




# Streaming Algorithms with Large Approximation Factors

Yi Li  


Division of Mathematical Sciences, Nanyang Technological University, Singapore

Honghao Lin 

Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, USA

David P. Woodruff 

Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, USA

Yuheng Zhang 

Zhiyuan College, Shanghai Jiao Tong University, China

---

## Abstract

---

We initiate a broad study of classical problems in the streaming model with insertions and deletions in the setting where we allow the approximation factor  $\alpha$  to be much larger than 1. Such algorithms can use significantly less memory than the usual setting for which  $\alpha = 1 + \epsilon$  for an  $\epsilon \in (0, 1)$ . We study large approximations for a number of problems in sketching and streaming, assuming that the underlying  $n$ -dimensional vector has all coordinates bounded by  $M$  throughout the data stream:

1. For the  $\ell_p$  norm/quasi-norm,  $0 < p \leq 2$ , we show that obtaining a  $\text{poly}(n)$ -approximation requires the same amount of memory as obtaining an  $O(1)$ -approximation for any  $M = n^{\Theta(1)}$ , which holds even for randomly ordered streams or for streams in the bounded deletion model.
2. For estimating the  $\ell_p$  norm,  $p > 2$ , we show an upper bound of  $O(n^{1-2/p}(\log n \log M)/\alpha^2)$  bits for an  $\alpha$ -approximation, and give a matching lower bound for linear sketches.
3. For the  $\ell_2$ -heavy hitters problem, we show that the known lower bound of  $\Omega(k \log n \log M)$  bits for identifying  $(1/k)$ -heavy hitters holds even if we are allowed to output items that are  $1/(\alpha k)$ -heavy, provided the algorithm succeeds with probability  $1 - O(1/n)$ . We also obtain a lower bound for linear sketches that is tight even for constant failure probability algorithms.
4. For estimating the number  $\ell_0$  of distinct elements, we give an  $n^{1/t}$ -approximation algorithm using  $O(t \log \log M)$  bits of space, as well as a lower bound of  $\Omega(t)$  bits, both excluding the storage of random bits, where  $n$  is the dimension of the underlying frequency vector and  $M$  is an upper bound on the magnitude of its coordinates.
5. For  $\alpha$ -approximation to the Schatten- $p$  norm, we give near-optimal  $\tilde{O}(n^{2-4/p}/\alpha^4)$  sketching dimension for every even integer  $p$  and every  $\alpha \geq 1$ , while for  $p$  not an even integer we obtain near-optimal sketching dimension once  $\alpha = \Omega(n^{1/q-1/p})$ , where  $q$  is the largest even integer less than  $p$ . The latter is surprising as it is unknown what the complexity of Schatten- $p$  norm estimation is for constant approximation; we show once the approximation factor is at least  $n^{1/q-1/p}$ , we can obtain near-optimal sketching bounds.

**2012 ACM Subject Classification** Theory of computation  $\rightarrow$  Streaming, sublinear and near linear time algorithms

**Keywords and phrases** streaming algorithms,  $\ell_p$  norm, heavy hitters, distinct elements

**Digital Object Identifier** 10.4230/LIPIcs.APPROX/RANDOM.2022.13

**Category** RANDOM

**Related Version** Full Version: <https://arxiv.org/abs/2207.08075>

**Funding** Yi Li: Supported in part by a Singapore Ministry of Education (MOE) AcRF Tier 1 grant RG75/21.

Honghao Lin: Supported in part by National Science Foundation (NSF) grant No. CCF-1815840.

David P. Woodruff: Supported in part by National Science Foundation (NSF) grant No. CCF-1815840.



© Yi Li, Honghao Lin, David P. Woodruff, and Yuheng Zhang;  
licensed under Creative Commons License CC-BY 4.0

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2022).

Editors: Amit Chakrabarti and Chaitanya Swamy; Article No. 13; pp. 13:1–13:23



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

## 1 Introduction

The data stream model is an important model for analyzing massive datasets, where the sheer size of the input imposes severe restrictions on the resources available to an algorithm. Such algorithms have only a small amount of memory and can only make a few passes over the data. Given a stream of elements from some universe, the algorithm maintains a short sketch, or summary, of what it has seen. Often such sketches are linear, which has multiple benefits, e.g., (1) the sketches can handle both insertions and deletions of items, and (2) the sketches are mergeable, meaning that given the sketch of a stream  $S$  and the sketch of a stream  $S'$ , the sketch of the concatenation of streams  $S$  and  $S'$  is the sum of the two sketches.

Many streaming algorithms have been developed to study fundamental problems in databases, such as estimating the number  $\ell_0$  of distinct elements, which is useful for query optimization and data mining. Among other things, this statistic can be used for selecting a minimum cost query plan [47], the design of databases [24], OLAP [44, 48], data integration [17, 20], and data warehousing [1]. Other important streaming problems include finding the heavy hitters, also known as the top- $k$ , most popular items, frequent items, elephants, or iceberg queries. These can be used in association rules and frequent itemsets [2, 27, 28, 46, 50], and for iceberg queries and iceberg datacubes [11, 23, 26]. Other important applications include estimating the frequency moments  $F_p$  [3], which for  $p \geq 1$  correspond to the  $p$ -th power of the  $\ell_p$  norm of the vector of frequencies of items, where the frequency of an item is its number of occurrences in the stream. For  $p \geq 2$ ,  $F_p$  indicates the degree of skew of the data, which may determine the selection of algorithms for data partitioning [21]. The case  $p = 2$  is the self-join size, which is useful for algorithms involving joining a relation with itself. The frequency moments of a vector are special cases of the Schatten- $p$  norms of a matrix, and there is a large body of work in the data stream model studying these intriguing norms [42, 43, 41, 15, 16], as well as the related cascaded norms [19, 5, 31, 6].

Given that the memory of a data stream algorithm is often significantly sublinear in the size of a stream  $\mathcal{S}$ , such algorithms are usually both randomized and approximate, and very often come with a guarantee that for a function  $f(\mathcal{S})$ , the output  $X$  of the algorithm satisfies that  $(1 - \epsilon)f(\mathcal{S}) \leq X \leq (1 + \epsilon)f(\mathcal{S})$ , with probability at least  $2/3$  over the coin tosses of the algorithm, where  $\epsilon \in (0, 1)$  is a parameter of the algorithm. Here the  $2/3$  probability can typically be amplified to  $1 - \delta$  by repeating the algorithm  $O(\log(1/\delta))$  times independently and outputting the median estimate. While a large body of work in the last two decades has resolved the space complexity of many of the aforementioned problems for  $\epsilon \in (0, 1)$ , for certain applications the lower bounds on the space complexity may be too large to be useful. For such applications it is therefore natural to allow for a larger approximation factor  $\alpha > 1$ , in the hope of obtaining a smaller amount of memory. Namely, one could instead ask for the output  $X$  of the streaming algorithm to satisfy  $f(\mathcal{S}) \leq X \leq \alpha \cdot f(\mathcal{S})$ . This motivates our main question:

*What is the space complexity of classical streaming problems when the approximation factor  $\alpha$  is allowed to be much larger than 1?*

Perhaps surprisingly, this question does not seem to be well-understood, and is in fact open for all of the abovementioned problems in a data stream. There are a few related works, such as [18], which studies large approximation factors for *deterministically* estimating the number of distinct elements,  $\ell_p$ -estimation, entropy estimation, as well as maximum matching size in a graph stream; see also [7] for large approximation factor lower bounds for randomized algorithms for maximum matching. Other streaming problems where large approximation

factors were studied include dynamic time warping [14], maximum  $k$ -coverage [29] and the  $p$ -to- $q$  norms [37]. In contrast to [18], our focus is on tight bounds for randomized algorithms, for which significantly less memory can be achieved than deterministic algorithms, and for a wide range of fundamental problems in the data stream model that do not appear to have been studied before for large approximation factors.

## 1.1 Our Results

A summary of our upper and lower bounds for a number of data stream problems can be found in Tables 1 and 2.

For estimating the  $\ell_p$  norm for  $0 < p \leq 2$ , we show that obtaining a  $\text{poly}(n)$ -approximation requires the same amount of memory as obtaining an  $O(1)$ -approximation, under the common assumption that  $M = \text{poly}(n)$ . Namely, we show an  $\Omega(\log n)$  lower bound even with a random oracle for these problems. Previously, only an  $\Omega(1)$  lower bound was known for  $\text{poly}(n)$ -approximation in this setting. Our result also holds if the stream is randomly ordered, or in the bounded deletion model [30], in which deletions are allowed but the norm should not drop by more than a constant factor from what it was at a previous point in time. Our lower bound can also be extended to a wide class of statistical  $M$ -estimators. We also show a two-pass algorithm that uses less space than the best existing one-pass algorithm.

For estimating the  $\ell_p$  norm of an underlying  $n$ -dimensional vector,  $p > 2$ , we show an upper bound of  $O(n^{1-2/p}(\log n \log M)/\alpha^2)$  bits for  $\alpha$ -approximation for any  $\alpha > 1$ , and a matching lower bound for almost the full range of  $\alpha$  on the bit complexity of linear sketches, which gives a matching streaming lower bound under the conditions of [40], though these conditions can be restrictive, see, e.g., [34] for discussion. One important motivation for studying such norms is to data-augmented streaming algorithms. For example, it was shown in [32] that for estimating the  $\ell_p$  norm with a so-called learned oracle, one can achieve an  $O(1)$ -approximation using  $\tilde{O}(n^{1/2-1/p})$  bits of space. However, this requires a successfully trained oracle, which could have an arbitrarily bad approximation in the worst case. By instead running our worst-case  $\tilde{O}(n^{1/4-1/(2p)})$ -approximation algorithm for  $\ell_p$  estimation with  $\tilde{O}(n^{1/2-1/p})$  bits of memory in parallel, we can ensure that we do at least as well as the learned algorithm in the same amount of memory (up to a constant factor), but we can ensure we never return worse than an  $\tilde{O}(n^{1/4-1/(2p)})$ -approximation. Another important consequence of our  $\ell_p$ -estimation algorithm is that it can be used as a subroutine to obtain large approximations for the  $(p, q)$ -cascaded-norm ( $p \geq 1, q > 2$ ) and rectangle  $\ell_p$  ( $p > 2$ ) problems, showing that the previous space bounds can be reduced by an  $\alpha^2$  factor for an  $\alpha$ -approximation. These results are shown in Sections E and F.

In the  $\ell_2$ -heavy hitters problem, the goal is to output a subset  $S$  of  $\{1, 2, \dots, n\}$  which contains every  $i$  for which  $x_i^2 \geq \frac{1}{k} \|x\|_2^2$ , and no  $i$  for which  $x_i^2 < \frac{1}{2k} \|x\|_2^2$ . It is known [8, 33] that the space complexity of this problem is  $\Theta(k \log n \log M)$  bits, if we are promised that  $x \in \{-M, \dots, M\}^n$ . A natural relaxation would be to instead require only that  $S$  contains every index  $i$  for which  $x_i^2 \geq \frac{1}{k} \|x\|_2^2$  and no  $i$  for which  $x_i^2 < \frac{1}{\alpha k} \|x\|_2^2$ . We show a strong negative result, that for any  $\alpha = O((n/k)(\log \log n)^2/(\log n)^2)$ , this problem still requires  $\Omega(k \log n \log M)$  bits of memory for any linear sketch, which gives a matching streaming lower bound under the conditions of [40]. For our bit complexity lower bound we assume the algorithm succeeds with probability  $1 - O(1/n)$ , while our sketching dimension lower bound only requires that the algorithm succeeds with constant probability. Interestingly, the proofs of our lower bounds do not use the usual hard instances for finding  $\ell_2$ -heavy hitters [8, 33], and instead use a hard instance for  $\ell_p$ -estimation in [51].

For estimating the number  $\ell_0$  of distinct elements, we show that to obtain an  $\alpha = n^{1/t}$ -approximation, an upper bound of  $O(t \log \log M)$  bits is possible and there is a lower bound of  $\Omega(t)$  bits, where  $n$  denotes the dimension of the underlying frequency vector and  $M$  is an upper bound on the absolute value of its coordinates. We state our results in the random oracle model, where a public random string is known to the algorithm. Without such a random string, a simple reduction from the Equality communication problem gives an  $\Omega(\log n)$  bit lower bound for any multiplicative approximation, see, e.g., [3] for similar arguments<sup>1</sup>. Nevertheless, our results are still interesting outside of the random oracle model, since in the common setting of  $M \leq \text{poly}(n)$ , setting  $t = (\log n) / \log \log n$  gives us an  $O(\log n)$ -approximation with  $O(\log n)$  bits of memory, and since  $O(\log n)$  bits of randomness is also sufficient, this matches the  $\Omega(\log n)$  bit lower bound from the Equality problem. The previous best algorithm [36] required at least  $O(\log n \log \log M)$  bits for any multiplicative approximation factor. We also study estimating the number of distinct elements in two and three passes, showing a separation for the problem between one and two passes and a near-optimal three-pass algorithm.

