Combinatorial Group Testing and Sparse Recovery Schemes with Near-Optimal Decoding Time

Mahdi Cheraghchi
EECS Department
University of Michigan
Ann Arbor, MI, USA
Email: mahdich@umich.edu

Vasileios Nakos
Computer Science Department
Saarland University and Max-Planck Institute for Informatics
Saarbrucken, Germany
Email: billynak@gmail.com

***Abstract*—In the long-studied problem of combinatorial group testing, one is asked to detect a set of k defective items out of a population of size n, using $m \ll n$ disjunctive measurements. In the non-adaptive setting, the most widely used combinatorial objects are disjunct and list-disjunct matrices, which define incidence matrices of test schemes. Disjunct matrices allow the identification of the exact set of defectives, whereas list disjunct matrices identify a small superset of the defectives. Apart from the combinatorial guarantees, it is often of key interest to equip measurement designs with efficient decoding algorithms. The most efficient decoders should run in sublinear time in n, and ideally near-linear in the number of measurements m.**

In this work, we give several constructions with an optimal number of measurements and near-optimal decoding time for the most fundamental group testing tasks, as well as for central tasks in the compressed sensing and heavy hitters literature. For many of those tasks, the previous measurement-optimal constructions needed time either quadratic in the number of measurements or linear in the universe size.

Among our results are the following: a construction of disjunct matrices matching the best-known construction in terms of the number of rows m, but achieving nearly linear decoding time in m; a construction of list disjunct matrices with the optimal $m = O(k \log(n/k))$ number of rows and nearly linear decoding time in m; error-tolerant variations of the above constructions; a non-adaptive group testing scheme for the "for-each" model with $m = O(k \log n)$ measurements and $O(m)$ decoding time; a streaming algorithm for the "for-all" version of the heavy hitters problem in the strict turnstile model with near-optimal query time, as well as a "list decoding" variant obtaining also near-optimal update time and $O(k \log(n/k))$ space usage; an ℓ_2/ℓ_2 weak identification system for compressed sensing with nearly optimal sample complexity and nearly linear decoding time in the sketch length.

Most of our results are obtained via a clean and novel approach that avoids list-recoverable codes or related complex techniques that were present in almost every state-of-the-art work on efficiently decodable constructions of such objects.

***Keywords*-group testing, heavy hitters, compressed sensing**

The full preprint of this work can be found in [1].

Vasileios Nakos: This work is part of the project TIPEA that has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement No. 850979).

I. Introduction

The study of combinatorial group testing dates back to the Second World War, suggested by Dorfman [2] in the context of testing blood samples collected from a large population of draftees. In an abstract formulation, a population of n individuals contains up to k, for a known parameter k, defectives and tests are conducted to identify the exact set of defectives. Each test identifies a subset of the individuals and returns positive if and only if the set contains at least one defective individual. The basic combinatorial goal is to minimize the number of tests required to identify the exact set of defectives in the worst case. This article focuses on non-adaptive tests where the tests are all pre-determined and can be conducted in parallel. In this case, the test design can be identified by a binary incidence matrix with n columns and one row per test.

Since its inception, group testing has found countless uses both in theory and practice. Practical applications include a wide range of areas such as molecular biology and DNA library screening (cf. [3]–[11] and the references therein), Human Genome Project (cf. [12, Section VI.46]), multiple access communication [13], data compression [14], pattern matching [15], secure key distribution [16], network tomography [17], quality control [18], among others. The reader is invited to consult [19], [20] for a more comprehensive discussion of the application areas. Finally, the original idea of using group testing for pooling samples in medical tests has recently gained renewed interest during the COVID-19 pandemic due to the prevalent shortage of test kits (cf. [21]–[28]).

In theoretical computer science, group testing falls under the broader umbrella of *sparse recovery*, where the general framework deals with the recovery of sparse structures (such as high-dimensional vectors with few nonzero entries or their approximations) via queries from a restricted class (such as linear queries, as in *compressed sensing* [29], disjunctive queries which define group testing, or by sampling Fourier coefficients of the underlying vector [30]–[33]). The general area of sparse recovery provides a fundamental toolkit

2575-8454/20/$31.00 ©2020 IEEE
DOI 10.1109/FOCS46700.2020.00115

for the study of streaming and sublinear time algorithms, and technology from that area lies at the heart of the latest improvements for Subset Sum [34], [35] and Linear programming [36]. As a combinatorial construct, sparse recovery, and more specifically group testing, is related to the notion of selectors [37] and related pseudorandom objects.

Disjunct Matrices: The combinatorial guarantee for a test design to allow for the identification of the set of defectives is studied in the literature under several essentially equivalent notions, such as superimposed codes, cover-free families (or codes), and disjunct matrices (Definition 2 (see [38, Chapter 19], [19, Chapter 4] and [39] for a detailed discussion). Roughly speaking, a disjunct test matrix for k defectives satisfies the following: for every set S of k columns and a column $i \notin S$, there is a row at which the columns in S have zeros whereas the ith column has a 1. A lower bound of $\Omega(k^2 \log_k n)$ on the number of rows has been proved several times in the literature [40]–[42]. The best-known construction achieves $m = O(k^2 \min\{\log n, (\log_k n)^2\})$ number of rows (by a combination of the Kautz-Singleton construction [43] and Porat-Rothschild [44]). The notion of disjunctness can be naturally extended to also allow for accurate recovery in presences of false positives and negatives in the test outcomes.

Two central problems in group testing are explicit construction of test designs and efficient recovery of the defectives from test outcomes. While a simple probabilistic argument can achieve an upper bound of $O(k^2 \log n)$ tests (cf. [19, Chapter 4]), an explicit construction (in polynomial time in the matrix size) matching this upper bound [44] can be significantly more challenging. From the recovery perspective, any disjunct matrix allows recovery in nearly linear time in the size of the matrix using the following *naive decoder*: the decoder can simply output the subset of the columns of the test matrix whose supports are contained in the support of the test outcomes. For large population sizes, however, it is desirable to have a *sublinear time* recovery algorithm that runs in polynomial time (or even nearly linear time) in the number of tests, which can potentially be exponentially faster than the naive decoder above.

List-Disjunct Matrices: Another important combinatorial object, introduced independently in [45], [46], is that of a list-disjunct matrix. List-disjunct matrices guarantee the recovery of a small superset of the defective items but feature the advantage that the number of rows can be much smaller than what a disjunct matrix would allow (essentially by a factor of k), among several additional notable advantages and applications. Using list-disjunct matrices, one can design *two-stage* group testing schemes, by first narrowing the universe down to a small set, and then performing a test on each one independently. Thus, in scenarios where two-stage testing is possible, for example in DNA library screening or data forensics [47], this results in a major savings. Moreover, list-disjunct matrices can be used for constructions of monotone encodings and multi-user tracing families [48], vote storage systems [49], and for designing state of the art heavy hitter sketches (as we show in this work). Last but not least, they can be used as an intermediate tool towards the construction of (efficiently decodable) disjunct matrices; indeed this was the main motivation in [45], [46].

