_2, ℓ_2" error guarantees in the "for each" model. Similar techniques were later utilized by Gilbert et. al. in [16] to create sub-linear time algorithms with the same error guarantees, but whose associated measurement matrices, $M \in \mathbb{R}^{m \times N}$, have a near-optimal number of rows up to constant factors (i.e., $m = O(k \log N)$). Other related compressed sensing methods with fast runtimes and "ℓ_2, ℓ_1" error guarantees in the "for all" model were also considered in [17]. Unlike these previous methods, the compressed sensing methods developed herein utilize the combinatorial properties of a new class of sparse binary measurement matrices formed by randomly selecting sub-matrices from larger incoherent matrices.

Perhaps the measurement matrices considered herein are most similar to previous compressed sensing matrices based on unbalanced expander graphs (see, e.g., [2, 18, 20]). Indeed, the measurement matrices used in this paper are created by randomly sampling rows from larger binary matrices that are, in fact, the adjacency matrices of a subclass of unbalanced expander graphs. However, unlike previous approaches which use the properties of general unbalanced expanders, we use different combinatorial techniques which allow us to develop $O(k \operatorname{polylog} N)$-time recovery algorithms. To the best of our knowledge, the runtimes we obtain by doing so are the best known for any such method having "ℓ_p, ℓ_q" error guarantees.

See Figure 1 for a comparison of the sub-linear time compressed sensing results proven herein (last two rows) with previous sub-linear time compressed sensing results discussed above (first four rows). The columns of Figure 1 list the following characteristics of each compressed sensing method: (i) the number of measurement matrix rows, m, (ii) the runtime complexity of the recovery algorithm, and (iii) the "ℓ_p, ℓ_q" error guarantee achieved by the method. All error guarantees hold in the "for each" model unless indicated otherwise by a ✓.

1.2. Techniques and Organization.

It has been shown that all binary matrices satisfying easily verifiable coherence conditions[4] have strong combinatorial properties capable of producing entirely

[3] More precisely, the number of random bits is $O(\log^2 k)$. To the best of our knowledge this represents the first fast recovery result which requires a number of random bits that is entirely independent of N, the length of $\mathbf{x}$.

[4] Any matrix whose maximum inner product between all pairs of columns is small compared to the minimal number of ones in each column satisfies the required coherence conditions. See Section 2 for details.

deterministic compressed sensing algorithms requiring $\Omega(k^2 \log N)$ runtime and measurements [1]. In this paper we demonstrate a general means for utilizing these same types of matrices to construct compressed sensing approximation schemes with near-optimal runtime and sampling complexities. Our new compressed sensing matrices are formed by randomly sampling a small number of rows from any sufficiently incoherent binary matrix. The resulting random sub-matrices are then shown to still satisfy sufficiently strong combinatorial properties with respect to any given input vector, $\mathbf{x}$, in order to allow standard fast compressed sensing techniques (i.e., similar of those utilized in [9]) to produce accurate results. Furthermore, the theory is developed in a modular fashion, making it easy to utilize different binary incoherent matrix constructions in order to generate new results. We take advantage of this modularity in order to generate the two new results listed in Figure 1, as well as to show that our new measurement matrices also allow for compressive sensing with an optimal number of measurements (up to constant factors) in $O(N \log N)$-time.[5] Each result is produced by utilizing a different combination of two incoherent binary matrix constructions: deterministic algebraic constructions due to DeVore [10], and randomly constructed incoherent binary matrices with fewer rows constructed below in Section 3.

The remainder of this paper is organized as follows: In Section 2 we fix notation and review existing results that are needed for later sections. In Section 3 we construct incoherent binary matrices with a near optimal number of rows. These new binary matrices ultimately allow the development of our $O\left((k \log k) \log N\right)$-time recovery result via the techniques developed later in Section 4. Section 4 constructs our compressed sensing measurement matrices by randomly sampling rows from the previously discussed binary incoherent matrices (i.e., from both the matrices reviewed in Section 2 as well as the matrices constructed in Section 3). Our main results are then proven in Section 5. Finally, we conclude with a short discussion in Section 6.

2. Preliminaries

Let $[N] = \{0, \dots, N-1\}$ for any $N \in \mathbb{N}$. We consider the elements of any given $\mathbf{x} \in \mathbb{R}^N$ to be ordered according to magnitude by the sequence $j_0, j_1, \dots, j_{N-1}$ so that $|x_{j_0}| \geq |x_{j_1}| \geq \cdots \geq \left|x_{j_{N-1}}\right|$. We set $S_k^{\text{opt}} = \{j_0, j_1, \dots, j_{k-1}\} \subset [N]$ for a given $\mathbf{x}$, and let $\mathbf{x}_{S_k^{\text{opt}}} = \mathbf{x}_k^{\text{opt}} \in \mathbb{R}^N$ denote the associated vector with exactly k nonzero entries:

$$\left(x_k^{\text{opt}}\right)_{j_0} = x_{j_0}, \left(x_k^{\text{opt}}\right)_{j_1} = x_{j_1}, \dots, \left(x_k^{\text{opt}}\right)_{j_{k-1}} = x_{j_{k-1}}.$$

All results below deal with randomly sampling rows from a rectangular binary matrix whose columns are all nearly pairwise orthogonal.

Definition 1. *Let $K, \alpha \in [N]$. An $m \times N$ matrix, $M \in \{0,1\}^{m \times N}$, is called (K, α)-coherent if both of the following properties hold:*

(1) *Every column of M contains at least K ones.*
(2) *For all $j, l \in [N]$ with $j \neq l$, the associated columns, $M_{\cdot,j}$ and $M_{\cdot,l} \in \{0,1\}^m$, have $\langle M_{\cdot,j}, M_{\cdot,l} \rangle \leq \alpha$.*

These matrices are closely related to nonadaptive group testing matrices, unbalanced expander graphs, binary matrices with the restricted isometry property, and codebook design problems in signal processing. Several (implicit) constructions of (K, α)-coherent matrices exist (e.g., the number theoretic and algebraic constructions of [9] and [10], respectively). In addition, every (K, α)-coherent matrix must have $\Omega\left(\min\left\{(K^2/\alpha^2) \log_{K/\alpha} N, N\right\}\right)$ rows. See [1] for details.

Given any binary matrix $M \in \{0,1\}^{m \times N}$ with at least $K \in [m]$ ones in column $n \in [N]$, let $M(K, n)$ denote a $K \times N$ submatrix of M created by selecting the first K rows of M with nonzero

[5] See Theorem 5 for details.

entries in the n^{th} column. The following useful fact concerning (K, α)-coherent matrices is proven in [1].

