On Low-Risk Heavy Hitters and Sparse Recovery Schemes

Yi Li
Nanyang Technological University, Singapore
yili@ntu.edu.sg

Vasileios Nakos[1]
Harvard University, USA
vasileiosnakos@g.harvard.edu

David P. Woodruff[2]
Carnegie Mellon University, USA
dwoodruf@cs.cmu.edu

Abstract

We study the heavy hitters and related sparse recovery problems in the *low failure probability regime*. This regime is not well-understood, and the main previous work on this is by Gilbert et al. (ICALP'13). We recognize an error in their analysis, improve their results, and contribute new sparse recovery algorithms, as well as provide upper and lower bounds for the heavy hitters problem with low failure probability. Our results are summarized as follows:

1. (Heavy Hitters) We study three natural variants for finding heavy hitters in the strict turnstile model, where the variant depends on the quality of the desired output. For the weakest variant, we give a randomized algorithm improving the failure probability analysis of the ubiquitous Count-Min data structure. We also give a new lower bound for deterministic schemes, resolving a question about this variant posed in Question 4 in the IITK Workshop on Algorithms for Data Streams (2006). Under the strongest and well-studied ℓ_∞/ℓ_2 variant, we show that the classical Count-Sketch data structure is optimal for very low failure probabilities, which was previously unknown.
2. (Sparse Recovery Algorithms) For non-adaptive sparse-recovery, we give sublinear-time algorithms with low-failure probability, which improve upon Gilbert et al. (ICALP'13). In the adaptive case, we improve the failure probability from a constant by Indyk et al. (FOCS '11) to $e^{-k^{0.99}}$, where k is the sparsity parameter.
3. (Optimal Average-Case Sparse Recovery Bounds) We give matching upper and lower bounds in all parameters, including the failure probability, for the measurement complexity of the ℓ_2/ℓ_2 sparse recovery problem in the spiked-covariance model, completely settling its complexity in this model.

2012 ACM Subject Classification Theory of computation → Streaming, sublinear and near linear time algorithms

Keywords and phrases heavy hitters, sparse recovery, turnstile model, spike covariance model, lower bounds

Digital Object Identifier 10.4230/LIPIcs.APPROX-RANDOM.2018.19

Related Version Full version available at `https://arxiv.org/abs/1709.02919`.

[1] Supported in part by NSF grant IIS-1447471
[2] Supported in part by NSF grant CCF-1815840

© Yi Li, Vasileios Nakos, and David P. Woodruff;
licensed under Creative Commons License CC-BY

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2018).
Editors: Eric Blais, Klaus Jansen, José D. P. Rolim, and David Steurer; Article No. 19; pp. 19:1–19:13

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Finding heavy hitters in data streams is one of the most practically and theoretically important problems in the streaming literature. Subroutines for heavy hitter problems, and in particular for the Count-Min sketch, are the basis for multiple problems on geometric data streams, including k-means and k-median clustering [14, 6, 2], as well as image acquisition [12]. Studying schemes for finding such heavy hitters has also led to important geometric insights in ℓ_p-spaces [1].

Abstractly, in the heavy hitters problem, we are asked to report all frequent items in a very long stream of elements coming from some universe. The main restriction is that the memory consumption should be much smaller than the universe size and the length of the stream. To rephrase the problem, consider a frequency vector $x \in \mathbb{R}^n$, where n is the size of the universe. Each element i in the data stream updates the frequency vector as $x_i \leftarrow x_i + 1$. At the end of the data stream, we wish to find the coordinates of x for which $|x_i| \geq \epsilon \|x\|_1$.

Oftentimes the problem is considered under a more general streaming model called the strict turnstile model, where arbitrary deletions and additions are allowed, but at all times the entries of x remain non-negative, that is $x_i \geq 0$. More formally, the frequency vector $x \in \mathbb{R}^n$ receives updates of the form (i, Δ), and each such update causes the change of x_i to $x_i + \Delta$, while ensuring that $x_i \geq 0$. The goal is to maintain a data structure such that upon query, the data structure returns the heavy hitters of the underlying vector x. The ℓ_p heavy hitters problem, for $p \geq 1$, then asks to find all coordinates i for which $x_i^p \geq \epsilon \|x\|_p^p$. The algorithm that treats the ℓ_1 case is the Count-min sketch [5], and the algorithm that treats the ℓ_2 case is the Count-Sketch [3]. Both algorithms are randomized and succeed with probability $1 - 1/\operatorname{poly}(n)$. In [5] the authors also suggest the "dyadic" trick for exchanging query time with space. Their "dyadic" trick allows for finding heavy hitters approximately in $\mathcal{O}(\frac{1}{\epsilon} \log^2 n)$ time, but with the downside of having a worse update time and a worse space consumption by an $\mathcal{O}(\log n)$ factor. The state of the art for heavy hitters is [17], where the authors give an algorithm that satisfies the ℓ_∞/l_p guarantee, has space $\mathcal{O}(\frac{1}{\epsilon} \log n)$, update time $\mathcal{O}(\log n)$, and query time $\mathcal{O}(\frac{1}{\epsilon} \operatorname{poly}(\log n))$. We note that the latter algorithm works in the more general setting of the turnstile model, where there is no constraint on x_i, in contrast to the strict turnstile model.

Another set of closely related problems occurs in the compressed sensing (CS) literature, which has seen broad applications to biomedical imaging, sensor networks, hand-held digital cameras, and monitoring systems, among other places. Sparse compressed sensing schemes were used for determining the attitudes[3], or 3-axis orientation, of spacecraft in [12].

Abstractly, in this problem we also seek to find the large coordinates of $x \in \mathbb{R}^n$ but with a different goal. Instead of finding all coordinates x_i for which $|x_i| \geq \epsilon \|x\|_1$, the CS problem seeks an approximation $\hat{x}$ to x such that the difference vector $x - \hat{x}$ has small norm. In particular, we consider the ℓ_2/ℓ_2 error metric, that is, we require that $\|x - \hat{x}\|_2^2 \leq (1+\epsilon)\|x_{-k}\|_2^2$, where x_{-k} is the vector x with the k largest entries (in magnitude) removed. If all ℓ_2 heavy hitters are found, it is clear that the norm of $x - \hat{x}$ can be made small, but the CS problem allows a small number of heavy hitters to be missed if their contribution to the approximation error $x - \hat{x}$ is small.

Gilbert et al. proposed the first sublinear-time algorithm for the ℓ_2/ℓ_2 problem that achieves $O(\frac{k}{\epsilon} \log \frac{n}{k})$ measurements with constant failure probability [9]. Earlier sublinear-time algorithms all contain several additional $\log n$ factors in their number of measurements.

[3] See `https://en.wikipedia.org/wiki/Attitude_control` for the notion of 'attitude' in this context.

The optimality of $O(\frac{k}{\epsilon}\log\frac{n}{k})$ measurements was shown by Price and Woodruff [21]. Later Gilbert et al. improved the failure probability to $n^{-k/\operatorname{poly}(\log k)}$ [10], while their number of measurements has a poor dependence on ϵ, which is at least ϵ^{-11}.

