THEORY OF COMPUTING, Volume 18 (20), 2022, pp. 1–32
www.theoryofcomputing.org

Universal Streaming of Subset Norms

Vladimir Braverman* Robert Krauthgamer† Lin F. Yang

Received March 9, 2020; Revised November 5, 2021; Published August 6, 2022

Abstract.

Most known algorithms in the streaming model of computation aim to approximate a single function such as an ℓ_p norm.

In 2009, Nelson [https://sublinear.info, Open Problem 30] asked if it is possible to design *universal algorithms*, that simultaneously approximate multiple functions of the stream. In this paper we answer the question of Nelson for the class of *subset-ℓ_0 norms* in the insertion-only frequency-vector model. Given a family of subsets, $\mathcal{S} \subset 2^{[n]}$, we provide a single streaming algorithm that can $(1 \pm \varepsilon)$-approximate the subset-ℓ_p norm for every $S \in \mathcal{S}$. Here, the subset-ℓ_p norm of $v \in \mathbb{R}^n$ with respect to the set $S \subseteq [n]$ is the ℓ_p norm of $v_{|S}$ (the vector v restricted to S by zeroing all other coordinates).

Our main result is a nearly tight characterization of the space complexity of the subset-ℓ_0 norm for every family $\mathcal{S} \subset 2^{[n]}$ in insertion-only streams, expressed in terms of the "heavy-hitter dimension" of $\mathcal{S}$, a new combinatorial quantity related to the VC-dimension of $\mathcal{S}$. We also show that the more general turnstile and sliding-window models require a much larger space usage. All these results easily extend to the ℓ_1 norm.

*This material is based upon work supported in part by the National Science Foundation under Grants No. 1447639, 1650041 and 1652257, Cisco faculty award, and by the ONR Award N00014-18-1-2364.

†Work partially supported by ONR Award N00014-18-1-2364, the Israel Science Foundation grant #1086/18, and a Minerva foundation grant from the Federal German Ministry for Education and Research.

ACM Classification: F.2.1

AMS Classification: 68Q25

Key words and phrases: universal streaming, subset norms

© 2022 Vladimir Braverman, Robert Krauthgamer, and Lin F. Yang
Licensed under a Creative Commons Attribution License (CC-BY)
DOI: 10.4086/toc.2022.v018.a020

In addition, we design algorithms for two other subset-ℓ_p variants. These can be compared to the famous Priority Sampling algorithm of Duffield, Lund and Thorup [JACM 2007], which achieves additive approximation $\varepsilon \|v\|_1$ for all possible subsets ($\mathcal{S} = 2^{[n]}$) in the entrywise update model. One of our algorithms extends their algorithm to handle turnstile updates, and another one achieves multiplicative approximation, given a family $\mathcal{S}$.

1 Introduction

The streaming model of computation, where a space-bounded algorithm makes only a single pass over an input stream, has gained popularity for its theoretical significance and usefulness in practice. Efficient streaming algorithms have been designed for many fundamental problems, including, for example, moments or norms of a frequency vector $v \in \mathbb{R}^n$ formed by a stream of additive updates [2, 31, 33]; clustering a stream of points in $\mathbb{R}^d$ [30]; and graph statistics for streams of edge updates [27].

Most algorithms designed for this model solve only a single problem. For instance, in the extensively studied area of streaming ℓ_p norms of a frequency vector, an algorithm usually makes a pass over the stream, and then it can use the summary it stores to compute only one particular norm—the one it was designed for. Designing a new algorithm for each statistic can be impractical in some applications. For example, in network monitoring, it is often desirable to maintain a single summary of the observed traffic and use this summary for multiple tasks such as approximating the entropy, finding elephant flows and heavy hitters, and detecting DDoS attacks [39, 45].

The importance of multi-functional summaries has been observed in the theory community as well. Nelson [42, Open Problem 30] asked in 2009 if it is possible to design *universal* algorithms for families of functions. More formally, given a family $\mathcal{F}$ of functions of the form $f : \mathbb{R}^n \to \mathbb{R}$, the goal is to compute, in one pass over a stream representing $v \in \mathbb{R}^n$, a single summary that can be used to evaluate $f(v)$ for every function $f \in \mathcal{F}$. Several algorithms [18, 19, 13, 11, 15] provide universal sketches for some families of functions, for example all symmetric norms in a certain class [11]. However, universal algorithms are an exception rather than the rule, and Nelson's question is still open in any reasonable generality.

A simple systematic method to generate a family $\mathcal{F}$ from a single function $f : \mathbb{R}^n \to \mathbb{R}$ is to apply this f to different subsets of coordinates. More precisely, for every subset $S \subset [n]$ define the function $f_S : v \mapsto f(v_{|S})$, where $v_{|S}$ denotes zeroing out the coordinates of v not in S. In this way, every set system $\mathcal{S} \subset 2^{[n]}$ describes a family of functions $\{f_S : S \in \mathcal{S}\}$. We focus on the basic case where f is an ℓ_p norm, and call such function families *subset-ℓ_p norms*. As usual, we wish to approximate each function *multiplicatively*, say within factor $1 \pm \varepsilon$ for a given $\varepsilon \in (0, 1)$; this is clearly stronger than approximating additively by $\varepsilon \|v\|_p$.

Subset-ℓ_p norms arise naturally in applications; for instance, $\mathcal{S}$ could represent supported queries to a database. Indeed, the well-known SUBSET SUM problem [1, 26, 48] and its variant, the DISAGGREGATED SUBSET SUM problem [24, 49], are equivalent to our subset-ℓ_1-norm problem in the entrywise and insertion-only models, respectively. Network-monitoring tasks, such as

worm detection, rely on subset-ℓ_p norms to approximate *flow* statistics [26]. Recall that a network flow is a subset of network traffic defined by a source address, a customer, an organization, an application or an arbitrary predicate. It is folklore that calculating flow volume is simply a subset-ℓ_1 query, and the number of distinct packets in a flow is a subset-ℓ_0 query. Recent work [29, 41] reiterated that approximating these subset-ℓ_p queries and more general filters is still an important open problem in network telemetry. In another recent example, Ting [49] argued that DISAGGREGATED SUBSET SUM is widely applicable in ad prediction, where future user behavior is inferred from historical aggregate queries that have the form of subset-ℓ_1 queries. In both examples, data collection is challenging since future queries can be arbitrary, and thus it is critical to answer large classes of subset-ℓ_1 queries. We refer the reader to [26, Section 1.4.1] and [49, Section 2.2] for detailed discussions of these and additional applications in machine learning [47], database query estimation [50] and denial of service attacks [44].

Our *main contribution* to universal streaming is a *near-tight* characterization, for every $\mathcal{S} \subset 2^{[n]}$, of the space complexity of subset-ℓ_0's in insertion-only streams. We stress that this problem asks to count the distinct items (non-zero coordinates of v) inside every subset $S \in \mathcal{S}$. Our characterization connects the space complexity for a set system $\mathcal{S}$ to a combinatorial notion that we call the *heavy-hitter dimension* of $\mathcal{S}$, which counts the maximum possible number of coordinates in a single $v \in \mathbb{R}^n$ that may be a "heavy hitter" for some $S \in \mathcal{S}$ (see Definition 2.1 for full details). This notion is related to VC-dimension by $\mathrm{VCdim}(\mathcal{S}) \le \mathrm{HHdim}(\mathcal{S})$, however the gap between the two is not bounded by any fixed factor. We in fact prove the above inequality and make use of it in Section 2.5. Throughout, $\widetilde{O}(\cdot)$ suppresses a $\mathrm{polylog}(n)$ factor, and $O_\varepsilon(\cdot)$ suppresses a factor depending only on ε; taken together, $\widetilde{O}_\varepsilon(f)$ stands for $O(g(\varepsilon)(\log^{O(1)} n)) \cdot f$ for some function g.

Theorem 1.1 (Informal Statement of Theorems 2.10 and 2.11)**.** *For every $\mathcal{S} \subset 2^{[n]}$ and $\varepsilon \in (0,1)$, there is a randomized universal algorithm for insertion-only streams, that makes one pass using $\widetilde{O}_\varepsilon(\mathrm{HHdim}(\mathcal{S}))$ words of storage, and can then $(1+\varepsilon)$-approximate each subset ℓ_0-norm from $\mathcal{S}$ with high probability. Moreover, every such algorithm requires $\Omega(\mathrm{HHdim}(\mathcal{S}))$ bits of storage.*

To illustrate the scope of this result, we present in Table 1 a few examples and properties of the heavy-hitter dimension (their proofs appear in Section 2.1). One interesting example is the family of all large intervals in $[n]$, namely, of size $\Omega(n)$, which has dimension $O(1)$. A second one is a family of $\mathrm{poly}(n)$-many uniformly-random sets (every set contains every index with probability $1/2$), which has dimension $O(\log n)$ with high probability. For both examples, our algorithm uses small space (polylogarithmic in n) to achieve multiplicative $(1+\varepsilon)$-approximation, which was not known before. (For intervals, additive approximation $\varepsilon \|v\|_1$ can be achieved by several known algorithms, including quantile estimates, range counting, and heavy-hitters over dyadic intervals.)

Another interesting example is a two-dimensional family derived from the last property in the table, as follows. Suppose each index $i \in [n]$ is actually a pair $(i_1, i_2) \in [n_1] \times [n_2]$, for instance the source and destination address of a packet. Let $\mathcal{S}_1$ be the family of all large intervals with respect to i_1, i. e., all $[a\,..\,b] \times [n_2]$ with $b - a = \Omega(n_1)$, and similarly $\mathcal{S}_2$ with respect to i_2, where throughout $[a\,..\,b] = \{a, a+1, \ldots, b\}$ denotes an interval of integers. Each of $\mathcal{S}_1, \mathcal{S}_2$

has heavy-hitter dimension $O(1)$, and by the bound in the table, also their union-product $\{S_1 \cup S_2 : S_1 \in \mathcal{S}_1, S_2 \in \mathcal{S}_2\}$. Our algorithm can then use small space (polylogarithmic in n) to estimate distinct elements in subsets of form $\{(i_1, i_2) : i_1 \in [a\,..\,b] \vee i_2 \in [c\,..\,d]\}$, capturing for instance the logical-or between a range of source addresses and a range of destination addresses.

Set System	HHdim	Description
$\{S_1, \ldots, S_k\}$	$\leq k$	any sets (tight for disjoint sets)
$\{S \subset [n] : \lvert S\rvert \geq n-k\}$	$k+1$	sets missing few coordinates
$\{[i\,..\,i'] : i'-i+1 \geq k\}$	$\Theta(n/k)$	intervals of size $\geq k$
$\{s_1, \ldots, s_k : \text{all } s_{i,j} \sim \mathcal{B}(p)\}$	whp $O(\log(nk)/p)$	k random subsets of density p
$\{S' \cup S'' : S', S'' \in \mathcal{S}\}$	$\mathrm{HHdim}(\mathcal{S})$	self union of $\mathcal{S}$
$\mathcal{S}_1 \cup \mathcal{S}_2$	$\leq \mathrm{HHdim}(\mathcal{S}_1) + \mathrm{HHdim}(\mathcal{S}_2)$	subadditivity
$\{S_1 \cup S_2 : S_1 \in \mathcal{S}_1, S_2 \in \mathcal{S}_2\}$	$\leq \mathrm{HHdim}(\mathcal{S}_1) + \mathrm{HHdim}(\mathcal{S}_2)$	union-product of $\mathcal{S}_1, \mathcal{S}_2$

Table 1: Simple examples and basic properties of heavy-hitter dimension over domain $[n]$.

Let us consider some natural extensions of the above theorem. First, our algorithm extends to subset-ℓ_1 as well, as shown in Theorem 2.22. Second, it is stated for insertion-only streams, however for the more general turnstile and sliding-window models (i. e., streams with both insertions and deletions or streams where old items expire), we show that the subset-ℓ_p problem, for any $p \geq 0$, requires space $\Omega(n)$ even if $\mathrm{HHdim}(\mathcal{S}) = O(1)$. This is striking because such a large separation between insertion-only and turnstile stream or sliding-window algorithms is rare.

Indeed, it may be instructive to see why the smooth-histograms technique of [17] fails in this case.

Theorem 1.2 (Informal Statement of Theorems 2.15 and 2.17)**.** *There exists $\mathcal{S} \subset 2^{[n]}$ with $\mathrm{HHdim}(\mathcal{S}) = O(1)$, such that every universal streaming algorithm achieving multiplicative approximation for subset-ℓ_p for $\mathcal{S}$ requires $\Omega(n)$ bits of space in both the turnstile and sliding-window models.*

Variants of the problem. Duffield, Lund and Thorup [26] consider a similar problem in the entrywise update model, in which each entry of the vector appears in the stream at most once. Their "subset-sum" problem is equivalent to our subset-ℓ_1 problem if all entries of the vector are non-negative.[1] They devise a Priority Sampling algorithm that approximates the subset-ℓ_1 norm of every subset $S \subset [n]$, achieving in fact an optimal space usage for this model [48]. However, their result actually guarantees an *additive approximation*, i. e., the error for every subset-ℓ_1 query is proportional to the ℓ_1 norm of the entire vector.[2] We additionally provide two extensions to their results in new directions.

[1]Non-negativity is a mild assumption in this entrywise update model, because one can easily separate the positive and negative entries and execute in parallel two algorithms.

[2]Indeed, Theorem 1 of [48] bounds the variance of the estimator by $\|v\|_1^2/(k-1)$, where k is number of samples being stored. This implies that with high probability, the estimator's additive error is at most $O(\|v\|_1/\sqrt{k-1})$.

The first extension is a full characterization of space complexity of *multiplicative approximation* of subset-ℓ_p's in the entrywise update model. In contrast to the results of [26, 48], once a multiplicative approximation is required, the space complexity depends on the query set system $\mathcal{S}$. Indeed, by modifying the priority sampling algorithm of [26, 48] and employing our lower bound, we show (in Theorem 2.23) that the space complexity is now precisely $\widetilde{\Theta}(\mathrm{HHdim}(\mathcal{S}))$.

Our second extension achieves the same *additive approximation*, but in the more general *turnstile model* (i. e., additive updates to entries). Similarly to the algorithm of [26, 48], our algorithm for the subset-ℓ_p problem achieves additive error $\varepsilon\|v\|_p$ with space complexity that does not depend on the query set system $\mathcal{S}$. This result is summarized in the following theorem; we note that a matching lower bound follows immediately from known results (e. g., [7]), as the case $S = [n]$ is the usual approximation of ℓ_p norm.