Lemma 1. *Suppose M is a (K, α)-coherent matrix. Let $n \in [N]$, $k \in \left[\frac{K}{\alpha}\right]$, $\epsilon \in (0, 1]$, $c \in [2, \infty) \cap \mathbb{N}$, and $\mathbf{x} \in \mathbb{R}^N$. If $K > c \cdot (k\alpha/\epsilon)$ then $(M(K, n) \cdot \mathbf{x})_j$ will be contained in the interval $\left(x_n - \frac{\epsilon \left\| \mathbf{x} - \mathbf{x}^{\text{opt}}_{(k/\epsilon)} \right\|_1}{k},\ x_n + \frac{\epsilon \left\| \mathbf{x} - \mathbf{x}^{\text{opt}}_{(k/\epsilon)} \right\|_1}{k} \right)$ for more than $\frac{c-2}{c} \cdot K$ values of $j \in [K]$.*

In addition to (K, α)-coherent matrices, we will also utilize a *bit-test matrix*, $\mathcal{B}_N \in \{0, 1\}^{(1+\lceil \log_2 N \rceil) \times N}$, whose n^{th}-column is a one followed by $n \in [N]$ written in base 2. These bit-test matrices will allow us to quickly identify large elements of a vector $\mathbf{x}$ using techniques from [9]. The *row tensor product* of two matrices, $\mathcal{A} \in \mathbb{R}^{m_1 \times N}$ and $\mathcal{B} \in \mathbb{R}^{m_2 \times N}$, denoted $\mathcal{A} \circledast \mathcal{B}$, is defined to be the $(m_1 \cdot m_2) \times N$ matrix with entries are given by

$$(\mathcal{A} \circledast \mathcal{B})_{i,j} = \mathcal{A}_{i \bmod m_1, j} \cdot \mathcal{B}_{\frac{(i-(i \bmod m_1))}{m_1}, j}.$$

The following Theorem was proven in [1].

Theorem 1. *Let $\epsilon \in (0, 1]$, $k \in \left[K \cdot \frac{\epsilon}{4\alpha}\right]$, and $\mathbf{x} \in \mathbb{R}^N$. Furthermore, suppose that M is an $m \times N$ binary matrix with the property that*

$$(M(K, n) \cdot \mathbf{x})_j \in \left(x_n - \frac{\epsilon \left\| \mathbf{x} - \mathbf{x}^{\text{opt}}_{(k/\epsilon)} \right\|_1}{k},\ x_n + \frac{\epsilon \left\| \mathbf{x} - \mathbf{x}^{\text{opt}}_{(k/\epsilon)} \right\|_1}{k} \right) \tag{3}$$

for more than $K/2$ values of $j \in [K]$ for all $n \in [N]$. Then, there exists an algorithm that takes M and $(M \circledast \mathcal{B}_N)\, \mathbf{x}$ as input, and outputs a vector $\mathbf{z} \in \mathbb{R}^N$ satisfying

$$\|\mathbf{x} - \mathbf{z}\|_2 \ \leq\ \left\| \mathbf{x} - \mathbf{x}_k^{\text{opt}} \right\|_2 + \frac{22\epsilon \left\| \mathbf{x} - \mathbf{x}^{\text{opt}}_{(k/\epsilon)} \right\|_1}{\sqrt{k}}.$$

Furthermore, the algorithm can be implemented to run in $O(m \log N)$ time.

Pseudocode for a faster randomized variant of the algorithm referred to by Theorem 1 can be found in the Appendix. This randomized variant and its associated measurement matrices are the focus of this paper. Briefly put, both algorithms operate in two phases. During the first phase all heavy entires of the input vector, $\mathbf{x}$, are identified using standard bit testing techniques [9]. These heavy vector elements are then estimated during the second round using an approach from the computer science streaming literature [8, 9]. The approximations provided by the binary matrices in Lemma 1 guarantee that taking the median of all K entires of $(M(K, n) \cdot \mathbf{x})$ will provide a good estimate of each important entry, $\mathbf{x}_n$.

2.1. **Efficiently Storing a $\left(K, \left\lfloor \frac{\ln N}{\ln K} \right\rfloor\right)$-coherent Matrix.** Let $P, N \in \mathbb{N}$, where P is prime. DeVore, using techniques along the lines of Kashin [21], gives a deterministic construction of $\left(P, \left\lfloor \frac{\ln N}{\ln P} \right\rfloor\right)$-coherent matrices having P^2 rows and N columns in [10]. This construction, together with Bertrand's postulate, yields general $\left(K, \left\lfloor \frac{\ln N}{\ln K} \right\rfloor\right)$-coherent matrices with $K^2 \leq m < 4K^2$ rows and N columns for any given $K, N \in \mathbb{N}$. In section 4 we will start to construct compressed sensing matrices of near-optimal size by randomly selecting a small set of rows from one of DeVore's deterministic $\left(K, \left\lfloor \frac{\ln N}{\ln K} \right\rfloor\right)$-coherent matrices. However, before doing so we will discuss the complexity of storing and regenerating submatrices of DeVore's $\left(K, \left\lfloor \frac{\ln N}{\ln K} \right\rfloor\right)$-coherent matrices.

As above, let $P, N \in \mathbb{N}$ with P prime. Furthermore, let F be the finite field of order P. Every column of a $P^2 \times N$ $\left(P, \left\lfloor \frac{\ln N}{\ln P} \right\rfloor\right)$-coherent matrix as constructed in [10] has an associated polynomial over F of degree at most $\lceil \log_P N \rceil - 1$. We will assume that these polynomials are ordered so that the polynomial associated j^{th}-column is

$$Q_j(x) := j_0 + j_1 x + j_2 x^2 + \cdots + j_{\lceil \log_P N \rceil - 1} x^{\lceil \log_P N \rceil - 1},$$

where $j_0, \dots, j_{\lceil \log_P N \rceil - 1} \in [P]$ are the digits of $j \in [N]$ base P. That is,

$$j = j_0 + j_1 P + j_2 P^2 + \cdots + j_{\lceil \log_P N \rceil - 1} P^{\lceil \log_P N \rceil - 1}.$$

Thus, Q_j with $j \in [N]$ can be obtained in $O(\log_P N)$-time by finding the representation of j base P.[6]

Let $M \in \{0,1\}^{P^2 \times N}$ be a $\left(P, \lfloor \frac{\ln N}{\ln P} \rfloor\right)$-coherent matrix as constructed in [10]. The P^2 rows of M are indexed by elements of $[P] \times [P]$, ordered lexicographically. Given $j \in [N]$ the ones in the j^{th}-column of M appear in rows

$$(0, Q_j(0)), (1, Q_j(1)), (2, Q_j(2)), \dots, (P-1, Q_j(P-1)).$$

Given $p \in [P]$ we will refer to the set of P rows of M,

$$\left\{(p, r) \mid r \in [P]\right\},$$

as the p^{th} *block of rows*. Every column of M, $j \in [N]$, will have exactly one 1 in each such block. Furthermore, this 1 can be located in $O(\log_P N)$-time by using Horner's rule on Q_j.