The Schatten- $p$  norm of an  $n \times n$  input matrix  $A$  is just the  $\ell_p$ -norm of the vector of singular values of  $A$ . For  $\alpha$ -approximation to the Schatten- $p$  norm, we give a linear sketch of dimension  $\tilde{O}(n^{2-4/p}/\alpha^4)$ , which is optimal up to logarithmic factors, for every even integer  $p$  and every  $\alpha \geq 1$ , while for  $p$  not an even integer we obtain a near-optimal sketch dimension of  $\tilde{O}(n^{2-4/p}/\alpha^4)$  once  $\alpha = \Omega(n^{1/q-1/p})$ , where  $q$  is the largest even integer less than  $p$ . Interestingly, we obtain the first near-optimal multiplicative approximations for Schatten- $p$  norms for non-integer  $p$  for a wide range of non-trivial approximation factors  $\alpha$ , whereas it is still unknown and a major open question (see, e.g., [41] for discussion) to obtain optimal multiplicative approximations for Schatten- $p$  norms for non-integer  $p$  when  $\alpha = O(1)$ . Our work highlights that surprisingly, the difficulty of this problem stems from the approximation factor rather than the problem being hard for every approximation factor.

## 1.2 Our Techniques

For our lower bound for estimating  $\ell_p$ -norms for  $0 < p \leq 2$  (or more generally for  $M$ -estimators), we give a reduction from the coin problem introduced in [12] and strengthened in [13]. Consider a sequence of independent coin flips with either a heads probability of  $1/2 + \beta$  or a heads probability of  $1/2 - \beta$ . The coin problem asks us to distinguish between the two cases with the fewest number of flips. Given a sequence of  $n$  coin flips, for an underlying vector  $x$  in a stream we can perform  $x_1 \leftarrow x_1 + 1$  or  $x_1 \leftarrow x_1 - 1$ , depending on whether the coin is a heads or a tail. To ensure a bounded deletion stream, we initialize  $x = (2n\beta, 0, \dots, 0)$ . Then, with constant probability, we have  $x_1 = 4n\beta \pm O(\sqrt{n})$  in one case and  $x_1 = \pm O(\sqrt{n})$  in the other, resulting in an  $\alpha$ -factor difference in the  $\ell_p$ -norm when  $4n\beta = \Omega(\alpha\sqrt{n})$ . Note that our goal is to obtain a lower bound for  $\alpha = \omega(1)$ . The earlier lower bound for the coin problem [12] instead considers  $\beta \sim 1/\sqrt{n}$ , which only translates into  $\alpha = \Theta(1)$  at best, for which we know an upper bound of  $O(\log n)$  words exists. The newer result [13] shows an  $O(\log n)$  bit lower bound for  $\beta < n^{1/3-\epsilon}$ . Such a  $\beta$  translates into  $\alpha = n^{\Omega(1)}$ , as desired. This is also the first application of the newer result [13] to data streams.

<sup>1</sup> Briefly, Alice has  $x \in \{0, 1\}^n$  and inserts  $i$  for which  $x_i = 1$ . Bob has  $y \in \{0, 1\}^n$  and deletes  $i$  for which  $y_i = 1$ . If  $x = y$  then  $\ell_0 = 0$ , otherwise it is non-zero, and the private coin randomized communication complexity of Equality is  $\Omega(\log n)$  bits.

■ **Table 1** Summary of previous results and the results obtained in this work. We assume that  $M = \text{poly}(n)$  and  $p$  is constant. The reported space bounds are measured in bits except for the Schatten- $p$  norm, where we consider the sketching dimension. In this table,  $\tilde{\Omega}(f)$  denotes  $\Omega(f \text{ poly log } f)$  and, for the rectangle  $\ell_p$  estimation problem,  $O^*(f)$  denotes  $f \cdot \text{poly}(d, \log(mn/\delta))$ . For the  $\ell_2$ -heavy hitters problem, both our lower bound and our upper bound for bit complexity assume that the success probability is at least  $1 - O(1/n)$ , while for the sketching dimension results we assume constant success probability.

Problem		Large Approx. Ratio		Constant Approx. Ratio	
$\ell_p$ Estimation ( $0 < p \leq 2$ )	$\text{poly}(n)$	$\Omega(\log n)$	Thm 7	$O(\log n)$ [35]	$\Omega(\log n)$ [35]
$\ell_p$ Estimation ( $p > 2$ )	$\alpha$	$\tilde{O}(n^{1-2/p}/\alpha^2)$	Thm 12	$\tilde{O}(n^{1-2/p})$	e.g. [6]
		$\tilde{\Omega}(n^{1-2/p}/\alpha^2)$	Thm 18	$\tilde{\Omega}(n^{1-2/p})$	e.g. [51]
$\ell_2$ Heavy Hitters	$\tilde{O}(n/k)$	$\Omega(k \log^2 n)$	Thm 21	$O(k \log^2 n)$	$\Omega(k \log^2 n)$ [33]
$\ell_2$ Heavy Hitters (Sketching Dimension)	$\tilde{O}(n/k)$	$\Omega(k \log n)$	Thm 24	$O(k \log n)$	$\Omega(k \log n)$ [45]
Distinct Elements	$n^{1/t}$	$O(t \log \log n)$	Thm 25	$O(\log n \log \log n)$ [36]	
		$\Omega(t)$	Thm 29	$\Omega(\log n \log \log n)$ [52]	
Schatten- $p$ Norm	$\alpha$	$\tilde{O}(n^{2-4/p}/\alpha^4)$	Thm 35, 36	$O(n^{2-4/p})$ even $p$ [41]	
		$\Omega(n^{2-4/p}/\alpha^4)$	Thm 38	$\Omega(n^{2-4/p})$	[41]
Cascaded Norm ( $p, q > 2$ )	$\alpha$	$\tilde{O}(n^{1-2/p}d^{1-2/q}/\alpha^2)$	Thm 39	$\tilde{O}(n^{1-2/p}d^{1-2/q})$	[6]
		$\Omega(n^{1-2/p}d^{1-2/q}/\alpha^2)$	Thm 40	$\Omega(n^{1-2/p}d^{1-2/q})$	[31]
Cascaded Norm ( $1 \leq p < 2, q > 2$ )	$\alpha$	$\tilde{O}(d^{1-2/q}/\alpha^2)$	Thm 39	$\tilde{O}(d^{1-2/q})$	[6]
		$\tilde{\Omega}(d^{1-2/q}/\alpha^2)$	Thm 40	$\Omega(d^{1-2/q})$	[31]
Rectangle $F_p$ Estimation ( $p > 2$ )	$\alpha$	$O^*(n^{d(1-2/p)}/\alpha^2)$	Thm 41	$O^*(n^{d(1-2/p)})$	[49]
		$\Omega(n^{d(1-2/p)}/\alpha^2)$	Thm 18	$\Omega(n^{d(1-2/p)})$	[49]

■ **Table 2** Summary of previous results and the results obtained in this work to obtain a  $(1 \pm \varepsilon)$ -approximation. The reported space bounds are measured in bit complexity.

Problem	Type	New Alg	Previous 1-pass Alg
Distinct Elements	2-pass	$O(\log n + \varepsilon^{-2} \log \log M (\log(1/\varepsilon) + \log \log M))$ Theorem 30	$O(\varepsilon^{-2} \log n \log \log nM)$ [36]
	3-pass	$O(\log n + \varepsilon^{-2} (\log(1/\varepsilon) + \log \log M))$ Theorem 31	$\Omega(\varepsilon^{-2} \log n \log \log nM)$ [52]
$\ell_p$ Moment ( $p \leq 2$ )	2-pass	$O(\log n + \varepsilon^{-2} (\log M + \log 1/\varepsilon))$ Theorem 32	$O(\varepsilon^{-2} \log nM)$ [35]

For our upper bound for estimating  $\ell_p$ -norms for  $p > 2$ , we connect the problem to an instance of the same problem with a different parameter. Namely, suppose that  $q$  is such that  $n^{1-2/q} = \Theta(n^{1-2/p}/\alpha^2)$ , where  $\alpha$  is the approximation factor. Then from relationships between norms we have  $\|x\|_p \leq \|x\|_q \leq \alpha \|x\|_p$ . Hence, a constant factor approximation to the  $\ell_q$  norm actually gives an  $\alpha$  approximation to the  $\ell_p$  norm. This “self-reduction” from an instance of the problem under one norm to an instance of the same problem under a different norm also helps us derive our algorithm for estimating the Schatten- $p$  norms of a matrix when  $\alpha = \Omega(n^{1/q-1/p})$ , where  $q$  is the largest even integer less than  $p$ . For our lower bound for  $\ell_p$ -norm estimation for  $p > 2$ , we consider the multiparty disjointness ( $\text{DISJ}_s^n$ ) problem in the public-coin simultaneous message passing model, which was initially proposed in [51]. We show that the hard instance can still give a matching lower bound for  $\alpha$ -approximation if we set the number of players appropriately.

For the  $\ell_2$  heavy hitters problem, the usual hard instances for this problem (see, e.g., [33] and [8]) fail to give an extra  $\log n$  factor for large approximations. The reason is that when reducing from the so-called Augmented Indexing problem, to make the two cases distinguishable for an  $\alpha$ -approximation, one would need to partition the vector into  $\log_\alpha(n)$  levels, which for  $\alpha = n^{\Omega(1)}$ , is only  $O(1)$ . In contrast, we consider the same multiparty disjointness problem we use for the  $\ell_p$  norm estimation problem and show that a similar hard instance gives a matching lower bound for the  $\ell_2$  heavy hitters problem with a large approximation factor. Thus, we use a fundamentally different hard instance for this problem.

For our upper bound for estimating the number  $\ell_0$  of distinct elements, suppose that the approximation factor  $\alpha = n^{1/t}$ . We sub-sample the input coordinates into  $t$  levels, with a geometrically decreasing sampling probability. In each level, the surviving coordinates are hashed into a constant number of buckets. If the  $\ell_0$  of the sub-sampled vector in a level is at most a constant, then only a small number of these buckets will be occupied. Based on this, we can find the specific level  $j^*$  for which the  $\ell_0$  norm in this level is between 0 and  $n^{1/t}$  and show that after rescaling it is a good estimator to the overall  $\ell_0$  of the original vector. To use less memory in each bucket, we choose a random prime  $p = \text{poly}(\log M)$  and only store each counter mod  $p$ . Our lower bound is based on a reduction from the Augmented Indexing problem mentioned above, which in more detail is a two player communication problem in which Alice holds a binary vector  $u \in \{0, 1\}^t$  and asks for Bob to recover  $u_i$  given  $u_{i+1}, \dots, u_t$ . We divide the vector  $x$  into  $l = \Theta(t)$  segments, where the  $i$ -th segment has length  $\Theta(n^{i/l})$ , and fill the  $i$ -th segment with all 1s if and only if  $u_i = 1$ . Then  $\|x\|_0$  differs by a factor of  $\Theta(n^{1/t})$  between the cases of  $u_i = 0$  and  $u_i = 1$ , whence an  $\Omega(t)$  lower bound follows. Despite the fact that a  $\log(1/\varepsilon)$ -factor gap remains in the upper and lower bounds for  $(1 \pm \varepsilon)$ -approximation for  $\ell_0$  (see, e.g., [22] for discussion), we obtain a tight  $\Theta(\log n)$  space bound for  $\alpha = \Theta(\log n)$  approximation, for example. Our bounds also show a separation between the estimation of the  $\ell_p$ -norm ( $0 < p \leq 2$ ) and the  $\ell_0$ -norm with an  $n^{\Theta(1)}$ -approximation factor, since we show an  $\Omega(\log n)$  lower bound via the coin problem for  $p > 0$  and  $n^{\Omega(1)}$  approximation, while we have an  $O(\log \log n)$  upper bound for  $p = 0$  and  $n^{O(1)}$  approximation.