Despite the above works, our understanding of the complexity of heavy hitter and compressed sensing schemes on the error probability is very limited. The question regarding failure probability of these schemes is a natural one for two reasons: first, it is strongly connected with the existence of uniform schemes via the probabilistic method, and, second, being able to amplify the failure probability of an algorithm in a non-trivial way without making parallel repetitions of it, makes the algorithm much more powerful application-wise. For sparse recovery schemes, our goal is to obtain the same measurements but with smaller failure probability, something we find important both from a practical and theoretical perspective- obtaining the correct number of measurements in terms of all parameters ϵ, k, n, δ would be the end of the story for compressed sensing tasks, and a challenging quest; we note that previous algorithms achieved optimality with respect to ϵ, k, n only. From the practical side, a small enough failure probability would allow to re-use the same measurements all the time, since an application of the union-bound would suffice for all vectors that might appear in a lifetime; thus, application-wise, we could treat such a scheme as uniform. We start with formal definitions of the problems and then state in detail our improvements over previous work.

1.1 Problem Formulation

For $x \in \mathbb{R}^n$, we define $H_k(x)$ to be the set of its largest k coordinates in magnitude, breaking ties arbitrarily. For a set S let x_S be the vector obtained from x by zeroing out every coordinate $i \notin S$. We also define $x_{-k} = x_{[n]\setminus H_k(x)}$. For the ℓ_2/ℓ_2-sparse recovery results we define $H_{k,\epsilon}(x) = \{i \in [n] : |x_i|^2 \geq \frac{\epsilon}{k}\|x_{-k}\|_2^2\}$.

Two common models in the literature are the strict turnstile model and the (general) turnstile model.

Strict Turnstile Model: Both insertions and deletions are allowed, and it is guaranteed that at all times $x_i \geq 0$.

(General) Turnstile Model: Both insertions and deletions are allowed, but there is no guarantee about the value of x_i at any point in time.

A sketch $f : \mathbb{R}^m \to \mathbb{R}^n$ is a function that maps an n-dimensional vector to m dimensions. In this paper, all sketches will be linear, meaning $f(x) = Ax$ for some $A \in \mathbb{R}^{m\times n}$. The sketch length m will be referred to as the space of our algorithms.

▶ **Definition 1** (Heavy hitters). For $x \in \mathbb{R}^n$ and $p \geq 1$, a coordinate x_i is called an ϵ-heavy hitter in ℓ_p norm if $|x_i|^p \geq \epsilon\|x\|_p^p$. We consider the following three variants of the heavy hitters problem with different **guarantees**:

1. Return a list containing all ϵ-heavy hitters but no $\epsilon/2$-heavy hitters.
2. Return a list L of size $\mathcal{O}(1/\epsilon)$ containing all ϵ-heavy hitters along with estimates $\hat{x}_i$ such that $|x_i - \hat{x}_i|^p \leq \epsilon\|x_{-\lceil 1/\epsilon\rceil}\|_p^p$ for all $i \in L$.
3. Return a list of size $\mathcal{O}(1/\epsilon)$ containing all ϵ-heavy hitters.

When the algorithm is randomized, it has a parameter δ of failure probability; that is, the algorithm succeeds with probability at least $1 - \delta$.

The variant with Guarantee 2 above is also referred to as the ℓ_∞/ℓ_p problem. In this paper we focus on $p = 1$ and $p = 2$, with corresponding Count-Min [5] and Count-Sketch [3] data structures that have been studied extensively.

We note that the strongest guarantee is Guarantee 2. It is folklore that Guarantee 2 implies Guarantees 1 and 3, and Guarantee 1 clearly implies Guarantee 3. In applications, such as sparse recovery tasks, it is often the case that one does not need the full power of Guarantees 1 and 2, but rather is satisfied with Guarantee 3. The natural question that arises is whether one can gain some significant advantage under this Guarantee. Indeed, we show that Guarantee 3 allows the existence of a uniform scheme, i.e., one that works for all vectors, in the strict turnstile model with the same space, in contrast to the other two guarantees.

▶ **Definition 2** (ℓ_2/ℓ_2 sparse recovery)**.** An ℓ_2/ℓ_2-recovery system $\mathcal{A}$ consists of a distribution $\mathcal{D}$ on $\mathbb{R}^{m\times n}$ and a recovery algorithm $\mathcal{R}$. We will write $\mathcal{A} = (\mathcal{D}, \mathcal{R})$. We say that $\mathcal{A}$ satisfies the ℓ_2/ℓ_2 guarantee with parameters $(n, k, \epsilon, m, \delta)$ if for a signal $x \in \mathbb{R}^n$, the recovery algorithm outputs $\hat{x} = \mathcal{R}(\Phi, \Phi x)$ satisfying

$$\mathbb{P}_{\Phi\sim\mathcal{D}}\left\{\|\hat{x} - x\|_2^2 \le (1+\epsilon)\|x_{-k}\|_2^2\right\} \ge 1 - \delta.$$

In the above definition, each coordinate of Φx is called a *measurement* and the vector Φx is referred to as the measurement vector or just as the measurements. The probability parameter δ is referred to as the failure probability.

1.2 Our Results

Heavy hitters. Our first result is an improved analysis of the Count-Min sketch [5] for Guarantee 3 under the change of the hash functions from 2-wise to $\mathcal{O}(\frac{1}{\epsilon})$-wise independence. Previous analyses for Guarantees 1 and 2 use $\mathcal{O}(\frac{1}{\epsilon}\log\frac{n}{\delta})$ space; in contrast our analysis shows that this version of the Count-Min sketch satisfies Guarantee 3 with only $\mathcal{O}(\frac{1}{\epsilon}\log(\epsilon n) + \log(\frac{1}{\delta}))$ space. Notably, the $\frac{1}{\epsilon}$ factor does not multiply the $\log(\frac{1}{\delta})$ factor. This result has two important consequences. First, it gives a uniform scheme for Guarantee 3; second, it implies an improved analysis of the classic dyadic trick [5] for Guarantee 3 using $\mathcal{O}(\frac{1}{\epsilon}\log(\epsilon n) + \log n \log(\frac{\log \epsilon n}{\delta}))$ space. For constant δ, previous analyses of the dyadic trick needed space $\mathcal{O}(\frac{1}{\epsilon}\log n \log(\frac{\log n}{\epsilon}))$ but our analysis shows that $\mathcal{O}(\frac{1}{\epsilon}\log(\epsilon n) + \log(\epsilon n)\log\log(\epsilon n))$ space suffices. These results are summarized in Table 1.

Regarding the lower bound, we give the first bound for Guarantee 2 with $p = 2$, which is simultaneously optimal in terms of n, any $\epsilon > \frac{1}{n^{.99}}$, and the failure probability δ. That is, we prove that the space has to be $\Omega(\frac{1}{\epsilon}\log\frac{\epsilon n}{\delta})$, which matches the upper bound of Count-Sketch [3] whenever $\epsilon > \frac{1}{n^{.99}}$. A lower bound of $\Omega(\frac{1}{\epsilon}\log(\epsilon n))$ was given in [16] and is valid for the full range of parameters of ϵ and n, but previous analyses cannot be adapted to obtain non-trivial lower bounds for $\delta < \frac{1}{n}$. Indeed, the lower bound instances used in arguments in previous work have deterministic upper bounds using $O(\frac{1}{\epsilon}\log n)$ space.