At least for noiseless testing, there is a simple *bit masking* trick that can augment any disjunct or list-disjunct matrix with additional rows to enable sublinear recovery (e.g., see [50]). The augmentation blows up the number of rows by a logarithmic factor in n, and thus a long line of work has been devoted to obtaining better trade-offs between rows and recovery time.

From now on, we shall refer to *decoding time*, as the time needed for recovery of the defectives or a small list containing them. We also stress the difference between the "for-all" guarantee (*uniform*) and the "for-each" guarantee (*non-uniform*). A matrix satisfies the first guarantee if it enables recovery for all vectors simultaneously, while a randomized matrix (i.e., a distribution over matrices) satisfies the for-each guarantee if it enables recovery of a fixed vector with some target probability. Disjunct and list-disjunct matrices are defined with the for-all guarantee in mind.

Work on sublinear-time group testing and related problems: Sublinear-time decoding on group testing (including disjunct, list-disjunct matrices, the probabilistic and the non-uniform case) has been explored in [46], [50]–[57]. In the context of the similar tasks of heavy hitters and compressed sensing (see below), sublinear-time has been investigated in [31], [32], [58]–[70], to name a few.

There is also a decent amount of literature on variants of the group testing problem, such as sparse group testing [71], graph-constrained group testing [72], [73], and threshold group testing [72]. Our focus in this paper is the most standard setup of the problem, although our techniques could potentially apply to the aforementioned settings as well.

Heavy Hitters and Compressed Sensing: A closely related problem is the task of finding heavy hitters in data streams. Given a long stream of updates (i, Δ) to a vector $x \in \mathbb{R}^n$ causing $x_i \leftarrow x_i + \Delta$, upon query detect the coordinates $i \in [n]$ which satisfy $|x_i| \geq (1/k)\|x\|_p$ (heavy hitters). The goal is to keep a small-space representation of x which allows finding the heavy hitters quickly, as well as rapid updates. The most interesting and well-studied cases correspond to $p = 1$ and $p = 2$. The heavy hitter problem is one of the core problems in streaming algorithms and has also served implicitly or explicitly as a subroutine in many streaming and compressed sensing algorithms; cf. [31], [58], [61], [68], [74]–[78] to name a few. It has also been an active area of research with many important results being discovered in the 2000s [79]–[81], as well as more recently [63], [82]–[88].

Another closely related area is *compressed sensing* [30],

[31], [59], [89], which focuses on understanding the design of a set of linear measurements $\Phi \in \mathbb{R}^{m \times n}$, such that given $y = \Phi x$ it is possible to recover an approximation to the best k-sparse approximation of x with respect to some ℓ_p norm. This problem is analogous to the heavy hitters problem, albeit with the difference that one desires to recover *most* of the heavy hitters in an ℓ_p sense, rather than all of them. Since the literature on the topic is vast, we refer the reader to a survey of Indyk and Gilbert [90], the introduction in [70], and the text [29]. Henceforth group testing, heavy hitters, and compressed sensing may be referred to using the umbrella term *sparse recovery*.

Our Contributions: We give several schemes for the sparse recovery problem, almost all of which feature near-optimal (nearly-*linear*) decoding time, improving upon several results in the literature, and setting the record straight for some of the most well-studied variants of the problem. We thus show that previous trade-offs in measurement complexity and decoding time can be greatly improved. In particular, we contribute the following.

- Combinatorial Group Testing
  1) A Monte-Carlo construction of list-disjunct matrices with the optimal $O(k \log(n/k))$ number of rows and $O(k \log^2(n/k))$ decoding time. The best previous sublinear-time scheme in terms of measurements suffered from quadratic decoding time in k and did not achieve the optimal number of rows. We thus essentially settle the measurement and the decoding time complexity of list-disjunct matrices.
  2) A Monte-Carlo construction of k-disjunct matrices with $m = O(k^2 \min\{\log n, (\log_k n)^2\})$ rows and $O(m + k \log^2(n/k))$ decoding time. Moreover, our construction can use an off-the-shelf construction of disjunct matrices (which may have an inefficient decoder) as a black box, so any improvement on the construction of disjunct matrices will immediately improve our result as well, resulting in a construction of disjunct matrices with the same number of rows and near-optimal decoding time.
  3) An explicit construction of k-disjunct matrices with $m = O(k^2 \log n)$ rows with decoding time nearly linear in m.
  4) State-of-the-art error-correcting disjunct and list-disjunct matrices, associated with decoding procedures which are faster by almost a factor k from previous schemes with the same number of rows.
  5) A (necessarily randomized) scheme with $O(k \log n)$ decoding time and measurements for the "for-each" version of the group testing problem, improving upon recent work which obtained the same number of measurements but with quadratic in k decoding time. This result essentially settles the "for-each" complexity of the group testing problem.
- Heavy Hitters
  6) A "for-all" streaming algorithm with $s = O(k \log(n/k))$ space usage for the heavy hitters problems in the strict turnstile model, allowing finding a list of size $O(k)$ that contains all $(1/k)$-heavy hitters. The query time is near-linear in s and the update time is $\tilde{O}(\log^2 k \cdot \log(n/k))$. In contrast, the previous algorithm with the same space required $\Omega(n \log n)$ query time and $\Omega(k \log(n/k))$ update time.
  7) A "for-all" streaming algorithm for the standard version of the heavy hitters problem in the strict turnstile model, matching the space usage s of previous constructions and allowing queries in time near-linear in s. Previous constructions suffered from $\Omega(nk)$ query time.
- Compressed Sensing
  8) A significantly stronger ℓ_2/ℓ_2 weak identification system than what was available before in the compressed sensing literature.

The most efficient previous sublinear-time schemes employed list-recoverable codes technology and the list-decoding view of pseudorandom objects such as expanders and extractors, or related ideas. On the other hand, most of our results stem from a unifying result (Theorem 11) which roughly is the following: "There exists a row-optimal $(k, 5k \log(n/k))$-list-disjunct matrix associated with a very efficient decoding procedure". Interestingly, in contrast to list-recoverable codes type of arguments which come with relatively large constants and many parameters to fine-tune, the aforementioned result and its implications require a minimal understanding of coding theory, being of potential practical impacts.

We bring the reader's attention to the concurrent work of Price and Scarlett [91], which arrives at the construction of our efficiently decodable list-disjunct matrix with a nearly identical algorithm, and uses it to obtain $O(k \log n)$ time for the "for-each" version of group testing, matching our contribution 5 listed above. Their analysis of the decoding algorithm is quite different, relying on bounds for sub-exponential random variables to control the branching process created by the execution of the algorithm. On the other hand, our argument is quite elementary and is based only on first principles. An advantage of their $O(k \log n)$-time algorithm is that they are able to guarantee correctness using limited independence for the hash functions [91, Section 3], thus obtaining a space-efficient variant of the $O(k \log n)$-time algorithm (i.e., our contribution 5 listed above).