Theorem 1.3 (Informal Statement of Theorem 3.1)**.** *There exists a one-pass streaming algorithm that, given a stream of additive updates to a vector $v \in \mathbb{R}^n$, uses only $\widetilde{O}(1)$ words of space for $0 \le p \le 2$ (and $\widetilde{O}_\varepsilon(n^{1-2/p})$ words for $p > 2$), and can then approximate $\|v_{|S}\|_p$ within additive error $\varepsilon\|v\|_p$ for each $S \subset [n]$ with high probability.*

Summary. Our results are summarized in Table 2, where the first row lists the main results.

Problem	Update Model	Approximation	Space	Theorems
subset-ℓ_0 or ℓ_1	insertion-only	multiplicative	$\widetilde{\Theta}(\mathrm{HHdim}(\mathcal{S}))$	2.10, 2.11, 2.22
subset-ℓ_p	turnstile	multiplicative	$\Omega(n)$	2.15
	sliding-window		$\Omega(n)$	2.17
	entrywise		$\widetilde{\Theta}(\mathrm{HHdim}(\mathcal{S}))$	2.23
subset-ℓ_p	turnstile	additive	$\widetilde{\Theta}(1)$ for $0 \le p \le 2$ $\widetilde{\Theta}(n^{1-2/p})$ for $p > 2$	3.1

Table 2: Summary of our results

1.1 Related work

There is a large body of literature that deals with approximating functions of a vector, i. e., norms and heavy hitters, in the streaming model of computation. This line of work was started by Alon, Mathias, and Szegedy [2]; they gave a surprising logspace algorithm to approximate the ℓ_2 norm, and proved that space $n^{\Omega(1)}$ is required for integer values $p \ge 6$. [31] gives the first near-optimal algorithm (in terms of n) for ℓ_p for all $0 \le p \le 2$; and [33] gives the first near-optimal algorithm for ℓ_p for all $p > 2$; [32, 51, 21, 7] give tight lower bounds on this problem with respect to the approximation parameter ε and dimension n. There is a sequence of papers gradually improving the space complexity with respect to other parameters and studying variants of the problem. Due to the lack of space we mention only a small subset of the relevant papers:

[23, 21, 9, 37, 36, 35, 5, 3, 16, 14, 15]; and references therein. Most of these papers design methods that approximate a single function such as an ℓ_p norm for a fixed p.

Our setting is also related to the "subset sum" problem [1, 26, 48, 49] where one is interested in approximating a sum of the entries of a vector indexed by a subset. It is not difficult to see that our problem is the same as the objective in [26] when the input is restricted to non-negative vectors; indeed the subset-sum problem is equivalent to the subset-ℓ_1 problem. However, the model in [26] is slightly different. In [26] the algorithm sees each coordinate of the frequency vector at most once. In this paper we consider the additive updates streaming model that allows incremental updates to the coordinates of the frequency vector. Thus, our model generalizes the model in [26]. In addition, the algorithms in [26] solve the subset-sum problem but with an additive error.

1.2 Preliminaries

We identify a binary vector $s \in \{0,1\}^n$ as a subset in $[n]$. For two vectors $u, v \in \mathbb{R}^n$, we denote $u \circ v \in \mathbb{R}^n$ as the Hadamard product, i. e., each $(u \circ v)_i = u_i v_i$. We denote the support of v, $\mathrm{supp}(v) \subset [n]$, as the set of non-zero coordinates in v, i. e., $\mathrm{supp}(v) = \{i \in [n] : v_i \neq 0\}$. For each $p > 0$, we denote the ℓ_p norm of a vector $v \in \mathbb{R}^n$ as $\|v\|_p = (\sum_{i\in[n]} |v_i|^p)^{1/p}$. For $p = 0$, $\|v\|_0 := |\mathrm{supp}(v)|$ is the size of the support of v. For $p = \infty$, $\|v\|_\infty := \max_{i\in[n]} |v_i|$. Note that for $p < 1$, ℓ_p is not a "norm" but was called a norm by convention.

In this paper we are focusing on the updates of a vector $v \in \mathbb{R}^n$. In the *insertion-only* model, the input is a stream $\langle a_1, \ldots, a_m\rangle$, where each item $a_j \in [n]$ represents an increment to coordinate a_j of a vector $v \in \mathbb{R}^n$, which is initialized to all zeros. Thus the accumulated vector is $v = \sum_{j=1}^m e_{a_j}$, where $\{e_i : i \in [n]\}$ is the standard basis. Here m is usually assumed to be upper bounded by $\mathrm{poly}(n)$. In the *turnstile* model, the input is a stream $\langle (a_1, \Delta_1), \ldots, (a_m, \Delta_m)\rangle$, where each item $(a_j, \Delta_j) \in [n] \times \{-1, 1\}$ represents an increment to coordinate a_j of a vector $v \in \mathbb{R}^n$ by Δ_j.[3] Thus the accumulated vector is $v = \sum_{j=1}^m \Delta_j \cdot e_{a_j}$.

We are interested in the following problem.

Problem 1.4 (Subset-ℓ_p)**.** Let $\alpha \geq 1$ be a parameter. Given a set of binary vectors $\mathcal{S} \subset \{0,1\}^n$, design an algorithm that makes a single pass over a stream of updates to a vector $v \in \mathbb{R}^n$, and at the end of the stream the algorithm outputs a function $E : \mathcal{S} \to \mathbb{R}$ that satisfies

$$\forall s \in \mathcal{S}, \qquad \Pr[\|v \circ s\|_p \leq E(s) \leq \alpha \|v \circ s\|_p] \geq 0.9.$$

We call this problem the *α-approximation subset-ℓ_p* problem w.r.t. $\mathcal{S}$.

Note that the above definition requires the algorithm to approximate, for each given $s \in \mathcal{S}$, the quantity $\|v \circ s\|_p$ multiplicatively. A standard parallel repeating argument can lead to the "for-all" guarantee, i. e., the algorithm succeeds on approximating $\|v \circ s\|_p$ for all $s \in \mathcal{S}$. However, we pay an additional $\log |\mathcal{S}|$ factor in the space—this can be linear in n if $|\mathcal{S}|$ is large. It would be interesting if one can design a for-all algorithm with space not depending on $\log |\mathcal{S}|$.

[3] A more general model allows $\Delta_i \in \{-M, \ldots, M\}$ for some $M = \mathrm{poly}(n)$. The space and time requirement of the case $M = 1$ is the same up to an $O(\log n)$ factor. We use $M = 1$ for sake of presentation.

Note that the set system $\mathcal{S}$ is given to the algorithm via a read-only tape, hence the space of storing $\mathcal{S}$ is not counted. A variant of this problem is the additive approximation problem.

Problem 1.5 (Additive Subset-ℓ_p)**.** For a set of binary vectors $\mathcal{S} \subset \{0,1\}^n$, design a one-pass algorithm over a stream of updates to some underlying vector $v \in \mathbb{R}^n$ such that after one pass over the stream, the algorithm outputs a function $E : \mathcal{S} \to \mathbb{R}$ satisfies,

$$\forall s \in \mathcal{S}, \quad \Pr[\|v \circ s\|_p - E(s)\big| \leq \varepsilon \|v\|_p] \geq 0.9.$$

We call this problem the *ε-additive-approximation subset-ℓ_p* problem (w.r.t. $\mathcal{S}$).

1.3 Technical overview

Multiplicative Subset-ℓ_p algorithm for $p \in \{0,1\}$. Estimating ℓ_0 of a stream is a well-studied problem, for instance the first streaming algorithm was given in [28], and the problem's space complexity was settled in [2, 51, 37, 10, 34]. Let us first recall a classical sample-and-estimate technique for this problem (see, e. g., [8]). Here, the algorithm samples each coordinate of v with some probability q, and then uses the sampled non-zero coordinates to estimate $\|v\|_0$ (simply count their number and divide by q). Suppose we could guess the correct rate q, such that number of non-zero samples is about $\Theta(\varepsilon^{-2})$; then we would obtain a good estimate to the ℓ_0 of the stream, i. e., a $(1 \pm \varepsilon)$-approximation with constant probability. The number of guesses is at most $\Theta(\log n)$, since $\|v\|_0 \leq n$, and we can try all of them in parallel. Observe that this algorithm actually stores all distinct samples up to a point—when the samples for a guess q exceeds the $O(\varepsilon^{-2})$ space bound, the algorithm starts rejecting any extra samples.

Consider now approximating $\|v \circ s\|_0$ for any s in a known set system $\mathcal{S} \subset 2^{[n]}$. To use the above sample-and-estimate technique, the guess q should be chosen according to $\|v \circ s\|_0$. However, an algorithm that is not tailored to s will store (distinct) samples from all $\mathrm{supp}(v)$, and thus it might reach its $O(\varepsilon^{-2})$ space bound and start rejecting samples, without storing enough samples from $\mathrm{supp}(v \circ s) \subseteq \mathrm{supp}(v)$. The challenge is thus to store enough samples from $\mathrm{supp}(v \circ s)$ for every $s \in \mathcal{S}$. Our idea is to rely on the structure of the set system $\mathcal{S}$, and store every sample that might be necessary for any $s \in \mathcal{S}$, which clearly maintains the correctness (accuracy guarantee). However, this might require large space, perhaps even linear in $|\mathcal{S}|$, and our solution is to *actively delete* samples that are not necessary.

To formalize this idea, we let the algorithm store a set $\mathcal{H} \subset [n]$ of (distinct) samples from the stream. Now whenever the number of samples from some $s \circ v$ is smaller than our $O(\varepsilon^{-2})$ bound, all these samples are stored, and then $\mathcal{H}$ can always be used to estimate $\|s \circ v\|_0$. However, when some $i \in \mathcal{H}$ is no longer necessary for any $s \in \mathcal{S}$ (which might happen as new samples are stored), the algorithm deletes this i from $\mathcal{H}$. The question is then: what is the maximum possible size of $\mathcal{H}$? Luckily, we can show that $|\mathcal{H}| = O[\varepsilon^{-2} \cdot \mathrm{HHdim}(\mathcal{S})]$ via an inductive argument, whose base case is precisely the heavy-hitter dimension. Maintaining $\mathcal{H}$ in a streaming fashion is straightforward and requires only $O[\varepsilon^{-2} \cdot \mathrm{HHdim}(\mathcal{S})]$ words of space. Recalling there are $O(\log n)$ guesses for q, the algorithm actually stores $O(\log n)$ sets of samples, which altogether can simulate the sample-and-estimate algorithm for any $s \in \mathcal{S}$ given at the query phase, to achieve a multiplicative approximation of $v \circ s$. This result is presented in Theorem 2.10.

In insertion-only streams, the ℓ_1 norm is just the sum of all (relevant) stream updates. We can thus reduce the ℓ_1 estimation problem to ℓ_0 estimation of a new vector of dimension nm, where m is the length of the stream. We show that the converted set system has exactly the same heavy-hitter dimension, yielding again an algorithm with space usage $O[\varepsilon^{-2} \cdot \mathrm{HHdim}(\mathcal{S})]$. This result is presented in Theorem 2.22.

The upper bound for subset-ℓ_p norm in the entrywise update model follows similar ideas to store a small subset that is important to the set system. The only difference is that we use the priority sampling technique [26, 48] as the bottom-level algorithm. This result is presented in Theorem 2.23.

Lower bound for subset-ℓ_p. Our lower bound is via reduction from the well-known INDEX problem.[4] Suppose we have a set system with heavy-hitter dimension $\mathrm{HHdim}(\mathcal{S})$ (formally defined in Definition 2.1), we can then find a vector v with $\mathrm{HHdim}(\mathcal{S})$ non-zero coordinates and for each coordinate v_i, there exists an $s^i \in S$ such that $\{i\} = \mathrm{supp}(s^i \circ v)$. Therefore, we can encode an INDEX instance into the non-zero coordinates of v and by approximating $\|v \circ s^i\|_p$ multiplicatively for any p, we can have a protocol for the INDEX problem. This implies an $\Omega(\mathrm{HHdim}(\mathcal{S}))$ lower bound. This result is presented in Theorem 2.11.

Strong lower bound in the turnstile model and sliding-window model. It is striking that in the turnstile model or sliding-window model, there does not exist sublinear one-pass multiplicative approximation subset-ℓ_p algorithms even for a very simple set system. We show that for a simple set system, e. g., a set system containing all the intervals with size $n/2$, which has heavy-hitter dimension $O(1)$, any multiplicative approximation of subset-ℓ_p for any $p \geq 0$ requires $\Omega(n)$ space. We show this via a reduction from a communication problem called Augmented INDEX. In this problem, Alice has a binary vector $x \in \{0,1\}^n$, and Bob has an index $j \in [n]$ together with $x_{j+1}, x_{j+2}, \ldots, x_n$. Alice sends one message to Bob (only one round), who needs to determine x_j. It has been shown in [6] that, any constant-probability success protocol for this problem requires $\Omega(n)$ bits of communication. We construct a protocol using the subset-ℓ_p algorithm. Alice simply maps each of its coordinates of x to some stream updates. Bob removes all $x_{j'}$ for $j' > j$. Bob then picks the interval that contains at most one non-zero coordinate, x_j, and asks the algorithm to compute the ℓ_p norm. Hence any multiplicative approximation can be used to decide whether x_j is 0. Thus, any algorithm in the turnstile model requires $\Omega(n)$ space for this simple set system. Similar lower bounds can be shown for the sliding-window model with a lower bound of $\Omega(\min(W, n))$, where W is the window size. These results are formally presented in Theorem 2.15 and Theorem 2.17.

Additive approximation. Our additive approximation to the subset-ℓ_p norm follows a similar flavor of the priority sampling algorithm [26, 48]. We use the algorithmic ideas that appeared in [20] (similar ideas also appear earlier in [5] and [3], but of different form). To approximate

[4] INDEX is a two-party communication problem, in which Alice's input is a binary vector $x \in \{0,1\}^n$ and Bob's input is an index $j \in [n]$. Alice sends one message to Bob (only one round), who needs to determine the coordinate x_j. This problem requires $\Omega(n)$ bits of communication, see, e. g., [6] for details.

the ℓ_p norm of a vector, we first generate n pseudorandom numbers to scale each entry of the input vector v, which can be implemented using $\Theta(\log n)$ space in the streaming setting. If the distribution of the random numbers has a nice tail, e.g, $\Pr[X > x] = 1/x^p$, the ℓ_2-heavy hitter of the scaled vector can be shown to be a good estimation of the ℓ_p norm. The scaling is "oblivious" to the subset, i. e., for each $s \in 2^{[n]}$, the ℓ_2-heavy hitter of $s \circ v'$ is a good estimator to $\|s \circ v\|_p$, where v' is the scaled version v'. This result is presented in Theorem 3.1.