3. Existence of Near Optimal (K, α)-coherent Matrices

In this section we will use standard probabilistic arguments to demonstrate the existence of (K, α)-coherent matrices having a near-optimal number of rows. In particular, we will demonstrate that a randomly generated matrix having $m = O\left(\frac{K^2}{\alpha}\right)$ rows will be (K, α)-coherent with high probability. The end result is that the methods herein can be utilized as the basis for a $O(mN)$-time Monte Carlo algorithm for building near-optimal (K, α)-coherent matrices.[7]

DeVore's construction yields $\left(K, \lfloor \frac{\ln N}{\ln K} \rfloor\right)$-coherent matrices having $O(K^2)$ rows, exceeding the lower bound, $\Omega\left(\frac{K^2}{\alpha^2} \log_{K/\alpha} N\right)$, by an $\alpha = O(\log_K N)$-factor (assuming that $K \gg \alpha$). In this section we demonstrate the existence of $(\Theta(K), \Theta(\ln N))$-coherent matrices having $O(K^2/\alpha)$ rows. These matrices exceed the lower bound by a $O(\log K)$ factor, and represent a general improvement over DeVore's construction with respect to row count.

We will build $M \in \{0,1\}^{m \times N}$ by letting each entry, $M_{i,j}$, be an independent and identically distributed Bernoulli random variable that is 1 with probability p and 0 with probability $1-p$. Note that the number of ones in a given column of M will be a binomial random variable in this case. Similarly, the inner product between any two given columns of M will also be binomial. Hence, we may bound both of these quantities using the Chernoff and union bounds. We have the following two lemmas.

Lemma 2. *Let $\sigma, p \in [0,1)$ and $m, N, K \in \mathbb{N}$. Randomly generate a matrix, $M \in \{0,1\}^{m \times N}$, each of whose entries is an i.i.d. Bernoulli random variable which is 1 with probability p. If K is $\Omega(\log N)$ and*

$$mp \;=\; K + \ln\left(\frac{3N}{1-\sigma}\right) + \sqrt{2K \ln\left(\frac{3N}{1-\sigma}\right) + \ln^2\left(\frac{3N}{1-\sigma}\right)} \tag{4}$$

then every column of M will contain $\Theta(K)$ ones with probability at least $1 - \frac{2(1-\sigma)}{3}$.

[6] We assume O(1)-time arithmetic operations (e.g., +, −, ·, /) throughout this paper.

[7] Furthermore, it is worth recalling that these same methods also provide Las Vegas algorithms which run in expected $O(mN^2)$-time.

Proof: Let S_j, $j \in [N]$, be the number of ones in column j of M. We have $\mathbb{E}[S_j] = mp$. The Chernoff bound now implies that

$$\mathbb{P}[S_j < K] = \mathbb{P}\left[S_j < \frac{K}{mp} \cdot \mathbb{E}[S_j]\right] < e^{-mp\left(1-\frac{K}{mp}\right)^2/2}$$

as long as $K < mp$. Thus, we can bound the probability that $S_j < K$ above by $\frac{1-\sigma}{3N}$ for any desired $\sigma \in [0,1)$ by ensuring that

$$\mathbb{P}[S_j < K] < e^{-mp\left(1-\frac{K}{mp}\right)^2/2} \le \frac{1-\sigma}{3N}. \tag{5}$$

Simplifying Equation 5 above, we obtain

$$mp\left(1-\frac{K}{mp}\right)^2 \ge 2\ln\left(\frac{3N}{1-\sigma}\right).$$

Solving for mp in terms of K and N, we learn that

$$(mp)^2 - 2\left(K + \ln\left(\frac{3N}{1-\sigma}\right)\right)(mp) + K^2 \ \ge\ 0.$$

This will hold whenever

$$mp \ \ge\ K + \ln\left(\frac{3N}{1-\sigma}\right) + \sqrt{2K\ln\left(\frac{3N}{1-\sigma}\right) + \ln^2\left(\frac{3N}{1-\sigma}\right)} \ >\ K.$$

Applying Equation 5 together with the union bound over all N choices of S_j yields the desired lower bound. A similar argument guarantees that every row will also have fewer than

$$eK + e\ln\left(\frac{3N}{1-\sigma}\right) + e\sqrt{2K\ln\left(\frac{3N}{1-\sigma}\right) + \ln^2\left(\frac{3N}{1-\sigma}\right)}$$

ones with probability at least $1 - \frac{1-\sigma}{3}$. □

Lemma 3. *Let $\sigma, p \in [0,1)$ and $m, N, K \in \mathbb{N}$. Randomly generate a matrix, $M \in \{0,1\}^{m\times N}$, each of whose entries is an i.i.d. Bernoulli random variable which is 1 with probability p. If $\alpha = 2mp^2 \ge 2\log_{4/e}\left(\frac{3N^2}{1-\sigma}\right)$ then all pairs of columns of M will have inner product at most α with probability at least $1 - \frac{1-\sigma}{3}$.*

Proof: Let $I_{i,j}$ be the inner product of the j^{th} column of M with the i^{th} column of M for a given $i,j \in [N]$ with $i \ne j$. We want $I_{i,j} \le \alpha$. Since $I_{i,j}$ is binomial with $\mathbb{E}[I_{i,j}] = mp^2$ the Chernoff bound implies that

$$\mathbb{P}[I_{i,j} > \alpha] = \mathbb{P}\left[I_{i,j} > \frac{\alpha}{mp^2} \cdot \mathbb{E}[I_{i,j}]\right] < \left[\frac{e^{\alpha/mp^2}}{e\left(\frac{\alpha}{mp^2}\right)^{\alpha/mp^2}}\right]^{mp^2}$$

as long as $\alpha > mp^2$. For the sake of simplicity, suppose that $\alpha = 2mp^2 = 2\log_{4/e}\left(\frac{3N^2}{1-\sigma}\right)$. Then,

$$\mathbb{P}[I_{i,j} > \alpha] < \left(\frac{e}{4}\right)^{\log_{4/e}\left(\frac{3N^2}{1-\sigma}\right)} = \frac{1-\sigma}{3N^2}.$$