II. Preliminaries

A. Notation

When referring to group testing, all matrices and vectors have entries in $\{0, 1\}$, with 0 corresponding to "false" and 1

to "true". Without loss of generality, we can assume that k, n are powers of two by rounding to the closest power of two from above, unless noted otherwise. We will associate $[n] := \{0, \ldots, n-1\}$ with $\{0,1\}^{\log n}$ via the obvious bijection. Moreover, all matrices and vectors are zero-indexed, that is for vector $x \in \{0,1\}^n$ the entries are $x_0, x_1, \ldots, x_{n-1}$. More generally, for a set $\mathcal{I} \subseteq [n]$, we denote by $x_{\mathcal{I}}$ the vector obtained by discarding the entries of x outside $\mathcal{I}$. We also denote the *support* of a vector by $\mathrm{supp}(x) = \{i \in [n] \colon x_i = 1\}$. For binary strings r, s, we write $r \| s$ to be the concatenation of r and s by writing r followed by s. For a test matrix $M \in \{0,1\}^{m \times n}$ (where the number of columns n is called the *population* or the *universe* size), and a vector $x \in \{0,1\}^n$, the vector $y = M \odot x$ corresponds to tests

$$y_q = \bigvee_{j \in [n] \colon M_{q,j}=1} x_j, \forall q \in [m],$$

where $M_{q,j}$ denotes the entry of M at row q and column j.

For $i \in [n]$ we denote by M^i the i-th column of M and M_i to be the i-th row of M. For $S \subseteq [n]$, we define $M^S = \bigcup_{i \in S} M^i$. Clearly $M^i = M^{\{i\}}$.

When referring to heavy hitters or compressed sensing we will work with the standard notion of addition and multiplication on numbers of $\Theta(\log n)$ bits. We will say that i is a $(1/k)$-heavy hitter for the vector $x \in \mathbb{R}^n$ if $|x_i| \geq (1/k)\|x\|_1$. We define $\|x\|_p = \left(\sum_{i=0}^{n-1} |x_i|^p\right)^{1/p}$. We denote by x_{-k} the *tail* vector that occurs after zeroing out the k largest in magnitude coordinates in x.

For non-negative integers α, ℓ such that $\alpha \leq 2^\ell - 1$, we denote by $\mathsf{bPref}_\ell(\alpha)$ to be the integer that is obtained from the first ℓ bits in the binary representation of α. For example, $\mathsf{bPref}_2((1100)_2) = (11)_2 = 3$, and $\mathsf{bPref}_3((11011)_2) = (110)_2 = 6$, where we have used the notation $(\cdot)_2$ for the binary representation of an integer).

B. Catalan Numbers

We shall use the following fact on Catalan numbers.

Lemma 1 (generalized Catalan numbers, [92])**.** *For natural integers $d, n \geq 2$, the number of rooted d-ary trees with exactly n nodes is*

$$\mathrm{Cat}_n^d = \frac{1}{n+1}\binom{dn}{n} \leq (ed)^n.$$

C. Disjunct and List-Disjunct Matrices

In this section, we review the standard notion of disjunctness and its variations that are instrumental for the design of group testing schemes (cf. [93, Chapter 4]).

Definition 2 (Disjunct Matrices)**.** *A matrix $M \in \{0,1\}^{m \times n}$ is called k-disjunct if for every set $S \subseteq [n]$ of size k, and every $j \in [n] \setminus S$ there exists a row $q \in [m]$ such that $M_{q,j} = 1$ and $M_{q,j'} = 0, \forall j' \in S$.*

A k-disjunct matrix essentially characterizes the combinatorial guarantee needed for noiseless group testing. The relaxes notion of *list-disjunct* matrices guarantees identification of a bounded-sized superset of the defective (thereby allowing a smaller number of rows by only requiring the recovery of a small list that is guaranteed to contain all defectives). The following definition is from [46] (while an essentially equivalent notion was formulated in [45]).

Definition 3 (List-Disjunct Matrices)**.** *A matrix $M \in \{0,1\}^{m \times n}$ is called a (k, ℓ)-list-disjunct matrix if for every two disjoint sets $S, T \subseteq [n]$ with $|S| = k, |T| = \ell + 1$ there exists an element $j \in T$ and a row $q \in [m]$ such that*

$$M_{q,j} = 1 \text{ and } \forall j' \in S, M_{q,j'} = 0.$$

The parameter ℓ captures the *list size* (so that it is always possible to output a list of size at most $k + \ell$ that contains the defective). For $\ell = 0$ the notion of list-disjunct matrices coincides with the classical notion of disjunct matrices (Definition 2). Given the measurement outcomes, one can naturally consider a list of possible defectives by selecting all columns of M the are covered by the vector of the measurement outcomes. More precisely, we can define the following.

Definition 4 (Associated List for List-Disjunct Matrices)**.** *Given $y = M \odot x$ with M being (k, ℓ)-list-disjunct matrix and $|\mathrm{supp}(x)| \leq k$, we will refer to L as the* associated list *of x with respect to M as the list of elements $i \in [n]$ satisfying the following:*

$$\forall q \in [m] \text{ such that } y_q = 1 \colon M_{q,i} = 1.$$

Put simply, L corresponds to the elements $i \in [n]$ which appear to be "defective" under measurements defined by M. Note that $|L| \leq k + \ell$.

The above notions can be strengthened to tolerate errors as follows:

Definition 5. *[54, Definition 1] A matrix $M \in \{0,1\}^{m \times n}$ is called (k, ℓ, e_0, e_1)-list disjunct if for every disjoint sets $S, T \subseteq [n]$ of size k and ℓ, respectively, the following holds. Let M^S and M^T respectively denote the unions of supports of the columns of M picked by S and T. Then, for every set $X \subseteq M^T \setminus M^S$ of size $|X| \leq e_0$, there is a column M^j picked by T such that $|\mathrm{supp}(M^j) \setminus (X \cup M^S)| > e_1$.*

In the above definition, e_0 (resp., e_1) captures the number of false positives (resp., negatives) that the matrix M can combinatorially tolerate in the measurement outcomes (and in the sequel, this is what we would mean by a matrix tolerating a certain number of false positives or negatives). We could have alternatively used the notion of error-correcting disjunct matrices in ([45, Definition 1]), but since our results behave differently in the case of false positives and false negatives, the definition in [45] is not the most suitable for

our needs. We refer the reader to Theorems 18 and 19 for the results on error-correcting k-disjunct matrices.

It is shown in [54, Proposition 2] that any (k, ℓ, e_0, e_1)-list disjunct matrix guarantees recovery of a set of size less than $k + \ell$ containing all defective items in presence of up to e_0 false positives and e_1 false negatives in the test outcomes. Lower bounds in [45], [54] show that $(k, \Theta(k), e_0, e_1)$-error-correcting list-disjunct matrices require $\Omega(k \log(n/k) + e_0 + ke_1)$ rows. Similarly, any k-disjunct matrix that can tolerate e_0 false positives and e_1 false negatives requires $\Omega(k^2 \log_k n + e_0 + ke_1)$ rows.

A natural decoder for disjunct and list-disjunct matrices is the following, often referred to as the *naive decoder*.

Definition 6 (Naive Decoder)**.** *Given $y = M \odot x$, for every $i \in [n]$ declare i defective if and only if $y_r = 1$ for every r such that $M_{r,i} = 1$. That is, output the set of columns that are covered by the measurement outcomes.*

The following are two well-known corollaries on the performance of the naive decoder in disjunct and list-disjunct matrices.