2 The streaming complexity of subset-ℓ_p norms

In this section we study algorithms for the subset-ℓ_p problem (namely, achieve multiplicative approximation) for $p = 0$ and $p = 1$. Our main finding is that the space complexity in insertion-only streams is characterized by the following combinatorial quantity.

Definition 2.1 (Heavy-Hitter Dimension)**.** For a set system $\mathcal{S} \subset 2^{[n]}$ and a vector $v \in \mathbb{R}^n$, denote by $H(\mathcal{S}, v)$ the set of *heavy hitters* induced by $\mathcal{S}$, i. e.,

$$H(\mathcal{S}, v) := \left\{ i \in [n] : \exists s \in \mathcal{S} \text{ s.t. } \operatorname{supp}(s \circ v) = \{i\} \right\},$$

and define the *heavy-hitter dimension* of $\mathcal{S} \subset \{0,1\}^n$ as

$$\mathrm{HHdim}(\mathcal{S}) := \sup_{v \in \mathbb{R}^n} |H(\mathcal{S}, v)|.$$

Let us provide some intuition for this definition. If $\operatorname{supp}(s \circ v) = \{i\}$ for some vector v and set $s \in \mathcal{S}$, then we call the index i a *heavy-hitter*, to reflect that it dominates the other (zero) coordinates in $s \circ v$. The heavy-hitter dimension then measures the number of heavy-hitters that a set system $\mathcal{S}$ could possibly induce (for an unknown v). It is known from previous sketching algorithms, cf. [32], that estimating a vector norm can essentially be reduced to estimating the heavy-hitters, which may provide an intuition why HHdim characterizes the space complexity of sketching subset norms. The above definition can be easily compared to that of VC-dimension (for the same set system), see Section 2.5 for details.

Our main result is a streaming algorithm with space complexity that is linear in the heavy-hitter dimension, i. e., $\widetilde{O}_\varepsilon(\mathrm{HHdim}(\mathcal{S}))$, see Theorem 2.10 in Section 2.2. We then provide several complementary results. From the direction of space lower bounds, we prove a linear lower bound $\Omega(\mathrm{HHdim}(\mathcal{S}))$, which matches our algorithm above (in Section 2.3), and also a much bigger bound for turnstile and sliding-window streams, which separates these richer models from insertion-only streams (in Section 2.4). From the direction of applications of our algorithmic techniques, we extend our algorithm to the "for-all" guarantee (in Section 2.5), and to the case $p = 1$ (in Section 2.6), and we also design a variant for the more restricted model of entrywise updates (in Section 2.7).

2.1 Examples and properties of heavy-hitter dimension

We present a few simple examples and basic properties of the heavy-hitter dimension that may be useful in applications, essentially proving the bounds shown in Table 1.

We begin with an alternative description of $\mathrm{HHdim}(\mathcal{S})$ where we view the set system $\mathcal{S} \subset 2^{[n]}$ as an *incidence matrix*, i. e., a $0-1$ matrix describing the incidence between sets $S \in \mathcal{S}$ and coordinates $i \in [n]$. Recall that a matrix $M \in \{0,1\}^{k \times k}$ is called a *permutation matrix* if every row and every column contain a single non-zero, i. e., exactly one 1. Clearly, up to reordering the rows and/or columns, such a matrix can be viewed as an identity matrix.

Lemma 2.2 (Permutation Submatrix)**.** *Let $\mathcal{S} \subset 2^{[n]}$, and let $M \in \{0,1\}^{|\mathcal{S}| \times n}$ be its incidence matrix. Then $\mathrm{HHdim}(\mathcal{S})$ is exactly the maximum size (number of rows/columns) in a submatrix of M that is a permutation matrix.*

Proof. Denote $\mathcal{S} = \{s_1, s_2, \ldots\}$, and let k be the largest size of permutation submatrix of M. Suppose that this submatrix is formed of rows $i_1, \ldots, i_k$ and columns $j_1, \ldots, j_k$. Consider a vector $v \in \{0,1\}^n$ in which coordinates $j_1, \ldots, j_k$ have value 1 and all other coordinates have value 0. Then it is straightforward to see that

$$\forall l \in [k], \quad \mathrm{supp}(s_{i_l} \circ v) = \{j_l\}.$$

Thus $k \leq \mathrm{HHdim}(\mathcal{S})$.

For the other direction, let $u \in \mathbb{R}^n$ be a vector that realizes $\mathrm{HHdim}(\mathcal{S})$. Then there are sets $s_{i_1}, \ldots, s_{i_{\mathrm{HHdim}(\mathcal{S})}} \in \mathcal{S}$ and coordinates $j_1, \ldots, j_{\mathrm{HHdim}(\mathcal{S})} \in [n]$ such that

$$\forall l \in [\mathrm{HHdim}(\mathcal{S})], \quad \mathrm{supp}(s_{i_l} \circ u) = \{j_l\}.$$

It is easily verified that rows $i_1, \ldots, i_{\mathrm{HHdim}(\mathcal{S})}$ and columns $j_1, \ldots, j_{\mathrm{HHdim}(\mathcal{S})}$ form a permutation submatrix of M. Thus, $\mathrm{HHdim}(\mathcal{S}) \leq k$, which completes the proof. □

Using the above lemma one can analyze the heavy-hitter dimension of several explicit set systems, as listed in the first four lines in Table 1. This lemma also implies bounds for several operations on set systems, as listed in the last three lines in Table 1. The proofs are rather straightforward and we give examples in Proposition 2.3 and Proposition 2.4 below.

Proposition 2.3 (Random Sets)**.** *Let $\mathcal{S} \subset 2^{[n]}$ be a set system of size $|\mathcal{S}| = k$, whose incidence matrix M is formed of entries that are independent Bernoulli random variables with parameter $p \in (0, 1/2]$. In other words, every set $S_i \in \mathcal{S}$ contains every coordinate $j \in [n]$ independently with probability p. Then*

$$\Pr\left[\, \mathrm{HHdim}(\mathcal{S}) \leq O(\log(nk)/p) \right] \geq 1 - 1/\mathrm{poly}(nk).$$

Proof. Fix t rows and $t = c \log(nk)/p$ columns of M for some absolute constant $c > 0$ and consider the corresponding submatrix M'. The probability that M' is an identity matrix is $p^t(1-p)^{t^2-t} \leq e^{-pt^2/2}$. For the event $\mathrm{HHdim}(\mathcal{S}) \geq t$ to occur there must be ordered sequences of t rows and t columns that yield an identity M'. Since the number of such choices is at most $n^t \cdot k^t$, we obtain

$$\Pr\left[\, \mathrm{HHdim}(\mathcal{S}) \geq t \right] \leq (nk)^t \cdot e^{-pt^2/2} \leq e^{-pt^2/4} \leq 1/\mathrm{poly}(nk),$$

where we use the fact that $(nk)^t = e^{pt^2/c}$ and choose c appropriately. □

Proposition 2.4 (Subadditivity). *For every $\mathcal{S}_1, \mathcal{S}_2 \subset 2^{[n]}$,*

$$\mathrm{HHdim}(\mathcal{S}_1 \cup \mathcal{S}_2) \le \mathrm{HHdim}(\mathcal{S}_1) + \mathrm{HHdim}(\mathcal{S}_2).$$

Proof. Let M_1 and M_2 be the incidence matrices of $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. Then the incidence matrix of $\mathcal{S}_1 \cup \mathcal{S}_2$ is, using block-matrix notation, simply $M = \left[\begin{smallmatrix} M_1 \\ M_2 \end{smallmatrix}\right]$. Every permutation submatrix of M can be partitioned into M_1 and M_2. As each part must contain a permutation submatrix that uses all its rows, the proposition follows. □

2.2 Streaming algorithm for the subset-ℓ_0 norm

We now design a one-pass streaming algorithm for the subset-ℓ_0 problem in an insertion-only stream. Recall that in this model the input is a stream $\langle a_1, \ldots, a_m \rangle$, where each item $a_j \in [n]$ represents an increment to coordinate a_j of a vector $v \in \mathbb{R}^n$. The streaming algorithm has two phases, an update phase that scans the stream, and a query phase that evaluates a query $s \in \mathcal{S}$. (In the case below, the query phase formally does not answer a specific query $s \in \mathcal{S}$, but rather reports a list that implicitly represents all queries $s \in \mathcal{S}$.) We assume that the set system $\mathcal{S}$ is given to the algorithm via a read-only tape, and thus requires no storage. See Section 1.2 for detailed definitions.

The algorithm uses a well-known technique of subsampling the coordinates of v (i. e., the set $[n]$) at a predetermined rate $p \in (0, 1]$, and producing an estimate only if the resulting vector has $\Omega(\varepsilon^{-2})$ non-zeros. Usually, counting the number of sampled non-zeros requires little space, but this is more challenging in our case of all norms $s \in \mathcal{S}$.

The key to bounding the total space usage is the following proposition, which bounds the global number of samples stored, when each of these samples is "needed" locally by some subset $s \in \mathcal{S}$. This condition has a parameter k, and the reader may initially think of $k = 1$.

Proposition 2.5. *Let $\mathcal{S} \subset \{0,1\}^n$ be a set system. Suppose $Z \subset [n]$ and $k \ge 1$ are such that for every $i \in Z$*

$$\exists s \in \mathcal{S}, \quad i \in s \text{ and } |Z \cap s| \le k \tag{2.1}$$

(in words, index i is "k-needed" by some $s \in \mathcal{S}$). Then $|Z| \le k \cdot \mathrm{HHdim}(\mathcal{S})$.

Proof. We proceed by induction on k. For the base case $k = 1$, consider a vector $v \in \{0,1\}^n$ whose support is exactly the given Z. Then for every $i \in Z$, there is $s \in \mathcal{S}$ such that $Z \cap s = \{i\}$, and thus $H(s, v) := \mathrm{supp}(s \circ v) = \{i\}$. It follows that $|Z| \le |H(\mathcal{S}, v)| \le \mathrm{HHdim}(\mathcal{S})$.

For the inductive step, consider $k \ge 2$. Given Z, construct $A \subset Z$ as follows. Start with $A = Z$, and then repeatedly, if A contains an index i for which there is no $s \in \mathcal{S}$ with $A \cap s = \{i\}$ (i. e., i is not 1-needed), then remove i from A. The repetitions stop when A contains no such index i. We claim that the final set A satisfies

$$\forall s \in \mathcal{S}, \qquad Z \cap s \ne \emptyset \iff A \cap s \ne \emptyset.$$

For the forward direction, observe that initially $|A \cap s| = |Z \cap s| \ge 1$, and that no iteration ever decreases any $|A \cap s|$ from 1 to 0. The reverse direction is obvious because $A \subset Z$.

We can verify that $Z \setminus A$ satisfies the induction hypothesis, i. e., that every $i \in Z \setminus A$ is $(k-1)$-needed. Indeed, since $i \in Z$, it is k-needed by some $s \in \mathcal{S}$, as expressed by (2.1), and by the claim, this same s also satisfies $|A \cap s| \geq 1$. Hence, $|(Z \setminus A) \cap s| \leq k-1$, which shows i is $(k-1)$-needed. Applying the induction hypothesis, we have $|Z \setminus A| \leq (k-1) \cdot \mathrm{HHdim}(\mathcal{S})$.

In addition, A satisfies the induction's base case $k = 1$, because the iterations stop when every $i \in A$ is 1-needed. Hence $|A| \leq \mathrm{HHdim}(\mathcal{S})$, and we conclude that $|Z| = |Z \setminus A| + |A| \leq k \cdot \mathrm{HHdim}(\mathcal{S})$. □

Our algorithm is based on simulating the simple estimator defined in the following lemma. (The difficulty will be to apply it to $s \circ v$ for $s \in \mathcal{S}$ that is not known in advance.)

Lemma 2.6. *Fix $v \in \mathbb{R}^n$, and sample its coordinates to form $v' \in \mathbb{R}^n$ as follows. Suppose each coordinate is $v'_i = v_i X_i$, where $X_1, \ldots, X_n$ are pairwise-independent identically distributed Bernoulli random variables with parameter $p \in (0,1]$. Then*

$$\mathbb{E}\left[\tfrac{1}{p}\|v'\|_0\right] = \|v\|_0, \quad \mathrm{Var}\left(\tfrac{1}{p}\|v'\|_0\right) = \tfrac{1-p}{p}\|v\|_0, \quad \textit{and}$$
$$\Pr\left[\left|\tfrac{1}{p}\|v'\|_0 - \|v\|_0\right| \geq 3(\tfrac{1-p}{p}\|v\|_0)^{1/2}\right] \leq \tfrac{1}{9}.$$

Proof. The expectation and variance follow from direct calculation (note that it suffices to assume pairwise independence). The tail bound is straightforward from Chebyshev's inequality. □

Our basic algorithm for storing a subsample of the coordinates is described in Algorithm 1. The idea is to sample the universe $[n]$ at rate $p \in (0,1]$ using pairwise independent random variables $\xi_1, \ldots, \xi_n \in \{0,1\}$ (this can be viewed as a hash function $\xi : [n] \to \{0,1\}$). We store the sampled items (coordinates) from the stream so long as they are "needed" by some $s \in \mathcal{S}$, or more precisely, U-needed in the sense of Proposition 2.5. Here, the budget parameter U represents a "local" bound that holds separately for each $s \in \mathcal{S}$. We show that at the end, for every $s \in \mathcal{S}$, if s contains at most U sampled coordinates, then all these samples are completely stored; otherwise, the number of samples stored is at least U. We will later use this algorithm to simulate a pairwise-independent sampling from a desired $s \in \mathcal{S}$.