We also consider multi-pass algorithms for  $\ell_0$  and  $\ell_p$  ( $0 < p \leq 2$ ) estimation. For the  $\ell_0$  estimation problem, we show that if we obtain an  $O(\log n)$ -approximation in the first pass, then we can obtain a  $(1 \pm \varepsilon)$ -approximation in the second pass using  $O(\varepsilon^{-2} \log \log M (\log(1/\varepsilon) + \log \log M))$  bits of space. This can be further reduced to  $O(\varepsilon^{-2} (\log(1/\varepsilon) + \log \log M))$  bits of space using a third pass. For  $\ell_p$  estimation, we show that if we can obtain a constant approximation  $Z$  in the first pass, then in the second pass, we can sample the coordinates with probability  $O(\varepsilon^{-2} M^p / Z)$ . Hence, we only need to generate certain  $p$ -stable random variables used in our algorithm with precision  $(M/\varepsilon)^{O(1)}$ , from which we obtain an  $O(\varepsilon^{-2} (\log M + \log(1/\varepsilon)))$  bits of space algorithm in the second pass, which is better than the previous result of  $O(\varepsilon^{-2} \log nM)$  when  $M$  is small.

## 2 Preliminaries

**Notation.** For a vector  $x \in \mathbb{R}^n$ , its  $\ell_p$  norm is  $\|x\|_p = (\sum_{i=1}^n |x_i|^p)^{\frac{1}{p}}$ , where  $p \geq 1$ . We also write  $F_p(x) = \|x\|_p^p$ . We also define  $\|x\|_\infty = \max_i |x_i|$ . When  $p < 1$ , the quantity  $\|x\|_p$  is not a norm though it is a well-defined quantity and  $\|x\|_p^p$  tends to the number of nonzero entries of  $x$  as  $p \rightarrow 0^+$ . In view of this limit, we denote the number of nonzero entries of  $x$  by  $\|x\|_0$  and also refer to it as  $\ell_0$ .



For a matrix  $A \in \mathbb{R}^{m \times n}$ , we define its Schatten- $p$  norm to be  $\|A\|_p = (\sum_{i=1}^{\min\{m,n\}} (\sigma_i(A))^p)^{\frac{1}{p}}$  for each  $p \geq 1$ , where  $\sigma_1(A) \geq \sigma_2(A) \geq \dots$  are the singular values of  $A$ . We also define the  $(p, q)$ -cascaded norm of  $A$  to be  $\|A\|_{p,q} = (\sum_i (\sum_j |A_{i,j}|^q)^{\frac{p}{q}})^{\frac{1}{p}}$  for  $p, q \geq 1$ .

**Turnstile streaming model.** In the turnstile model of data streams, there is an underlying  $n$ -dimensional vector  $x$  which is initialized to 0 and keeps receiving updates of the form  $(i, \Delta) \in [n] \times \mathbb{R}$ , which represents  $x_i \leftarrow x_i + \Delta$ . Here  $\Delta$  can be either positive or negative. In this paper we assume that the underlying vector is guaranteed to be bounded by  $M$ , i.e., it always holds that  $\|x\|_\infty \leq M$  throughout the data stream. The length of the stream is denoted by  $m$ . When the vector  $x$  is given by a stream  $\mathcal{S}$  in the turnstile model, we abuse notation and also write  $\ell_p(\mathcal{S})$  for  $\|x\|_p$ .

When the input describes a matrix  $A \in \mathbb{R}^{m \times n}$ , we can view the matrix as an  $mn$ -dimensional vector and each item in the stream updates an entry of the matrix.

A variant of the streaming model for a matrix  $A$  concerns rectangular updates. Here  $x$  is a tensor indexed by  $[n]^d$  and each update has the form  $(R, \Delta)$ , where  $R \subseteq [n]^d$  is a rectangle, representing the update  $x_i \leftarrow x_i + \Delta$  for all  $i \in R$ . The rectangle  $\ell_p$  problem is considered under this model (see, e.g., [49]), which asks to estimate  $\|A\|_p = (\sum_{i \in [n]^d} |x_i|^p)^{1/p}$ .

**Subspace Embeddings.** Suppose that  $A \in \mathbb{R}^{n \times d}$ . A matrix  $S \in \mathbb{R}^{m \times n}$  is called an  $(\varepsilon, \delta)$ -subspace-embedding for  $A$  if it holds with probability at least  $1 - \delta$  that  $(1 - \varepsilon) \|Ax\|_2 \leq \|SAx\|_2 \leq (1 + \varepsilon) \|Ax\|_2$  for all  $x \in \mathbb{R}^d$  simultaneously. A classical construction is to take  $S$  to be a Gaussian random matrix of i.i.d.  $N(0, 1/m)$  entries, where  $m = O((d + \log(1/\delta))/\varepsilon^2)$ . Recall the minimax characterization of singular values of a matrix  $A$ :  $\sigma_i(A) = \sup_H \inf_{x \in H: \|x\|_2=1} \|Ax\|_2$ , where the supremum is taken over all subspaces  $H$  such that  $\dim(H) = i$ . This implies (see e.g., Lemma 7.2 of [41]) that  $(1 - \varepsilon)\sigma_i(A) \leq \sigma_i(SA) \leq (1 + \varepsilon)\sigma_i(A)$  with probability at least  $1 - \delta$ , for all  $i = 1, \dots, \min\{m, n\}$ , i.e.,  $S$  preserves all singular values of  $A$  if  $S$  is an  $(\varepsilon, \delta)$ -subspace-embedding for  $A$ .

### 3 Lower Bound for $M$ -Estimators

We start by giving a very general lower bound for  $M$ -estimator estimation with a large approximation factor.  $M$ -estimators can be seen as generalizations of the  $p$ -th frequency moments of the underlying vector  $x$ . We first show this lower bound in the turnstile streaming model and later we will show that it still holds even in the bounded deletion and random order models.

► **Definition 1** ( $M$ -estimator with parameter  $\gamma$ ). Suppose  $G : \mathbb{R} \rightarrow \mathbb{R}_{\geq 0}$  is a function. We say  $\|y\|_G = \sum_i G(y_i)$  is an  $M$ -estimator with parameter  $\gamma$  if  $G$  satisfies the following conditions:

- $G(0) = 0$ ;
- $G(x) = G(-x)$ ;
- $G(x)$  is non-decreasing in  $|x|$ ;
- For all  $x, y$  with  $|y| > |x| > 0$ ,  $\frac{G(y)}{G(x)} \geq \left(\frac{|y|}{|x|}\right)^\gamma$ .

We will give a reduction from the following coin problem. In [13], the authors show an  $\Omega(\log n)$  lower bound even when the parameter  $\beta$  is allowed to be very small:

► **Definition 2** (Coin Problem). Let  $X_1, \dots, X_n$  be a stream of i.i.d. random bits, which either (1) comes from a distribution with heads probability  $\frac{1}{2} + \beta$  or (2) comes from a distribution with heads probability  $\frac{1}{2} - \beta$ . We are asked to distinguish these two cases at the end of the stream, with probability  $2/3$ .

► **Theorem 3** ([13]). *For all constant  $\varepsilon > 0$ , any length- $n$  Read-Once Branching Program that solves the coin problem for bias  $\beta < n^{-1/3-\varepsilon}$ , requires  $n^{\Omega(\varepsilon)}$  width.*

► **Corollary 4.** *For all constants  $\varepsilon > 0$ , any randomized streaming algorithm that solves the coin problem with bias  $\beta < n^{-1/3-\varepsilon}$  requires  $\Omega(\log n)$  space. This holds even if we give the algorithm access to an arbitrarily long random tape.*

Suppose that we are given an  $M$ -estimator with parameter  $\gamma$ . We define a distribution  $\mathcal{D}$  on the sequences of  $n$  random bits as follows: suppose that  $\beta = n^{-1/3-\varepsilon}$  for a small constant  $\varepsilon$ . Let  $X_1, \dots, X_n$  be a binary sequence coming from a distribution with heads probability  $\frac{1}{2}$  or a distribution with heads probability  $\frac{1}{2} + \beta$ , where  $X_i = 1$  if the  $i$ -th coin is a head and  $X_i = 0$  if the  $i$ -th coin is a tail. Let  $x$  be the underlying vector in the streaming algorithm. During the stream we perform updates  $x_1 \leftarrow x_1 + 1$  if  $X_i = 1$ , or  $x_1 \leftarrow x_1 - 1$  otherwise. We will show that any streaming algorithm that gives an  $O(n^{(1/6-\varepsilon)\gamma})$ -approximation for  $\|x\|_G$  can distinguish the above two distributions with large constant probability. We first analyze the sum  $|\sum_{i=1}^n X_i|$  for these two distributions. The following two lemmas can be easily proved using Chebyshev's inequality and thus the proofs are omitted.

► **Lemma 5.** *Suppose that the sequence  $(X_1, \dots, X_n) \in \{\pm 1\}^n$  comes from the distribution with heads probability  $\frac{1}{2}$ . Then with probability at least  $1 - 1/(4k)$ , we have  $|\sum_{i=1}^n X_i| \leq \sqrt{kn}$ .*

► **Lemma 6.** *Suppose that the sequence  $(X_1, \dots, X_n) \in \{\pm 1\}^n$  comes from the distribution with heads probability  $\frac{1}{2} + n^{-1/3-\varepsilon}$ . Then with probability at least  $1 - 1/(4k)$ , we have  $\sum_{i=1}^n X_i \geq 2n^{2/3-\varepsilon} - \sqrt{kn}$ .*

We are now ready to give our lower bound.

► **Theorem 7.** *Suppose that  $G$  is an  $M$ -estimator with parameter  $\gamma$ . Then any randomized streaming algorithm which outputs an  $O(M^{(1/6-\varepsilon)\gamma})$ -approximation to  $\|x\|_G$  with probability at least  $2/3$  requires  $\Omega(\log M)$  bits of space, excluding the storage for random bits.*

**Proof.** Suppose that we have a streaming algorithm which outputs an  $O(M^{(1/6-\varepsilon)\gamma})$ -approximation to  $\|x\|_G$ . We shall show that we can distinguish the two distributions with bias  $\beta$  in the coin problem with large constant probability.

We initialize the vector  $x = (0, 0, \dots, 0)$ . Suppose we have a stream of bits  $X_1, \dots, X_M$  coming from the distribution with heads probability  $\frac{1}{2}$  or with heads probability  $\frac{1}{2} + \beta = \frac{1}{2} + M^{-1/3-\varepsilon}$ . Then, during the stream, we perform the update  $x_1 \leftarrow x_1 + 1$  if  $X_i = 1$  and  $x_1 \leftarrow x_1 - 1$  otherwise.

Let  $x^{(0)}$  be the underlying vector if the heads probability for the distribution is  $\frac{1}{2}$  and  $x^{(1)}$  be the underlying vector if the heads probability is  $\frac{1}{2} + \beta$ . From Lemmas 5 and 6 we have that with probability at least  $9/10$ ,  $|x_1^{(0)}| = O(\sqrt{M})$  at the end of the stream, while  $|x_1^{(1)}| = \Omega(M^{2/3-\varepsilon})$  in the second case. It follows from the definition of  $\gamma$  that  $\frac{\|x^{(1)}\|_G}{\|x^{(0)}\|_G} = \Omega(M^{(1/6-\varepsilon)\gamma})$  for the two cases. This implies that if the streaming algorithm can output an  $O(n^{(1/6-\varepsilon)\gamma})$ -approximation to  $\|x\|_G$ , then we can distinguish the two distributions with bias  $\beta$  in the coin problem. From Corollary 4, such a streaming algorithm needs  $\Omega(\log M)$  bits of space. ◀

► **Corollary 8.** *Any randomized streaming algorithm outputting an  $O(M^{1/6-\varepsilon})$ -approximation to  $\|x\|_p^p$  requires  $\Omega(p \cdot \log M)$  bits of space, excluding the storage for random bits. Moreover, under the assumption that  $M = \text{poly}(n)$ , any randomized streaming algorithm outputting a  $\text{poly}(n)$ -approximation to  $\|x\|_p^p$  requires  $\Omega(p \cdot \log n)$  bits of space.*



### 3.1 Lower Bound in Other Streaming Models

**Bounded Deletions.** Our  $\Omega(\log M)$  lower bound still holds even with the assumption of bounded deletions. In this model, the updates  $\Delta_j$  can be positive or negative, but one is promised that the norm  $\|x\|_2$  never drops by more than an  $\alpha$ -fraction of what it was at any earlier point in the stream, for a constant parameter  $\alpha$ . We only state the theorem below, whose proof can be found in the full version.