Lemma 7 (Naive Decoder and Point-Queries for Disjunct Matrices)**.** *Given $y = M \odot x$, where M is a k-disjunct matrix and $|\mathrm{supp}(x)| \leq k$ the following holds. Given $i \in [n]$, we can decide in time $O(|\mathrm{supp}(M^i)|)$ whether i is defective or not. Moreover, the naive decoder returns $\mathrm{supp}(x)$ in time $O(n \cdot \max_{i \in [n]} |\mathrm{supp}(M^i)|)$.*

Lemma 8 (Naive Decoder and Point-Queries for List-Disjunct Matrices)**.** *Given $y = M \odot x$, where M is a (k, ℓ) list-disjunct matrix, and $|\mathrm{supp}(x)| \leq k$ the following holds. Given $i \in [n]$ we can decide whether i belongs to the associated list L in time $O(|\mathrm{supp}(M^i)|)$ whether i is defective or not. Moreover, the naive decoder returns L in time $O(n \cdot \max_{i \in [n]} |\mathrm{supp}(M^i)|)$.*

We shall use the following well-known constructions of k-disjunct matrices, which follows from standard constructions of incoherent matrices (based on either Reed-Solomon codes by Kautz and Singleton [43] or codes on the Gilbert-Varshamov bound [44] by Porat and Rothschild)[1]

Theorem 9 (Disjunct Matrices)**.** *There exists a k-disjunct matrix with*

$$m = O(k^2 \min\left\{\log n, (\log_k n)^2\right\})$$

rows. In particular, there exist explicit k-disjunct matrices with (i) $O(k^2 \log n)$ rows and $O(k \log n)$ non-zeros per column (via [44]), and strongly explicit k-disjunct matrices with (ii) $O(k(\log_k n)^2)$ rows and $O(k \log_k n)$ non-zeros per column (via [43]).

[1]The construction of [43] is strongly explicit, in the sense that each entry of the matrix can be computed in $\mathrm{poly}(k, \log n)$ time, whereas [44] is explicit in the sense of being computable in $\mathrm{poly}(n)$ time.

We will use the following existential bound on list-disjunct matrices, that can be derived from [45] via a probabilistic argument:

Theorem 10. *[45] There exists a (k, k)-list-disjunct matrix with $O(k \log(n/k))$ rows and $O(\log(n/k))$ non-zeros per column.*

III. Results

In this section, we present our results on disjunct matrices, list-disjunct matrices, group testing, and heavy hitters, based on the definitions given in the preliminaries. In what follows, $C, C_{\mathsf{L}}, C_{\mathsf{FP}} > 1$ are absolute constants. All our results assume, without loss of generality, that $k \leq \gamma n$ for some absolute constant γ, as otherwise storing the identity matrix is asymptotically the best solution. Our starting point and one of our strongest tools is the following theorem.

Theorem 11. *There exists a Monte-Carlo construction of a $(k, C_{\mathsf{L}} k \log(n/k))$-list-disjunct matrix $M \in \{0,1\}^{m \times n}$ with $m = C \cdot k \log(n/k)$, allowing decoding in time $O(k \log(n/k))$. M is the vertical concatenation of $\log(n/k)$ matrices $M^{(\log k)}, \ldots, M^{(\log n)}$ such that (i) each such submatrix has Ck rows and exactly 1 non-zero per column, (ii) every such submatrix can tolerate up to $C_{\mathsf{FP}} \cdot k$ false positives.*

Furthermore, M can be stored in $O(k \log(n/k))$ space, and for every ℓ and every choice of $B = O(k \log(n/k))$ columns $i_1, i_2, \ldots, i_B \subseteq [n]$, we can find the rows $q_{i_1}, \ldots, q_{i_B} \subseteq [Ck]$ where the aforementioned columns have the non-zero element in $M^{(\ell)}$ in time

$$O\left(k \log^2\left(\frac{n}{k}\right) \cdot \log^2\left(k \log\left(\frac{n}{k}\right)\right) \cdot \log\log\left(k \log\left(\frac{n}{k}\right)\right)\right).$$

The last sentence of the above theorem, namely the claim about storing the matrix M in small space and the fast batch location, is particularly important for our application to the heavy hitters problem. For the group testing applications, this property will be irrelevant. The matrix M will be also called the *identification* matrix.

A. Disjunct and List-Disjunct Matrices

Theorem 12 (List-Disjunct Matrices)**.** *There exists a Monte-Carlo construction of a (k, k)-list-disjunct matrix $M \in \{0,1\}^{m \times n}$ with $m = O(k \log(n/k))$, that allows decoding in $O(k \log^2(n/k))$ time[2].*

In comparison, the best previous construction of efficiently decodable list-disjunct matrices requires either $k^2 \,\mathrm{poly}(\log n)$ decoding time and $O(k \log n \cdot \log\log_k n)$ rows [54], or $O(k \log^2(n/k))$ rows and decoding time (we note that [54] gives another construction using Parvaresh-Vardy codes with much less clean time and measurement bounds and polynomial in k decoding time).

[2]For ease of exposition, we chose to give a construction of list-disjunct matrices with $\ell = k$.

Theorem 13 (Disjunct Matrices)**.** *There exists a Monte-Carlo construction of a k-disjunct matrix $M \in \{0,1\}^{m\times n}$ with $m = O(k^2 \min\{\log n, (\log_k n)^2\})$, that allows decoding in $O(m + k\log^2(n/k))$ time.*

The best previous constructions of efficiently decodable disjunct matrices are: (i) The results from [46], which achieves $m = O(k^2 \log n)$ rows and $\Omega(k^4 \log n)$ decoding time, (ii) The result from [54] which achieves $O(k^2 \log n + k \log n \cdot \log\log_k n)$ rows and $m \log^2 n$ decoding time. We strictly improve upon the measurement and the decoding time complexity of previous work, obtaining the cleanest bounds.

Theorem 14 (Explicit Disjunct Matrices)**.** *We can construct in polynomial time in n a k-disjunct matrix with $m = O(k^2 \log n)$ rows that allows decoding in time $m \cdot \mathrm{poly}(\log n)$, unless*

$$k \in \left[\frac{C \log n}{\log\log n}, \left(\frac{C \log n}{\log\log n} \right)^{1+o(1)} \right],$$

where the $o(1)$ term is $\Theta\left(\frac{(\log\log\log n)^2}{\log\log n}\right)$.

Of course, the small intermediate range of k where the above result does not apply can be eliminated by slightly rounding k up to $k^{1+o(1)}$, resulting in $m = k^{2+o(1)} \log n$ rows and $m \cdot \mathrm{poly}(\log n)$ decoding time for all k.

B. Heavy Hitters in the Strict Turnstile Model

Theorem 15. *("For-all" Heavy Hitters) There exists a streaming algorithm with space usage $O(k \log(n/k))$, which keeps a (non-linear) representation of a vector $x \in \mathbb{R}^n$, and upon query, if $x \in \mathbb{R}^n_+$ then* always *returns a list L of size $O(k)$ which contains every $(1/k)$-heavy hitter. The query time is $O(k\,\mathrm{poly}(\log n))$ and the update time is $\tilde{O}(\log(n/k) \cdot \log^2 k)$.*

In contrast to the result appearing in [94] which achieved $\Omega(n \log n)$ query time and $O(k \log(n/k))$ update time, our algorithm achieves nearly optimal query and update time. The non-linearity of the sketch does not play a role in the number of measurements, but only to achieve the desired update time. It is shown in [94, Theorems 4,5] that if we drop the assumption of the strict turnstile model or additionally demand accurate estimates (up to $(1/k)\|x\|_1$) of the coordinates in L, then there exists no such linear sketch unless it has $\Omega(k^2)$ rows.