Lemma 2.7. *Consider Algorithm 1 for $\mathcal{S} \subset \{0,1\}^n$ with parameters $p \in (0,1]$ and $U \in [1,n]$. When run on an insertion-only stream accumulating to $v \in \mathbb{R}^n$, it makes one pass, uses $O(U) \cdot \mathrm{HHdim}(\mathcal{S})$ words of space, and outputs $\mathcal{H} \subset [n]$ of size $|\mathcal{H}| \leq U \cdot \mathrm{HHdim}(\mathcal{S})$. Moreover, suppose that $\xi \in \{0,1\}^n$ is the sampling vector from the algorithm. Then for every $s \in \mathcal{S}$, if $\|\xi \circ s \circ v\|_0 \leq U$ then*

$$\mathrm{supp}(\xi \circ s \circ v) \subset \mathcal{H};$$

and otherwise $|\mathcal{H} \cap s| \geq U$.

Proof. Observe that after each update operation, every $i \in \mathcal{H}$ is U-needed, i. e.,

$$\exists s \in \mathcal{S}, \quad i \in s \text{ and } |s \cap \mathcal{H}| \leq U.$$

Algorithm 1 Bounded-Sampler for $\mathcal{S} \subset \{0,1\}^n$

Input: $p \in (0,1]$, $U \in [1,n]$, an insertion-only stream $\langle a_1, \ldots, a_m \rangle$, where each item $a_j \in [n]$
Initialize:
  $\mathcal{H} \leftarrow \emptyset$
  pick random $\xi_1, \ldots, \xi_n \in \{0,1\}^n$, with each $\Pr[\xi_i = 1] = p$ and pairwise independent
Update(a_j):
if $\xi_{a_j} = 1$ and $a_j \notin \mathcal{H}$ **then**
  $\mathcal{H} \leftarrow \mathcal{H} \cup \{a_j\}$
  while there is $i \in \mathcal{H}$ such that all $s \in \mathcal{S}$ with $i \in s$ satisfy $|s \cap \mathcal{H}| > U$ **do**
    remove this i from $\mathcal{H}$ ▷ remove i that is not U-needed
Query():
  return $\mathcal{H}$ ▷ implicit answer $\mathrm{supp}(\xi \circ s \circ v)$ for each $s \in \mathcal{S}$

By applying Proposition 2.5 to this $\mathcal{H}$ we have that $|\mathcal{H}| \le U \cdot \mathrm{HHdim}(\mathcal{S})$. This bound applies in particular to the output $\mathcal{H}$, and also implies that the algorithm uses $O(U) \cdot \mathrm{HHdim}(\mathcal{S})$ words of space.

Now consider $\mathcal{H}$ at the end of the stream, and some $s \in \mathcal{S}$. Let $Z = \mathrm{supp}(\xi \circ s \circ v)$. If $|Z| \le U$, then every $i \in Z$ must have been stored in the final $\mathcal{H}$ (because it must be added at some update and can never be removed), which proves that $Z \subset \mathcal{H}$.

Next, suppose that $|Z| > U$. Observe that every index $i \in Z$ is added at some point to $\mathcal{H}$, hence $|s \cap \mathcal{H}|$ is increased more than U times. Moreover, whenever any $i \in s \cap \mathcal{H}$ is removed from $\mathcal{H}$, it may only decrease $|s \cap \mathcal{H}|$ from $U+1$ to U, but never below U. It follows that at the end of the execution, $|s \cap \mathcal{H}| \ge U$. □

We shall use Algorithm 1 as a subroutine twice, first to compute an $O(1)$-approximation to a query $s \in \mathcal{S}$, and then (a refined version of it) to compute a $(1 \pm \varepsilon)$-approximation. En route to an $O(1)$-approximation, we introduce Algorithm 2, which simply runs Algorithm 1 in parallel $\Theta(\log\log n)$ times, and then when given a query $s \in \mathcal{S}$, it outputs one bit. The next lemma shows that this algorithm solves a promise (gap) version of the subset-ℓ_0 problem: if $\|v \circ s\|_0 \le 1/(2p)$ then with high probability it outputs 0, and if $\|v \circ s\|_0 \ge 2/p$ then with high probability it outputs 1.

Lemma 2.8. *Consider Algorithm 2 for $\mathcal{S} \subset \{0,1\}^n$ with parameter $r \in [\frac{1}{4}, n]$. When run on an insertion-only stream accumulating to $v \in \mathbb{R}^n$, it makes one pass and uses $O(\log\log n \cdot \mathrm{HHdim}(\mathcal{S}))$ words of space, and then when queried for $s \in \mathcal{S}$, with probability at least $1 - O(1/\log^2 n)$ its output satisfies: if $\|v \circ s\|_0 \le r/2$ the output is 0, and if $\|v \circ s\|_0 \ge 2r$ the output is 1.*

Proof. The algorithm's space usage is dominated by the t instances of the Bounded-Sampler, and thus follows from Lemma 2.7.

If $r \le U$, then $p = 1$ and the algorithm is deterministic, with the following guarantee by Lemma 2.7. If $\|s \circ v\|_0 \le r/2 \le U$, then $z = \|s \circ v\|_0$ is an exact estimator, and the output is 0. If $\|s \circ v\|_0 \ge 2r$, then $z \ge \min(\|s \circ v\|_0, U) \ge r$ and the output is 1.

Algorithm 2 Constant-Detector for $\mathcal{S} \subset \{0,1\}^n$

Input: $r \in [\frac{1}{4}, n]$, an insertion-only stream $\langle a_1, \ldots, a_m \rangle$, where each item $a_j \in [n]$
Initialize:
  $U \leftarrow 100$, $p \leftarrow \min(1, U/r)$, $t \leftarrow \Theta(\log \log n)$
  let $\mathcal{A}_1, \ldots, \mathcal{A}_t$ be instances of Bounded-Sampler with parameters p and U
Update(a_j):
  update each $\mathcal{A}_i$ with a_j
Query($s \in \mathcal{S}$):
  query each $\mathcal{A}_i$ for s and let $\mathcal{H}_i$ be its output
  $\bar{z} \leftarrow \frac{1}{p} \cdot \text{median}\left(|\mathcal{H}_1 \cap s|, \ldots, |\mathcal{H}_t \cap s|\right)$
  return $\mathbb{1}_{\{\bar{z} \geq r\}}$

We thus assume henceforth that $r > U$ and thus $p = U/r$. Consider an instance $\mathcal{A}_i$ of the Bounded-Sampler, let $\mathcal{H}_i$ be its output, and let $\xi^i \in \{0,1\}^n$ be its sampling vector. We would like to analyze the quantity $\frac{1}{p}|\mathcal{H}_i \cap s|$ used in the algorithm.

Now suppose $\|s \circ v\|_0 \geq 2r = \frac{2U}{p}$. Then by Lemma 2.6, the expectation is $\mathbb{E}[\frac{1}{p}\left\|\xi^i \circ s \circ v\right\|_0] = \|s \circ v\|_0$, and with probability at least 8/9,

$$\tfrac{1}{p}\left\|\xi^i \circ s \circ v\right\|_0 \geq \|s \circ v\|_0 - 3\big(\tfrac{1-p}{p}\|s \circ v\|_0\big)^{1/2} \geq \|s \circ v\|_0 - \tfrac{3}{(2U)^{1/2}}\|s \circ v\|_0 \geq \tfrac{3}{4}\|s \circ v\|_0 \geq \tfrac{3U}{2p}.$$

In this event, $\left\|\xi^i \circ s \circ v\right\|_0 \geq \frac{3U}{2} > U$, which by Lemma 2.7 implies that $|\mathcal{H}_i \cap s| \geq U$, and therefore the estimate obtained from the instance $\mathcal{A}_i$ is $\frac{1}{p}|\mathcal{H}_i \cap s| \geq \frac{1}{p}U \geq r$. By a standard probability amplification argument, i. e., Chernoff bound, with probability at least $1 - O(1/\log^2 n)$, the median of t independent repetitions is $\bar{z} \geq r$, and the output is 1.

Suppose next that $\|s \circ v\|_0 \leq \frac{r}{2} = \frac{U}{2p}$. Then by Lemma 2.6, with probability at least 8/9,

$$\tfrac{1}{p}\left\|\xi^i \circ s \circ v\right\|_0 \leq \|s \circ v\|_0 + 3\big(\tfrac{1-p}{p}\|s \circ v\|_0\big)^{1/2} \leq \tfrac{U}{2p} + \tfrac{3}{p}\big(\tfrac{U}{2}\big)^{1/2} \leq \tfrac{3U}{4p}.$$

In this event, $\left\|\xi^i \circ s \circ v\right\|_0 \leq \frac{3U}{4}$ which by Lemma 2.7 implies that $\mathcal{H}_i \cap s = \text{supp}(\xi^i \circ s \circ v)$, and therefore the estimate obtained from the instance $\mathcal{A}_i$ is $\frac{1}{p}|\mathcal{H}_i \cap s| = \frac{1}{p}\left\|\xi^i \circ s \circ v\right\|_0 \leq \frac{3U}{4p} < r$. By a standard probability amplification argument, i. e., Chernoff bound, with probability at least $1 - O(1/\log^2 n)$, the median of t independent repetitions is $\bar{z} < r$, and the output is 0. □

Using Algorithm 2 we can now easily design an algorithm achieving $O(1)$-approximation. This algorithm uses standard ideas from the literature, like subsampling a stream at different rates, which goes back to [28]). We adapt it here to our setting of a set system.

Lemma 2.9 (8-approximation algorithm)**.** *There is an algorithm that when run for $\mathcal{S} \subset \{0,1\}^n$ with parameter $r \in [\frac{1}{4}, n]$ on an insertion-only stream accumulating to $v \in \mathbb{R}^n$, makes one pass and uses $O(\log n \cdot \log\log n \cdot \text{HHdim}(\mathcal{S}))$ words of space, and then when queried for $s \in \mathcal{S}$, it output a number z that with probability at least 0.99 satisfies $\|s \circ v\|_0 < z < 8\|s \circ v\|_0$.*

Proof. The algorithm consists of $l = O(\log n)$ parallel independent instances of Algorithm 2, denoted $\mathcal{B}_0, \mathcal{B}_1, \ldots, \mathcal{B}_l$. For each instance $\mathcal{B}_j$, the parameter is $r_j = 2^{j-2}$. To process a query $s \in \mathcal{S}$, the algorithm queries every instance for this s. Let x_j denote the output from instance $\mathcal{B}_j$. By Lemma 2.8, $x_0 = 0$ if and only if $\|s \circ v\|_0 = 0$. Hence, x_0 can be used to distinguish whether $\|s \circ v\|_0 = 0$, and we may assume henceforth that $\|s \circ v\|_0 > 0$.

The algorithm computes j^* which is the smallest $j \geq 1$ such that $x_j = 0$, and outputs $z = 2^{j^*}$. By Lemma 2.8 and a union bound, with probability at least 0.99, for all $j = 1, \ldots, l$, if $\|s \circ v\|_0 \geq 2r_j = 2^j$ then $x_j = 1$, and if $\|s \circ v\|_0 \leq r_j/2 = 2^{j-2}$ then $x_j = 0$. Assuming this event happens, and by applying the first condition to j^* and the second one to $j^* - 1$ (both in the contrapositive form), we obtain $2^{(j^*-1)-2} < \|s \circ v\|_0 < 2^{j^*} = z$.

It is easy to verify the algorithm's space requirement, and this completes the proof. □

We now design an $(1 \pm \varepsilon)$-approximation algorithm, by using the above $O(1)$-approximation algorithm and a variant of Algorithm 1.

Algorithm 3 Multiplicative Approximation for $\mathcal{S} \subset \{0,1\}^n$

Input: $\varepsilon \in (0,1)$, an insertion-only stream $\langle a_1, \ldots, a_m \rangle$, where each item $a_j \in [n]$
Initialize:
  $t \leftarrow \Theta(\log n)$
  let $\{\mathcal{A}_i\}_{i=1,\ldots,t}$ be instances of Bounded-Sampler with parameters $p_i = 2^{1-i}$ and $U = \lceil 400\varepsilon^{-2} \rceil$
  let $\mathcal{B}$ be an 8-approximation algorithm (from Lemma 2.9)
Update(a_j):
  Update $\mathcal{B}$ and each $\mathcal{A}_i$ with a_j
Query($s \in \mathcal{S}$):
  query $\mathcal{B}$ for s and let z be its output
  $l \leftarrow \max(1, \lceil \log(z\varepsilon^2/100) \rceil)$;
  query $\mathcal{A}_l$ for s and let $\mathcal{H}_l$ be its output
  return $\frac{1}{p_l}|\mathcal{H}_l \cap s|$;

Theorem 2.10. *Consider Algorithm 3 for $\mathcal{S} \subset \{0,1\}^n$ with parameter $\varepsilon \in (0,1)$. When run on an insertion-only stream accumulating to $v \in \mathbb{R}^n$, it makes one pass and uses $O((\varepsilon^{-2} + \log\log n)\log n \cdot \mathrm{HHdim}(\mathcal{S}))$ words of space, and then when queried for $s \in \mathcal{S}$, its output $\widehat{z}(s)$ satisfies*[5]

$$\forall s \in \mathcal{S}, \qquad \Pr\left[\widehat{z}(s) \in (1 \pm \varepsilon)\|s \circ v\|_0\right] \geq 0.8.$$

Proof. We assume $\|s \circ v\|_0 > 0$, since otherwise the algorithm specified in Lemma 2.9 already gives the correct answer. Next, by Lemma 2.9, with probability at least 0.99, the value z reported by $\mathcal{B}$ is an 8-approximation to $\|s \circ v\|_0$, and for the rest of the proof we assume this event

[5]Our sketching algorithm uses pairwise-independent random bits, and its success probability can be amplified by independent parallel repetitions of the algorithm. A possibly better approach is to use k-wise independent random bits for larger k to amplify the success probability directly.

happens. Consider l as in the algorithm, and its corresponding $p_l = 2^{1-l} = \min(1, 200\varepsilon^{-2}/z)$, then

$$\min(1, 25\varepsilon^{-2}/\|s \circ v\|_0) \le p_l \le \min(1, 200\varepsilon^{-2}/\|s \circ v\|_0).$$

Let $Z_l \subset \mathrm{supp}(s \circ v)$ be the set of indices sampled in algorithm $\mathcal{A}_l$ by the hash function ξ, and recall it samples each index in $\mathrm{supp}(s \circ v)$ with probability p_l pairwise independently. By Lemma 2.6, with probability at least 0.8,

$$\left|\tfrac{1}{p_l}|Z_l| - \|s \circ v\|_0\right| \le 3(\tfrac{1-p_l}{p_l}\|s \circ v\|_0)^{1/2} \le \varepsilon\|s \circ v\|_0,$$

which implies that $|Z_l| \le (1+\varepsilon)p_l\|s \circ v\|_0 \le 400\varepsilon^{-2} = U$. When this happens, by Lemma 2.7 instance $\mathcal{A}_l$ of Algorithm 1 outputs $\mathcal{H}_l$ that satisfies $\mathcal{H}_l \cap s = Z_l$, and we conclude that our algorithm's output is $\frac{1}{p_l}|\mathcal{H}_l \cap s| = \frac{1}{p_l}|Z_l| \in (1 \pm \varepsilon)\|s \circ v\|_0$.