In this case the union bound now guarantees that our randomly constructed matrix M will also satisfy the second (K,α)-coherent property with probability at least $1 - \frac{1-\sigma}{3}$. □

Lemma 2 guarantees that a randomly constructed binary matrix will satisfy the first (K,α)-coherent property with high probability. Similarly, Lemma 3 guarantees the second (K,α)-coherent property. Solving for p in light of Equation 4 and Lemma 3 we get that we can set

$$p = \frac{mp^2}{mp} = \frac{\log_{4/e}\left(\frac{3N^2}{1-\sigma}\right)}{K + \ln\left(\frac{3N}{1-\sigma}\right) + \sqrt{2K\ln\left(\frac{3N}{1-\sigma}\right) + \ln^2\left(\frac{3N}{1-\sigma}\right)}} \tag{6}$$

and

$$m = \frac{mp}{p} = \frac{\left(K + \ln\left(\frac{3N}{1-\sigma}\right) + \sqrt{2K\ln\left(\frac{3N}{1-\sigma}\right) + \ln^2\left(\frac{3N}{1-\sigma}\right)}\right)^2}{\log_{4/e}\left(\frac{3N^2}{1-\sigma}\right)}. \tag{7}$$

Note these equations both make sense whenever $K \geq \alpha \geq 2\log_{4/e}\left(\frac{3N^2}{1-\sigma}\right)$. We have the following.

Theorem 2. *Fix $\sigma \in [0,1)$. Let $m, K, \alpha \in [N]$ be such that $K \geq \alpha \geq 2\log_{4/e}\left(\frac{3N^2}{1-\sigma}\right)$, and let $m = \Theta\left(K^2/\alpha\right)$ as per Equation 7. Randomly generate a matrix, $M \in \{0,1\}^{m\times N}$, each of whose entries is an i.i.d. Bernoulli random variable which is 1 with the probability, p, given in Equation 6. Then, M will be both (K,α)-coherent and have $\Theta(K)$ ones per column with probability at least σ.*

Although the matrices developed above have fewer rows than DeVore's, we hasten to point out that they are generally less structured. This ultimately means that they will be difficult to store in compact form, and, therefore, of limited use when space complexity is a dominant concern.

4. Sampling Rows from a (K,α)-coherent Matrix

Given an $m \times N$ matrix M and a subset $s \subset [m]$, we define M_s to be the $|s| \times N$ sub matrix of M consisting of the rows of M contained in s. If s is explicitly specified to be a multiset as opposed to a set, we will (implicitly) repeat rows from M contained in s as necessary. Let l_n denote the number of nonzero entries in the n^{th} column of M_s. We define $M_s(l,n)$, $1 \leq l \leq l_n$, to be the $l \times N$ sub matrix of M_s consisting of the first l rows in M_s with nonzero entries in the n^{th} column.

4.1. Identification Matrix.

The following corollary to Lemma 1 will be used to construct matrices for the identification of the largest magnitude entries in $\mathbf{x}$. Note that the corollary is essentially a coupon collection result (i.e., we want to collect, for each element in $S^{\text{opt}}_{2k/\epsilon}$, a "good row" satisfying Equation 9).

Corollary 1. *Suppose M is an $m \times N$ (K,α)-coherent matrix. Let $\epsilon^{-1} \in \mathbb{N}^+$, $k \in [\epsilon K/\alpha]$, $c \in [14,\infty) \cap \mathbb{N}$, $\sigma \in [2/3,1)$, and $\mathbf{x} \in \mathbb{R}^N$. Select a subset of the rows of M, $s' \subset [m]$, by independently choosing*

$$\gamma \;\geq\; \frac{7}{6} \cdot \frac{m}{K} \ln\left(\frac{2k/\epsilon}{1-\sigma}\right) \tag{8}$$

values from $[m]$ uniformly at random with replacement. If $K > c \cdot (k\alpha/\epsilon)$ then with probability at least σ every $n \in S^{\text{opt}}_{2k/\epsilon} \subset [N]$ will have an associated row of $M_{s'}$, $i_n \in [\gamma]$, for which

$$\left|(M_{s'} \cdot \mathbf{x})_{i_n} - x_n\right| \leq \frac{\epsilon \cdot \left\|\mathbf{x} - \mathbf{x}^{\text{opt}}_{(k/\epsilon)}\right\|_1}{k}. \tag{9}$$

Proof: Fix $n \in S^{\text{opt}}_{2k/\epsilon}$. Lemma 1 implies that each randomly selected row of M, $j \in [m]$, will satisfy Equation 9 with probability at least $\frac{6}{7} \cdot \frac{K}{m}$. Hence, the probability that none of the γ selected rows will satisfy Equation 9 is at most

$$\left(1 - \frac{6}{7} \cdot \frac{K}{m}\right)^{\gamma}.$$

Let $x = \frac{7}{6} \cdot \frac{m}{K}$. If γ satisfies Equation 8 we have that

$$\gamma \left(1 + \sum_{h=2}^{\infty} \frac{1}{hx^{h-1}}\right) \geq x \cdot \ln\left(\frac{2k/\epsilon}{1-\sigma}\right).$$

This in turn implies that

$$\gamma \left(\sum_{h=1}^{\infty} \frac{1}{hx^{h}}\right) \;=\; -\gamma \cdot \ln\left(1 - \frac{1}{x}\right) \geq \ln\left(\frac{2k/\epsilon}{1-\sigma}\right).$$

Thus,

$$\left(1 - \frac{6}{7} \cdot \frac{K}{m}\right)^{\gamma} \;\leq\; \frac{1-\sigma}{2k/\epsilon}$$

whenever γ satisfies Equation 8. Taking the union bound over all $2k/\epsilon$ elements of $S^{\text{opt}}_{2k/\epsilon}$ finishes the proof. □

It is straightforward to show that a random sub matrix, $M_{s'}$, will have $O(\log N)$ ones in every column with high probability when it is constructed as per Corollary 1 from a (K, α)-coherent matrix having $\Theta(K)$ ones per column.[8] It is also important to note that an analogous variant of Corollary 1 can be proven for DeVore's $(\Theta(K), \Theta(\log_K N))$-coherent matrices by randomly selecting $O\left(\ln\left(\frac{2k/\epsilon}{1-\sigma}\right)\right)$ *blocks of* $\Theta(K)$ *rows* (see Section 2.1). Randomly selecting rows from a DeVore matrix in blocks both (i) guarantees that every column of the resulting sub matrix will have $O\left(\ln\left(\frac{2k/\epsilon}{1-\sigma}\right)\right)$ ones, and (ii) requires only $O\left(\ln\left(\frac{2k/\epsilon}{1-\sigma}\right)\ln K\right)$ random bits. The following theorem is proven via standard bit testing techniques (see, e.g., [9, 1]).