The next result is a streaming algorithm for the more common version of the heavy hitters problem, where one wants to find every $(1/k)$-heavy hitter and no i with $x_i \le (1/(2k))\|x\|_1$. This greatly improves upon the scheme appearing in [95] which has the same space usage but requires $\Omega(nk)$ query time[3].

Theorem 16. *("For-all" Heavy Hitters with Estimates) There exists a streaming algorithm using space usage*

$$O\left(k^2 \cdot \min\left\{ \log n, \left(\frac{\log n}{\log k + \log\log n} \right)^2 \right\} \right),$$

which keeps a (non-linear) representation of a vector $x \in \mathbb{R}^n$, and upon query, if $x \in \mathbb{R}^n_+$ then always *returns a list L containing every $(1/k)$-heavy hitter, and no $i \in [n]$ with $x_i \le (1/(2k))\|x\|_1$. Moreover, for every $i \in L$ it returns an estimate x'_i with $x_i \le x'_i \le x_i + (c/k)\|x\|_1$, where c is an arbitrarily small absolute constant $c < 1$. The query time is $k^2\,\mathrm{poly}(\log n)$ and the update time is $O(k \cdot \min\left\{ \log n, \left(\frac{\log n}{\log k + \log\log n} \right)^2 \right\}) + \tilde{O}(\log^3 n)$.*

Remark 17. *One could also ask whether the update time of k on the above theorem is necessary, or more interestingly, one can decode k-disjunct matrices and perform queries for heavy hitters faster than quadratic time in k. After all, as one can observe in the full version, we need to point query only $O(k)$ coordinates, so it is not immediately evident that the quadratic time-bound in k is necessary (we might need $\Omega(k^2)$ measurements, but an algorithm might not need to read all of them). However, it seems that performing point-queries is indeed a bottleneck, since (i) an easy argument (which we leave to the reader) shows that any k-disjunct matrix must have at least $n - m$ columns of sparsity at least k, and (ii) any $(1/k)$-incoherent matrix (from which known heavy hitters sketches follow) must have column sparsity $\Omega(k)$ as long as $m \le n/\log k$ [96, Theorem 10]. This constitutes strong evidence that it is impossible to beat quadratic decoding/ query time and linear (in k) update time, unless using a near-linear in n number of measurements.*

It is also worth noting that any subsequent improvement of sketches that enable ℓ_1 point-queries immediately translates, via our framework, to a streaming algorithm with sublinear query time. Thus, we may consider the problem of sublinear-time query time essentially closed, up to logarithmic factors.

C. Error-Correcting Disjunct Matrices

We give the following two constructions of efficiently decodable matrices. The first is particularly efficient for false negatives, while the second for false positives. Both results are significantly faster than what was attainable by previous techniques using the same number of rows. We find it intriguing that while we are able to construct fast error-correcting disjunct matrices with respect to either false positives or false negatives, we cannot construct fast

[3]The results in [95] satisfy a stronger guarantee, referred to as the "tail" guarantee in the sparse recovery literature. It is not hard to see that our arguments can facilitate that guarantee as well, but for ease of exposition we chose to present only the more standard guarantee of the heavy hitters problem.

error-correcting disjunct matrices that can facilitate both simultaneously.

Theorem 18. *(False Positives) There exist Monte-Carlo constructions of*

1) *A $(k, k, e_0, 0)$-list-disjunct matrix with*

$$m = O(k \log(n/k) + \log_k n \cdot e_0)$$

rows which allows decoding in time $O(k^\alpha \cdot m)$, for any constant $\alpha > 0$.

2) *A k-disjunct matrix with $m = O(k^2 \log n + \log_k n \cdot e_0)$, which can tolerate up to e_0 false positives and allows decoding in time $O(m \cdot \log n))$.*

The first result of the preceding theorem improves by almost a k factor what is attainable by the techniques in [54], [57] (see also the comment following); the techniques in [50], [55], [97] result in schemes with a strictly larger number of rows. For both results in Theorem 18, the argument in [54] obtains near-linear decoding albeit with a slight loss of $\log \log_k n$; [50], [55], [97] can be modified to obtain near-linear decoding but with a $\log n$ factor overhead in the measurement complexity.

Theorem 19. *(False Negatives) There exists a Monte-Carlo construction of a k-error-correcting disjunct matrix achieving $m = O(k^2 \log n + k \cdot e_1)$, which can tolerate up to ke_1 false negatives and allows decoding in time $O(m \cdot \text{poly}(\log n))$.*

This theorem improves upon what was known and achievable using previous techniques, both in terms of measurements and decoding time, and achieves the optimal dependence in terms of e_1, the number of false negatives.

D. Resolving the "For-Each" Case of Group Testing

Theorem 20. *There exists a randomized construction of a matrix $\{0, 1\} \in \mathbb{R}^{m \times n}$ with $m = O(k \log n)$ such that the following holds. Given $y = M \odot x$ with $|\text{supp}(x)| \leq k$, we can find x in time $O(k \log n)$, with failure probability $e^{-\Omega(k)} + \frac{1}{\text{poly}(n)}$.*

This theorem improves upon the recent work of [57], which achieved the same number of rows but required quadratic running time in k. Our result essentially settles the non-uniform case of the group testing problem.

E. ℓ_2/ℓ_2 Compressed Sensing

One of the central problems in compressed sensing is the design of an ℓ_2/ℓ_2 scheme, which is a matrix $\Phi \in \mathbb{R}^{m \times n}$, such that given $y = \Phi x$ we can find x' satisfying

$$\mathbb{P}\left\{\|x - x'\|_2^2 \leq (1 + \epsilon)\text{min}_{k-\text{sparse } z}\|x - z\|_2^2\right\} \geq 1 - \delta.$$

The goal is to randomly design Φ satisfying the above with the optimal number of rows, and enabling computation of such an x' in sublinear-time (it can be proved that it suffices to pick x' to be $O(k)$-sparse). Almost all sublinear-time algorithms (precisely, all but [70]) proceed by reducing the problem to the construction of an ℓ_2/ℓ_2 weak identification system[4]. This is a matrix $\Psi \in \mathbb{R}^{m \times n}$ such that given $y = \Psi x$ we can find x' satisfying $\|(x - x')_{-k/2}\|_2 \leq (1 + \epsilon)\|x_{-k}\|_2$ with probability $1 - \delta$; recalling that x_{-k} is the vector that occurs after zeroing out the k largest in magnitude coordinates in x. For yet another intriguing consequence of our techniques, we give the best weak identification ℓ_2/ℓ_2 system available in the literature. On how that translates to ℓ_2/ℓ_2 schemes, we refer the reader to the full version [1].