The proof of Theorem 2.10 is completed by easily verifying the space usage of the algorithm. □

2.3 Matching lower bound

We prove that for every set system $\mathcal{S} \subset \{0,1\}^n$, the space complexity of every universal streaming algorithm must be $\Omega(\mathrm{HHdim}(\mathcal{S}))$, which matches Theorem 2.10 in terms of the linear dependence on the heavy-hitter dimension.

Theorem 2.11. *Let $\mathcal{S} \subset \{0,1\}^n$ be a non-empty set system. Suppose $\mathcal{A}$ is a (randomized) one-pass streaming algorithm that solves the subset-ℓ_p problem for $\mathcal{S}$ within approximation factor $\alpha \ge 1$ for some $p \ge 0$. Then $\mathcal{A}$ requires $\Omega(\mathrm{HHdim}(\mathcal{S}))$ bits of space for some insertion-only stream input. Moreover, if $\alpha = 1+\varepsilon$ for some $\varepsilon \ge 1/\sqrt{\max_{s\in\mathcal{S}}\|s\|_0}$ and $p \ne 1$, then $\mathcal{A}$ requires $\Omega(\mathrm{HHdim}(\mathcal{S}) + \varepsilon^{-2})$ bits of space.*

Proof. We begin with the lower bound for $\Omega(\mathrm{HHdim}(\mathcal{S}))$. Let $k = \mathrm{HHdim}(\mathcal{S}) \ge 1$, then there exists a vector $v \in \mathbb{R}^n$ such that $|H(S, v)| \ge k$. Without loss of generality, we can assume $v \in \{0,1\}^n$ since replacing each non-zero coordinate of v with 1 does not change $H(s, v) := \mathrm{supp}(s) \cap \mathrm{supp}(v)$ for any $s \in \{0,1\}^n$.

Since $\mathcal{S}$ is given to the algorithm before the stream begins, we can use the vector v and the algorithm $\mathcal{S}$ to design a one-way communication protocol that solves INDEX(k), in which Alice is holding a binary vector $x \in \{0,1\}^k$ of dimension k and Bob is holding index $i \in [k]$. Alice needs to send one round of messages to Bob. Bob needs to figure out the i-th coordinate in Alice's string. It is well-known that any protocol with at least constant probability of success requires $\Omega(k)$ bits (see, e. g., [38]).

We now describe the protocol for the INDEX problem using the algorithm for the subset-ℓ_p problem. Firstly, since $|H(\mathcal{S}, v)| \ge k$, there exists $s_1, s_2, \ldots, s_k \in \mathcal{S}$ such that each $\|s_j \circ v\|_0 = 1$ and $H(s_j, v)$s are disjoint. Therefore, each s_j uniquely picks up a coordinate in v. We denote the non-zero coordinate of $s_j \circ v$ as z_j. Alice and Bob (without communication) then map the j-th element in $[k]$ to the z_j-th coordinate in v. Alice then modifies v such that $v_{z_j} = x_j$ for each j. As such, Alice obtains a vector v'. She then converts the vector v' to a insertion-only

stream and runs the algorithm $\mathcal{A}$ on vector v and sends the memory content to Bob. Bob recovers the instance of the algorithm $\mathcal{A}$ and runs a query on approximating the ℓ_p norm of vector $v' \circ s_i$. Since $\|v' \circ s_i\|_p = x_i$, Bob can recover an answer for the INDEX problem from any α-multiplicative approximation of $\|v' \circ s_i\|_p$. Thus algorithm $\mathcal{A}$ must use $\Omega(k)$ bits of space in the worst case.

Lastly, the $\Omega(\varepsilon^{-2})$ lower bound follows from the standard lower bound for ℓ_p norm [32]. □

Remark 2.12. For every $p \leq 2$, the lower bound $\Omega(\mathrm{HHdim}(\mathcal{S}) + \varepsilon^{-2})$ is existentially tight up to polylog(n)-factor. Indeed, the system $\mathcal{S}^* = \{e_1, \ldots, e_{k-1}, [k\,..\,n]\}$ has $\mathrm{HHdim}(\mathcal{S}^*) = |\mathcal{S}^*| = k$, but admits an algorithm with space usage $O(\mathrm{HHdim}(\mathcal{S}^*) + 1/\varepsilon^2)$ by explicitly storing the first $k-1$ coordinates and running an ℓ_p-norm algorithm for the coordinates-subset $[k\,..\,n]$. Thus, Theorem 2.11 provides the best-possible dependence on ε.

Remark 2.13. Some set systems $\mathcal{S} \subset \{0,1\}^n$ admit a stronger lower bound hand of $\Omega(\mathrm{HHdim}(\mathcal{S}) \cdot \varepsilon^{-2})$. Indeed, consider a set system with $\mathrm{HHdim}(\mathcal{S})$ disjoint subsets where each subset has cardinality ε^{-2}, then a lower bound follows by a reduction from $\mathrm{HHdim}(\mathcal{S})$ independent instances of Gap Hamming Distance, and apply [4], which is based on [22, 12]. Thus, the space complexity in Theorem 2.10 provides the best-possible dependence on ε.

2.4 Strong lower bounds for turnstile and sliding-window models

We now show an impossibility result for the subset-ℓ_p problem in richer data streams, namely, the strict turnstile and sliding-window models. Specifically, we exhibit a family of subsets that has a small heavy-hitter dimension but does not admit efficient (nontrivial) streaming algorithms in those richer data streams. This shows a strong separation from the insertion-only model.

Recall that in the turnstile model, the stream contains additive updates to a vector $v \in \mathbb{R}^n$, which is initialized to all-zeros. As the updates may be negative, it captures both insertions and deletions. In the strict turnstile model, the coordinates of v must remain non-negative at all times. For an even integer n, let $\mathcal{S}_{\mathrm{int}} \subset 2^{[n]}$ be the family of all intervals of length $n/2$ (and thus cardinality $n/2+1$), i. e.,

$$\mathcal{S}_{\mathrm{int}} := \big\{[a..a + n/2] : \ a = 1, \ldots, n/2\big\}.$$

We next show that $\mathcal{S}_{\mathrm{int}}$ has a small heavy-hitter dimension (much smaller than its cardinality $|\mathcal{S}_{\mathrm{int}}| = n/2$), and thus admits efficient algorithms for insertion-only streams.

Proposition 2.14. $\mathrm{HHdim}(\mathcal{S}_{\mathrm{int}}) \leq 2$.

Proof. Consider any vector $v \in \mathbb{R}^n$. We claim that $H(\mathcal{S}_{\mathrm{int}}, v)$ contains at most one index from $[1..n/2]$. Indeed, if there is such an index at all, let j be the smallest one. Then there is $s \in \mathcal{S}_{\mathrm{int}}$ such that $\mathrm{supp}(s \circ v) = \{j\}$, which implies that $v_{j+1}, \ldots, v_{n/2}$ must all be equal to 0. It follows that $j+1, \ldots, n/2 \notin H(\mathcal{S}_{\mathrm{int}}, v)$, and the claim follows. A symmetric claim applies to the indices from $[n/2+1..n]$, and thus $|H(\mathcal{S}_{\mathrm{int}}, v)| \leq 2$. □

We now show a strong space lower bound for every streaming algorithm in the turnstile model.

Theorem 2.15. *Suppose $\mathcal{A}$ is a (randomized) one-pass streaming algorithm that solves the subset-ℓ_p problem for $\mathcal{S}_{\text{int}}$ within approximation factor $\alpha \geq 1$ for some $p \geq 0$. Then for some turnstile stream input, $\mathcal{A}$ requires $\Omega(n)$ bits of space.*

Before proving the theorem, we recall a known communication lower bound, which was introduced and proved in [6], following the classical Index problem from [40]. In the *Augmented Index* problem, denoted AUG_n, Alice holds a binary vector $x \in \{0,1\}^n$, and Bob holds an index $j \in [n]$ and a sequence $x_{j+1}, \ldots, x_n \in \{0,1\}$ (part of Alice's vector). Alice then sends a single round of message to Bob, who is required to output x_j.

Theorem 2.16 (Lower Bound for Augmented Index [6])**.** *In every shared-randomness protocol for AUG_n (with success probability at least 0.9), Alice must send $\Omega(n)$ bits.*

We are now ready to prove our theorem.

Proof of Theorem 2.15. Without loss of generality, suppose $n > 0$ is an even integer. Suppose we have an algorithm $\mathcal{A}$ that solves, with α-approximation, the subset-ℓ_p problem of $\mathcal{S}_{\text{int}}$. We now describe a protocol that solves the $\text{AUG}_{n/2}$ problem. Now Alice has a binary vector $x \in \{0,1\}^{n/2}$, and Bob has an index j and $x_{j+1}, x_{j+2}, \ldots x_{n/2}$. Alice treats her vector x as a stream of updates and feeds it to $\mathcal{A}$. She then sends the memory content of $\mathcal{A}$ to Bob. Bob continues running $\mathcal{A}$ based on the memory content received from Alice. He then feeds $-x_{j+1}, -x_{j+2}, \ldots, -x_{n/2}$ to the stream. After this, Bob queries the set $s_j = [j \,..\, j + n/2]$. If the algorithm answers a number > 0, Bob then claims $x_j = 1$. Otherwise he claims $x_j = 0$.

To show the correctness, we observe that after Bob's updates, the vector in the stream is exactly $x' = (x_1, x_2, \ldots, x_j, 0, 0, \ldots, 0)$. Therefore, $x' \circ s_j = (0, \ldots, 0, x_j, 0, \ldots, 0)$. Hence if $x_j = 0$, then with probability at least 0.9, $\mathcal{A}$ outputs 0 and if $x_j \neq 0$, then with probability at least 0.9, $\mathcal{A}$ outputs > 0. Thus, the lower bound of $\text{AUG}_{n/2}$ implies a space lower bound of $\mathcal{A}$. □

A similar argument applies to the sliding-window model, where the stream has a parameter $W \geq 1$ called window-size, and the input vector $v \in \mathbb{R}^n$ at any time t is determined by the last W additive updates, i. e., items from time $t - W$ or earlier in the stream expire (are ignored).

Theorem 2.17. *Suppose $\mathcal{A}$ is a (randomized) one-pass streaming algorithm that solves the subset-ℓ_p problem for $\mathcal{S}_{\text{int}}$ within approximation factor $\alpha \geq 1$ for some $p \geq 0$. Then for some sliding-window stream input, $\mathcal{A}$ requires $\Omega(\min(n, W))$ bits of space.*

Proof. Suppose we have an algorithm $\mathcal{A}$ that solves, with α-approximation, the subset-ℓ_p problem for $\mathcal{S}_{\text{int}}$ for the most recent W updates at any time t. Let $d = \min(n/2, W)$. We now describe a protocol that solves the AUG_d problem. Now Alice has a binary vector $x \in \{0,1\}^d$, and Bob has an index j and $x_{j+1}, x_{j+2}, \ldots x_d$. Alice treats her vector x as a stream of updates and feeds it to $\mathcal{A}$ in an order $x_d, x_{d-1}, \ldots, x_1$. She then sends the memory content of $\mathcal{A}$ to Bob. Bob continues running $\mathcal{A}$ based on the memory content received from Alice. He then feeds

arbitrary updates for x_{j-1} to the stream such that all updates of $x_d, x_{d-1}, \ldots, x_{j+1}$ expire except the update of x_j, i. e., Bob sends $W - \sum_{j'=j+1}^{d} x_{j'}$ updates to the $(j-1)$-st coordinate of v. After this, Bob queries the set $s_j = [j, j + n/2]$. If the algorithm answers a number > 0, Bob then claims $x_j = 1$. Otherwise he claims $x_j = 0$.

It is easy to verify the correctness of the protocol and hence proves a $\Omega(\min(n, W))$ space lower bound of the algorithm. □

2.5 Streaming algorithm with "for all" guarantee

We now show how to extend our algorithm to achieve the "for all" guarantee using space usage $\widetilde{O}_\varepsilon(\mathrm{HHdim}(\mathcal{S})^2)$. The key is to establish a connection between the heavy-hitter dimension and the VC-dimension (of every set system $\mathcal{S}$), and the algorithm then follows by the standard technique of amplifying the success probability by independent repetitions.

Recall that the VC-dimension of $\mathcal{S}$ is defined as the maximum cardinality of a set $A \subset [n]$ that is shattered, where A is called shattered if every subset of A can be realized as $A \cap S$ for some $S \in \mathcal{S}$. The heavy-hitter dimension can be defined analogously, by modifying the definition of being shattered to this: for every element $a \in A$, there is $S \in \mathcal{S}$ such that $S \cap A = \{a\}$. It then follows easily that $\mathrm{VCdim}(\mathcal{S}) \le \mathrm{HHdim}(\mathcal{S})$, proved formally in Proposition 2.18 below. However the gap between them cannot be bounded by any fixed factor. For instance, when $\mathcal{S}$ is the set of $k \in [n]$ singleton sets, $\mathrm{VCdim}(\mathcal{S}) = 1$ whereas $\mathrm{HHdim}(\mathcal{S}) = k$.

Proposition 2.18. *Let $\mathcal{S} \subset 2^{[n]}$, and denote its VC-dimension by $\mathrm{VCdim}(\mathcal{S})$. Then*

$$\mathrm{VCdim}(\mathcal{S}) \le \mathrm{HHdim}(\mathcal{S}).$$

Proof. We show that $\mathcal{S}$ cannot shatter any set of size $\mathrm{HHdim}(\mathcal{S}) + 1$. Suppose $\mathcal{S}$ can shatter a set $\mathcal{S}$ with $|S| = \mathrm{HHdim}(\mathcal{S}) + 1$. Then for each $a \in \mathcal{S}$, we have a set $s_a \in \mathcal{S}$ such that $\{a\} = s_a \cap S$. This in turn indicates $\mathrm{HHdim}(\mathcal{S}) \ge \mathrm{HHdim}(\mathcal{S}) + 1$, a contradiction. □

Lemma 2.19 (Sauer-Shelah Lemma [43, 46])**.** *Every $\mathcal{S} \subset 2^{[n]}$ with VC-dimension k has cardinality $|\mathcal{S}| = O(n^k)$.*

The next theorem achieves the "for all" guarantee by standard amplification, namely, by reporting the median value of $O(\log|\mathcal{S}|)$ independent repetitions, and using the above to bound this number of repetitions.