Theorem 3. *Suppose M is an $m \times N$ (K, α)-coherent matrix. Let $\epsilon \in (0, 1]$, $\sigma \in [2/3, 1)$, $k \in \left[K \cdot \frac{\epsilon}{14\alpha}\right]$, and $\mathbf{x} \in \mathbb{R}^N$. Construct $M_{s'}$ as per Corollary 1. Then, with probability at least σ, $(M_{s'} \circledast \mathcal{B}_N)\,\mathbf{x}$ will allow Phase 1 (i.e., lines 4 through 14) of Algorithm 1 in the appendix to recover all $n \in [N]$ for which*

$$(10) \qquad |x_n| \geq 4 \frac{\epsilon \cdot \left\|\mathbf{x} - \mathbf{x}^{\text{opt}}_{(k/\epsilon)}\right\|_1}{k}.$$

The required Phase 1 runtime is $O\left(\frac{m}{K} \ln\left(\frac{2k/\epsilon}{1-\sigma}\right) \ln N\right)$.

The only new observation required for the proof of Theorem 3 beyond those used to prove the analogous results in [9, 1, 19] involves noting that any n satisfying Equation 10 also belongs to $S^{\text{opt}}_{2k/\epsilon}$.

To finish, we note that applying (a variant of) Corollary 1 to a $(\Theta(K), \Theta(\log_K N))$-coherent matrix from Section 2.1 produces a random matrix, $M_{s'}$, having

$$O\left(K \ln\left(\frac{2k/\epsilon}{1-\sigma}\right)\right) = O\left(\frac{k}{\epsilon} \cdot \log_{k/\epsilon} N \cdot \ln\left(\frac{2k/\epsilon}{1-\sigma}\right)\right)$$

[8] This result follows via techniques analogous to those utilized in Section 3 in order to establish Theorem 2 (i.e., via the Chernoff and union bounds).

rows. For σ fixed, this reduces to $O\left((k/\epsilon)\log N\right)$ rows. Furthermore, $M_{s'}$ will have $O\left(\log(k/\epsilon)\right)$ ones in all columns. Applying Corollary 1 to a $(\Theta(K), \Theta(\log N))$-coherent matrix from Section 3 produces a random matrix, $M_{s'}$, having

$$O\left(\frac{m}{K}\ln\left(\frac{2k/\epsilon}{1-\sigma}\right)\right) = O\left(\frac{K}{\log N}\cdot\ln\left(\frac{2k/\epsilon}{1-\sigma}\right)\right) = O\left(\frac{k}{\epsilon}\cdot\ln\left(\frac{2k/\epsilon}{1-\sigma}\right)\right)$$

rows. For σ fixed, this reduces to $O\left((k/\epsilon)\log(k/\epsilon)\right)$ rows. Furthermore, $M_{s'}$ will have $O\left(\log N\right)$ ones in all columns with high probability.

4.2. **Estimation Matrix.** The following corollary constructs measurements capable of estimating every entry of $\mathbf{x}$ that is identified as large in magnitude during Phase 1 of Algorithm 1. Furthermore, the estimation procedure is simple, requiring only median operations (see Phase 2 of Algorithm 1).

Corollary 2. *Suppose M is an $m \times N$ (K, α)-coherent matrix. Let $\epsilon^{-1} \in \mathbb{N}^+$, $k \in [\epsilon K/\alpha]$, $c \in [14, \infty) \cap \mathbb{N}$, $\sigma \in [2/3, 1)$, $S \subseteq [N]$, and $\mathbf{x} \in \mathbb{R}^N$. Select a multiset of the rows of M, $\tilde{s} \subset [m]$, by independently choosing*

$$\beta \geq 28.56 \cdot \frac{m}{K}\ln\left(\frac{2|S|}{1-\sigma}\right) \tag{11}$$

values from $[m]$ uniformly at random with replacement. If $K > c \cdot (k\alpha/\epsilon)$ then $M_{\tilde{s}}$ will have both of the following properties with probability at least σ:

1. *There will be at least $\tilde{l} = 21 \cdot \ln\left(\frac{2|S|}{1-\sigma}\right)$ nonzero values in every column of $M_{\tilde{s}}$ indexed by S. Hence, $M_{\tilde{s}}(\tilde{l}, n)$ will be well defined for all $n \in S$.*
2. *For all $n \in S$ more than $l_n/2$ of the entries in $M_{\tilde{s}}(l_n, n) \cdot \mathbf{x}$ (i.e., more than half of the values $j \in [l_n]$, counted with multiplicity) will have*

$$\left|\left(M_{\tilde{s}}(l_n, n)\cdot\mathbf{x}\right)_j - x_n\right| \leq \frac{\epsilon \cdot \left\|\mathbf{x} - \mathbf{x}^{\text{opt}}_{(k/\epsilon)}\right\|_1}{k}.$$

Proof: Fix $n \in S$. We select our multiset, $\tilde{s} \subset [m]$, of the rows of M by independently choosing β elements of $[m]$ uniformly at random with replacement. Denote the j^{th} element chosen for $\tilde{s}$ by $\tilde{s}_j$. Finally, let P^n_j be the random variable indicating whether $M_{\tilde{s}_j, n} > 0$, and let Q^n_j be the random variable indicating whether $\tilde{s}_j$ satisfies

$$\left|\left(M \cdot \mathbf{x}\right)_{\tilde{s}_j} - x_n\right| \leq \frac{\epsilon \cdot \left\|\mathbf{x} - \mathbf{x}^{\text{opt}}_{(k/\epsilon)}\right\|_1}{k} \tag{12}$$

conditioned on P^n_j. Thus, $P^n_j = 1$ if $M_{\tilde{s}_j, n} > 0$, and 0 otherwise. Similarly,

$$Q^n_j = \begin{cases} 1 & \text{if } \tilde{s}_j \text{ satisfies Equation 12 and } P^n_j = 1 \\ 0 & \text{otherwise} \end{cases}.$$