► **Theorem 9.** *Suppose  $G$  is an  $M$ -estimator with parameter  $\gamma$ . Then any randomized streaming algorithm outputting an  $O(M^{(1/6-\varepsilon)\gamma})$ -approximation of  $\|x\|_G$  in the bounded deletion model needs  $\Omega(\log M)$  bits, excluding the storage for random bits.*

**Random Order.** In the random order model, we assume the updates  $\Delta_j$  come in a random order. We note that the updates for the distribution in Theorem 7 are a sequence of random  $\pm 1$  variables. Hence it satisfies the random order assumption automatically, which means we obtain the following theorem.

► **Theorem 10.** *Suppose  $G$  is an  $M$ -estimator with parameter  $\gamma$ . Then any randomized streaming algorithm which outputs an  $O(M^{(1/6-\varepsilon)\gamma})$ -approximation to  $\|x\|_G$  in the random order model requires  $\Omega(\log M)$  bits of space, excluding the storage for its random bits.*

## 4 $\ell_p$ Estimation $p > 2$

In this section, we consider the problem of estimating  $\|x\|_p$  with a large approximation factor when  $p > 2$ . We present an algorithm that gives an  $\alpha$ -approximation to  $\|x\|_p$  using  $\tilde{O}(n^{1-2/p}/\alpha^2)$  bits of space. We will also give a matching lower bound for this problem.

**Upper bound.** Suppose that we want an  $\alpha$ -approximation where  $n^{1-2/p}/\alpha^2 = \Omega(1)$  (otherwise there is a trivial  $\Omega(1)$  lower bound) and let  $q$  be the number such that  $n^{1-2/q} = \Theta(n^{1-2/p}/\alpha^2)$ . Then we have  $2 \leq q < p$ . The following lemma shows that  $\|x\|_q$  is an  $\alpha$ -approximation to  $\|x\|_p$ .

► **Lemma 11.** *Suppose that  $p \geq q \geq 2$  and  $\alpha \geq 1$  satisfies  $n^{1-2/q} = n^{1-2/p}/\alpha^2 = \Omega(1)$ . Then it holds that  $\|x\|_p \leq \|x\|_q \leq \alpha \|x\|_p$ .*

**Proof.** From our choice of  $q$ , we have that  $\alpha = n^{1/q-1/p}$ . The  $\ell_p$  norm is decreasing in  $p$ , and thus  $\|x\|_q \geq \|x\|_p$ . By Hölder's inequality, it also holds that  $\|x\|_q \leq n^{1/q-1/p} \|x\|_p = \alpha \|x\|_p$ . ◀

The preceding lemma shows that we can use any  $O(1)$ -approximation algorithm for  $\ell_q$  to obtain an  $\alpha$ -approximation to the  $\ell_p$  norm. For example, we can use the  $O(n^{1-2/q} \log^2 n)$ -bit algorithm of [4], or the algorithm of [25]. Our theorem follows immediately.

► **Theorem 12.** *Suppose that  $p > 2$  is a constant. There is an algorithm whose output is  $Z$ , which satisfies that  $\|x\|_p \leq Z \leq \alpha \|x\|_p$  with probability at least 0.9. Furthermore, the algorithm uses  $O(n^{1-2/p} \log n \log M/\alpha^2)$  bits of space.*

**Application to Data-augmented Algorithm Design.** One important motivation for  $\ell_p$  estimation with large approximation is worst-case guarantees for learning-augmented data stream algorithm design. In [32], it was shown that given a heavy hitter oracle which can decide, for each input  $i$ , whether or not  $|x_i| \geq n^{-p/2} \|x\|_p$ , one can estimate  $\|x\|_p$  up to

## 13:10 Streaming Algorithms with Large Approximation Factors

a constant factor with probability at least 0.9 using  $O(n^{1/2-1/p} \log n \log M)$  bits of space. In this case, we say the oracle is successful. However, when the oracle is not successful, there is no worst-case guarantee on the quality of approximation. An observation here is that when the oracle is not successful, the estimation will be an under-estimate with high probability. Letting  $\alpha = \Theta(n^{1/4-1/(2p)})$  in the preceding theorem, we obtain an  $\alpha$ -approximation algorithm whose output  $Z$  satisfies  $\frac{1}{\alpha} \|x\|_p \leq Z \leq \|x\|_p$  with probability at least 0.9 using the same  $O(n^{1/2-1/p} \log n \log M)$  bits of space. Hence we can run our algorithm and the oracle algorithm in parallel and take a maximum. This guarantees an  $\alpha$ -approximation in  $O(n^{1/2-1/p} \log n \log M)$  bits of space with probability at least 0.8.

► **Theorem 13.** *Assuming a successful oracle, there is a streaming algorithm which runs in  $O(n^{1/2-1/p} \log M \log n)$  bits of space, and for which the output  $Z$  satisfies  $\|x\|_p \leq Z \leq 2 \|x\|_p$ . Moreover, even if the oracle is not successful, the output  $Z$  always satisfies  $\|x\|_p \leq Z \leq n^{1/4-1/(2p)} \|x\|_p$ .*

**Lower Bound.** We next show an  $\Omega(n^{1-2/p}(\log(M) \log(1/\delta))/\alpha^2)$  lower bound for obtaining an  $\alpha$ -approximation to  $\|x\|_p$ , or, equivalently, an  $\Omega(n^{1-2/p}(\log(M) \log(1/\delta))/\alpha^{2/p})$  lower bound for obtaining an  $\alpha$ -approximation of  $F_p(x)$ . We first note that it is easy to get an  $\Omega(n^{1-2/p}/\alpha^{2/p})$  lower bound from the following  $\ell_\infty^k$  communication problem in [9]: there are two parties, Alice and Bob, holding vectors  $x, y \in \mathbb{Z}^n$  respectively, and their goal is to decide if  $\|x - y\|_\infty \leq 1$  or  $\|x - y\|_\infty \geq k$ . This problem requires  $\Omega(n/k^2)$  bits of communication [9]. Let  $k = 2^{1/p} \alpha^{1/p} n^{1/p}$ . For the case where  $\|x - y\|_\infty \leq 1$ , we have  $\|x - y\|_p^p \leq n$ . For the case where  $\|x - y\|_\infty \geq k$ , we have  $\|x - y\|_p^p \geq k^p = 2\alpha n$ . Suppose there is an algorithm  $\mathcal{A}$  which can output a number  $Z$  such that  $\|x\|_p \leq Z \leq \alpha \|x\|_p$  with probability at least  $2/3$ . Then Alice can perform the update  $x$  to the algorithm  $\mathcal{A}$  and send the memory contents of  $\mathcal{A}$  to Bob. Bob then performs the update  $-y$  to  $\mathcal{A}$ . From the discussion above, Bob can determine which of the two cases it is with probability at least  $2/3$ .

To obtain a stronger lower bound, we consider the following version of multiparty disjointness ( $\text{DIS}_s^n$ ), coupled with an input distribution, in the public-coin simultaneous message passing model of communication (SMP), as proposed in [51]. In this setting, there are  $s$  players, each of whom has a bit string  $\mathbf{X}_i \in \{0, 1\}^n$  ( $i \in [s]$ ) as input. The inputs are generated according to the following distribution  $\eta$ .

► **Definition 14** (Distribution  $\eta$ ). *The distribution  $\eta$  is the joint distribution of  $(\mathbf{X}_1, \dots, \mathbf{X}_s) \in (\{0, 1\}^n)^s$ , generated as follows.*

- For each  $i \in [n], j \in [s]$ , set  $\mathbf{X}_{j,i} \sim B(1/s)$  independently at random.
- Pick a uniformly random coordinate  $I \in [n]$ .
- Pick a  $Z \in \{0, 1\}$ . If  $Z = 1$ , set  $\mathbf{X}_{j,I} = 1$  for all  $j \in [s]$ . (If  $Z = 0$ , keep all coordinates as before.)

We call the instance of the inputs  $\{\mathbf{X}_i\}_{i \in [s]}$  a “YES” instance when  $Z = 1$ , and a “NO” instance when  $Z = 0$ .

The players simultaneously send a message  $M_i(\mathbf{X}_i, R)$  to a referee, where  $R$  denotes the public coins shared among the players. The referee then decides, based on  $M_1(\mathbf{X}_1, R), \dots, M_s(\mathbf{X}_s, R)$  and  $R$ , whether  $\{\mathbf{X}_i\}_{i \in [s]}$  forms a YES instance or a NO instance. As observed in [51], if  $X \sim \text{Bin}(s, 1/s)$ , then  $\Pr[X > \ell] \leq (e/\ell)^\ell$ . Hence, by a union bound for all coordinates  $i \in [n]$ , it holds in a NO instance, with probability at least  $1 - 1/\text{poly}(n)$ , that  $\sum_{j=1}^s \mathbf{X}_{j,i} \leq c \log n / (\log \log n)$  for all  $i \in [n]$ . On the other hand, in a YES instance it always holds that  $\sum_{j=1}^s \mathbf{X}_{j,I} = s$ . Thus, YES and NO instances are distinguishable for  $s = \Omega(\log n / \log \log n)$ .

The following is an augmented version of this problem.

► **Definition 15** (Aug-DISJ( $r, s, \delta$ )). *The augmented disjointness problem Aug-DISJ( $r, s, \delta$ ) is the following  $s$ -party communication problem. The players receive  $r$  instances of  $\text{DISJ}_s^n(\mathbf{X}_1^1, \dots, \mathbf{X}_s^1), (\mathbf{X}_1^2, \dots, \mathbf{X}_s^2), \dots, (\mathbf{X}_1^r, \dots, \mathbf{X}_s^r)$  and the referee, in addition, receives an index  $T \in [r]$  which is unknown to the players, along with the last  $(r - T)$  inputs  $\{(\mathbf{X}_1^t, \dots, \mathbf{X}_s^t)\}_{t=T+1}^r$ . The inputs are generated according to the following distribution:*

- (i)  $T$  is chosen uniformly at random from  $[r]$ ;
- (ii)  $(\mathbf{X}_1^T, \dots, \mathbf{X}_s^T) \sim \eta$ ;
- (iii) For each  $t \neq T$ ,  $(\mathbf{X}_1^t, \dots, \mathbf{X}_s^t) \sim \eta_0$  independently, where  $\eta_0$  is the conditional distribution of  $\eta$  given  $Z = 0$ .

At the end of the protocol, the referee should output whether the  $T$ -th instance  $(\mathbf{X}_1^T, \dots, \mathbf{X}_s^T)$  is a YES or a NO instance, i.e., the players need to solve  $\text{DISJ}_s^n(\mathbf{X}_1^T, \dots, \mathbf{X}_s^T)$ , with probability  $1 - \delta$ .

► **Theorem 16** ([51]). *Suppose that  $\delta \geq n \cdot 2^{-s}$ . Any deterministic protocol that solves Aug-DISJ( $r, s, \delta$ ) (as defined in Definition 15) requires  $\Omega(rn \min(\log \frac{1}{\delta}, \log s)/s)$  bits of total communication.*

*A Reduction to Streaming:* To lower bound the space complexity of a streaming algorithm we need a way of relating it to the communication cost of a protocol for this communication problem. In [51], the authors use a result of [40], showing under certain conditions that any streaming algorithm  $\mathcal{A}$  which solves the problem  $P$  with probability at least  $1 - \delta$  can be converted to a “path-independent” streaming algorithm  $\mathcal{B}$  which solves  $P$  with probability at least  $1 - 7\delta$ , and which uses the same space up to an additive  $(\log n + \log \log m + \log 1/\delta)$  factor. The latter then gives a protocol for the Aug-DISJ( $r, s, \delta$ ) problem. Here path-independence means that the output of the algorithm only depends on the initial state and the underlying frequency vector. In other words, the order of the updates of the same frequency vector will not cause different outputs to such an algorithm. We now assume that the algorithm  $\mathcal{A}$  we have enjoys this path-independence property. For a more detailed discussion, we refer the readers to Section 5 in [51].