Theorem 2.20. *For every $\mathcal{S} \subset \{0,1\}^n$ there is a one-pass streaming algorithm, that when run and $\varepsilon \in (0,1)$ on an insertion-only stream accumulating to $v \in \mathbb{R}^n$, it uses $O((\varepsilon^{-2} + \log\log n) \cdot \mathrm{HHdim}(\mathcal{S})^2 \cdot \log^2 n)$ words of space, and then when queried for $s \in \mathcal{S}$, its output $\widehat{z}(s)$ satisfies*

$$\Pr\left[\forall s \in \mathcal{S},\ \widehat{z}(s) \in (1 \pm \varepsilon)\|s \circ v\|_0\right] \ge 0.9.$$

Proof. We apply Algorithm 2.10 $O(\log|\mathcal{S}|)$ times independently in parallel, and answer any query using the median value of all the repetitions. For the space bound, we use Lemma 2.19 and Proposition 2.18 to bound the number of repetitions by

$$O(\log|\mathcal{S}|) \le O(\mathrm{VCdim}(\mathcal{S}) \cdot \log n) \le O(\mathrm{HHdim}(\mathcal{S}) \cdot \log n).$$

For the desired error bound we use a union bound over all $s \in \mathcal{S}$, where the error probability for a fixed query $s \in \mathcal{S}$ is at most $2^{-\Omega(\log|\mathcal{S}|)} \leq 0.1/|\mathcal{S}|$ by a standard Chernoff bound and a suitable choice of constants. □

2.6 Generalizing the algorithm to the subset-ℓ_1 norm

In the insertion-only streaming model, the ℓ_1 norm of a vector v is simply the sum of all updates (effectively without absolute values). Thus, we can then reduce ℓ_1 to ℓ_0, and obtain an algorithm for subset-ℓ_1, as follows. Assuming that the stream length m is bounded by $\overline{m} = \text{poly}(n)$, we convert each update $a_j \in [n]$ to an update of the form (a_j, j) to $v' \in \mathbb{R}^{n\times\overline{m}}$, a binary vector in a larger dimension. It is easy to verify that $\|v'\|_0 = \|v\|_1$. We also convert the set system $\mathcal{S} \in 2^{[n]}$ to the new universe as follows. For each $s \in \mathcal{S}$, we expand each of its entries s_i to $\overline{m}$ duplicates of s_i, which yields a new set system $\mathcal{S}' \subset 2^{[n\times\overline{m}]}$. The following lemma shows that the new set system $\mathcal{S}'$ has the same-heavy hitter dimension as $\mathcal{S}$.

Lemma 2.21. $\text{HHdim}(\mathcal{S}') = \text{HHdim}(\mathcal{S})$.

Proof. We first show $\text{HHdim}(\mathcal{S}) \leq \text{HHdim}(\mathcal{S}')$. For each vector $v \in \mathbb{R}^n$, we can pad each coordinate with $\overline{m} - 1$ zeros to obtain a vector v'. Therefore, $|H(v, \mathcal{S})| = |H(v', \mathcal{S}')|$ and thus $\text{HHdim}(\mathcal{S}) \leq \text{HHdim}(\mathcal{S}')$. We now show the other direction, $\text{HHdim}(\mathcal{S}') \leq \text{HHdim}(\mathcal{S})$. For any vector $v' \in \mathbb{R}^{n\times\overline{m}}$, suppose an $s' \in \mathcal{S}'$ satisfies $\text{supp}(s' \circ v') = \{(i, j)\}$ for some $(i, j) \in [n]\times[\overline{m}]$. Then it must be the case that $v_{(i,j')} = 0$ for any $j' \neq j$ since $(i, j') \in s'$ for all $j' \in [\overline{m}]$. Suppose $|H(v', S')| = k$, then there exists k distinct indices $i_1, i_2, \ldots, i_k \in [n]$ and some k indices $j_1, j_2, \ldots j_k \in [\overline{m}]$ such that for each $l \in [k]$ there exists $s'_l \in \mathcal{S}'$ with $\text{supp}(s'_l \circ v') = \{(i_l, j_l)\}$. Consider a corresponding vector $v \in \mathbb{R}^n$ with only $v_{i_l} = 1$ for $l = 1, 2, \ldots, k$ and other places 0. Then we have, for each $l \in [k]$, $\text{supp}(s_l \circ v) = \{i_l\}$, where s_l is the corresponding set of s'_l. Hence $\text{HHdim}(\mathcal{S}') \leq \text{HHdim}(\mathcal{S})$. □

We can now apply our algorithm for subset-ℓ_0 on v', and obtain an algorithm for the subset-ℓ_1 norm.

Theorem 2.22. *There is an algorithm that when run on an insertion-only stream that accumulates to $v \in \mathbb{R}^n$ and has length* $\text{poly}(n)$*, the algorithm makes one pass using $O((\varepsilon^{-2} + \log\log n)\log n \cdot \text{HHdim}(\mathcal{S}))$ words of space, and then when queried for $s \in \mathcal{S}$, its output $\widehat{z}(s)$ satisfies*

$$\forall s \in \mathcal{S}, \quad \Pr[\widehat{z}(s) \in (1 \pm \varepsilon)\|s \circ v\|_1] \geq 0.9.$$

2.7 The entrywise update model

In this section, we show an algorithm that computes the subset-ℓ_p in the entrywise update model. For the entrywise update model, the algorithm for ℓ_p is essentially equivalent for all $p \geq 0$, because when an entry v_i arrives, the algorithm can simply raise it to power p for free. Therefore, we show only an algorithm for the subset-ℓ_1 problem, and the algorithms for all other p follows automatically. Unlike the priority sampling algorithm of [48, 1, 26], we simply reduce each

entrywise update to a sequence of updates to an insertion-only stream of a higher-dimensional vector.

Specifically, we assume that all entrywise updates are integers bounded by $\overline{m} = \text{poly}(n)$, which ensures that every entry can be stored using $O(\log n)$ bits. Given an entrywise update, say entry v_i in a vector $v \in \mathbb{R}^n$, our algorithm converts it to a sequence of v_i updates to a vector $v' \in \mathbb{R}^{n\times\overline{m}}$; similarly to Section 2.6, these updates are of the form $((i,j),1)$ for $j = 1, 2, \ldots, v_i$, and we also convert the set system $\mathcal{S} \in 2^{[n]}$ to a corresponding set system $\mathcal{S}' \subset 2^{[n\times\overline{m}]}$. By applying Theorem 2.22, we obtain the following corollary.

Corollary 2.23. *Let $\mathcal{S} \subset 2^{[n]}$ be an arbitrary set system. Let $p \geq 0$, $\varepsilon \in (0,1)$. In the entrywise update model of an vector $v \in \mathbb{R}^n$, there exists an algorithm that uses $O((\varepsilon^{-2} + \log\log n)\log n \cdot \text{HHdim}(\mathcal{S}))$ words of space and for each query $s \in \mathcal{S}$, it outputs a $(1\pm\varepsilon)$ approximation to $\|s \circ v\|_p$ with probability at least 0.9.*

3 Additive error for subset-ℓ_p norms

In this section we design additive-error approximation algorithms for subset-norms (in contrast to multiplicative approximation in the preceding section). Consider a set system $\mathcal{S} \subset 2^{[n]}$. If $\mathcal{S}$ includes the all-ones vector (i. e., the set $[n]$), then the space complexity of $(\varepsilon\|v\|_p)$-additive approximation of subset-ℓ_p norm is clearly at least that of $(1+\varepsilon)$-multiplicative approximation of subset-ℓ_p norm (simply because of the entire vector v). We present an algorithm that matches this lower bound, up to $\text{poly}(\varepsilon^{-1}\log n)$ factors, for every $p \in (0,\infty)$. It works for all possible subsets, i. e., the set system $\mathcal{S} = 2^{[n]}$.

Theorem 3.1. *Given $p \in (0,\infty)$ and $\varepsilon \in (0,1)$, Algorithm 4 makes a single pass over a data stream of additive updates to a vector $v \in \mathbb{R}^n$, and outputs a function $F : \{0,1\}^n \to \mathbb{R}$ that satisfies*

$$\forall s \in \{0,1\}^n, \qquad \Pr\left[F(s) \in \|s \circ v\|_p \pm 2\varepsilon\|v\|_p\right] \geq 0.75,$$

where the probability is over the algorithm's randomness. Moreover, the space complexity of the algorithm and the function F is $O_p(\varepsilon^{-3}\,\text{polylog}(n))$ bits for $p \leq 2$ and $O_p(\varepsilon^{-3}n^{1-2/p}\,\text{polylog}(n))$ bits for $p > 2$.

At a high-level, Algorithm 4 follows the framework developed in [36, 5, 20], where every coordinate is scaled at random and then the Count-Sketch algorithm is used to find heavy-hitters. We employ a specific simple method established in [20], and thus need the following definition and lemma from their work.

Definition 3.2 (α-inverse distribution [20])**.** Let $\alpha \in (0,\infty)$ be a parameter. The distribution of a random variable $X \in \mathbb{N} = \{1,2,\ldots\}$ is called *α-inverse* if

$$\forall x \in \mathbb{N}, \quad \Pr[X < x] = 1 - \frac{1}{x^\alpha}. \tag{3.1}$$

Lemma 3.3 (Lemma 4 in [20])**.** *Let $p \in (0,\infty)$ and $\varepsilon \in (0,1)$. Let $k \geq c_p\varepsilon^{-2}$ be an even integer for large enough $c_p > 0$ that depends only on p. Let $X \in \mathbb{R}^{k\times n}$ be a random matrix, whose entries have a*

p-inverse distribution and they are pairwise independent. Given $v \in \mathbb{R}^n$, let $X \circ v \in \mathbb{R}^{k\times n}$ be a matrix given by $(X \circ v)_{i,j} := X_{i,j}v_j$. Let Z be the $(k/2)$-largest entry in absolute value in $X \circ v$. Then

$$\Pr\left[2^{-1/p}|Z| \in (1 \pm \varepsilon)\|v\|_p\right] \geq 0.9.$$

We henceforth define $V := X \circ v \in \mathbb{R}^{kn}$, and view it as a vector by simply flattening the $k \times n$ matrix. Throughout, let $V_{\mathrm{tail}(k)}$ denote the vector obtained from V by zeroing the k entries of largest absolute value. Thus,

$$\|V_{\mathrm{tail}(k)}\|_2^2 := \sum_{j=k+1}^{kn} V_{[j]}^2,$$

where $U_{[j]}$ denotes the j-largest coordinate in absolute value in U.

Another ingredient is the famous Count-Sketch algorithm of [23], but we need a well-known and slightly stronger guarantee from [25], as follows.

Proposition 3.4 (Lemma 7 in [25])**.** *There is a one-pass algorithm, Count-Sketch, with parameters $k \in \mathbb{N}$ and $\varepsilon' \in (0,1)$, that given a stream of additive updates to a vector $V \in \mathbb{R}^{n'}$, uses space of $O(k/\varepsilon'^2 \cdot \log^3 n')$ bits to output an estimate $\widehat{V} \in \mathbb{R}^{n'}$, such that with high probability $1 - 1/n'^2$,*

$$\|\widehat{V} - V\|_\infty \leq \frac{\varepsilon'}{\sqrt{k}}\|V_{\mathrm{tail}(k)}\|_2.$$

We can now present our Algorithm 4 which is used in Theorem 3.1. The idea is to use the estimator from Lemma 3.3, but in order to save space, instead of storing $V = X \circ v \in \mathbb{R}^{kn}$ explicitly, it estimates this vector using the Count-Sketch algorithm (where the parameters k and ε' in Proposition 3.4 are set to k and $\varepsilon'(p,\varepsilon,n)$ given in line 5 of Algorithm 4). We write $V_{|s}$, for $s \in \{0,1\}^n$ (representing $s \subset [n]$), to denote the vector obtained by restricting V to entries corresponding to (i,j) where $i \in [k]$ and $j \in s$ (i. e., zeroing all other entries).

Before proving Theorem 3.1, we need the next lemma to bound the error of the Count-Sketch algorithm in our setting. Its proof follows from a direct calculation and appears in Section 3.1.

Lemma 3.5. *Let $V := X \circ v \in \mathbb{R}^{kn}$ be a vector defined as in Lemma 3.3. Then with probability at least 0.9,*

$$\left\|V_{\mathrm{tail}(k)}\right\|_2^2 \leq C_p \cdot \begin{cases} k\|v\|_p^2 & \text{if } p < 2, \\ k\|v\|_2^2 \leq k\|v\|_p^2 \cdot n^{1-2/p} & \text{if } p > 2, \\ k\|v\|_2^2 \cdot \log n & \text{if } p = 2. \end{cases}$$

holds for a suitable $C_p > 0$ that depends only on p.

Proof (of Theorem 3.1). We know by Lemma 3.3 that to estimate $\|v \circ s\|_p$, it suffices to find the $(k/2)$-largest coordinate in absolute value in $V_{|_s}$ for $k = \Theta(\varepsilon^{-2})$, and report its absolute value scaled by $2^{-1/p}$. Observe that the input to the Count-Sketch instance CS is exactly the vector $V = X \circ v$, and thus, with probability at least 0.99, its output vector $\widehat{V}$ satisfies

$$\|\widehat{V} - V\|_\infty \leq \frac{\varepsilon'}{\sqrt{k}}\|V_{\mathrm{tail}(k)}\|_2, \tag{3.2}$$

where ε' is defined in line 5 of Algorithm 4, and we assume henceforth this event occurs.

Now let $\widehat{z}$ be as in the algorithm, i. e., the $(k/2)$-largest coordinate in absolute value in $\widehat{V}_{|_S}$, and let z be similarly in $V_{|_S}$. It follows easily from (3.2) that

$$|\widehat{z} - z| \leq \frac{\varepsilon'}{\sqrt{k}} \|V_{\mathrm{tail}(k)}\|_2. \tag{3.3}$$

Indeed, by definition of z, at least $k/2$ coordinates in $V_{|_S}$ have value at least z, and (3.2) implies that all the corresponding coordinates in $\widehat{V}_{|_S}$ have value at least $z - \frac{\varepsilon'}{\sqrt{k}}\|V_{\mathrm{tail}(k)}\|_2$, which implies $\widehat{z} \geq z - \frac{\varepsilon'}{\sqrt{k}}\|V_{\mathrm{tail}(k)}\|_2$. The other direction is proved similarly.