Theorem 21. *There exists a randomized construction of an ℓ_2/ℓ_2 weak identification system with*

$$m = O\left((k/\epsilon)\log(n/k) + \frac{1}{\epsilon} \cdot \frac{\log(n/k)}{\log\log(n/k)} \cdot \log(1/\delta)\right),$$

which allows finding the desired x' in time $O(m \log^2 m)$.

A comparison with previous work follows.

- The construction in [59] requires

$$m = \Theta((k/\epsilon)\log(n/k) \cdot \log(1/\delta)).$$

- The construction in [94] achieves

$$m = O((k/\epsilon)\log(n/k) + \epsilon^{-1}\log(n/k)\log(1/\delta)),$$

but in order to run in near-linear time in m storing an additional inversion table of size $\Omega(n)$ is required.

- The strongest result in [60], [62] obtains

$$m = O(\epsilon^{-4}k\log(n/k)(\log_k n)^\alpha + \epsilon^{-1}\,\text{poly}(\log n)\log(1/\delta))$$

and decoding time $\Omega((k/\epsilon)^{2^{1/\alpha}}\,\text{poly}(\log n))$, for any $a < 1$. The main source of sub-optimality is the invocation of a list-recoverable code based on the Loomis-Whitney inequality [98].

To the best of our knowledge, our work is the first to construct a near-optimal weak system with near-optimal decoding time (without using an additional inversion table as in [94]). In fact, we are able to obtain stronger results for the general ℓ_2/ℓ_2 problem. However, the argument turns out to be lengthy and somewhat outside the scope of the technical contribution of this paper. We have therefore decided to leave the most general result for a future publication.

REFERENCES

[1] M. Cheraghchi and V. Nakos, "Combinatorial group testing and sparse recovery schemes with near-optimal decoding time," *arXiv preprint arXiv:2006.08420*, 2020.

[2] R. Dorfman, "The detection of defective members of large populations," *Annals of Mathematical Statistics*, vol. 14, pp. 436–440, 1943.

[4]The authors in [60], [68] define it in a slightly different way, and use slightly different terminology at places, but the essence of the property they demand is the same.

[3] W. Bruno, E. Knill, D. Balding, D. Bruce, N. Doggett, W. Sawhill, R. Stallings, C. Whittaker, and D. Torney, "Efficient pooling designs for library screening," *Genomics*, vol. 26, no. 1, pp. 21–30, 1995.

[4] Y. Cheng and D.-Z. Du, "New constructions of one-and two-stage pooling designs," *Journal of Computational Biology*, vol. 15, no. 2, pp. 195–205, 2008.

[5] E. Knill and S. Muthukrishnan, "Group testing problems in experimental molecular biology," *arXiv preprint arXiv:math/9505211*, 1995.

[6] M. Farach, S. Kannan, E. Knill, and S. Muthukrishnan, "Group testing problems with sequences in experimental molecular biology," in *Proceedings of Compression and Complexity of Sequences*, 1997, pp. 357–367.

[7] A. Macula, "Probabilistic nonadaptive group testing in the presence of errors and DNA library screening," *Annals of Combinatorics*, vol. 3, no. 1, pp. 61–69, 1999.

[8] H.-Q. Ngo and D.-Z. Du, "A survey on combinatorial group testing algorithms with applications to DNA library screening," *DIMACS Series on Discrete Math. and Theoretical Computer Science*, vol. 55, pp. 171–182, 2000.

[9] A. Schliep, D. Torney, and S. Rahmann, "Group testing with DNA chips: Generating designs and decoding experiments," in *Proceedings of Computational Systems Bioinformatics*, 2003.

[10] W. Wu, Y. Huang, X. Huang, and Y. Li, "On error-tolerant DNA screening," *Discrete Applied Mathematics*, vol. 154, no. 12, pp. 1753–1758, 2006.

[11] W. Wu, Y. Li, C. Huang, and D. Du, "Molecular biology and pooling design," *Data Mining in Biomedicine*, vol. 7, pp. 133–139, 2008.

[12] C. J. Colbourn and J. H. Dinitz, *Handbook of Combinatorial Designs, Second Edition (Discrete Mathematics and Its Applications)*. Chapman & Hall/CRC, 2006.

[13] J. Wolf, "Born-again group testing: multiaccess communications," *IEEE Transactions on Information Theory*, vol. 31, pp. 185–191, 1985.

[14] E.-S. Hong and R. Ladner, "Group testing for image compression," in *Data Compression Conference*, 2000, pp. 3–12.

[15] R. Clifford, K. Efremenko, E. Porat, and A. Rothschild, "k-mismatch with don't cares," in *Proceedings of the* 15*th European Symposium on Algorithm (ESA)*, ser. Lecture Notes in Computer Science, vol. 4698, 2007, pp. 151–162.

[16] H.-B. Chen, D.-Z. Du, and F.-K. Hwang, "An unexpected meeting of four seemingly unrelated problems: graph testing, DNA complex screening, superimposed codes and secure key distribution," *Journal of Combinatorial Optimization*, vol. 14, no. 2-3, pp. 121–129, 2007.

[17] M. Cheraghchi, A. Karbasi, S. Mohajer, and V. Saligrama, "Graph-constrained group testing," *IEEE Transactions on Information Theory*, vol. 58, no. 1, pp. 248–262, 2012.

[18] M. Sobel and P. Groll, "Group-testing to eliminate efficiently all defectives in a binomial sample," *Bell Systems Technical Journal*, vol. 38, pp. 1179–1252, 1959.

[19] D.-Z. Du and F.-K. Hwang, *Combinatorial Group Testing and its Applications*, 2nd ed. World Scientific, 2000.

[20] ——, *Pooling Designs and Nonadaptive Group Testing*. World Scientific, 2006.

[21] J. Zhu, K. Rivera, and D. Baron, "Noisy pooled PCR for virus testing," *arXiv preprint arXiv:2004.02689*, 2020.

[22] A. Z. Broder and R. Kumar, "A note on double pooling tests," *arXiv preprint arXiv:2004.01684*, 2020.

[23] I. Yelin, N. Aharony, E. Shaer-Tamar, A. Argoetti, E. Messer, D. Berenbaum, E. Shafran, A. Kuzli, N. Gandali, T. Hashimshony, Y. Mandel-Gutfreund, M. Halberthal, Y. Geffen, M. Szwarcwort-Cohen, and R. Kishony, "Evaluation of COVID-19 RT-qPCR test in multi-sample pools," *medRxiv*, 2020.

[24] Technion, "Pooling method for accelerated testing of COVID-19," https://www.technion.ac.il/en/2020/03/pooling-method-for-accelerated-testing-of-covid-19, 2020, accessed: 2020-04-15.

[25] EurekAlert, "Pool testing of SARS-CoV-02 samples increases worldwide test capacities many times over," https://www.eurekalert.org/pub_releases/2020-03/guf-pto033020.php, 2020, accessed: 2020-04-15.

[26] Omaha News, "Gov. Ricketts provides update on coronavirus testing," https://www.3newsnow.com/news/coronavirus/live-gov-ricketts-provides-coronavirus-briefing-3-24-20, 2020, accessed: 2020-04-15.