Lemma 1 implies that $\mathbb{P}\left[Q^n_j = 1 \mid P^n_j = 1\right] > \frac{6}{7}$. Furthermore,

$$\mu = \mathbb{E}\left[\sum_{j=1}^{\beta} Q^n_j \mid P^n_1, \dots, P^n_\beta\right] \geq \frac{6}{7}\left(\sum_{j=1}^{\beta} P^n_j\right).$$

Let $l_n = \sum_{j=1}^{\beta} P^n_j$. The Chernoff bound (see, e.g., [24]) guarantees that

$$\mathbb{P}\left[\sum_{j=1}^{\beta} Q^n_j < \frac{4 \cdot l_n}{7} \;\middle|\; l_n\right] \leq e^{-\frac{\mu}{18}} \leq e^{-\frac{l_n}{21}}.$$

Thus, if $l_n > 21$ we can see that $\sum_{j=1}^{\beta} Q_j^n$ will be less than $\frac{l_n+1}{2}$ with probability less than $e^{-\frac{l_n}{21}}$. Hence, if $l_n \geq 21 \ln\left(\frac{2|S|}{1-\sigma}\right)$ then Property 2 will fail to be satisfied for n with probability less than $\frac{1-\sigma}{2|S|}$. Focusing now on l_n, we note that $\mathbb{P}\left[P_j^n = 1\right] \geq \frac{K}{m}$ so that $\tilde{\mu} = \mathbb{E}\left[l_n\right] \geq \frac{K}{m}\beta$.

Let $\tilde{l} = 21 \ln\left(\frac{2|S|}{1-\sigma}\right)$. Applying the Chernoff bound one additional time reveals that $\mathbb{P}\left[l_n < \tilde{l}\right] < e^{-\tilde{\mu}\cdot\left(1-\frac{\tilde{l}}{\tilde{\mu}}\right)^2/2}$. Hence, if we wish to bound $\mathbb{P}\left[l_n < \tilde{l}\right]$ from above by $\frac{1-\sigma}{2|S|}$ it suffices to have $\tilde{\mu}^2 - \frac{44}{21}\tilde{\mu}\tilde{l} + \tilde{l}^2 \geq 0$. Setting $\beta \geq 1.36 \cdot \frac{m}{K}\tilde{l} = 28.56 \cdot \frac{m}{K}\ln\left(\frac{2|S|}{1-\sigma}\right)$ achieves this goal. The end result is that $M_{\tilde{s}}$ will fail to satisfy both Properties 1 and 2 for any $n \in S$ with probability less than $\frac{1-\sigma}{|S|}$. Applying the union bound over all $n \in S$ finishes the proof. □

Note that corollary 2 considers selecting a multiset of rows from a (K, α)-coherent matrix. Hence, some rows may be selected more than once. If this occurs, rows should be considered to be selected multiple times for counting purposes only. That is, all computations involving a row which is selected several times should still be carried out only once. However, the results of these computations should be considered with greater weight during subsequent reconstruction efforts (e.g., multiplely selected rows should be considered as generating multiple duplicate entries in $M_{\tilde{s}} \cdot \mathbf{x}$).

As above, it is straightforward to show that a random sub matrix, $M_{\tilde{s}}$, will have $O(\log N)$ ones in every column with high probability when it is constructed as per Corollary 2 from a (K, α)-coherent matrix having $\Theta\left(K\right)$ ones per column. In addition, an analogous variant of Corollary 2 can be proven for DeVore's $(\Theta(K), \Theta(\log_K N))$-coherent matrices by randomly selecting $O\left(\ln\left(\frac{2|S|}{1-\sigma}\right)\right)$ blocks of $\Theta(K)$ rows. Randomly selecting rows from a DeVore matrix in blocks this way both guarantees that all columns of the resulting sub matrix will have $O\left(\ln\left(\frac{2|S|}{1-\sigma}\right)\right)$ ones, and also requires only $O\left(\ln\left(\frac{2|S|}{1-\sigma}\right)\ln K\right)$ random bits. Note that we must be able to quickly construct arbitrary columns of $M_{\tilde{s}}$ in order to execute Phase 2 of Algorithm 1 in the low memory setting (i.e., when we can not explicitly store either the entire matrix M, or the randomly selected sub matrix $M_{\tilde{s}}$ in memory). In this setting DeVore's $(\Theta(K), \Theta(\log_K N))$-coherent matrices allow us to reconstruct any column of a random sub matrix containing $O\left(\ln\left(\frac{2|S|}{1-\sigma}\right)\right)$ blocks of rows, $M_{\tilde{s}}$, in just $O\left(\ln\left(\frac{2|S|}{1-\sigma}\right)\cdot\log_K N\right)$-time (see Section 2.1 for details).

Corollary 2 will generally be applied with $S \subset [N]$ set to the subset discovered by Phase 1 of Algorithm 1.[9] Hence, we will generally have $|S|$ equal to the number of rows in a matrix $M_{s'}$ constructed via Corollary 1. In more extreme settings, where we want to be able to estimate *all* entries of $\mathbf{x}$ with high probability, we will set $S = [N]$. Corollary 2 implies the following theorem.

Theorem 4. *Suppose M is an $m \times N$ (K, α)-coherent matrix. Let $\epsilon \in (0, 1]$, $\sigma \in [2/3, 1)$, $k \in \left[K \cdot \frac{\epsilon}{14\alpha}\right]$, and $\mathbf{x} \in \mathbb{R}^N$. Construct $M_{\tilde{s}}$ as per Corollary 2. Then, with probability at least σ, $M_{\tilde{s}}\mathbf{x}$ will allow Phase 2 (i.e., lines 15 through 19) of Algorithm 1 in the appendix to estimate all x_n with $n \in S$ with a z_n satisfying*

$$|z_n - x_n| \leq \frac{\epsilon \cdot \left\|\mathbf{x} - \mathbf{x}_{(k/\epsilon)}^{\text{opt}}\right\|_1}{k}. \tag{13}$$

[9] In fact, we select the rows for $M_{\tilde{s}}$ independently of the subset, S, found during Phase 1 of Algorithm 1, *before* the subset has been identified. Note that we only require an upper bound on the size of S before selecting rows from M for our estimation matrix. Such an upper bound is supplied in advance by Corollary 1.

The required Phase 2 runtime (and memory complexity) is $O\left(|S|\ln\left(\frac{2|S|}{1-\sigma}\right)\cdot\log_K N\right)$ when M is a $(\Theta(K),\Theta(\log_K N))$-coherent DeVore matrix. Phase 2 requires $O\left(|S|\cdot\log N\right)$-time if M is a $(\Theta(K),\Theta(\log N))$-coherent matrix from Section 3.