Suppose there is a path-independent 1-pass streaming algorithm  $\mathcal{A}$  which gives an  $\alpha$ -approximation to  $\|x\|_p^p$  with probability  $1 - \delta$ . We shall use this to solve the Aug-DISJ( $r, s, \delta$ ) problem for  $s = \Theta(\alpha^{1/p} n^{1/p})$  and  $r = \log(M/s)$ , from which a space lower bound of  $\Omega(n^{1-2/p} \log(M) \log(1/\delta)/\alpha^{2/p})$  bits follows if  $M = \Omega((n\alpha)^{1/p+O(1)})$ .

We design the following protocol  $\pi$  between the players and the referee. For each  $i \in [s]$ , player  $i$  has  $r$  instances  $(\mathbf{X}_i^1, \mathbf{X}_i^2, \dots, \mathbf{X}_i^r)$ . Player  $i$  performs the update  $10^{j-1} \cdot \mathbf{X}_i^j$  to the algorithm  $\mathcal{A}$ , for each  $j \in [r]$ , and sends the memory contents of  $\mathcal{A}$  to the referee. Under the path-independence assumption, the referee can determine an equivalent frequency vector (i.e., leading to the same state of the algorithm) from each player and then add up the corresponding updates itself. After receiving  $T$  and  $\{(\mathbf{X}_1^t, \dots, \mathbf{X}_s^t)\}_{t=T+1}^r$ , the referee performs the update  $-10^{j-1} \cdot (\sum_{i=1}^s \mathbf{X}_i^j)$  to the algorithm  $\mathcal{A}$ , for each  $j \geq T + 1$ . Suppose that  $\mathcal{A}$  outputs a set  $S$ . The referee will output YES if  $|S| = 1$  and NO if  $S = \emptyset$ .

Next we analyze correctness of the above protocol  $\pi$ . We recall that the referee needs to output the answer to the  $T$ -th instance. For simplicity, we define  $\mathbf{Y}^j = \sum_{i=1}^s \mathbf{X}_i^j$  for the  $j$ -th instance. After taking a union bound, for every instance  $j$ ,  $\|\mathbf{Y}^j\|_\infty \leq c \log n / \log \log n$  if it is a NO instance. Also from a Chernoff bound, it is easy to see that  $\|\mathbf{Y}^j\|_2^2 = \Omega(n)$  for all  $j$  with probability at least  $1 - e^{-\Omega(n)}$ . Note that the actual underlying vector that  $\mathcal{A}$  maintains has the same output as the frequency vector  $\mathbf{Y} = \sum_{t=1}^T 10^{t-1} \mathbf{Y}^t$  after the referee performs the updates. We need the following concentration bounds for  $\mathbf{Y}$  [51]. We note an omission in the proof in that paper and included a corrected one in Appendix A.

## 13:12 Streaming Algorithms with Large Approximation Factors

► **Lemma 17** ([51]). Let  $\sigma_r(\|\mathbf{Y}_{-I}\|_p^p) = (\mathbb{E}[\|\mathbf{Y}_{-I}\|_p^p] - \mathbb{E}[\|\mathbf{Y}_{-I}\|_p^p]^r])^{1/r}$ . It holds that

$$\mathbb{E}[\|\mathbf{Y}_{-I}\|_p^p] \leq K_1^p p^p n \cdot 10^{pT}, \quad (1)$$

$$\sigma_r(\|\mathbf{Y}_{-I}\|_p^p) \leq K_2^p p^p \frac{r}{\ln r} \max\{2^p \sqrt{n}, r^p n^{1/r}\} \cdot 10^{pT}, \quad (2)$$

where  $r \geq 2$  is arbitrary and  $K_1, K_2 > 0$  are absolute constants.

Taking  $r = 3 \ln n$  in (2) gives that

$$\begin{aligned} & \Pr[|\|\mathbf{Y}_{-I}\|_p^p - \mathbb{E}[\|\mathbf{Y}_{-I}\|_p^p]| > 0.1n \cdot 10^{pT}] \\ & \leq \Pr[|\|\mathbf{Y}_{-I}\|_p^p - \mathbb{E}[\|\mathbf{Y}_{-I}\|_p^p]| > 2\sigma_r(\|\mathbf{Y}_{-I}\|_p^p)] \\ & \leq 2^{-r} \leq 1/n^2. \end{aligned} \quad (3)$$

We condition on all of the events above. Notice that in all cases, the value  $\|x\|_\infty$  of the underlying vector  $x$  the algorithm  $\mathcal{A}$  maintains is less than  $(\sum_{i=1}^{r-1} 10^{i-1} \cdot \frac{\log n}{\log \log n} + 10^{r-1} \cdot s) < 10^r \cdot s = O(M)$  for  $r = \log(M/\alpha^{1/p} n^{1/p})$ .

We first consider the case for which the  $T$ -th instance is a YES instance. In this case,  $\mathbf{Y}_I^T = s$  and thus,  $\|\mathbf{Y}\|_p^p \geq 10^{(T-1)p} \cdot s = \Omega(10^{(T-1)p} \cdot \alpha n)$ .

Next consider the case in which the  $T$ -th instance is a NO instance. In this case, we have from (1) and (3) that  $\|\mathbf{Y}\|_p^p = \|\mathbf{Y}_{-I}\|_p^p + \mathbf{Y}_I^p \leq K_p \cdot 10^{pT} n + 10^{pT} (\frac{\log n}{\log \log n})^p \leq 1.1K_p \cdot 10^{pT} n$ , where  $K_p$  is a constant that depends only on  $p$ .

From the same argument in Section 5 we know that if there is an algorithm that can output a  $Z$  such that  $\|x\|_p^p \leq Z \leq K'_p \alpha \|x\|_p^p$ , we can use this algorithm to solve the Aug-DISJ( $r, s, \delta$ ) problem. From Theorem 16, we obtain the following theorem.

► **Theorem 18.** Suppose that  $p$  is a constant and  $M = \Omega((\alpha n)^{1/p+O(1)})$ . Then, for  $\delta \geq 2^{-\Theta((n\alpha)^{1/p})}$ , any one-pass streaming algorithm which outputs a number  $Z$  for which  $\|x\|_p^p \leq Z \leq \alpha \|x\|_p^p$  with probability at least  $1 - \delta$  requires  $\Omega(n^{1-2/p} \log(M) \log(1/\delta)/\alpha^{2/p})$  bits of space. In particular, when  $\delta = \Theta(1/n)$ , any one-pass streaming algorithm requires  $\Omega(n^{1-2/p} \log(M) \log(n)/\alpha^{2/p})$  bits of space.

### 5 $\ell_2$ Heavy Hitters

In the heavy hitters problem, we want to find a set  $S \in [n]$  of indices for the underlying vector  $x$  such that:

- (i)  $S$  contains every  $i$  such that  $|x_i|^2 \geq \frac{1}{k} \|x\|_2^2$ ;
- (ii)  $S$  does not contain any  $i$  such that  $|x_i|^2 < \frac{1}{2k} \|x\|_2^2$ .

We call  $S$  a  $(1/k)$ -heavy set of  $x$  if  $S$  satisfies the above conditions. Using the classical Count-Sketch, we can solve the above problem in  $O(k \log n \log M)$  bits of space with high probability.

► **Lemma 19.** There is a randomized one-pass streaming algorithm which can be implemented in  $O(k \log n \log M)$  bits of space such that with probability  $1 - 1/\text{poly}(n)$ , it can output a  $\frac{1}{k}$ -heavy set  $S$  of  $x$ .

In this section, we consider the following relaxation of the heavy hitters problem, where we want to find a set  $S$  of indices such that:

- (i)  $S$  contains every  $i$  such that  $|x_i|^2 \geq \frac{1}{k} \|x\|_2^2$ ;
- (ii)  $S$  does not contain any  $i$  such that  $|x_i|^2 < \frac{1}{\alpha k} \|x\|_2^2$ .

We call such a set  $S$  a  $(\frac{1}{k}, \alpha)$ -heavy set of  $S$ . Our result is negative, where we show that any one-pass streaming algorithm outputting a  $(\frac{1}{k}, \alpha)$ -heavy set of  $x$  with probability at least  $1 - 1/n$  still requires  $\Omega(k \log n \log M)$  bits of space if  $\alpha = O((n/k)(\log \log n)^2 / (\log n)^2)$ .

We  
tion 15.

Suppose that there is a path-independent one-pass streaming algorithm  $\mathcal{A}$  which can solve the  $(\frac{1}{k}, \alpha)$ -heavy hitters problem with probability  $1 - O(1/n)$ , where  $\alpha = O(n/k \cdot (\log \log n / \log n)^2)$ . Then we can use it to solve the Aug-DISJ( $r, s, \delta$ ) problem for  $r = \log(M/n^{1/2})$ ,  $s = \Theta(\sqrt{n/k})$ ,  $\delta = 1/n$ , from which a space lower bound of  $\Omega(k \log n \log M)$  bits follows if  $M = \Omega(n^{1/2+O(1)})$ .

We design the following protocol  $\pi$  between the players and referee. For each  $i \in [s]$ , player  $i$  has the  $r$  instances  $(\mathbf{X}_i^1, \mathbf{X}_i^2, \dots, \mathbf{X}_i^r)$ . Player  $i$  then performs the update  $10^{j-1} \cdot \mathbf{X}_i^j$  to the algorithm  $\mathcal{A}$  for each  $j \in [r]$  and sends the memory of  $\mathcal{A}$  to the referee. Under the path-independence assumption, the referee can determine an equivalent frequency vector (i.e., leading to the same state of the algorithm) from each player and then add up the corresponding updates. After receiving  $T$  and  $\{(\mathbf{X}_1^t, \dots, \mathbf{X}_s^t)\}_{t=T+1}^r$ , the referee performs the update  $-10^{j-1} \cdot (\sum_{i=1}^s \mathbf{X}_i^j)$  to the algorithm  $\mathcal{A}$  for each  $j \geq T+1$ . Suppose that  $\mathcal{A}$  outputs a set  $S$ . The referee will output YES if  $|S| = 1$  and NO if  $S = \emptyset$ .

Now we analyze the correctness of the above protocol  $\pi$ . We recall that the referee needs to output the answer to the  $T$ -th instance. For simplicity, we define  $\mathbf{Y}^j = \sum_{i=1}^s \mathbf{X}_i^j$  for the  $j$ -th instance. Recall that after taking a union bound, for every instance  $j$ ,  $\|\mathbf{Y}^j\|_\infty \leq c \log n / \log \log n$  if it is a NO instance. Also from a Chernoff bound, it is easy to see that  $\|\mathbf{Y}^j\|_2^2 = \Omega(n)$  for all  $j$  with probability at least  $1 - e^{-\Omega(n)}$ . Note that the actual underlying vector that algorithm  $\mathcal{A}$  maintains has the same output as the frequency vector  $\mathbf{Y} = \sum_{t=1}^T 10^{t-1} \mathbf{Y}^t$  after the referee performs the updates. We need the following concentration bounds, which are a special case of Lemma 17 with  $p = 2$ .

► **Lemma 20** ([51], special case of Claim 6.2). *It holds that*

$$\mathbb{E} \left[ \|\mathbf{Y}_{-I}\|_2^2 \right] \leq K_1 n \cdot 10^{2T}, \quad (4)$$

$$\sigma_\ell \left( \|\mathbf{Y}_{-I}\|_2^2 \right) \leq K_2 \frac{\ell}{\ln \ell} \max\{4\sqrt{n}, \ell^2 n^{1/\ell}\} \cdot 10^{2T}, \quad \forall \ell \geq 2, \quad (5)$$

where  $K_1, K_2 > 0$  are absolute constants.

Taking  $\ell = 3 \ln n$  in (5) gives that

$$\begin{aligned} & \Pr \left[ \left| \|\mathbf{Y}_{-I}\|_2^2 - \mathbb{E} \left[ \|\mathbf{Y}_{-I}\|_2^2 \right] \right| > 0.1n \cdot 10^{2T} \right] \\ & \leq \Pr \left[ \left| \|\mathbf{Y}_{-I}\|_2^2 - \mathbb{E} \left[ \|\mathbf{Y}_{-I}\|_2^2 \right] \right| > 2\sigma_\ell \left( \|\mathbf{Y}_{-I}\|_2^2 \right) \right] \\ & \leq 2^{-\ell} \leq 1/n^2. \end{aligned} \quad (6)$$

Condition on all of the events above occurring. In all cases, the value  $\|x\|_\infty$  of the underlying vector  $x$  that algorithm  $\mathcal{A}$  maintains is less than  $\left( \sum_{i=1}^{r-1} 10^{i-1} \cdot \frac{\log n}{\log \log n} + 10^{r-1} \cdot \sqrt{n/k} \right) < 10^r \cdot \sqrt{n} = O(M)$  for  $r = \log(M/n^{1/2})$ .