We also show a new randomized lower bound of $\Omega(\frac{1}{\epsilon}(\log n + \sqrt{\log\frac{1}{\delta}}))$ space for $p = 1$, provided that $\frac{1}{\epsilon} > \sqrt{\log\frac{1}{\delta}}$. Although not necessarily optimal, this lower bound is the first to show that a term involving $\log\frac{1}{\delta}$ must multiply the $\frac{1}{\epsilon}$ factor for $p = 1$. The assumption that $\frac{1}{\epsilon} > \sqrt{\log\frac{1}{\delta}}$ is necessary, as there exist deterministic $O(\frac{1}{\epsilon^2})$ space algorithms for $p = 1$ [7, 20]. For deterministic algorithms satisfying Guarantee 3 with $p = 1$, we also show a lower bound of $\Omega(\frac{1}{\epsilon^2})$ measurements, which resolves Question 4 in the IITK Workshop on Algorithms for Data Streams [18].

Table 1 Summary of previous heavy hitter algorithms. The notation $\mathcal{O}(\cdot)$ for space complexity is suppressed, $\tilde{\mathcal{O}}(\cdot)$ hides logarithmic factors in n, $1/\epsilon$ and $1/\delta$.

Algorithm	Space	Guarantee	Query time
Count-Min [5]	$\frac{1}{\epsilon}\log n + \frac{1}{\epsilon}\log(\frac{1}{\delta})$	1, 2	$\tilde{\mathcal{O}}(n)$
This paper	$\frac{1}{\epsilon}\log(\epsilon n) + \log(\frac{1}{\delta})$	3	$\tilde{\mathcal{O}}(n)$
Dyadic Trick [5]	$\frac{1}{\epsilon}\log n \log(\frac{\log n}{\delta\cdot\epsilon})$	1, 2	$\tilde{\mathcal{O}}(\frac{1}{\epsilon})$
This paper	$\frac{1}{\epsilon}\log(\epsilon n) + \log(\epsilon n)\log(\frac{\log(\epsilon n)}{\delta})$	3	$\tilde{\mathcal{O}}(\frac{1}{\epsilon})$

Sparse Recovery. We summarize previous algorithms in Table 2. We give algorithms for the ℓ_2/ℓ_2 problem with failure probability much less than $1/\operatorname{poly}(n)$ whenever $k = \Omega(\log n)$. We present two novel algorithms, one running in $\mathcal{O}(k\operatorname{poly}(\log n))$ time and the other in $O(k^2\operatorname{poly}(\log n))$ time with a trade-off in failure probability. Namely, the first algorithm has a larger failure probability than the second one. The algorithms follow a similar overall framework to each other but are instantiated with different parameters. We also show how to modify the algorithm of [10] to obtain an optimal dependence on ϵ, achieving a smaller failure probability along the way. All of these results are included in Table 2. Our algorithms, while constituting a significant improvement over previous work, are still not entirely optimal. We show, however, that at least in the spiked covariance model, which is a standard average-case model of input signals, we can obtain optimal upper and lower bounds in terms of the measurement complexity. Combined with the identification scheme from [10] we also obtain a scheme with decoding time nearly linear in k, assuming that $k = n^{\Omega(1)}$.

Besides the above non-adaptive schemes, we also make contributions, in terms of the failure probability, for adaptive schemes. For adaptive sparse recovery, Indyk et al. gave an algorithm under the ℓ_2/ℓ_2 guarantee [15] using $\mathcal{O}((k/\epsilon)\log(\epsilon n/k))$ measurements and achieving constant failure probability. In followup work [19] adaptive schemes were designed for other ℓ_p/ℓ_p error guarantees and improved bounds on the number of rounds were given; here our focus, as with the non-adaptive schemes we study, is on the error probability. We give a scheme that achieves failure probability $e^{-k^{1-\gamma}}$ for any constant γ, using the same number of measurements. Moreover, we present an algorithm for the regime when $k/\epsilon \leq \operatorname{poly}(\log n)$. Our scheme achieves the stronger ℓ_∞/ℓ_2 guarantee and fails with probability $1/\operatorname{poly}(\log n)$. Thus, our algorithms improve upon [15] in both regimes: in the high-sparsity regime we get an almost exponential improvement in k, and in the low-sparsity regime we get $1/\operatorname{poly}(\log n)$.

2 Our Techniques

Heavy hitters. All the schemes we give are for the strict turnstile model. Our first algorithm is based on a small but important tuning to the Count-Min sketch: we change the amount of independence in the hash functions from 2-wise to $\mathcal{O}(1/\epsilon)$-wise. Since the estimate of any coordinate is an overestimate of it, we are able to show that the set of $\mathcal{O}(1/\epsilon)$ coordinates with the largest estimates is a superset of the set of the ϵ-heavy hitters. Although changing the amount of independence might increase both the update and the query time by a multiplicative factor of $1/\epsilon$, we show that using fast multipoint evaluation of polynomials, we can suffer only a $\log^2(1/\epsilon)$ factor in the update time in the amortized case, and a $\log^2(1/\epsilon)$ factor in query time in the worst case. The above observation for the Count-Min sketch gives also an improvement on the well-known dyadic trick which appeared in the seminal paper of Cormode and Hadjieleftheriou [4].

Table 2 Summary of previous sparse recovery results and the results obtained in this paper. The notation $\mathcal{O}(\cdot)$ is suppressed. The paper [10] and the third result of our paper require $k = n^{\Omega(1)}$. The constants γ, α should be thought as arbitrarily small constants, say .001. We also note that in the regime $k/\epsilon \leq n^{1-\alpha}$, the decoding time of our first algorithm becomes $(k/\epsilon)\log^{2+\gamma} n$. The exponents in the poly$(\cdot)$ factors in [9] and [10] are at least 5, though the authors did not attempt at an optimization of these quantities. The exponent in the poly$(\cdot)$ factors in [17] is at least 3. We also note that our first result, in the regime $k/\epsilon \leq n^{1-\beta}$, gives also $k\log^{2+\gamma} n$ running time, thus improving the running time of [9].