We now bound the additive error in (3.3) using Lemma 3.5. By plugging in the value of ε' (depending on whether $p < 2$, $p > 2$, or $p = 2$) and setting a sufficiently small $c'_p > 0$, with probability at least 0.9,

$$\frac{\varepsilon'}{\sqrt{k}}\|V_{\mathrm{tail}(k)}\|_2 \leq c'_p C_p^{1/2} \cdot \varepsilon \, \|v\|_p \leq 2^{1/p} \cdot \varepsilon \|v\|_p,$$

thus $2^{-1/p}|\widehat{z} - z| \leq \varepsilon \, \|v\|_p$. The claimed overall accuracy now follows, via a union bound, from this last bound and Lemma 3.3.

To complete the proof, observe that the space complexity is dominated by that of the Count-Sketch instance, which is $O(k/\varepsilon'^2 \cdot \mathrm{polylog}(n))$ bits (Proposition 3.4), as required. □

Boosting the probability. The algorithm in Theorem 3.1 is oblivious to the vector s, and thus one can boost its success probability by parallel repetition and reporting the median estimate. In particular, with only an $O(\log |\mathcal{S}|)$ factor increase in the space, the additive approximation will hold simultaneously for all the sets $S \in \mathcal{S}$.

3.1 Proof of Lemma 3.5

Proof of Lemma 3.5. We will prove the lemma separately for $p > 2$ and for $p \leq 2$.

Case $p > 2$: By definition of V,

$$\|V\|_2^2 = \sum_{i\in[k],j\in[n]} V_{i,j}^2 \quad \text{and} \quad \mathbb{E}[\|V\|_2^2] = \sum_{i\in[k],j\in[n]} v_j^2 \, \mathbb{E}[X_{i,j}^2].$$

To calculate $\mathbb{E}[X_{i,j}^2]$, observe that (3.1) implies $\Pr[X_{i,j} = x] = 1/x^p - 1/(x+1)^p$ for all $x \in \mathbb{N}$, and therefore

$$\mathbb{E}[X_{i,j}^2] = \sum_{x=1}^{\infty} \left(\frac{x^2}{x^p} - \frac{x^2}{(x+1)^p}\right) = \sum_{x=1}^{\infty} \frac{x^2}{x^p} - \sum_{x=2}^{\infty} \frac{(x-1)^2}{x^p} \leq C'_p \sum_{x=1}^{\infty} \frac{1}{x^{p-1}} \leq C''_p.$$

Algorithm 4 Additive Subset-ℓ_p of $v \in \mathbb{R}^n$

Input: $p \in (0, \infty)$ and $\varepsilon \in (0, 1)$
Initialize:
$k \leftarrow \Theta(\varepsilon^{-2})$ ▷ k is an even integer
generate a random matrix $X \in \mathbb{R}^{k \times n}$ whose entries have a p-inverse distribution and they are pairwise independent
initialize a Count-Sketch instance CS for a vector $V \in \mathbb{R}^{kn}$ with parameters k and

$$\varepsilon' = c'_p \cdot \begin{cases} \varepsilon & \text{for } p < 2; \\ \varepsilon/(\log n)^{1/2} & \text{for } p = 2; \\ \varepsilon/n^{1/2-1/p} & \text{for } p > 2 \end{cases}$$

for a suitable constant $c'_p > 0$ that depends on p
Update(i, Δ**):**
feed the Count-Sketch instance CS with k updates: ▷ maintain $V = X \circ v$

$$\big((1,i), X_{1,i}\Delta\big), \big((2,i), X_{2,i}\Delta\big), \ldots, \big((k,i), X_{k,i}\Delta\big).$$

Query($s \in \{0,1\}^n$**):**
let $\widehat{V}$ be the estimate of $V \in \mathbb{R}^{kn}$ provided by CS
let $\widehat{z}$ be the $(k/2)$-largest coordinate in absolute value in $\widehat{V}_{|_S}$.
return $2^{-1/p}|\widehat{z}|$.

for some constants $C'_p, C''_p > 0$ that depend on p. Thus, $\mathbb{E}[\|V\|_2^2] \le C''_p k \|v\|_2^2$, and by Markov's inequality, we obtain as claimed

$$\Pr\left[\, \|V\|_2^2 \ge 10C''_p k \|v\|_2^2 \,\right] \le \frac{1}{10},$$

and we can also plug in the well-known comparison of norms $\|v\|_2 \le \|v\|_p^{1/2-1/p}$.

Case $p \le 2$: Define the set

$$W := \left\{(i,j) \in [k] \times [n] : \ |V_{i,j}| \ge 20^{1/p}\|v\|_p\right\}.$$

A simple calculation shows that

$$\mathbb{E}[|W|] = k \sum_{j\in[n]} \Pr\left[|v_j|X_{i,j} \ge 20^{1/p}\|v\|_p\right] = k \sum_{j\in[n]} \frac{|v_j|^p}{20\|v\|_p^p} = \frac{k}{20}.$$

By Markov's inequality, the event $\mathcal{E} = \{|W| \ge k\}$ has probability $\Pr[\mathcal{E}] \le \frac{1}{20}$. When the complement event $\bar{\mathcal{E}}$ occurs, every coordinate of $V_{\mathrm{tail}(k)}$ is not in W, i. e., has magnitude smaller than $20^{1/p}\|v\|_p$. Let us define the random variable

$$R := \sum_{i\in[k], j\in[n]} v_j^2 X_{i,j}^2 \cdot \mathbb{1}_{\{|v_j|X_{i,j} \le 20^{1/p}\|v\|_p\}}.$$

When $\bar{\mathcal{E}}$ occurs, clearly $\|V_{\mathrm{tail}(k)}\|_2^2 \le R$, and we thus wish to bound R (with high probability).

To this end, observe that for all $i \in [k]$, $j \in [n]$ with $v_j \ne 0$,

$$\begin{aligned}
\mathbb{E}\left[X_{i,j}^2 \cdot \mathbb{1}_{\{|v_j|X_{i,j} \le 20^{1/p}\|v\|_p\}}\right] &= \sum_{x=1}^{(20)^{1/p}\|v\|_p/|v_j|} \left(\frac{x^2}{x^p} - \frac{x^2}{(x+1)^p}\right) \\
&\le C'_p \cdot \sum_{x=1}^{(20)^{1/p}\|v\|_p/|v_j|} \frac{1}{x^{p-1}} \\
&\le C''_p \cdot \begin{cases} (\|v\|_p/|v_j|)^{2-p} & \text{if } p < 2, \\ \log n & \text{if } p = 2, \end{cases}
\end{aligned}$$

for some constants $C'_p, C''_p > 0$ that depend on p, where we used the fact that $m = \mathrm{poly}(n)$ and thus $\log m = O(\log n)$. It immediately follows that

$$\mathbb{E}[R] \le C''_p \cdot \begin{cases} k\|v\|_p^2 & \text{if } p < 2, \\ k\|v\|_2^2 \cdot \log n & \text{if } p = 2, \end{cases}$$

By Markov's inequality,

$$\Pr\left[R \geq 20C''_p \cdot k\|v\|_p^2\right] \leq \frac{1}{20} \qquad \text{for } p < 2$$
$$\text{and} \quad \Pr\left[R \geq 20C''_p \cdot k\|v\|_2^2 \log n\right] \leq \frac{1}{20} \qquad \text{for } p = 2.$$

Now by a union bound on the above event and $\bar{\mathcal{E}}$, with probability at least 0.9 we have

$$\|V_{\text{tail}(k)}\|_2^2 \leq R < \begin{cases} 20C''_p \cdot k\|v\|_p^2 & \text{if } p < 2, \\ 20C''_p \cdot k\|v\|_2^2 \log n & \text{if } p = 2, \end{cases}$$

which completes the proof of this case. □

4 Concluding remarks

To conclude, we study the universal streaming problem for subset-ℓ_p norms, i. e., providing a single summary of a stream of *insertion-only* updates to an input vector $v \in \mathbb{R}^n$, which suffices to approximate any *subset-ℓ_0 norm* from a given family $\mathcal{S} \subset 2^{[n]}$. (Recall that a subset-$\ell_p$ norm of v is the ℓ_p norm of the vector induced by a subset of coordinates $S \in \mathcal{S}$.) We prove that the space complexity of this problem is characterized by the *heavy-hitter dimension* of the set $\mathcal{S}$, a notion that we introduce and define as the maximum number (over all $v \in \mathbb{R}^n$) of distinct heavy-hitters with respect to all subsets $S \in \mathcal{S}$. We further show that this characterization holds also for subset-ℓ_1 norms in the same insertion-only setting. However, it *does not* hold for more general streaming models, namely, for the *turnstile setting* and the *sliding-window setting*, and thus there is a strict separation between these models.

For subset-ℓ_p norms with general p, namely, every $p \in (0, \infty)\backslash\{1\}$, we prove that the *heavy-hitter dimension* characterizes the space complexity of universal streaming in the *entrywise* updates model, where each coordinate of the vector is updated at most once. In the more general model of insertion-only updates, it is remains open whether subset-ℓ_p norms for $p \neq 0, 1$ admits uniform streaming with space complexity $\widetilde{O}(\text{HHdim}(\mathcal{S}))$. For example, the major obstacle for subset-ℓ_2 norms is how to maintain the distinct ℓ_2-heavy-hitters for every subset of coordinates $S \in \mathcal{S}$. We leave this problem for future investigations.

References

[1] NOGA ALON, NICK DUFFIELD, CARSTEN LUND, AND MIKKEL THORUP: Estimating arbitrary subset sums with few probes. In *Proc. 24th Symp. on Principles of Database Systems (PODS'05)*, pp. 317–325. ACM Press, 2005. [doi:10.1145/1065167.1065209] 2, 6, 20

[2] NOGA ALON, YOSSI MATIAS, AND MARIO SZEGEDY: The space complexity of approximating the frequency moments. *J. Comput. System Sci.*, 58(1):137–147, 1999. Preliminary version in STOC'96. [doi:10.1006/jcss.1997.1545] 2, 5, 7

[3] ALEXANDR ANDONI: High frequency moments via max-stability. In *Proc. 42nd Internat. Conf. on Acoustics, Speech and Signal Processing (ICASSP'17)*, pp. 6364–6368. IEEE Comp. Soc., 2017. [doi:10.1109/ICASSP.2017.7953381] 6, 8

[4] ALEXANDR ANDONI, JIECAO CHEN, ROBERT KRAUTHGAMER, BO QIN, DAVID P. WOODRUFF, AND QIN ZHANG: On sketching quadratic forms. In *Proc. 7th Innovations in Theoret. Comp. Sci. Conf. (ITCS'16)*, pp. 311–319. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016. [doi:10.1145/2840728.2840753, arXiv:1511.06099] 17

[5] ALEXANDR ANDONI, ROBERT KRAUTHGAMER, AND KRZYSZTOF ONAK: Streaming algorithms via precision sampling. In *Proc. 52nd FOCS*, pp. 363–372. IEEE Comp. Soc., 2011. [doi:10.1109/FOCS.2011.82] 6, 8, 21

[6] ZIV BAR-YOSSEF, T. S. JAYRAM, ROBERT KRAUTHGAMER, AND RAVI KUMAR: The sketching complexity of pattern matching. In *Proc. 8th Internat. Workshop on Randomization and Computation (RANDOM'04)*, pp. 261–272. Springer, 2004. [doi:10.1007/978-3-540-27821-4_24] 8, 18

[7] ZIV BAR-YOSSEF, T. S. JAYRAM, RAVI KUMAR, AND D. SIVAKUMAR: An information statistics approach to data stream and communication complexity. *J. Comput. System Sci.*, 68(4):702–732, 2004. Preliminary version in FOCS'02. [doi:10.1016/j.jcss.2003.11.006] 5

[8] ZIV BAR-YOSSEF, T. S. JAYRAM, RAVI KUMAR, D. SIVAKUMAR, AND LUCA TREVISAN: Counting distinct elements in a data stream. In *Proc. 6th Internat. Workshop on Randomization and Computation (RANDOM'02)*, pp. 1–10. Springer, 2002. [doi:10.1007/3-540-45726-7_1] 7

[9] LAKSHMINATH BHUVANAGIRI, SUMIT GANGULY, DEEPANJAN KESH, AND CHANDAN SAHA: Simpler algorithm for estimating frequency moments of data streams. In *Proc. 17th Ann. ACM–SIAM Symp. on Discrete Algorithms (SODA'06)*, pp. 708–713. SIAM, 2006. [doi:10.1145/1109557.1109634] 6

[10] JAROSŁAW BŁASIOK: Optimal streaming and tracking distinct elements with high probability. *ACM Trans. Algorithms*, 16(1):3:1–28, 2020. Preliminary version in SODA'18. [doi:10.1145/3309193, arXiv:1804.01642] 7

[11] JAROSŁAW BŁASIOK, VLADIMIR BRAVERMAN, STEPHEN R. CHESTNUT, ROBERT KRAUTHGAMER, AND LIN F. YANG: Streaming symmetric norms via measure concentration. In *Proc. 49th STOC*, pp. 716–729. ACM Press, 2017. [doi:10.1145/3055399.3055424, arXiv:1511.01111] 2

[12] MARK BRAVERMAN, ANKIT GARG, DENIS PANKRATOV, AND OMRI WEINSTEIN: Information lower bounds via self-reducibility. *Theory Computing Sys.*, 59(2):377–396, 2016. Preliminary version in CSR'13. [doi:10.1007/s00224-015-9655-z, ECCC:TR12-177] 17

[13] VLADIMIR BRAVERMAN AND STEPHEN R. CHESTNUT: Universal sketches for the frequency negative moments and other decreasing streaming sums. In *Proc. 19th Internat. Workshop on Randomization and Computation (RANDOM'15)*, pp. 591–605. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015. [doi:10.4230/LIPIcs.APPROX-RANDOM.2015.591] 2