We first consider the case in which the  $T$ -th instance is a YES instance. In this case,  $\mathbf{Y}_I^T = s$ , and thus

$$\mathbf{Y}_I \geq 10^{T-1} \cdot s.$$

## 13:14 Streaming Algorithms with Large Approximation Factors

Meanwhile, for all  $j \neq I$ ,

$$\mathbf{Y}_j \leq c \sum_{t=1}^T 10^{t-1} \frac{\log n}{\log \log n} < c \cdot 10^T \frac{\log n}{\log \log n}. \quad (7)$$

It follows from (4) and (6) that

$$\Omega(10^{2T} \cdot n) \leq \|\mathbf{Y}\|_2^2 = \|\mathbf{Y}_{-I}\|_2^2 + \mathbf{Y}_I^2 \leq (K_1 + 0.1)n \cdot 10^{2T} + s^2.$$

It thus holds that  $\mathbf{Y}_I^2 \geq (1/k)\|\mathbf{Y}\|_2^2$ , or equivalently,  $s^2/100 \geq s^2/k + (K_1 + 0.1)n/k$  when  $k > 100$  and  $s = \Omega(\sqrt{n/k})$ . Furthermore, for  $j \neq I$ ,  $\mathbf{Y}_j^2 \leq \|\mathbf{Y}\|_2^2/(\alpha k)$  when  $\alpha = O((n/k)(\log \log n / \log n)^2)$ . Therefore, our choices of  $k$ ,  $s$  and  $\alpha$  imply that the set  $S = \{I\}$ .

Now we consider the case when the  $T$ -th instance is a NO instance. In this case, (7) holds for all  $j \in [n]$ . Since  $\|\mathbf{Y}\|_2^2 \geq \Omega(10^{2T} \cdot n)$ , it follows that  $\mathbf{Y}_j^2 \leq \|\mathbf{Y}\|_2^2/(\alpha k)$  for all  $j$ , provided that  $\alpha = O((n/k)(\log \log n / \log n)^2)$ . It follows that  $S = \emptyset$ .

To conclude, we have proved the following theorem.

► **Theorem 21.** *Suppose that  $k = \Omega(1)$ ,  $\alpha = O(\frac{n}{k}(\frac{\log \log n}{\log n})^2)$  and  $M = \Omega(n^{1/2+O(1)})$ . Then, any one-pass streaming algorithm that solves the  $(1/k, \alpha)$ -heavy hitters problem with failure probability  $O(1/n)$  requires  $\Omega(k \log n \log M)$  bits of space, where the algorithm can store any number of random bits.*

**Sketching dimension lower bound.** One limitation of the above theorem is that it requires the algorithm  $\mathcal{A}$  to succeed with high probability. Below we show that any algorithm  $\mathcal{A}$  using a linear sketch to solve the  $(1/k, \alpha)$ -heavy hitters problem with constant probability requires the sketching dimension to be  $O(k \log(n/k))$  if  $\alpha = O(n/(k \log n))$ .

We will consider the following communication game in [45]. Let  $\mathcal{F} \subset \{S \subset [n] \mid |S| = k/2\}$  be a family of  $k$ -sparse supports such that:

- $|S \Delta S'| \geq k$  for  $S \neq S' \in \mathcal{F}$ ,
- $\Pr_{S \in \mathcal{F}}[i \in S] = k/(2n)$  for all  $i \in [n]$ , and
- $\log |\mathcal{F}| = \Omega(k \log(n/k))$ .

Let  $X = \{x \in \{0, \pm 2\sqrt{n/k}\}^n \mid \text{supp}(x) \in \mathcal{F}\}$ . Let  $w \sim \mathcal{N}(0, I_n)$ . Consider the following process. First, Alice chooses  $S \in \mathcal{F}$  uniformly at random. Then  $x \in X$  is uniformly at random subject to  $\text{supp}(x) = S$ , and then  $w \sim \mathcal{N}(0, I_n)$ . Then, Alice computes  $y = Az = A(x + w)$ , where  $A \in \mathbb{R}^{m \times n}$  is the sketching matrix in  $\mathcal{A}$ , and Alice sends  $y$  to Bob. Then Bob needs to recover  $S$  from  $y$ .

► **Theorem 22** ([45]). *Suppose that Bob can recover  $S$  with probability at least  $2/3$ . Then  $m = \Omega(k \log(n/k))$ .*

Next we will show that Alice and Bob can use a  $(\frac{1}{k}, \alpha)$ -heavy hitters algorithm to solve the communication game above if  $\alpha = O(n/(k \log n))$ . To show correctness, we need the following bounds for  $w$ .

► **Lemma 23** (folklore). *Suppose that  $w \sim \mathcal{N}(0, I_n)$ . Then with probability  $9/10$  we have the following:*

- (i)  $0.9n \leq \|w\|_2^2 \leq 1.1n$ ;
- (ii)  $\|w\|_\infty \leq c \cdot \sqrt{\log n}$ .



Condition on the events above. For each  $i \in [n]$ , we have

$$\begin{aligned} z_i &\geq 2\sqrt{n/k} - c\sqrt{\log n} \geq 1.9\sqrt{n/k}, & i \in S, \\ z_i &\leq c\sqrt{\log n}, & i \notin S. \end{aligned}$$

We also have from Lemma 23 that

$$0.9n < \|w\|_2^2 < \|z\|_2^2 \leq \|w\|_2^2 + \|x\|_2^2 + 4ck\sqrt{\log n}\sqrt{n/k} < 4n.$$

It follows that any  $(\frac{1}{k}, \alpha)$ -heavy set  $T$  will exactly be the support set  $S$  if  $\alpha = O(n/(k \log n))$ . The following theorem is immediate.

► **Theorem 24.** *Suppose that  $\alpha = O(n/(k \log n))$ . Then, any linear sketching algorithm that solves the  $(1/k, \alpha)$ -heavy hitters problem with constant probability requires a sketching dimension of  $\Omega(k \log(n/k))$ .*

## 6 Additional Results

We present improved one-pass and multipass algorithms for the  $\ell_0$  estimation problem in Section B.1, our two-pass algorithm for the  $F_p$  estimation ( $0 < p \leq 2$ ) problem in Section C, our results for Schatten- $p$  norm estimation in Section D, and finally, our results for cascaded norms and rectangle-efficient  $F_p$  estimation in Section E and Section F, respectively. All proofs are omitted and can be found in the full version of this paper.

---

### References

- 1 Swarup Acharya, Phillip B. Gibbons, Viswanath Poosala, and Sridhar Ramaswamy. The aqua approximate query answering system. In Alex Delis, Christos Faloutsos, and Shahram Ghandeharizadeh, editors, *SIGMOD 1999, Proceedings ACM SIGMOD International Conference on Management of Data*, pages 574–576, June 1-3, 1999, Philadelphia, Pennsylvania, USA, 1999. ACM Press.
- 2 Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proc. 20th International Conference on Very Large Data Bases*, pages 487–499, 1994.
- 3 Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *J. Comput. Syst. Sci.*, 58(1):137–147, 1999. doi:10.1006/jcss.1997.1545.
- 4 Alexandr Andoni. High frequency moments via max-stability. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017*, pages 6364–6368. IEEE, 2017. doi:10.1109/ICASSP.2017.7953381.
- 5 Alexandr Andoni, Khanh Do Ba, Piotr Indyk, and David P. Woodruff. Efficient sketches for earth-mover distance, with applications. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*, pages 324–330. IEEE Computer Society, 2009. doi:10.1109/FOCS.2009.25.
- 6 Alexandr Andoni, Robert Krauthgamer, and Krzysztof Onak. Streaming algorithms via precision sampling. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 363–372. IEEE, 2011.
- 7 Sepehr Assadi, Sanjeev Khanna, and Yang Li. On estimating maximum matching size in graph streams. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1723–1742. SIAM, 2017.

- 8 Khanh Do Ba, Piotr Indyk, Eric Price, and David P. Woodruff. Lower bounds for sparse recovery. In Moses Charikar, editor, *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 1190–1197. SIAM, 2010. doi:10.1137/1.9781611973075.95.
- 9 Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *J. Comput. Syst. Sci.*, 68(4):702–732, 2004.
- 10 Ziv Bar-Yossef, Thathachar S Jayram, Robert Krauthgamer, and Ravi Kumar. The sketching complexity of pattern matching. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 261–272. Springer, 2004.
- 11 Kevin S. Beyer and Raghuram Ramakrishnan. Bottom-up computation of sparse and iceberg CUBEs. In *Proc. ACM SIGMOD International Conference on Management of Data*, pages 359–370, 1999.
- 12 Mark Braverman, Sumegha Garg, and David P Woodruff. The coin problem with applications to data streams. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 318–329. IEEE, 2020.
- 13 Mark Braverman, Sumegha Garg, and Or Zamir. Tight space complexity of the coin problem. *Electron. Colloquium Comput. Complex.*, 2021. URL: <https://eccc.weizmann.ac.il/report/2021/083>.
- 14 Vladimir Braverman, Moses Charikar, William Kuszmaul, David P. Woodruff, and Lin F. Yang. The one-way communication complexity of dynamic time warping distance. In Gill Barequet and Yusu Wang, editors, *35th International Symposium on Computational Geometry, SoCG 2019, June 18-21, 2019, Portland, Oregon, USA*, volume 129 of *LIPICs*, pages 16:1–16:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.
- 15 Vladimir Braverman, Stephen R. Chestnut, Robert Krauthgamer, Yi Li, David P. Woodruff, and Lin F. Yang. Matrix norms in data streams: Faster, multi-pass and row-order. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 648–657, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, 2018. PMLR.
- 16 Vladimir Braverman, Robert Krauthgamer, Aditya Krishnan, and Roi Sinoff. Schatten norms in matrix streams: Hello sparsity, goodbye dimension. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020*, volume 119 of *Proceedings of Machine Learning Research*, pages 1100–1110, Virtual Event, 2020. PMLR.
- 17 Paul Brown, Peter Haas, Jussi Myllymaki, Hamid Pirahesh, Berthold Reinwald, and Yannis Sismanis. Toward automated large-scale information integration and discovery. In *Data Management in a Connected World*, pages 161–180. Springer, 2005.
- 18 Amit Chakrabarti and Sagar Kale. Strong fooling sets for multi-player communication with applications to deterministic estimation of stream statistics. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 41–50. IEEE, 2016.
- 19 Graham Cormode and S Muthukrishnan. Space efficient mining of multigraph streams. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 271–282, 2005.
- 20 Tamraparni Dasu, Theodore Johnson, Shanmugauelayut Muthukrishnan, and Vladislav Shkapenyuk. Mining database structure; or, how to build a data quality browser. In *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, pages 240–251, 2002.
- 21 David J DeWitt, Jeffrey F Naughton, Donovan A Schneider, and Srinivasan Seshadri. Practical skew handling in parallel joins. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 1992.
- 22 Elbert Du, Michael Mitzenmacher, David P. Woodruff, and Guang Yang. Separating k-Player from t-Player One-Way Communication, with Applications to Data Streams. *arXiv e-prints*, page arXiv:1905.07135, May 2019. arXiv:1905.07135.