Paper	Measurements	Decoding Time	Failure Probability
[3]	$\epsilon^{-1}k\log n$	$n\log n$	$1/\operatorname{poly}(n)$
[9]	$\epsilon^{-1}k\log(n/k)$	$\epsilon^{-1}k\operatorname{poly}(\log(n/k))$	$\Omega(1)$
[10]	$\epsilon^{-11}k\log(n/k)$	$k^2\cdot\operatorname{poly}(\epsilon^{-1}\log n)$	$(n/k)^{-k/\log^{13} k}$
[17]	$\epsilon^{-1}k\log n$	$\epsilon^{-1}k\cdot\operatorname{poly}(\log n)$	$1/\operatorname{poly}(n)$
This paper	$\epsilon^{-1}k\log(n/k)$	$\epsilon^{-1}k^{1+\alpha}\log^3 n$	$e^{-\sqrt{k}/\log^3 k}$
	$\epsilon^{-1}k\log(n/k)$	$\epsilon^{-1}k^2(\log k)\log^{2+\gamma}(n/k)$	$e^{-k/\log^3 k}$
	$\epsilon^{-1}k\log(\frac{n}{\epsilon k})$	$\epsilon^{-1}k^2\operatorname{poly}(\log n)$	$(n/k)^{-k/\log k}$

For the deterministic case, our improved analysis of the Count-Min sketch implies a deterministic algorithm that finds heavy hitters of all vectors $x \in \mathbb{R}^n_+$; moreover, we show how expanders that expand only on sets of size $\Theta(1/\epsilon)$ (or equivalently lossless condensers) lead to schemes in the strict turnstile model under Guarantee 3. Then we instantiate the Guruswami-Umans-Vadhan expander [13] properly to obtain an explicit algorithm. The idea of using expanders in the context of heavy hitters has been employed by Ganguly [8], although his result was for the ℓ_∞/ℓ_1 problem with $\Omega(1/\epsilon^2)$ space. Known constructions of these combinatorial objects are based on list decoding, and do not achieve optimal parameters. Any improvement on explicit constructions of these objects would immediately translate to an improved explicit heavy hitters scheme for the strict turnstile model.

Our deterministic lower bounds are based on choosing "bad input vectors" for the sketching matrix S based on several properties of S itself. Since the algorithm is deterministic, it must succeed even for these vectors.

Our randomized lower bounds come from designing a pair of distributions which must be distinguished by a heavy hitters algorithm with the appropriate guarantee. They are based on distinguishing a random Gaussian input from a random Gaussian input with a large coordinate planted in a uniformly random position. By Lipschitz concentration of the ℓ_1-norm and ℓ_2-norm in Gaussian space, we show that the norms in the two cases are concentrated, so we have a heavy hitter in one case but not the other. Typically, the planted large coordinate corresponds to a column in S of small norm, which makes it indistinguishable from the noise on remaining coordinates. The proof is carried out by a delicate analysis of the total variation distance of the distribution of the image of the input under the sketch in the two cases.

Non-Adaptive Sparse Recovery. Our result follows a similar framework to that of [9], though chooses more carefully the main primitives it uses and balances the parameters in a more effective way. Both schemes consist of $\mathcal{O}(\log k)$ so-called weak systems: a scheme that takes as input a vector x and returns a vector $\hat{x}$ which contains a 2/3 fraction of the heavy hitters of x (the elements with magnitude larger than $\frac{1}{\sqrt{k}}\|x_{-k}\|_2$) along with accurate estimates of (most of) them. Then it proceeds by considering the vector $x - \hat{x}$, which contains

at most $1/3$ of the heavy hitters of x. We then feed the vector $x - \hat{x}$ to the next weak-level system to obtain a new vector which contains at most $(2/3)(1/3)k = 2k/9$ of the heavy hitters. Proceeding in a similar fashion, after the i-th stage we will be left with at most $k/3^i$ heavy hitters.

Each weak system consists of an identification and an estimation part. The identification part finds a $2/3$ fraction of the heavy hitters while the estimation part estimates their values. For the identification part, the algorithm in [9] hashes n coordinates to $\Theta(k)$ buckets using a 2-wise independent hash function and then uses an error-correcting code in each bucket to find the heaviest element. Since, with constant probability, a heavy hitter will be isolated and its value will be larger than the 'noise' level in the bucket it is hashed to, it is possible to find a $2/3$ fraction of the heavy hitters with constant probability in $\mathcal{O}(k \operatorname{poly}(\log n))$ decoding time. Moreover, in each bucket we use a b-tree, which is a folklore data structure in the data stream literature, the special case of which ($b = 2$) first appeared in [5]. The estimation part is a different analysis of the folklore Count-Sketch data structure: we show that the estimation scheme can be implemented with an optimal dependence on ϵ, in contrast to the the expander-based scheme in [10], which gave a sub-optimal dependence on ϵ.

In this paper, we first design an algorithm with running time $\mathcal{O}(k^2 \operatorname{poly}(\log n))$, as in [10], and then improve it to $\mathcal{O}(k^{1+\alpha} \operatorname{poly}(\log n))$ time, but with a slightly larger failure probability. The key observation is that in the first round we find a constant fraction of the heavy hitters with e^{-k} failure probability, in the second round we find a constant fraction of the remaining heavy hitters with $e^{-k/2}$ failure probability, and so on, with polynomially decreasing number of measurements. In later rounds we can decrease the number of measurements at a slower rate so that the failure probability can be reduced by using more measurements while the optimality of the number of measurements is retained. Our suffering from the quadratic dependence on k in the runtime is due to the fact that our sensing matrix is very dense, with $\mathcal{O}(k)$ non-zeros per column. Hence, updating measurements $y \leftarrow y - \Phi\hat{x}$ will incur a running time proportional to $\|\hat{x}\|_0 \cdot k$, where $\hat{x}$ is a $\mathcal{O}(k)$-sparse vector.[4]

But, how do we achieve an almost linear time algorithm while beating the constant failure probability of [9]? The idea is to use again the same analysis, but without sharpening the failure probability in the first $(1/2)\log k$ steps. The first $(1/2)\log k$ rounds still fail with tiny probability, and once we reach round $(1/2)\log k$, we can afford to run the quadratic-time algorithm above, since our sparsity is now $\mathcal{O}(\sqrt{k})$. Hence we would expect the total algorithm to run in time $\mathcal{O}((\sqrt{k})^2 \operatorname{poly}(\log n)) = \mathcal{O}(k \operatorname{poly}(\log n))$. Putting everything together, we obtain substantial improvements over both [9] and [10].

A caveat of our approach, which is the reason we obtain $k^{1+\alpha}$ dependence on the decoding time, is the following. Since we do not want to store the whole matrix, our algorithms are implemented differently when $k \leq n^{1-\alpha/2}$ and $k \geq n^{1-\alpha/2}$. In the former case, we use $\log n = \Theta(\log(n/k))$ measurements per bucket in the identification step, in order to avoid inverting an $O(k)$-wise independent hash function. In the latter case, to compute the pre-image of an $O(k)$-wise independent hash function we just evaluate the hash function, which corresponds to a degree-$O(k)$ polynomial, in all places in time $O(n \log^2 k)$, and trivially

[4] We note an omission in the runtime analysis in [10]. Their measurement matrix contains $s = 2^i/i^c$ (where c is a constant) repetitions of an expander-based identification matrix (see Lemma 4.10 and Theorem 4.9 of [10]). Each repetition has at least one non-zero entry per column and thus the measurement matrix for the i-th iteration has at least s non-zero entries per column, which implies that when $i = \log k - 1$, each column has at least $\Omega(k/\operatorname{poly}(\log k))$ nonzero entries. Updating measurements $y \leftarrow y - \Phi\hat{x}$ will then take $\Omega(k^2/\operatorname{poly}(\log k))$ time, where $\hat{x}$ has $\Omega(k)$ nonzero coordinates. Therefore we would expect that the overall running time of the recovery algorithm will be $\tilde{\Omega}(k^2)$ instead of their claimed $\tilde{\mathcal{O}}(k^{1+\alpha})$.

find the pre-images. The asymptotic complexity of our algorithm in its full generality is dominated by the latter case, where we obtain an $\tilde{\mathcal{O}}(k^{1+\alpha})$ decoding time. We note that in the regime $k \leq n^{1-\alpha}$, the running time becomes $\mathcal{O}(k \log^{2+\gamma} n)$.