Proof: Equation 13 follows from the second property of $M_{\tilde{s}}$ guaranteed by Corollary 2. Lines 15 through 17 can be accomplished in $O\left(|S|\cdot\ln\left(\frac{2|S|}{1-\sigma}\right)\cdot\log_K N\right)$-time using a median-of-medians algorithm when M is a $(\Theta(K),\Theta(\log_K N))$-coherent DeVore matrix. When M is a $(\Theta(K),\Theta(\log N))$-coherent matrix from Section 3, lines 15 through 17 can be accomplished in $O\left(|S|\cdot\log N\right)$-time.[10] Lines 18 and 19 can always be accomplished in $O(|S|\log|S|)$-time. □

We conclude this section by noting that applying (a variant of) Corollary 2 to a $(\Theta(K),\Theta(\log_K N))$-coherent matrix from Section 2.1 produces a random matrix, $M_{\tilde{s}}$, having

$$O\left(K\ln\left(\frac{2|S|}{1-\sigma}\right)\right)=O\left(\frac{k}{\epsilon}\cdot\log_{k/\epsilon}N\cdot\ln\left(\frac{2|S|}{1-\sigma}\right)\right)$$

rows. Applying Corollary 2 to a $(\Theta(K),\Theta(\log N))$-coherent matrix from Section 3 produces a random matrix, $M_{\tilde{s}}$, having

$$O\left(\frac{m}{K}\ln\left(\frac{2|S|}{1-\sigma}\right)\right)=O\left(\frac{K}{\log N}\cdot\ln\left(\frac{2|S|}{1-\sigma}\right)\right)=O\left(\frac{k}{\epsilon}\cdot\ln\left(\frac{2|S|}{1-\sigma}\right)\right)$$

rows. For σ fixed, this reduces to $O\left((k/\epsilon)\log(|S|)\right)$ rows. Furthermore, $M_{\tilde{s}}$ will have $O\left(\log N\right)$ ones in all columns with high probability.

5. Main Results

We may now prove the three new results mentioned in Section 1.1. We have the following theorem.

Theorem 5. *Let $\epsilon\in(0,1]$, $\sigma\in[2/3,1)$, $\mathbf{x}\in\mathbb{R}^N$, and $k\in[N]$.[11] With probability at least σ Algorithm 1 will output a vector $\mathbf{z}\in\mathbb{R}^N$ satisfying*

$$\|\mathbf{x}-\mathbf{z}\|_2 \;\leq\; \left\|\mathbf{x}-\mathbf{x}_k^{\mathrm{opt}}\right\|_2+\frac{22\epsilon\left\|\mathbf{x}-\mathbf{x}_{(k/\epsilon)}^{\mathrm{opt}}\right\|_1}{\sqrt{k}} \tag{14}$$

when executed using any of the following identification and estimation matrices:

1. *A $(\Theta(k\log N/\epsilon),\Theta(\log N))$-coherent matrix from Section 3 used for estimation via Corollary 2 with $S=[N]$. Only Phase 2 of Algorithm 1 need be applied (i.e., no identification will be performed). The resulting number of measurements is $O\left(\frac{k}{\epsilon}\cdot\ln\left(\frac{N}{1-\sigma}\right)\right)$. The required runtime is $O(N\log N)$.*
2. *A $\left(\Theta(\frac{k}{\epsilon}\log_{k/\epsilon}N),\Theta(\log_{k/\epsilon}N)\right)$-coherent matrix from Section 2.1 used for both identification (via Corollary 1 variant) and estimation (via Corollary 2 variant with $|S|=O(\frac{k}{\epsilon}\ln\left(N/(1-\sigma)\right))$). The resulting number of measurements is $O\left(\frac{k}{\epsilon}\cdot\ln\left(\frac{N}{1-\sigma}\right)\ln N\right)$. The required runtime is $O\left(\frac{k}{\epsilon}\cdot\ln^2\left(\frac{N}{1-\sigma}\right)\right)$.*

[10] However, using the matrixes from Section 3 requires $O(N\log N)$-memory since their columns contain ones in random locations that must be remembered.

[11] For the sake of simplicity, we assume $k=\Omega(\log N)$ when stating the measurement and runtime bounds below.

(3) *A* $(\Theta(k\log N/\epsilon), \Theta(\log N))$*-coherent matrix from Section 3 used for identification (via Corollary 1), and a* $\left(\Theta(\frac{k}{\epsilon}\log_{k/\epsilon} N), \Theta(\log_{k/\epsilon} N)\right)$*-coherent matrix from Section 2.1 used for estimation (via Corollary 2 variant with* $|S| = O(\frac{k}{\epsilon}\ln(k/\epsilon(1-\sigma)))$*). The resulting number of measurements is* $O\left(\frac{k}{\epsilon}\cdot\ln\left(\frac{k/\epsilon}{1-\sigma}\right)\ln N\right)$*. The required runtime is* $O\left(\frac{k}{\epsilon}\cdot\ln\left(\frac{k/\epsilon}{1-\sigma}\right)\ln\left(\frac{N}{1-\sigma}\right)\right)$*.*

Proof: The runtime and measurement bounds follow from Theorem 3, Theorem 4, and the subsequent Section 4 discussions. The error guarantee for $\mathbf{z}$ follows from Theorem 3, Theorem 4, and the proof of Theorem 7 in [19]. □

It is interesting to consider the possibility of improving the runtime bounds obtained in Theorem 5 by using iterative recovery techniques akin to those employed in [16]. This appears to be difficult. In particular, such iterative recovery methods generally require the contributions of partial solutions to be subtracted from the input measurements of the original vector, $\mathbf{x}$, after each of $O(\log k)$ rounds. Assuming that one must subtract some partial solution containing at least $\Omega(k)$ nonzero entries from a large (constant fraction) of the initial measurements of $\mathbf{x}$ at some point during reconstruction, it becomes clear that updating our measurements will not be $O(k\log N)$-time unless our measurement matrix contains $O(\log N)$ nonzero entries per column. Unfortunately, fast nonadaptive identification of previously undiscovered heavy elements of $\mathbf{x}$ (e.g., via bit-testing methods) requires the use of matrices having $\Omega(\log N)$ nonzero entries in many columns during each new round of iterative approximation. Hence, it appears as if only $O(1)$ rounds of identification may be performed using the techniques considered herein before the required measurement matrices have too many ones per column in order to allow $O(k\log N)$-time recovery. The author considers this as (a weak) justification for utilizing only one round of identification in Algorithm 1.