- 23 Min Fang, Narayanan Shivakumar, Hector Garcia-Molina, Rajeev Motwani, and Jeffrey D. Ullman. Computing iceberg queries efficiently. In *Proc. 24rd International Conference on Very Large Data Bases*, pages 299–310, 1998.
- 24 Schkolnick Finkelstein, Mario Schkolnick, and Paolo Tiberio. Physical database design for relational databases. *ACM Transactions on Database Systems (TODS)*, 13(1):91–128, 1988.
- 25 Sumit Ganguly and David P. Woodruff. High probability frequency moment sketches. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, volume 107 of *LIPICs*, pages 58:1–58:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.
- 26 Jiawei Han, Jian Pei, Guozhu Dong, and Ke Wang. Efficient computation of iceberg cubes with complex measures. In *Proc. 2001 ACM SIGMOD International Conference on Management of Data.*, pages 1–12, 2001.
- 27 Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. In *Proc. 2000 ACM SIGMOD International Conference on Management of Data*, pages 1–12, 2000.
- 28 Christian Hidber. Online association rule mining. In *SIGMOD 1999, Proc. ACM SIGMOD International Conference on Management of Data*, pages 145–156, 1999.
- 29 Piotr Indyk and Ali Vakilian. Tight trade-offs for the maximum k-coverage problem in the general streaming model. In Dan Suciu, Sebastian Skritek, and Christoph Koch, editors, *Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019*, pages 200–217. ACM, 2019.
- 30 Rajesh Jayaram and David P. Woodruff. Data streams with bounded deletions. *CoRR*, abs/1803.08777, 2018. [arXiv:1803.08777](https://arxiv.org/abs/1803.08777).
- 31 T. S. Jayram and David P. Woodruff. The data stream space complexity of cascaded norms. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*, pages 765–774. IEEE Computer Society, 2009. [doi:10.1109/FOCS.2009.82](https://doi.org/10.1109/FOCS.2009.82).
- 32 Tanqiu Jiang, Yi Li, Honghao Lin, Yisong Ruan, and David P Woodruff. Learning-augmented data stream algorithms. In *International Conference on Learning Representations*, 2019.
- 33 Hossein Jowhari, Mert Saglam, and Gábor Tardos. Tight bounds for  $L_p$  samplers, finding duplicates in streams, and related problems. In Maurizio Lenzerini and Thomas Schwentick, editors, *Proceedings of the 30th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2011, June 12-16, 2011, Athens, Greece*, pages 49–58. ACM, 2011. [doi:10.1145/1989284.1989289](https://doi.org/10.1145/1989284.1989289).
- 34 John Kallaugher and Eric Price. Separations and equivalences between turnstile streaming and linear sketching. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 1223–1236. ACM, 2020.
- 35 Daniel M. Kane, Jelani Nelson, and David P. Woodruff. On the exact space complexity of sketching and streaming small norms. In Moses Charikar, editor, *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 1161–1178. SIAM, 2010.
- 36 Daniel M. Kane, Jelani Nelson, and David P. Woodruff. An optimal algorithm for the distinct elements problem. In Jan Paredaens and Dirk Van Gucht, editors, *Proceedings of the Twenty-Ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2010, June 6-11, 2010, Indianapolis, Indiana, USA*, pages 41–52. ACM, 2010. [doi:10.1145/1807085.1807094](https://doi.org/10.1145/1807085.1807094).
- 37 Aditya Krishnan, Sidhanth Mohanty, and David P. Woodruff. On sketching the  $q$  to  $p$  norms. *CoRR*, abs/1806.06429, 2018. [arXiv:1806.06429](https://arxiv.org/abs/1806.06429).

- 38 Rafał Latała. Estimation of moments of sums of independent real random variables. *The Annals of Probability*, 25(3):1502–1513, 1997. doi:10.1214/aop/1024404522.
- 39 Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: Isoperimetry and Processes*. Springer, 1991.
- 40 Yi Li, Huy L. Nguyen, and David P. Woodruff. Turnstile streaming algorithms might as well be linear sketches. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 174–183. ACM, 2014.
- 41 Yi Li, Huy L. Nguyen, and David P. Woodruff. On approximating matrix norms in data streams. *SIAM J. Comput.*, 48(6):1643–1697, 2019. doi:10.1137/17M1152255.
- 42 Yi Li and David P. Woodruff. Tight bounds for sketching the operator norm, Schatten norms, and subspace embeddings. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.
- 43 Yi Li and David P Woodruff. Embeddings of Schatten norms with applications to data streams. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *Proceedings of ICALP*, volume 80 of *LIPICs*, pages 60:1–60:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPICs.ICALP.2017.60.
- 44 Sriram Padmanabhan, Bishwaranjan Bhattacharjee, Tim Malkemus, Leslie Cranston, and Matthew Huras. Multi-dimensional clustering: A new data layout scheme in db2. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 637–641, 2003.
- 45 Eric Price and David P. Woodruff.  $(1 + \epsilon)$ -approximate sparse recovery. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 295–304. IEEE Computer Society, 2011. doi:10.1109/FOCS.2011.92.
- 46 Ashok Savasere, Edward Omiecinski, and Shamkant B. Navathe. An efficient algorithm for mining association rules in large databases. In *Proc. 21th International Conference on Very Large Data Bases*, pages 432–444, 1995.
- 47 P Griffiths Selinger, Morton M Astrahan, Donald D Chamberlin, Raymond A Lorie, and Thomas G Price. Access path selection in a relational database management system. In *Readings in Artificial Intelligence and Databases*, pages 511–522. Elsevier, 1989.
- 48 Amit Shukla, Prasad Deshpande, Jeffrey F Naughton, and Karthikeyan Ramasamy. Storage estimation for multidimensional aggregates in the presence of hierarchies. In *VLDB*, volume 96, pages 522–531. Citeseer, 1996.
- 49 Srikanta Tirthapura and David Woodruff. Rectangle-efficient aggregation in spatial data streams. In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGAI symposium on Principles of Database Systems*, pages 283–294. ACM, 2012.
- 50 Hannu Toivonen. Sampling large databases for association rules. In *Proc. 22th International Conference on Very Large Data Bases*, pages 134–145, 1996.
- 51 Omri Weinstein and David P. Woodruff. The simultaneous communication of disjointness with applications to data streams. In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, volume 9134 of *Lecture Notes in Computer Science*, pages 1082–1093. Springer, 2015.
- 52 David P. Woodruff and Guang Yang. Separating  $k$ -player from  $t$ -player one-way communication, with applications to data streams. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPICs*, pages 97:1–97:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

## A Proof of Lemma 17

The first result, Equation (1), was proved in [51]. Now we prove the second result.

By a standard symmetrization technique (see, e.g., [39, p153]),

$$\left(\mathbb{E} \left\| \mathbf{Y}_{-I} \|_p^p - \mathbb{E} \left[ \left\| \mathbf{Y}_{-I} \|_p^p \right\|^r \right] \right\|^{\frac{1}{r}} = \left( \mathbb{E} \left| \sum_{i \neq I} (\mathbf{Y}_i^p - \mathbb{E} \mathbf{Y}_i^p) \right|^r \right)^{\frac{1}{r}} \leq 2 \left( \mathbb{E} \left| \sum_{i \neq I} \varepsilon_i \mathbf{Y}_i^p \right|^r \right)^{\frac{1}{r}}, \quad (8)$$

multiline where the  $\varepsilon_i$  are independent Rademacher variables.

By Latała's inequality ([38, Corollary 3]), it holds for  $r \geq 2$  that

$$\left( \mathbb{E} \left| \sum_{i \neq I} \varepsilon_i \mathbf{Y}_i^p \right|^r \right)^{\frac{1}{r}} \leq K_1 \frac{r}{\ln r} \max \left\{ \left( \mathbb{E} \sum_{i \neq I} \mathbf{Y}_i^{2p} \right)^{\frac{1}{2}}, \left( \mathbb{E} \sum_{i \neq I} \mathbf{Y}_i^{rp} \right)^{\frac{1}{r}} \right\}, \quad (9)$$

where  $K_1 > 0$  is an absolute constant.

It was shown in [51, Lemma 6.3] that

$$\mathbb{E} \mathbf{Y}_i^p \leq K_2^p p^p 10^{Tp}, \quad p \geq 1,$$

for some absolute constant  $K_2 > 0$ . It then follows that

$$\left( \mathbb{E} \sum_{i \neq I} \mathbf{Y}_i^{rp} \right)^{\frac{1}{r}} \leq K_2^p (rp)^p n^{1/r} 10^{pT} \quad (10)$$

The result follows from combining (8), (9) and (10).

*Remark.* We note an omission in [51]. In that paper, the proof of the second result, i.e., Equation (5), assumes that the larger term in (9) is  $(\mathbb{E} \sum_{i \neq I} \mathbf{Y}_i^{2r})^{1/2}$ , which is not necessarily the case. Lemma 2.5 in that paper is also an incorrect citation from [38], since the conclusion should be  $\max\{\Delta_1(X), \Delta_\ell(X)\}$  for nonnegative variables  $X$ , but this would be too large for the proof. Hence we first symmetrize the variables, which allows for a better bound on  $\max\{\Delta_2(X), \Delta_\ell(X)\}$ .

## B $\ell_0$ Estimation

### B.1 One-pass Algorithm

We describe a randomized algorithm which gives an  $n^{1/t}$ -approximation with constant probability using  $O(t \log \log M)$  bits of space, excluding the storage for random bits. We assume that  $n^{1/t} \geq c_2$  for some constant  $c_2$ , otherwise an optimal algorithm is known [35].

The algorithm is presented in Algorithm 1. The idea behind the algorithm is to subsample the coordinates at  $t$  levels, with a geometrically decreasing sampling probability. In each level, the surviving coordinates are hashed into a constant number of buckets. If the  $\ell_0$  of the subvector (which is the vector of surviving coordinates) at a level is at most a constant, then only a small number of these buckets will be occupied. Otherwise, all the buckets will be occupied with high probability. Based on this, we design a criterion to determine the occupancy of these buckets to infer the  $\ell_0$  of the subvector at a level. Finally, we find the specific level  $J$  such that the  $\ell_0$  in level  $J$  is between 0 and at most  $n^{1/t}$ , and then it can shown that  $n^{J/t}$  is a good estimator to the overall  $\ell_0$ .

■ **Algorithm 1**  $n^{1/t}$ -approximator for  $\ell_0$ .

---

```

1 Initialize  $cKt$  counters  $C_{1,1,1}, \dots, C_{t,K,c}$  to 0;
2  $c_1 \leftarrow \beta\sqrt{c}$ ;
3 Initialize pairwise independent hash functions  $h : [n] \rightarrow [n], g : [n] \rightarrow [c]$ ;
4 Initialize  $K$  4-wise independent hash functions  $s_i : [n] \rightarrow \{-1, 1\}$ ;
5 Pick a prime  $p \in \Theta(c^3 \log^2 M)$ ;
6 foreach  $(x, v)$  in the data stream do
7    $b \leftarrow$  the largest  $j$  such that  $h(x) \bmod \lfloor n^{1/t} \rfloor^j = 0$ ;
8   for  $i \leftarrow 1$  to  $b$  do
9     for  $j \leftarrow 1$  to  $K$  do
10       $C_{i,j,g(x)} \leftarrow (C_{i,j,g(x)} + v \cdot s_j(x)) \bmod p$ ;
11    end
12  end
13 end
14 if there exists  $j$  such that  $|\{k \mid \exists l, C_{j,l,k} \neq 0\}| > c_1$  then
15    $J \leftarrow$  the largest  $j$  such that  $|\{k \mid \exists l, C_{j,l,k} \neq 0\}| > c_1$ ;
16 else
17    $J \leftarrow 0$ ;
18 end
19 return  $c_2 n^{J/t}$ ;

```

---

► **Theorem 25.** *Algorithm 1 outputs  $Z$ , which with probability at least 0.9 satisfies that  $\ell_0/n^{1/t} \leq Z \leq L_0 n^{1/t}$ . Furthermore, Algorithm 1 uses  $O(t \log \log M)$  bits of space, excluding its random tape.*

► **Remark 26.** Algorithm 1 uses  $O(\log n)$  random bits since the hash functions  $h, g$  are pairwise independent and the  $s_i$  are 4-wise independent.

## B.2 Lower Bound

We now prove a space lower bound of  $\Omega(t)$  bits for estimating  $\ell_0$  up to an  $n^{1/t}$ -approximation factor. Our lower bound holds even if the algorithm has access to an arbitrarily long random tape, which we do not charge for in its space. We reduce the  $\ell_0$  estimation problem to the Augmented Indexing communication problem, in the one-way public coin model, which we now define. We assume that  $t = O(\log n)$ .

► **Definition 27 (Augmented Indexing).** *Alice has a string  $u \in \{0, 1\}^l$ , Bob has an index  $i^* \in [l]$  and  $u_{i^*+1}, \dots, u_l$ . Alice is allowed to send a single message to Bob, and Bob wants to learn  $u_{i^*}$  from Alice with probability at least  $2/3$ .*

► **Lemma 28 ([10]).** *The one-way communication complexity of Augmented Indexing is  $\Omega(l)$  in the public coin model.*

Assume we have a streaming algorithm  $\mathcal{A}$ . Alice runs  $\mathcal{A}$  on her stream  $s(a)$ , then sends the state of  $\mathcal{A}$  to Bob. Bob feeds his stream  $s(b)$  into  $\mathcal{A}$  and obtains an estimate of  $\ell_0$ . We show how to design  $s(a)$  and  $s(b)$  so that Bob can solve the Augmented Indexing problem.