Adaptive Compressed Sensing. We start by implementing a $1/\text{poly}(\log n)$ failure probability version of the 1-sparse routine of [15]. We apply a preconditioning step before running [15] with a different setting of parameters; this preconditioning step gives us power for the next iteration, enabling us to achieve the desired failure probability in each round.

The lemma above leads to a scheme for ℓ_2/ℓ_2 in the low-sparsity regime, when $k < \text{poly}(\log n)$. The algorithm operates by hashing into $\text{poly}(\log n)$ buckets, determining the heavy buckets using a standard variant of Count-Sketch, and then running the 1-sparse recovery in each of these buckets. The improved algorithm for 1-sparse recovery is crucial here since it allows for a union bound over all buckets found.

For the general case of k-sparsity, we show that the main iterative loop of [15] can be modified so that it gives exponentially smaller failure probability in k. The idea is that, as more and more heavy hitters are found, it is affordable to use more measurements to reduce the failure probability. Interestingly and importantly for us, the failure probability per round is minimized in the first round, and in fact is increasing exponentially, although this was not exploited in [15]. Therefore, in the beginning we have exponentially small failure probability, but in later rounds we can use more measurements to boost the failure probability by making more repetitions. This part needs care in order not to blow up the number of measurements while achieving the best possible failure probability. We use a martingale argument to handle the dependency issue that arises from hashing coordinates into buckets, and thus avoid additional repetitions that would otherwise increase the number of of measurements.

The two algorithms above show how we can beat the failure probability of [15] for all values of k: we have $1/\text{poly}(\log n)$ for small k and $\exp(-k^{0.999})$ for large k, thus achieving asymptotic improvements in every case.

We note that although in the heavy hitters schemes we take into account the space to store the hash functions, in the sparse recovery we adopt the standard practice of not counting the space needed to store the measurement matrix, and therefore we use full randomness.

3 Formal Statement of Results

In this section we state all of our results and in subsequent sections we shall only give an outline of our improved analysis of Count-Min and our lower bound for Count-Sketch. The proofs of all other theorems can be found in the full version. The notations $\mathcal{O}_{a,b,\ldots}, \Omega_{a,b,\ldots}$ indicate that the constant in $\mathcal{O}$- and Ω-notations depend on $a, b, \ldots$.

3.1 Heavy Hitters

3.1.1 Upper Bounds

▶ **Theorem 3** (ℓ_1 Heavy Hitters Under Guarantee 3)**.** *There exists a data structure* DS *which finds the ℓ_1 heavy hitters of any $x \in \mathbb{R}^n$ in the strict turnstile model under Guarantee 3. In other words, we can sketch x, such that we can find a list L of $\mathcal{O}(k)$ coordinates that contains all ε-heavy hitters of x. The space usage is $\mathcal{O}(\frac{1}{\epsilon}\log(\epsilon n))$, the update time is amortized $\mathcal{O}(\log^2(\frac{1}{\epsilon})\log(\epsilon n))$ and the query time is $\mathcal{O}(n\log^2(\frac{1}{\epsilon})\log(\epsilon n))$.*

The following theorem follows by an improved analysis of the dyadic trick [4].

▶ **Theorem 4.** *There exists a data structure with space* $\mathcal{O}(\frac{1}{\epsilon}\log(\epsilon n)+\log(\epsilon n)\cdot\log(\frac{\log(\epsilon n)}{\delta}))$ *that finds the* ℓ_1 *heavy hitters of* $x\in\mathbb{R}^n$ *in the strict turnstile model under Guarantee 3 with probability at least* $1-\delta$*. The update time is* $\mathcal{O}(\log^2(\frac{1}{\epsilon})\log(\epsilon n)+\epsilon\log(\epsilon n)\log^2(\frac{1}{\epsilon})\log(\frac{\log(\epsilon n)}{\delta}))$ *amortized and the query time is* $\mathcal{O}(\frac{1}{\epsilon}(\log^2(\frac{1}{\epsilon})\log(\epsilon n)+\log(\epsilon n)\log^2(\frac{1}{\epsilon})\log(\frac{\log(\epsilon n)}{\delta}))))$.

▶ **Theorem 5** (Explicit ℓ_1 Heavy Hitters in the Strict Turnstile Model)**.** *There exists a fully explicit algorithm that finds the* ϵ*-heavy hitters of any vector* $x\in\mathbb{R}^n$ *using space* $\mathcal{O}(k^{1+\alpha}(\log(\frac{1}{\epsilon})\log n)^{2+2/\alpha})$*. The update time is* $\mathcal{O}(\mathrm{poly}(\log n))$ *and the query time is* $\mathcal{O}(n\cdot\mathrm{poly}(\log n))$*.*

3.1.2 Lower Bounds

▶ **Theorem 6** (Strict turnstile deterministic lower bound for Guarantees 1,2)**.** *Assume that* $n=\Omega(\epsilon^{-2})$*. Any sketching matrix* S *must have* $\Omega(\epsilon^{-2})$ *rows if, in the strict turnstile model, it is always possible to recover from* Sx *a set which contains all the* ϵ*-heavy hitters of* x *and contains no items which are not* $(\epsilon/2)$*-heavy hitters.*

▶ **Theorem 7** (Turnstile deterministic lower bound for Guarantee 3)**.** *Assume that* $n=\Omega(\epsilon^{-2})$*. Any sketching matrix* S *must have* $\Omega(\epsilon^{-2})$ *rows if, in the turnstile model, some algorithm never fails in returning a superset of size* $O(1/\epsilon)$ *containing the* ϵ*-heavy hitters. Note that it need not return approximations to the values of the items in the set which it returns.*

▶ **Theorem 8** (Randomized Turnstile ℓ_1-Heavy Hitters Lower Bound for Guarantees $1,2$)**.** *Assume that* $1/\epsilon\geq C\sqrt{\log(1/\delta)}$ *and suppose* $n\geq\left\lceil 64\epsilon^{-1}\sqrt{\log(1/\delta)}\right\rceil$*. Then for any sketching matrix* S*, it must have* $\Omega(\epsilon^{-1}\sqrt{\log(1/\delta)})$ *rows if, in the turnstile model, it succeeds with probability at least* $1-\delta$ *in returning a set containing all the* ϵ ℓ_1*-heavy hitters and containing no items which are not* $(\epsilon/2)$ ℓ_1*-heavy hitters.*