[27] K. Bennhold, "A German exception? why the country's coronavirus death rate is low," https://www.nytimes.com/2020/04/04/world/europe/germany-coronavirus-death-rate.html, 2020, accessed: 2020-04-15.

[28] K. R. Narayanan, A. Heidarzadeh, and R. Laxminarayan, "On accelerated testing for COVID-19 using group testing," *arXiv preprint arXiv:2004.04785*, 2020.

[29] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*. Birkhäuser Basel, 2013.

[30] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[31] H. Hassanieh, P. Indyk, D. Katabi, and E. Price, "Nearly optimal sparse Fourier transform," in *Proceedings of the forty-fourth annual ACM symposium on Theory of Computing*. ACM, 2012, pp. 563–578.

[32] M. Kapralov, "Sample efficient estimation and recovery in sparse FFT via isolation on average," in *Proceedings of the Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2017.

[33] V. Nakos, Z. Song, and Z. Wang, "(nearly) sample-optimal sparse Fourier Transform in any dimension; RIPless and filterless," in *60th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, 2019, pp. 1568–1577.

[34] K. Axiotis, A. Backurs, C. Jin, C. Tzamos, and H. Wu, "Fast modular Subset Sum using linear sketching," in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*. SIAM, 2019, pp. 58–69.

[35] K. Bringmann and V. Nakos, "Top-k-convolution and the quest for near-linear output-sensitive subset sum," in *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*. ACM, 2020, pp. 982–995.

[36] J. van den Brand, Y. T. Lee, A. Sidford, and Z. Song, "Solving tall dense linear programs in nearly linear time," *STOC 2020, To appear*, 2020.

[37] B. S. Chlebus, L. Gçasieniec, A. Östlin, and J. M. Robson, "Deterministic radio broadcasting," in *Automata, Languages and Programming*. Springer Berlin Heidelberg, 2000, pp. 717–729.

[38] V. Guruswami, A. Rudra, and M. Sudan, "Essential coding theory," https://cse.buffalo.edu/faculty/atri/courses/coding-theory/book/, 2019, accessed: 2020-04-15.

[39] A. G. D'yachkov and V. V. Rykov, "A survey of superimposed code theory," *Problems of Control and Information Theory*, vol. 12, no. 4, pp. 1–13, 1983.

[40] ——, "Bounds on the length of disjunctive codes," *Problemy Peredachi Informatsii*, vol. 18, no. 3, pp. 7–13, 1982.

[41] A. D'yachkov, V. Rykov, and A. Rashad, "Superimposed distance codes," *Problems of Control and Information Theory-problemy Upravleniya i Teorii Informatsii*, vol. 18, no. 4, pp. 237–250, 1989.

[42] M. Ruszinkó, "On the upper bound of the size of the r-cover-free families," *Journal of Combinatorial Theory, Series A*, vol. 66, no. 2, pp. 302–310, 1994.

[43] W. Kautz and R. Singleton, "Nonrandom binary superimposed codes," *IEEE Transactions on Information Theory*, vol. 10, pp. 363–377, 1964.

[44] E. Porat and A. Rothschild, "Explicit non-adaptive combinatorial group testing schemes," in *International Colloquium on Automata, Languages, and Programming*. Springer, 2008, pp. 748–759.

[45] M. Cheraghchi, "Noise-resilient group testing: Limitations and constructions," *Discrete Applied Mathematics*, vol. 161, no. 1–2, pp. 81–95, 2012, preliminary version in Proceedings of FCT, LNCS:5699, pp. 62–73, 2009, arXiv version (arXiv:0811.2609) in 2008.

[46] P. Indyk, H. Q. Ngo, and A. Rudra, "Efficiently decodable non-adaptive group testing," in *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*. SIAM, 2010, pp. 1126–1142.

[47] M. T. Goodrich, M. J. Atallah, and R. Tamassia, "Indexing information for data forensics," in *International Conference on Applied Cryptography and Network Security*. Springer, 2005, pp. 206–221.

[48] N. Alon and R. Hod, "Optimal monotone encodings," *IEEE Transactions on Information Theory*, vol. 55, no. 3, pp. 1343–1353, 2009.

[49] T. Moran, M. Naor, and G. Segev, "Deterministic history-independent strategies for storing information on write-once memories," in *International Colloquium on Automata, Languages, and Programming*. Springer, 2007, pp. 303–315.

[50] K. Lee, K. Chandrasekher, R. Pedarsani, and K. Ramchandran, "Saffron: A fast, efficient, and robust framework for group testing based on sparse-graph codes," *IEEE Transactions on Signal Processing*, 2019.

[51] V. Guruswami and P. Indyk, "Linear-time list decoding in error-free settings: (extended abstract)," in *Proceedings of ICALP*, ser. Lecture Notes in Computer Science, vol. 3142. Springer, 2004, pp. 695–707.

[52] G. Cormode and S. Muthukrishnan, "What's hot and what's not: tracking most frequent items dynamically," *ACM Transactions on Database Systems*, vol. 30, no. 1, pp. 249–278, 2005.

[53] H.-B. Chen and F.-K. Hwang, "A survey on nonadaptive group testing algorithms through the angle of decoding," *Journal of Combinatorial Optimization*, vol. 15, no. 1, pp. 49–59, 2008.

[54] H. Q. Ngo, E. Porat, and A. Rudra, "Efficiently decodable error-correcting list disjunct matrices and applications," in *International Colloquium on Automata, Languages, and Programming*. Springer, 2011, pp. 557–568.

[55] S. Cai, M. Jahangoshahi, M. Bakshi, and S. Jaggi, "Efficient algorithms for noisy group testing," *IEEE Transactions on Information Theory*, vol. 63, no. 4, pp. 2113–2136, 2017.

[56] A. Vem, N. T. Janakiraman, and K. R. Narayanan, "Group testing using left-and-right-regular sparse-graph codes," *arXiv preprint arXiv:1701.07477*, 2017.

[57] S. Bondorf, B. Chen, J. Scarlett, H. Yu, and Y. Zhao, "Sublinear-time non-adaptive group testing with $O(k \log n)$ tests via bit-mixing coding," *arXiv preprint arXiv:1904.10102*, 2019.

[58] A. C. Gilbert, S. Muthukrishnan, and M. Strauss, "Improved time bounds for near-optimal sparse Fourier representations," in *Optics & Photonics 2005*. International Society for Optics and Photonics, 2005, pp. 398–412.

[59] A. C. Gilbert, Y. Li, E. Porat, and M. J. Strauss, "Approximate sparse recovery: optimizing time and measurements," *SIAM Journal on Computing 2012*, vol. 41, no. 2, pp. 436–453, 2010.

[60] 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*. Society for Industrial and Applied Mathematics, 2012, pp. 1215–1227.

[61] H. Hassanieh, P. Indyk, D. Katabi, and E. Price, "Simple and practical algorithm for sparse Fourier transform," in *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*. SIAM, 2012, pp. 1183–1194.

[62] A. C. Gilbert, H. Q. Ngo, E. Porat, A. Rudra, and M. J. Strauss, "ℓ_2/ℓ_2-foreach sparse recovery with low risk," in *International Colloquium on Automata, Languages, and Programming*. Springer, 2013, pp. 461–472.