Without loss of generality, we assume that  $t$  is divisible by 8. Let  $u$  be the vector in an instance of the Augmented Indexing problem with  $l = t/8$ . We shall create an input vector  $x$  for the  $\ell_0$  estimation problem.



► **Theorem 29.** *Estimating  $\ell_0$  with approximation factor  $n^{1/t}$  requires  $\Omega(t)$  bits, even if the algorithm has an arbitrarily long random tape.*

### B.3 Multi-pass $\ell_0$ Estimation

We state our results for two-pass and three-pass algorithms below.

► **Theorem 30.** *There exists an absolute constant  $\varepsilon_0$  and a two-pass algorithm such that the following holds. For all  $\varepsilon \in (0, \varepsilon_0)$ , the algorithm outputs  $Z$  satisfying  $(1-\varepsilon)\ell_0 \leq Z \leq (1+\varepsilon)\ell_0$  with probability at least 0.8. The algorithm uses  $O(\log n + \varepsilon^{-2} \log \log M(\log(1/\varepsilon) + \log \log M))$  bits of space.*

► **Theorem 31.** *There exists an absolute constant  $\varepsilon_0$  and a three-pass algorithm such that the following holds. For all  $\varepsilon \in (0, \varepsilon_0)$ , the algorithm outputs  $Z$  satisfying  $(1-\varepsilon)\ell_0 \leq Z \leq (1+\varepsilon)\ell_0$  with probability at least 0.75. Furthermore, the algorithm uses  $O(\log n + \varepsilon^{-2}(\log(1/\varepsilon) + \log \log M))$  bits of space.*

### C Two-Pass Algorithm for $F_p$ ( $0 < p \leq 2$ )

As we have shown in the previous section, for the  $\|x\|_p^p$  estimation problem, even a large approximation also requires  $\Omega(\log n)$  bits of space. In this section, we will show that after obtaining a constant approximation to  $\|x\|_p^p$  in the first pass using  $O(\log n)$  bits, we can obtain a  $(1 \pm \varepsilon)$ -approximation to  $\|x\|_p^p$  using  $O(\log n + \varepsilon^{-2}(\log M + \log \frac{1}{\varepsilon}))$  bits. This is better than the previous  $O(\varepsilon^{-2} \log nM)$  space bound in one-pass if  $M$  is small.

► **Theorem 32.** *Suppose that  $0 < p \leq 2$ . There is a two-pass streaming algorithm which can be implemented in  $O(\log n + \varepsilon^{-2}(\log M + \log \frac{1}{\varepsilon}))$  bits of space and which outputs a  $(1 \pm \varepsilon)$ -approximation to  $\|x\|_p^p$  with probability at least  $9/10$ .*

### D Schatten- $p$ Norm Estimation

In this section, we consider approximating the Schatten- $p$  norm  $\|A\|_p$  of a given matrix  $A$  with large approximation factor  $\alpha$ , where  $\sigma_i(A)$  is the  $i$ -th singular value of  $A$ . We assume  $A \in \mathbb{R}^{n \times n}$  here because for a general matrix  $A \in \mathbb{R}^{n \times d}$ , we can first apply a subspace embedding to the left or to the right of  $A$  to preserve each of its singular values up to a constant factor and then pad with zero rows or columns (see, e.g., Appendix C of [40] for the details of this argument). As in the majority of previous work on Schatten norm estimation, we focus on the sketching dimension complexity.

**Upper Bound.** We will show that for an even integer  $p$  and an arbitrary  $\alpha = \Omega(1)$ , there is an  $O(n^{2-4/p}/\alpha^4)$  dimension sketching algorithm, while for  $p$  not an even integer, the  $O(n^{2-4/p}/\alpha^4)$  dimension bound still holds if  $\alpha$  is not too small. Our algorithm is based on a constant approximation algorithm for  $\|A\|_p$  when  $p$  is an even integer.

► **Lemma 33** (Theorem 8.2, [41]). *Suppose that  $p$  is an even integer. There is a sketching algorithm whose output  $Z$  satisfies  $\|A\|_p \leq Z \leq 2\|A\|_p$  with probability at least  $2/3$ . Furthermore, the sketching dimension of this algorithm is  $O(n^{2-4/p})$ .*

Our algorithm is given in Algorithm 2. For an even integer  $p$ , we maintain the matrix  $GAH^T$  where  $G$  and  $H$  are defined in algorithm 2 and use the constant approximation algorithm  $\mathcal{A}_q$  to estimate the Schatten- $q$  norm of  $GAH^T$ . The following lemma shows that  $\|GA\|_q$  can be an  $\alpha$ -approximation to  $\|A\|_p$ .

■ **Algorithm 2**  $\alpha$ -approximation for  $\|A\|_p$ .

---

```

1 Set  $q = p$  if  $p \in 2\mathbb{Z}$ , or to be the largest even integer less than  $p$  otherwise;
2 Let  $\mathcal{A}_q$  be a streaming algorithm that can output a constant-factor approximation to
    $\|A\|_q$ ;
3 Set  $t = (n^{1/2-1/p}/\alpha)^{1/(1/2-1/q)}$ ;
4 Let  $G$  be an  $r = t \text{ poly}(\log(n/t)) \times n$  matrix with i.i.d  $\mathcal{N}(0, 1/r)$  entries (these are i.i.d.
   normal random variables with mean 0 and variance  $1/r$ ) and let  $H$  be an
   independent  $r = O(t) \times n$  matrix with i.i.d  $\mathcal{N}(0, 1/r)$  entries.
5 foreach  $\Delta_{i,j}$  in the data stream do
6   | Compute the matrix  $G\Delta_{i,j}H^T$ ;
7   | Add  $G\Delta_{i,j}H^T$  to the input stream for  $\mathcal{A}_q$ ;
8 end
9 Let  $Z$  be the output from  $\mathcal{A}_q$ ;
10 return  $Z$ ;
```

---

► **Lemma 34** (rewording of Theorem 22, [43]). *Suppose that  $p \geq q \geq 2$ ,  $q$  is an even integer, and  $t = O(n)$ . Let  $G$  be an  $r \times n$  matrix with i.i.d.  $\mathcal{N}(0, 1/r)$  entries, where  $r = O(t)$  when  $q = 2$  and  $r = O(t \log^{1/(1/2-1/q)}(n/t))$  when  $q \geq 4$ . Then, with probability at least  $1 - \exp(-c't)$ , we have  $\|A\|_p \leq \|\gamma GA\|_q \leq (n^{1/2-1/p})/(t^{1/2-1/q}) \|A\|_p$ , where  $\gamma$  is an appropriate scaling factor.*

If  $H$  is a  $(1/2)$ -subspace embedding of  $GA$ , then the singular values of  $GAH^T$  are different from those of  $GA$  by at most a constant factor (see Section 2), and thus  $\|GAH^T\|_q$  is a constant approximation to  $\|GA\|_q$ . Recall that our sketch is a matrix of dimensions  $r \times O(t)$ , where  $r = t \text{ poly}(\log t)$ , so the sketching dimension of our algorithm is  $\tilde{O}(t^{2-4/p}) = \tilde{O}(n^{2-4/p}/\alpha^4)$ .

► **Theorem 35.** *Suppose that  $p \geq 2$  is an even integer. Then there is a sketching algorithm whose output  $Z$  satisfies  $\|A\|_p \leq Z \leq \alpha \|A\|_p$  with probability at least  $2/3$ . Furthermore, the sketching dimension of this algorithm is  $\tilde{O}(n^{2-4/p}/\alpha^4)$ .*

When  $p$  is not an even integer (and could even be a non-integer), let  $q$  be the largest even integer that is smaller than  $p$ . Then our choice of  $t$  still satisfies that  $t = O(n)$  if  $\alpha = \Omega(n^{1/p-1/q})$ . Our arguments above continue to hold and we obtain the following theorem.

► **Theorem 36.** *Suppose that  $p \geq 2$  is not an even integer. Let  $q$  be the largest even integer less than  $p$  and  $\alpha = \Omega(n^{1/q-1/p})$ . Then there is a sketching algorithm whose output  $Z$  satisfies  $\|A\|_p \leq Z \leq \alpha \|A\|_p$  with probability at least  $2/3$ . Furthermore, the sketching dimension of this algorithm is  $\tilde{O}(n^{2-4/p}/\alpha^4)$ .*

**Lower Bound.** Below we show that our upper bound is optimal up to  $\text{polylog}(n)$  factors. In [41], the authors give the following  $n^2/\alpha^4$  lower bound for  $\alpha$ -approximating  $\|A\|_{\text{op}}$ .

► **Lemma 37** (Corollary 3.3, [41]). *Suppose that  $\alpha \geq 1 + c$  where  $c$  is an arbitrarily small constant. Then, any sketching algorithm estimating  $\|A\|_{\text{op}}$  within a factor  $\alpha$  with failure probability smaller than  $1/6$  requires sketching dimension  $n^2/\alpha^4$ .*

Since  $\|x\|_\infty \leq \|x\|_p \leq n^{1/p} \|x\|_\infty$ , an  $\alpha$ -approximation of  $\|A\|_p$  implies an  $\alpha n^{1/p}$  approximation to  $\|A\|_\infty = \sigma_1(A)$ . The following lower bound follows.

► **Theorem 38.** *Suppose that  $\alpha \geq 1 + c$ , where  $c > 0$  is an arbitrarily small constant. Then, any sketching algorithm estimating  $\|A\|_p$  within a factor  $\alpha$  with failure probability smaller than  $1/6$  requires sketching dimension  $O(n^{2-4/p}/\alpha^4)$ .*

## E Cascaded Norms

In this section, we consider approximating the cascaded  $(p, q)$ -norm of a matrix  $X$ , defined as  $\|X\|_{p,q} = (\sum_i (\sum_j |x_{ij}|^q)^{p/q})^{1/p}$ , for a large approximation factor  $\alpha$ , when  $p \geq 1$  and  $q > 2$ . We have the following upper bound and show it is tight up to  $\text{poly}(\log n)$  factors.

► **Theorem 39.** *Suppose that  $\alpha \geq 8$ . Then there is an algorithm whose output is  $Z$ , which satisfies that  $\|X\|_{p,q} \leq Z \leq \alpha \|X\|_{p,q}$  with probability at least  $2/3$ . Furthermore, the algorithm uses  $O(n^{1-2/p} d^{1-2/q} \cdot (pq \log n)^{O(1)}/\alpha^2)$  bits of space when  $p, q > 2$  and uses  $O(d^{1-2/q} \cdot (q \log n)^{O(1)}/\alpha^2)$  bits of space when  $1 \leq p < 2$  and  $q > 2$ .*

► **Theorem 40.** *For the case that  $p, q > 2$ , any one-pass streaming algorithm which outputs a number  $Z$  such that  $\|X\|_{p,q} \leq Z \leq \alpha \|X\|_{p,q}$  with probability at least  $2/3$  requires  $\Omega(n^{1-2/p} d^{1-2/q}/\alpha^2)$  bits of space. For the case that  $1 \leq p < 2$  and  $q > 2$ , any one-pass streaming algorithm which outputs a  $Z$  such that  $\|X\|_{p,q} \leq Z \leq \alpha \|X\|_{p,q}$  with probability at least  $1 - \delta$  requires  $\Omega(d^{1-2/q} \log(M) \log(1/\delta)/\alpha^2)$  bits of space.*

## F Rectangle $F_p$ ( $p > 2$ )

In this section, we consider the rectangle  $F_p$  problem. A rectangle-efficient algorithm was proposed in [49]. Instead of updating the counter in each coordinate inside a rectangle, they develop a rectangle-efficient data structure called RECTANGLECOUNTSKETCH. We follow their notation that  $O^*(f)$  denotes a function of the form  $f \cdot \text{poly}(\log(mn/\delta))$  for constant rectangle dimension  $d$ .

► **Theorem 41.** *Suppose that  $p > 2$ . There is a rectangle-efficient one-pass streaming algorithm which outputs a number  $Z$  that is an  $\alpha$ -approximation to  $\|x\|_p^p$ , i.e.,  $\|x\|_p^p \leq Z \leq \alpha \|x\|_p^p$ , with probability at least  $1 - \delta$ . It uses  $O^*(n^{d(1-2/p)}/\alpha^{2/p})$  bits of space and  $O^*(n^{d(1-2/p)}/\alpha^{2/p})$  time to process each rectangle in the stream.*