▶ **Theorem 9** (Randomized ℓ_2-Heavy Hitters Lower Bound with Guarantee 2)**.** *Suppose that* $\delta<\delta_0$ *and* $\epsilon<1/\epsilon_0$ *for sufficiently small absolute constants* $\delta_0,\epsilon_0\in(0,1)$ *and* $n\geq\left\lceil 64\epsilon^{-1}\log(6/\delta)\right\rceil$*. Then for any sketching matrix* S*, it must have* $\Omega(\epsilon^{-1}\log(1/\delta))$ *rows if it succeeds with probability at least* $1-\delta$ *in returning a set containing all the* ϵ*-heavy hitters and containing no items which are not* $(\epsilon/2)$*-heavy hitters.*

3.2 Non-Adaptive Sparse Recovery

▶ **Theorem 10.** *Let* $1\leq k\leq n$ *be integers and* $\gamma>0$ *be a constant. There exists an* ℓ_2/ℓ_2 *sparse recovery system* $\mathcal{A}=(\mathcal{D},\mathcal{R})$ *with parameters* $(n,k,\epsilon,\mathcal{O}_\gamma(k/\epsilon\log(n/\epsilon k)),\exp(-k/\log^3 k))$*. Moreover,* $\mathcal{R}$ *runs in time* $\mathcal{O}_\gamma(k^2\log^{2+\gamma}n)$*.*

In other words, there exists an ℓ_2/ℓ_2 *sparse recovery system that uses* $\mathcal{O}(k\epsilon\log(n/\epsilon k))$ *measurements, runs in time* $\mathcal{O}_\gamma(k^2\log^{1+\gamma}n)$*, and fails with probability* $\exp(-k/\log^3 k)$*.*

▶ **Theorem 11.** *Let* $1\leq k\leq n$ *be integers and* $\gamma>0$ *be a constant. There exists an* ℓ_2/ℓ_2 *sparse recovery system* $\mathcal{A}=(\mathcal{D},\mathcal{R})$ *with parameters* $(n,k,\epsilon,\mathcal{O}_\gamma(k/\epsilon\log(n/k)),\exp(-\sqrt{k}/\log^3 k))$*. Moreover,* $\mathcal{R}$ *runs in time* $\mathcal{O}_\gamma(k/\epsilon\log^{2+\gamma}n)$*. In other words, there exists an* ℓ_2/ℓ_2 *sparse recovery system that uses* $\mathcal{O}(k\epsilon\log(n/\epsilon k))$ *measurements, runs in time* $\mathcal{O}_\gamma(k\log^{2+\gamma}n)$*, and fails with probability* $\exp(-\sqrt{k}/\log^3 k)$*.*

▶ **Theorem 12.** *Suppose that* $k=n^{\Omega(1)}$*. There exists an* ℓ_2/ℓ_2 *sparse recovery system* $\mathcal{A}=(\mathcal{D},\mathcal{R})$ *with parameters* $\left(n,k,\epsilon,\mathcal{O}(\frac{k}{\epsilon}\log\frac{n}{\epsilon k}),(\frac{n}{k})^{-\frac{k}{\log k}}\right)$*. Moreover,* $\mathcal{R}$ *runs in* $\mathcal{O}(k^2/\epsilon\,\mathrm{poly}(\log n))$ *time. In other words, there exists an* ℓ_2/ℓ_2 *sparse recovery system that uses* $\mathcal{O}(k\epsilon\log(n/\epsilon k))$ *measurements, runs in time* $\mathcal{O}(k^2/\epsilon\,\mathrm{poly}(\log n))$*, and fails with probability* $(n/k)^{-k/\log k}$*.*

3.3 Adaptive Sparse Recovery

▶ **Theorem 13** (Entire regime of parameters). *Let $x \in \mathbb{R}^n$ and $\gamma > 0$ be a constant. There exists an algorithm that performs $\mathcal{O}((k/\epsilon)\log\log(\epsilon n/k))$ adaptive linear measurements on x in $\mathcal{O}(\log^* k \cdot \log\log(\epsilon n/k))$ rounds, and finds a vector $\hat{x} \in \mathbb{R}^n$ such that $\|x-\hat{x}\|_2^2 \le (1+\epsilon)\|x_{-k}\|_2^2$. The algorithm fails with probability at most $\exp(-k^{1-\gamma})$.*

▶ **Theorem 14** (low sparsity regime). *Let $x \in \mathbb{R}^n$ and parameters k, ϵ be such that $k/\epsilon \le c\log n$, for some absolute constant c. There exists an algorithm that performs $\mathcal{O}((k/\epsilon)\log\log n)$ adaptive linear measurements on x in $\mathcal{O}(\log\log n)$ rounds, and finds a vector $\hat{x} \in \mathbb{R}^n$ such that $\|x-\hat{x}\|_2 \le (1+\epsilon)\|x_{-k/\epsilon}\|_2$. The algorithm fails with probability at most $1/\operatorname{poly}(\log n)$.*

3.4 Spiked Covariance Model

In the spiked covariance model, the signal x is subject to the following distribution: we choose k coordinates uniformly at random, say, $i_1, \ldots, i_k$. First, we construct a vector $y \in \mathbb{R}^n$, in which each y_{k_i} is a uniform Bernoulli variable on $\{-\sqrt{\epsilon/k}, +\sqrt{\epsilon/k}\}$ and these k coordinate values are independent of each other. Then let $z \sim N(0, \frac{1}{n}I_n)$ and set $x = y + z$. We now present a non-adaptive algorithm (although the running time is slow) that uses $\mathcal{O}((k/\epsilon)\log(\epsilon n/k) + (1/\epsilon)\log(1/\delta))$ measurements and present a matching lower bound.

▶ **Theorem 15** (Upper Bound). *Assume that $(k/\epsilon)\log(1/\delta) \le \beta n$, where $\beta \in (0,1)$ is a constant. There exists an ℓ_2/ℓ_2 algorithm for the spiked-covariance model that uses $\mathcal{O}\left(\frac{k}{\epsilon}\log\frac{\epsilon n}{k} + \frac{1}{\epsilon}\log\frac{1}{\delta}\right)$ measurements and succeeds with probability $\ge 1-\delta$. Here the randomness is over both the signal and the algorithm.*

▶ **Theorem 16** (Lower Bound). *Suppose that $\delta < \delta_0$ for a sufficiently small absolute constant $\delta_0 \in (0,1)$ and $n \ge \lceil 64\epsilon^{-1}\log(6/\delta)\rceil$. Then any ℓ_2/ℓ_2-algorithm that solves with probability $\ge 1-\delta$ the ℓ_2/ℓ_2 problem in the spiked-covariance model must use $\Omega(\epsilon^{-1}\log(1/\delta))$ measurements.*

Combining with the lower bound from [21] (Section 4) we get a lower bound for the spiked covariance model of $\Omega((k/\epsilon)\log(n/k) + \log(1/\delta)/\epsilon)$. We note that although the lower bound is not stated for the spiked covariance model, inspection of the proof indicates that the hard instance is designed in that model.