[63] K. G. Larsen, J. Nelson, H. L. Nguyễn, and M. Thorup, "Heavy hitters via cluster-preserving clustering," in *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*. IEEE, 2016, pp. 61–70.

[64] M. Kapralov, "Sparse Fourier transform in any constant dimension with nearly-optimal sample complexity in sublinear time," in *Symposium on Theory of Computing Conference, STOC'16, Cambridge, MA, USA, June 19-21, 2016*, 2016.

[65] V. 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*, ser. LIPIcs, vol. 80, 2017, pp. 61:1–61:14.

[66] ——, "Almost optimal phaseless compressed sensing with sublinear decoding time," in *2017 IEEE International Symposium on Information Theory, ISIT 2017, Aachen, Germany, June 25-30, 2017*. IEEE, 2017, pp. 1142–1146.

[67] V. Cevher, M. Kapralov, J. Scarlett, and A. Zandieh, "An adaptive sublinear-time block sparse Fourier transform," in *Proceedings of the 49th Annual Symposium on the Theory of Computing (STOC)*. ACM, 2017.

[68] A. C. Gilbert, Y. Li, E. Porat, and M. J. Strauss, "For-all sparse recovery in near-optimal time," *ACM Transactions on Algorithms (TALG)*, vol. 13, no. 3, p. 32, 2017.

[69] Y. Li and V. Nakos, "Sublinear-time algorithms for compressive phase retrieval," in *2018 IEEE International Symposium on Information Theory, ISIT 2018, Vail, CO, USA, June 17-22, 2018*. IEEE, 2018, pp. 2301–2305.

[70] V. Nakos and Z. Song, "Stronger ℓ_2/ℓ_2 compressed sensing; without iterating," in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*. ACM, 2019, pp. 289–297.

[71] V. Gandikota, E. Grigorescu, S. Jaggi, and S. Zhou, "Nearly optimal sparse group testing," *IEEE Trans. Inf. Theory*, vol. 65, no. 5, pp. 2760–2773, 2019.

[72] M. Cheraghchi, "Improved constructions for non-adaptive threshold group testing," *Algorithmica*, vol. 67, no. 3, pp. 384–417, 2013.

[73] B. Spang and M. Wootters, "Unconstraining graph-constrained group testing," in *Proceedings of APPROX/RANDOM*, ser. LIPIcs, vol. 145, 2019, pp. 46:1–46:20.

[74] N. J. A. Harvey, J. Nelson, and K. Onak, "Sketching and streaming entropy via approximation theory," in *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science, (FOCS)*, 2008, pp. 489–498.

[75] D. M. Kane, J. Nelson, E. Porat, and D. P. Woodruff, "Fast moment estimation in data streams in optimal space," in *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*. ACM, 2011, pp. 745–754.

[76] M. A. Iwen, "Improved approximation guarantees for sublinear-time Fourier algorithms," *Applied And Computational Harmonic Analysis*, vol. 34, no. 1, pp. 57–82, 2013.

[77] P. Indyk and M. Kapralov, "Sample-optimal Fourier sampling in any constant dimension," in *Proceedings of the Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2014, pp. 514–523.

[78] R. Jayaram and D. P. Woodruff, "Perfect lp sampling in a data stream," in *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, 2018, pp. 544–555.

[79] M. Charikar, K. Chen, and M. Farach-Colton, "Finding frequent items in data streams," in *Automata, Languages and Programming*. Springer, 2002, pp. 693–703.

[80] G. Cormode and S. Muthukrishnan, "An improved data stream summary: the count-min sketch and its applications," in *Proceedings of the Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, 2004.

[81] G. Cormode and M. Hadjieleftheriou, "Finding the frequent items in streams of data," *Communications of the ACM*, vol. 52, no. 10, pp. 97–105, 2009.

[82] V. Braverman, S. R. Chestnut, N. Ivkin, and D. P. Woodruff, "Beating CountSketch for heavy hitters in insertion streams," in *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC*, 2016, pp. 740–753.

[83] D. P. Woodruff, "New algorithms for heavy hitters in data streams," *arXiv preprint arXiv:1603.01733*, 2016.

[84] V. Braverman, S. R. Chestnut, N. Ivkin, J. Nelson, Z. Wang, and D. P. Woodruff, "BPTree: an ℓ_2 heavy hitters algorithm using constant memory," in *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS)*, 2017, pp. 361–376.

[85] Y. Li and V. Nakos, "Deterministic heavy hitters with sublinear query time," in *Proceedings of APPROX/RANDOM*, 2018, pp. 18:1–18:18.

[86] A. Aamand, P. Indyk, and A. Vakilian, "(learned) frequency estimation algorithms under Zipfian distribution," *arXiv preprint arXiv:1908.05198*, 2019.

[87] V. Braverman, E. Grigorescu, H. Lang, D. P. Woodruff, and S. Zhou, "Nearly optimal distinct elements and heavy hitters on sliding windows," in *Proceedings of APPROX/RANDOM 2018*, ser. LIPIcs, vol. 116, 2018, pp. 7:1–7:22.

[88] A. Bhattacharyya, P. Dey, and D. P. Woodruff, "An optimal algorithm for ℓ_1-heavy hitters in insertion streams and related problems," *ACM Trans. Algorithms*, vol. 15, no. 1, pp. 2:1–2:27, 2019.

[89] D. L. Donoho, "Compressed sensing," *IEEE Trans. Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[90] A. Gilbert and P. Indyk, "Sparse recovery using sparse matrices," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 937–947, 2010.

[91] E. Price and J. Scarlett, "A fast binary splitting approach to non-adaptive group testing," in *Proceedings of APPROX/RANDOM*, ser. LIPIcs, vol. 176. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, pp. 13:1–13:20.

[92] N. J. Sloane, "The on-line encyclopedia of integer sequences," in *International Conference on Mathematical Knowledge Management*. Springer, 2007, pp. 130–130.

[93] D.-Z. Du and F.-K. Hwang, *Combinatorial group testing and its applications*. World Scientific, 2000, vol. 12.

[94] Y. Li, V. Nakos, and D. P. Woodruff, "On low-risk heavy hitters and sparse recovery schemes," in *Proceedings of APPROX/RANDOM*, 2018, pp. 19:1–19:13.

[95] J. Nelson, H. L. Nguyễn, and D. P. Woodruff, "On deterministic sketching and streaming for sparse recovery and norm estimation," in *Proceedings of APPROX/RANDOM*, ser. Lecture Notes in Computer Science, vol. 7408. Springer, 2012, pp. 627–638.

[96] J. Nelson and H. L. Nguyễn, "Sparsity lower bounds for dimensionality reducing maps," in *Proceedings of the forty-fifth annual ACM symposium on Theory of Computing*, 2013, pp. 101–110.

[97] S. Cai, M. Jahangoshahi, M. Bakshi, and S. Jaggi, "GROTESQUE: noisy group testing (quick and efficient)," in *Proceedings of the 51st Annual Allerton Conference on Communication, Control, and Computing*. IEEE, 2013, pp. 1234–1241.

[98] H. Q. Ngo, E. Porat, C. Ré, and A. Rudra, "Worst-case optimal join algorithms," *J. ACM*, vol. 65, no. 3, pp. 16:1–16:40, 2018.