[14] VLADIMIR BRAVERMAN, STEPHEN R. CHESTNUT, NIKITA IVKIN, JELANI NELSON, ZHENGYU WANG, AND DAVID P. WOODRUFF: BPTree: An ℓ_2 heavy hitters algorithm using constant memory. In *Proc. 36th Symp. on Principles of Database Systems (PODS'17)*, pp. 361–376. ACM Press, 2017. [doi:10.1145/3034786.3034798, arXiv:1603.00759] 6

[15] VLADIMIR BRAVERMAN, STEPHEN R. CHESTNUT, DAVID P. WOODRUFF, AND LIN F. YANG: Streaming space complexity of nearly all functions of one variable on frequency vectors. In *Proc. 35th Symp. on Principles of Database Systems (PODS'16)*, pp. 261–276. ACM Press, 2016. [doi:10.1145/2902251.2902282, arXiv:1601.07473] 2, 6

[16] VLADIMIR BRAVERMAN, JONATHAN KATZMAN, CHARLES SEIDELL, AND GREGORY VORSANGER: An optimal algorithm for large frequency moments using $O(n^{1-2/k})$ bits. In *Proc. 18th Internat. Workshop on Randomization and Computation (RANDOM'14)*, pp. 531–544. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2014. [doi:10.4230/LIPIcs.APPROX-RANDOM.2014.531] 6

[17] VLADIMIR BRAVERMAN AND RAFAIL OSTROVSKY: Smooth histograms for sliding windows. In *Proc. 48th FOCS*, pp. 283–293. IEEE Comp. Soc., 2007. [doi:10.1109/FOCS.2007.55] 4

[18] VLADIMIR BRAVERMAN AND RAFAIL OSTROVSKY: Zero-one frequency laws. In *Proc. 42nd STOC*, pp. 281–290. ACM Press, 2010. [doi:10.1145/1806689.1806729] 2

[19] VLADIMIR BRAVERMAN, RAFAIL OSTROVSKY, AND ALAN ROYTMAN: Zero-one laws for sliding windows and universal sketches. In *Proc. 19th Internat. Workshop on Randomization and Computation (RANDOM'15)*, pp. 573–590. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015. [doi:10.4230/LIPIcs.APPROX-RANDOM.2015.573] 2

[20] VLADIMIR BRAVERMAN, EMANUELE VIOLA, DAVID P. WOODRUFF, AND LIN F. YANG: Revisiting frequency moment estimation in random order streams. In *Proc. 45th Internat. Colloq. on Automata, Languages, and Programming (ICALP'18)*, pp. 25:1–14. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018. [doi:10.4230/LIPIcs.ICALP.2018.25, arXiv:1803.02270] 8, 21

[21] AMIT CHAKRABARTI, SUBHASH KHOT, AND XIAODONG SUN: Near-optimal lower bounds on the multi-party communication complexity of set disjointness. In *Proc. 18th IEEE Conf. on Comput. Complexity (CCC'03)*, pp. 107–117. IEEE Comp. Soc., 2003. [doi:10.1109/CCC.2003.1214414] 5, 6

[22] AMIT CHAKRABARTI AND ODED REGEV: An optimal lower bound on the communication complexity of GAP-HAMMING-DISTANCE. *SIAM J. Comput.*, 41(5):1299–1317, 2012. [doi:10.1137/120861072] 17

[23] MOSES CHARIKAR, KEVIN CHEN, AND MARTIN FARACH-COLTON: Finding frequent items in data streams. *Theoret. Comput. Sci.*, 312(1):3–15, 2004. Preliminary version in ICALP'02. [doi:10.1016/S0304-3975(03)00400-6] 6, 22

[24] EDITH COHEN, NICK DUFFIELD, HAIM KAPLAN, CARSTENT LUND, AND MIKKEL THORUP: Algorithms and estimators for summarization of unaggregated data streams. *J. Comput. System Sci.*, 80(7):1214–1244, 2014. [doi:10.1016/j.jcss.2014.04.009] 2

[25] GRAHAM CORMODE AND S. MUTHUKRISHNAN: Combinatorial algorithms for compressed sensing. In *Proc. 13th Internat. Colloq. on Structural Information and Communication Complexity (SIROCCO'06)*, pp. 280–294. Springer, 2006. [doi:10.1007/11780823_22] 22

[26] NICK DUFFIELD, CARSTEN LUND, AND MIKKEL THORUP: Priority sampling for estimation of arbitrary subset sums. *J. ACM*, 54(6):32, 2007. [doi:10.1145/1314690.1314696] 2, 3, 4, 5, 6, 8, 20

[27] JOAN FEIGENBAUM, SAMPATH KANNAN, ANDREW MCGREGOR, SIDDHARTH SURI, AND JIAN ZHANG: On graph problems in a semi-streaming model. *Theoret. Comput. Sci.*, 348(2–3):207–216, 2005. Preliminary version in ICALP'04. [doi:10.1016/j.tcs.2005.09.013] 2

[28] PHILIPPE FLAJOLET AND G. NIGEL MARTIN: Probabilistic counting algorithms for data base applications. *J. Comput. System Sci.*, 31(2):182–209, 1985. [doi:10.1016/0022-0000(85)90041-8] 7, 14

[29] ARPIT GUPTA, ROB HARRISON, MARCO CANINI, NICK FEAMSTER, JENNIFER REXFORD, AND WALTER WILLINGER: Sonata: Query-driven streaming network telemetry. In *Proc. 31st SIG Data Communication Conf. (SIGCOMM'18)*, pp. 357–371. ACM Press, 2018. [doi:10.1145/3230543.3230555, arXiv:1705.01049] 3

[30] SARIEL HAR-PELED AND SOHAM MAZUMDAR: On coresets for k-means and k-median clustering. In *Proc. 36th STOC*, pp. 291–300. ACM Press, 2004. [doi:10.1145/1007352.1007400] 2

[31] PIOTR INDYK: Stable distributions, pseudorandom generators, embeddings, and data stream computation. *J. ACM*, 53(3):307–323, 2006. Preliminary version in FOCS'00. [doi:10.1145/1147954.1147955] 2, 5

[32] PIOTR INDYK AND DAVID P. WOODRUFF: Tight lower bounds for the distinct elements problem. In *Proc. 44th FOCS*, pp. 283–288. IEEE Comp. Soc., 2003. [doi:10.1109/SFCS.2003.1238202] 5, 9, 17

[33] PIOTR INDYK AND DAVID P. WOODRUFF: Optimal approximations of the frequency moments of data streams. In *Proc. 37th STOC*, pp. 202–208. ACM Press, 2005. [doi:10.1145/1060590.1060621] 2, 5

[34] T. S. JAYRAM AND DAVID P. WOODRUFF: Optimal bounds for Johnson-Lindenstrauss transforms and streaming problems with subconstant error. *ACM Trans. Algorithms*, 9(3):26:1–17, 2013. Preliminary version in SODA'11. [doi:10.1145/2483699.2483706] 7

[35] DANIEL M. KANE, JELANI NELSON, ELY PORAT, AND DAVID P. WOODRUFF: Fast moment estimation in data streams in optimal space. In *Proc. 43rd STOC*, pp. 745–754. ACM Press, 2011. [doi:10.1145/1993636.1993735, arXiv:1007.4191] 6

[36] DANIEL M. KANE, JELANI NELSON, AND DAVID P. WOODRUFF: On the exact space complexity of sketching and streaming small norms. In *Proc. 21st Ann. ACM–SIAM Symp. on Discrete Algorithms (SODA'10)*, pp. 1161–1178. SIAM, 2010. [doi:10.1137/1.9781611973075.93] 6, 21

[37] DANIEL M. KANE, JELANI NELSON, AND DAVID P. WOODRUFF: An optimal algorithm for the distinct elements problem. In *Proc. 29th Symp. on Principles of Database Systems (PODS'10)*, pp. 41–52. ACM Press, 2010. [doi:10.1145/1807085.1807094] 6, 7

[38] ILAN KREMER, NOAM NISAN, AND DANA RON: On randomized one-round communication complexity. *Comput. Complexity*, 8(1):21–49, 1999. Preliminary version in STOC'95, Errata in *Comput. Complexity* 10 (2001), 314–315. [doi:10.1007/s000370050018] 16

[39] ZAOXING LIU, ANTONIS MANOUSIS, GREGORY VORSANGER, VYAS SEKAR, AND VLADIMIR BRAVERMAN: One sketch to rule them all: Rethinking network flow monitoring with univmon. In *Proc. 29th SIG Data Communication Conf. (SIGCOMM'16)*, pp. 101–114. ACM Press, 2016. [doi:10.1145/2934872.2934906] 2

[40] PETER BRO MILTERSEN, NOAM NISAN, SHMUEL SAFRA, AND AVI WIGDERSON: On data structures and asymmetric communication complexity. *J. Comput. System Sci.*, 57(1):37–49, 1998. Preliminary version in STOC'95. [doi:10.1006/jcss.1998.1577] 18

[41] SRINIVAS NARAYANA, ANIRUDH SIVARAMAN, VIKRAM NATHAN, PRATEESH GOYAL, VENKAT ARUN, MOHAMMAD ALIZADEH, VIMALKUMAR JEYAKUMAR, AND CHANGHOON KIM: Language-directed hardware design for network performance monitoring. In *Proc. 30th SIG Data Communication Conf. (SIGCOMM'17)*, pp. 85–98. ACM Press, 2017. [doi:10.1145/3098822.3098829] 3

[42] JELANI NELSON: List of open problems in sublinear algorithms: Problem 30. `sublinear.info`, accessed 2018-06-20. 2

[43] NORBERT SAUER: On the density of families of sets. *J. Combin. Theory–A*, 13(1):145–147, 1972. [doi:10.1016/0097-3165(72)90019-2] 19

[44] VYAS SEKAR, NICK G. DUFFIELD, OLIVER SPATSCHECK, JACOBUS E. VAN DER MERWE, AND HUI ZHANG: LADS: Large-scale automated DDoS detection system. In *Proc. 12th USENIX Annual Technical Conference, General Track (UATC'06)*, pp. 171–184. USENIX Association, 2006. USENIX. 3

[45] VYAS SEKAR, MICHAEL K. REITER, AND HUI ZHANG: Revisiting the case for a minimalist approach for network flow monitoring. In *Proc. 10th SIGCOMM Internet Measurement Conf. (IMC'10)*, pp. 328–341. ACM Press, 2010. [doi:10.1145/1879141.1879186] 2

[46] SAHARON SHELAH: A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific J. Math*, 41(1):247–261, 1972. [doi:10.2140/pjm.1972.41.247] 19

[47] ANSHUMALI SHRIVASTAVA, ARND CHRISTIAN KONIG, AND MIKHAIL BILENKO: Time adaptive sketches (ada-sketches) for summarizing data streams. In *Proc. 41st Internat. Conf. on Management of Data (SIGMOD'16)*, pp. 1417–1432. ACM Press, 2016. [doi:10.1145/2882903.2882946] 3

[48] MARIO SZEGEDY: The DLT priority sampling is essentially optimal. In *Proc. 38th STOC*, pp. 150–158. ACM Press, 2006. [doi:10.1145/1132516.1132539] 2, 4, 5, 6, 8, 20

[49] DANIEL TING: Data sketches for disaggregated subset sum and frequent item estimation. In *Proc. 44th Internat. Conf. on Management of Data (SIGMOD'18)*, pp. 1129–1140. ACM Press, 2018. [doi:10.1145/3183713.3183759, arXiv:1709.04048] 2, 3, 6

[50] DAVID VENGEROV, ANDRE CAVALHEIRO MENCK, MOHAMED ZAIT, AND SUNIL P. CHAKKAPPEN: Join size estimation subject to filter conditions. *Proc. VLDB Endowment*, 8(12):1530–1541, 2015. [doi:10.14778/2824032.2824051] 3

[51] DAVID P. WOODRUFF: Optimal space lower bounds for all frequency moments. In *Proc. 15th Ann. ACM–SIAM Symp. on Discrete Algorithms (SODA'04)*, pp. 167–175. SIAM, 2004. ACM DL. 5, 7

AUTHORS

Vladimir Braverman
Associate professor
Department of Computer Science
Johns Hopkins University
Baltimore, Maryland, USA
vova@cs.jhu.edu
https://www.cs.jhu.edu/~vova/

Robert Krauthgamer
Professor
Department of Computer Science and Applied Mathematics
The Weizmann Institute of Science
Rehovot, Israel
robert.krauthgamer@weizmann.ac.il
http://www.wisdom.weizmann.ac.il/~robi/

Lin F. Yang
Assistant Professor
Department of Electrical and Computer Engineering
University of California, Los Angeles
Los Angeles, California, USA
linyang@ee.ucla.edu
http://drlinyang.net

ABOUT THE AUTHORS

VLADIMIR BRAVERMAN is an associate professor in the Department of Computer Science at Johns Hopkins University, with a secondary appointment in the Department of Applied Mathematics and Statistics. He is a member of the Algorithms and Complexity group, the Johns Hopkins Mathematical Institute for Data Science (MINDS), the Institute for Data Intensive Engineering and Science (IDIES) and the Machine Learning group. Braverman graduated from UCLA; before that, he was leading a research group at HyperRoll, a startup that was acquired by Oracle in 2009. Braverman's research interests include efficient sublinear algorithms (such as sketches and coresets) and their applications to data science. He is a recipient of an NSF CAREER award, a Google Faculty Award and a Cisco Faculty Award.

ROBERT KRAUTHGAMER (called "Robi" by his friends and colleagues) received his Ph. D. at the Weizmann Institute of Science in 2001 under Uriel Feige. He was subsequently a postdoc in Berkeley's theory group, and then a Research Staff Member at the theory group in the IBM Almaden Research Center. Since 2007, he has been a faculty member at the Weizmann Institute of Science. Robi's main research area is the design of algorithms for problems involving combinatorial optimization, finite metric spaces, high-dimensional geometry, data analysis, and related areas. His favorite sport since youth has been swimming. Once he swam across the Sea of Galilee in a 10 km competitive race, and was the last one to arrive at the finish line.

LIN F. YANG is an assistant professor in the Electrical and Computer Engineering Department at the University of California, Los Angeles. His research focuses on developing and applying fast algorithms for machine learning and data science. His current research focuses on reinforcement learning theory and applications, learning for control, non-convex optimization, and streaming algorithms. He received a Ph. D. in physics and a Ph. D. in computer science from Johns Hopkins University and was a postdoctoral fellow at the Department of Operations Research and Financial Engineering, Princeton University. He was a recipient of the Simons Research Fellowship, the Dean Robert H. Roy Fellowship, and the JHU MINDS best dissertation award.