4 ℓ_∞/ℓ_2 lower bound

This section is devoted to the proof of Theorem 9. The proof is based on designing a pair of hard distributions which cannot be distinguished by a small sketch. We show this by using rotational properties of the Gaussian distribution to reduce our problem to a univariate Gaussian mean estimation problem, which we show is hard to solve with low failure probability.

4.1 Toolkit

We list some facts in measure of concentration phenomenon in this subsection and omit their proofs owing to space limitations. Hereinafter we use $D_{TV}(\cdot,\cdot)$ to denote the total variation distance between two distributions.

▶ **Fact 1** (TVD Between Gaussians). $D_{TV}(N(0,I_r), N(\tau,I_r)) = \mathbb{P}_{g\sim N(0,1)}\{|g| \le \|\tau\|_2/2\}$.

▶ **Fact 2** (Concentration of ℓ_2-Norm). *Suppose $x \sim N(0,I_n)$ and $n \ge 18\ln(6/\delta)$. Then $\mathbb{P}\{\sqrt{n}/2 \le \|x\|_2 \le 3\sqrt{n}/2\} \ge 1-\delta/3$.*

▶ **Fact 3** (Univariate Tail Bound). *Let $g \sim N(0,1)$. There exists $\delta_0 > 0$ such that it holds for all $\delta < \delta_0$ that $\mathbb{P}\{|g| \leq 4\sqrt{\log(1/\delta)}\} \geq 1 - \delta/3$.*

In our proofs we are interested in lower bounding the number r of rows of a sketching matrix S.

4.2 Proof of the lower bound

Proof. Let the universe size be $n = \left\lceil 64\epsilon^{-1}\log(6/\delta)\right\rceil$, which is large enough in order for us to apply Fact 2 (note if the actual universe size is larger, we can set all but the first $\left\lceil 64\epsilon^{-1}\log(6/\delta)\right\rceil$ coordinates of our input to 0).

Let r be the number of rows of the sketch matrix S, where $r \leq n$. If $r > n$, then we immediately obtain an $\Omega(\epsilon^{-1}\log(1/\delta))$ lower bound. We can assume that S has orthonormal rows, since a change of basis to the row space of S can always be performed in a post-processing step.

Hard Distribution. Let I be a uniformly random index in $[n]$.

Case 1: Let η be the distribution $N(0, I_n)$, and suppose $x \sim \eta$. By Fact 2, $\|x\|_2 \geq \sqrt{n/2}$ with probability $1 - \delta/3$. By Fact 3, $|x_I| \leq 4\sqrt{\log(1/\delta)}$ with probability $1 - \delta/3$. Let $\mathcal{E}$ be the joint occurrence of these events, so that $\mathbb{P}(\mathcal{E}) \geq 1 - 2\delta/3$.

By our choice of n, it follows that if $\mathcal{E}$ occurs, then $x_I^2 \leq 16\log(1/\delta) \leq \frac{\epsilon}{2}\|x\|_2^2$, and therefore I cannot be output by an ℓ_2-heavy hitters algorithm.

Case 2: Let $y \sim N(0, I_n)$ and $x = \sqrt{\epsilon n}e_I + y$, where e_I denotes the standard basis vector in the I-th direction. By Fact 2, $\|y\|_2 \leq \frac{3\sqrt{n}}{2}$ with probability $1 - \delta/3$. By Fact 3, $|y_I| \leq 4\sqrt{\log(1/\delta)} < \sqrt{\epsilon n}/2$ with probability $1 - \delta/3$. Let $\mathcal{F}$ be the joint occurrence of these events, so that $\mathbb{P}(\mathcal{F}) \geq 1 - 2\delta/3$.

If event $\mathcal{F}$ occurs, then $|x_I| \geq 3\sqrt{\epsilon n} - 4\sqrt{\log(1/\delta)} \geq \frac{5\sqrt{\epsilon n}}{2}$, and so $x_I^2 \geq \frac{25\epsilon n}{4}$. We also have $\|x\|_2 \leq 3\sqrt{\epsilon n} + \frac{3\sqrt{n}}{2} \leq 2\sqrt{n}$, provided $\epsilon \leq 1/36$, and so $\|x\|_2^2 \leq 4n$. Consequently, $x_I^2 \geq \epsilon\|x\|_2^2$. Consequently, if $\mathcal{F}$ occurs, for an ℓ_2-heavy hitters algorithm to be correct, it must output I.

Conditioning. Let η' be the distribution of η conditioned on $\mathcal{E}$, and let γ' be the distribution of γ conditioned on $\mathcal{F}$. For a distribution μ on inputs y, we let $\bar{\mu}$ be the distribution of Sy.

Note that any ℓ_2-heavy hitters algorithm which succeeds with probability at least $1-\delta$ can decide, with probability at least $1 - \delta$, whether $x \sim \eta'$ or $x \sim \gamma'$. Hence, $D_{TV}(\bar{\eta'}, \bar{\gamma'}) \geq 1 - \delta$. Observe for any measurable set $A \subseteq \mathbb{R}^m$ it holds that

$$\left|\frac{\bar{\eta}(A) - \bar{\mu}(A)}{1 - \frac{2}{3}\delta} - (\bar{\eta}'(A) - \bar{\mu}'(A))\right| \leq \frac{2}{3}\delta,$$

and so it then follows that $D_{TV}(\bar{\eta}, \bar{\gamma}) \geq \left(D_{TV}(\bar{\eta'}, \bar{\gamma'}) - \frac{2\delta}{3}\right)\left(1 - \frac{2}{3}\delta\right) \geq 1 - \frac{7\delta}{3}$. Therefore, to obtain our lower bound, it suffices to show if the number r of rows of S is too small, then it cannot hold that $D_{TV}(\bar{\eta}, \bar{\gamma}) \geq 1 - 7\delta/3$.

Bounding the Total Variation Distance. Since S has orthonormal rows, by rotational invariance of the Gaussian distribution, the distribution of $\bar{\eta}$ is identical to $N(0, I_r)$ and the distribution of $\bar{\gamma}$ identical to $(3\sqrt{\epsilon n})S_I + N(0, I_r)$, where S_I is the I-th column of S.

Since S has orthonormal rows, by a Markov bound, for $9/10$ fraction of values of I, it holds that $\|S_I\|_2^2 \leq \frac{10r}{n}$. Call this set of columns T.