6. Conclusion

In this paper we present a compressed sensing recovery algorithm with an "ℓ_2, ℓ_1" error guarantee that runs in only $O((k\log k)\log N)$-time. This runtime is within a $O(\log k)$ factor of the known lower $\Omega(k\log N)$ runtime bound. Demonstrating (or refuting) the existence of a $O(k\log N)$-time (i.e., linear-time in its required input size) compressed sensing recovery algorithm with similar error guarantees remains an open problem.

References

[1] J. Bailey, M. A. Iwen, and C. V. Spencer. On the design of deterministic matrices for fast recovery of fourier compressible functions. *SIAM J. Matrix Anal. Appl.*, 33(1):263 – 289, 2012.

[2] R. Berinde, A. Gilbert, P. Indyk, H. Karloff, and M. Strauss. Combining geometry and combinatorics: A unified approach to sparse signal recovery. In *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, pages 798–805. IEEE, 2008.

[3] T. Blumensath and M. E. Davies. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, 27(3):265 – 274, 2009.

[4] E. J. Candes, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Comm. Pure Appl. Math.*, 59(8):12081223, 2006.

[5] M. Charikar, K. Chen, and M. Farach-Colton. Finding frequent items in data streams. *Automata, Languages and Programming*, pages 784–784, 2002.

[6] A. Cohen, W. Dahmen, and R. DeVore. Compressed Sensing and Best k-term Approximation. *Journal of the American Mathematical Society*, 22(1):211–231, January 2008.

[7] G. Cormode and S. Muthukrishnan. What's hot and what's not: tracking most frequent items dynamically. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 296–306. ACM, 2003.

[8] 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.

[9] G. Cormode and S. Muthukrishnan. Combinatorial algorithms for compressed sensing. *Structural Information and Communication Complexity*, pages 280–294, 2006.

[10] R. DeVore. Deterministic constructions of compressed sensing matrices. *Journal of Complexity*, 23(4-6):918–925, 2007.

[11] K. Do Ba, P. Indyk, E. Price, and D. Woodruff. Lower bounds for sparse recovery. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1190–1197. Society for Industrial and Applied Mathematics, 2010.

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

[13] D. Du and F. Hwang. *Combinatorial group testing and its applications*. World Scientific Pub Co Inc, 2000.

[14] Y. Erlich, K. Chang, A. Gordon, R. Ronen, O. Navon, M. Rooks, and G. Hannon. Dna sudokuharnessing high-throughput sequencing for multiplexed specimen analysis. *Genome research*, 19(7):1243–1253, 2009.

[15] S. Foucart and H. Rauhut. *A Mathematical Introduction to Compressive Sensing*. Springer, to appear.

[16] A. Gilbert, Y. Li, E. Porat, and M. Strauss. Approximate sparse recovery: optimizing time and measurements. In *Proceedings of the 42nd ACM symposium on Theory of computing*, pages 475–484. ACM, 2010.

[17] A. C. Gilbert, M. J. Strauss, J. A. Tropp, and R. Vershynin. One sketch for all: fast algorithms for compressed sensing. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, STOC '07, pages 237–246, New York, NY, USA, 2007. ACM.

[18] P. Indyk and M. Ruzic. Near-optimal sparse recovery in the l1 norm. In *Foundations of Computer Science, 2008. FOCS'08. IEEE 49th Annual IEEE Symposium on*, pages 199–207. IEEE, 2008.

[19] M. A. Iwen. Improved approximation guarantees for sublinear-time fourier algorithms. *Applied and Computational Harmonic Analysis*, 34(1):57 – 82, 2013.

[20] S. Jafarpour, W. Xu, B. Hassibi, and R. Calderbank. Efficient and robust compressed sensing using optimized expander graphs. *Information Theory, IEEE Transactions on*, 55(9):4299–4308, 2009.

[21] B. S. Kashin. The diameters of octahedra. *Uspekhi Mat. Nauk*, 30(4):251–252, 1975.

[22] E. Porat and M. 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, 2012.

[23] E. Price and D. Woodruff. (1+ eps)-approximate sparse recovery. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 295–304. IEEE, 2011.

[24] P. Raghavan and R. Motwani. *Randomized Algorithms*. Cambridge Univ. Press, 1995.

[25] Y. Zheng, N. Pitsianis, and D. Brady. Nonadaptive group testing based fiber sensor deployment for multiperson tracking. *Sensors Journal, IEEE*, 6(2):490–494, 2006.

Appendix A. The Recovery Algorithm

See Algorithm 1 below.

Algorithm 1 Approximate $\mathbf{x}$

```
Input: $M_{\tilde{s}}$ and $M_{\tilde{s}}\mathbf{x}$ for estimation, and $(M_{s'} \circledast \mathcal{B}_N)\,\mathbf{x}$ for identification
Output: $\mathbf{z}$, an approximation to $\mathbf{x}_k^{\text{opt}}$
Initialize multiset $S \leftarrow \emptyset,\ \mathbf{z} \leftarrow \mathbf{0} \in \mathbb{R}^N,\ \mathbf{b} \leftarrow \mathbf{0} \in \mathbb{R}^{\lceil \log_2 N \rceil}$
                    Phase 1: Identify All Heavy $n \in [0, N) \cap \mathbb{N}$
for $j$ from 1 to $|s'|$ do
    for $i$ from 1 to $\lceil \log_2 N \rceil$ do
        if $\left|\left(M_{s'} \circledast (\mathcal{B}_N)_{i+1}\,\mathbf{x}\right)_j\right| > \left|\left(M_{s'}\mathbf{x} - M_{s'} \circledast (\mathcal{B}_N)_{i+1}\,\mathbf{x}\right)_j\right|$ then
            $b_i \leftarrow 1$
        else
            $b_i \leftarrow 0$
        end if
    end for
    $n \leftarrow \sum_{i=0}^{\lceil \log_2 N \rceil - 1} b_{i+1} 2^i$
    $S \leftarrow S \cup \{n\}$
end for
                    Phase 2: Estimate $\mathbf{x}_S \approx \mathbf{x}_k^{\text{opt}}$ Using Equation 3
for each $n$ value belonging to $S$ do
    $z_n \leftarrow$ median of multiset $\left\{(M_{\tilde{s}}(l_n, n) \cdot \mathbf{x})_h \;\middle|\; 1 \le h \le l_n\right\}$
end for
Sort nonzero $\mathbf{z}$ entries by magnitude so that $|z_{n_1}| \ge |z_{n_2}| \ge |z_{n_3}| \ge \dots$
$S \leftarrow \{n_1, n_2, \dots, n_{2k}\}$
Output: $\mathbf{z}_S$
```