Let $\mathcal{G}$ be the event that $I \in T$, then $\mathbb{P}(\mathcal{G}) \geq 9/10$. It follows that

$$\begin{aligned} D_{TV}(\bar{\eta}, \bar{\gamma}) \leq \mathbb{P}(G) D_{TV}(\bar{\eta}, \bar{\gamma}|\mathcal{G}) + \mathbb{P}(\neg G) D_{TV}(\bar{\eta}, \bar{\gamma}|\neg\mathcal{G}) &\leq \mathbb{P}(G) D_{TV}(\bar{\eta}, \bar{\gamma}|\mathcal{G}) + 1 - \mathbb{P}(G) \\ &= 1 - \mathbb{P}(G)(1 - D_{TV}(\bar{\eta}, \bar{\gamma}|\mathcal{G})) \\ &\leq 1 - \frac{9}{10}(1 - D_{TV}(\bar{\eta}, \bar{\gamma}|\mathcal{G})). \end{aligned}$$

Hence, in order to deduce a contradiction that $D_{TV}(\bar{\eta}, \bar{\gamma}) < 1 - 7\delta/3$, it suffices to show that $D_{TV}(\bar{\eta}, \bar{\gamma}|\mathcal{G}) < 1 - 70\delta/27$.

The total variation distance between $N(0, I_r)$ and $(3\sqrt{\epsilon n})S_i + N(0, I_r)$ for a fixed $i \in T$ is, by rotational invariance and by rotating S_i to be in the same direction as the first standard basis vector e_1, the same as the total variation distance between $N(0, I_r)$ and $(3\sqrt{\epsilon n})\|S_i\|_2 e_1 + N(0, I_r)$, which is equal to the total variation distance between $N(0, 1)$ and $N(3\sqrt{\epsilon n}\|S_i\|_2, 1)$.

Using that $i \in T$ and so $\|S_i\|_2 \leq \sqrt{10r/n}$, we apply Fact 1 to obtain that the variation distance is at most $\mathbb{P}[|N(0,1)| \leq (3/2)\sqrt{\epsilon n} \cdot \sqrt{10r/n}]$. It follows that

$$D_{TV}(\bar{\eta}, \bar{\gamma} \mid \mathcal{G}) \leq \sum_{i \in T} \frac{1}{|T|} D_{TV}(\bar{\eta}, \bar{\gamma} \mid I = i) \leq \mathop{\mathbb{P}}_{g \sim N(0,1)} \left\{ |g| \leq \frac{3}{2}\sqrt{\epsilon n} \cdot \sqrt{\frac{10r}{n}} \right\},$$

and thus it suffices to show, when r is small, that $\mathbb{P}_{g \sim N(0,1)} \left\{ |g| \geq \frac{3}{2}\sqrt{10\epsilon r} \right\} > \frac{70\delta}{27}$. Observe that the left-hand is a decreasing function in r, and so it suffices to show the inequality above for $r = \alpha\epsilon^{-1}\log(1/\delta)$ for some $\alpha > 0$.

Invoking the well-known bound that (see, e.g., [11])

$$\mathop{\mathbb{P}}_{g \sim N(0,1)} \{g \geq t\} \geq \frac{1}{\sqrt{2\pi}} \cdot \frac{1}{2t} e^{-\frac{t^2}{2}}, \quad t \geq \sqrt{2},$$

we have that

$$\mathop{\mathbb{P}}_{g \sim N(0,1)} \left\{ |g| \geq \frac{3}{2}\sqrt{10\epsilon r} \right\} \geq \frac{3}{4\sqrt{5\pi}} \delta^{\frac{45}{4}\alpha} \frac{1}{\sqrt{\alpha \log(1/\delta)}} > \frac{70}{27}\delta$$

when α is small enough. Therefore it must hold that $r \geq \alpha\epsilon^{-1}\log(1/\delta)$ and the proof is complete. ◀

References

1 Zeyuan Allen Zhu, Rati Gelashvili, and Ilya P. Razenshteyn. Restricted isometry property for general p-norms. *IEEE Trans. Information Theory*, 62(10):5839–5854, 2016.

2 Vladimir Braverman, Gereon Frahling, Harry Lang, Christian Sohler, and Lin F. Yang. Clustering high dimensional dynamic data streams. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 576–585, 2017.

3 Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. *Theoretical Computer Science*, 312(1):3–15, 2004.

4 Graham Cormode and Marios Hadjieleftheriou. Finding frequent items in data streams. *Proceedings of the VLDB Endowment*, 1(2):1530–1541, 2008.

5 Graham Cormode and Shan Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithms*, 55(1):58–75, 2005.

6 Gereon Frahling and Christian Sohler. Coresets in dynamic geometric data streams. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 209–217, 2005.

7 S. Ganguly and A. Majumder. CR-precis: A deterministic summary structure for update data streams. *eprint arXiv:cs/0609032*, 2006.

8 Sumit Ganguly. Data stream algorithms via expander graphs. In *International Symposium on Algorithms and Computation*, pages 52–63. Springer, 2008.

9 Anna C Gilbert, Yi Li, Ely Porat, and Martin J Strauss. Approximate sparse recovery: optimizing time and measurements. *SIAM Journal on Computing*, 41(2):436–453, 2012.

10 Anna C Gilbert, Hung Q Ngo, Ely Porat, Atri Rudra, and Martin J Strauss. ℓ_2/ℓ_2-foreach sparse recovery with low risk. In *International Colloquium on Automata, Languages, and Programming*, pages 461–472. Springer, 2013.

11 Robert D. Gordon. Values of mills' ratio of area to bounding ordinate and of the normal probability integral for large values of the argument. *Ann. Math. Statist.*, 12(3):364–366, 09 1941.

12 Rishi Gupta, Piotr Indyk, Eric Price, and Yaron Rachlin. Compressive sensing with local geometric features. *Int. J. Comput. Geometry Appl.*, 22(4):365, 2012.

13 Venkatesan Guruswami, Christopher Umans, and Salil Vadhan. Unbalanced expanders and randomness extractors from parvaresh–vardy codes. *Journal of the ACM (JACM)*, 56(4):20, 2009.

14 Piotr Indyk. Algorithms for dynamic geometric problems over data streams. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 373–380, 2004.

15 Piotr Indyk, Eric Price, and David P Woodruff. On the power of adaptivity in sparse recovery. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 285–294. IEEE, 2011.

16 Hossein Jowhari, Mert Sağlam, and Gábor Tardos. Tight bounds for lp samplers, finding duplicates in streams, and related problems. In *Proceedings of the thirtieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 49–58. ACM, 2011.

17 Kasper Green Larsen, Jelani Nelson, Huy L Nguyen, and Mikkel Thorup. Heavy hitters via cluster-preserving clustering. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 61–70. IEEE, 2016.

18 Andrew McGregor. Open problems in data streams and related topics: Iitk workshop on algorithms for data streams, 2006, 2007.

19 Vasileios Nakos, Xiaofei Shi, David P. Woodruff, and Hongyang Zhang. Improved algorithms for adaptive compressed sensing. In *ICALP*, 2018.

20 Jelani Nelson, Huy L. Nguyên, and David P. Woodruff. On deterministic sketching and streaming for sparse recovery and norm estimation. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 15th International Workshop, APPROX 2012, and 16th International Workshop, RANDOM 2012, Cambridge, MA, USA, August 15-17, 2012. Proceedings*, pages 627–638, 2012.

21 E. Price and D. P. Woodruff. $(1 + \epsilon)$-approximate sparse recovery. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 295–304, Oct 2011.