Towards Optimal Moment Estimation in Streaming and Distributed Models

Rajesh Jayaram
Carnegie Mellon University, Pittsburgh, PA, USA
http://rajeshjayaram.com/
rkjayara@cs.cmu.edu

David P. Woodruff
Carnegie Mellon University, Pittsburgh, PA, USA
http://www.cs.cmu.edu/~dwoodruf/
dwoodruf@cs.cmu.edu

Abstract

One of the oldest problems in the data stream model is to approximate the p-th moment $\|\mathcal{X}\|_p^p = \sum_{i=1}^n \mathcal{X}_i^p$ of an underlying non-negative vector $\mathcal{X} \in \mathbb{R}^n$, which is presented as a sequence of poly(n) updates to its coordinates. Of particular interest is when $p \in (0, 2]$. Although a tight space bound of $\Theta(\epsilon^{-2} \log n)$ bits is known for this problem when both positive and negative updates are allowed, surprisingly there is still a gap in the space complexity of this problem when all updates are positive. Specifically, the upper bound is $O(\epsilon^{-2} \log n)$ bits, while the lower bound is only $\Omega(\epsilon^{-2} + \log n)$ bits. Recently, an upper bound of $\tilde{O}(\epsilon^{-2} + \log n)$ bits was obtained under the assumption that the updates arrive in a *random order*.

We show that for $p \in (0, 1]$, the random order assumption is not needed. Namely, we give an upper bound for worst-case streams of $\tilde{O}(\epsilon^{-2} + \log n)$ bits for estimating $\|\mathcal{X}\|_p^p$. Our techniques also give new upper bounds for estimating the empirical entropy in a stream. On the other hand, we show that for $p \in (1, 2]$, in the natural coordinator and blackboard distributed communication topologies, there is an $\tilde{O}(\epsilon^{-2})$ bit max-communication upper bound based on a randomized rounding scheme. Our protocols also give rise to protocols for heavy hitters and approximate matrix product. We generalize our results to arbitrary communication topologies G, obtaining an $\tilde{O}(\epsilon^2 \log d)$ max-communication upper bound, where d is the diameter of G. Interestingly, our upper bound rules out natural communication complexity-based approaches for proving an $\Omega(\epsilon^{-2} \log n)$ bit lower bound for $p \in (1, 2]$ for streaming algorithms. In particular, any such lower bound must come from a topology with large diameter.

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

Keywords and phrases Streaming, Sketching, Message Passing, Moment Estimation

Digital Object Identifier 10.4230/LIPIcs.APPROX-RANDOM.2019.29

Category APPROX

Related Version A full version of the paper is available at https://arxiv.org/abs/1907.05816.

Funding The authors thank the partial support by the National Science Foundation under Grant No. CCF-1815840.

1 Introduction

The streaming and distributed models of computation have become increasingly important for the analysis of massive datasets, where the sheer size of the input imposes stringent restrictions on the resources available to algorithms. Examples of such datasets include internet traffic logs, sensor networks, financial transaction data, database logs, and scientific data streams (such as huge experiments in particle physics, genomics, and astronomy). Given

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2019).
Editors: Dimitris Achlioptas and László A. Végh; Article No. 29; pp. 29:1–29:21
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

their prevalence, there is a large body of literature devoted to designing extremely efficient algorithms for analyzing streams and enormous datasets. We refer the reader to [4, 52] for surveys of these algorithms and their applications.

Formally, the data stream model studies the evolution of a vector $\mathcal{X} \in \mathbb{Z}^n$, called the frequency vector. Initially, $\mathcal{X}$ is initialized to be the zero-vector. The frequency vector then receives a stream of m coordinate-wise updates of the form $(i_t, \Delta_t) \in [n] \times \{-M, \ldots, M\}$ for some $M > 0$ and time step $t \in [m]$. Each update (i_t, Δ_t) causes the change $\mathcal{X}_{i_t} \leftarrow \mathcal{X}_{i_t} + \Delta_t$. If we restrict that $\Delta_t \geq 0$ for all $t \in [m]$, this is known as the *insertion-only* model. If the updates $\Delta_t \in \{-M, \ldots, M\}$ can be both positive and negative, then this is known as the *turnstile*-model. The p-th frequency moment of the frequency vector at the end of the stream, F_p, is defined as $F_p = \sum_{i=1}^n |\mathcal{X}_i|^p$. For simplicity (but not necessity), it is generally assumed that $m, M = \text{poly}(n)$.

The study of frequency moments in the streaming model was initiated by the seminal 1996 paper of Alon, Matias, and Szegedy [1]. Since then, nearly two decades of research have been devoted to understanding the space and time complexity of this problem. An incomplete list of works which study frequency moments in data streams includes [16, 36, 6, 58, 35, 45, 12, 44, 11, 15, 11, 7, 13]. For $p > 2$, it is known that polynomial in n (rather than logarithmic) space is required for F_p estimation [16, 36]. In the regime of $p \in (0, 2]$, the space complexity of F_p estimation in the turnstile model is now understood, with matching upper and lower bounds of $\Theta(\epsilon^{-2} \log(n))$ bits to obtain a $(1 \pm \epsilon)$ approximation of F_p. Here, for $\epsilon > 0$, a $(1 \pm \epsilon)$ approximation means an estimate $\tilde{F}_p$ such that $(1 - \epsilon)F_p \leq \tilde{F}_p \leq (1 + \epsilon)F_p$. For insertion only streams, however, the best known lower bound is $\Omega(\epsilon^{-2} + \log(n))$ [58]. Moreover, if the algorithm is given query access to an arbitrarily long string of random bits (known as the random oracle model), then the lower bound is only $\Omega(\epsilon^{-2})$. On the other hand, the best upper bound is to just run the turnstile $O(\epsilon^{-2} \log(n))$-space algorithm.

In this work, we make progress towards resolving this fundamental problem. For $p < 1$, we resolve the space complexity by giving an $\tilde{O}(\epsilon^{-2} + \log n)$[1]-bits of space upper bound. In the random oracle model, our upper bound is $\tilde{O}(\epsilon^{-2})$[2], which also matches the lower bound in this setting. Prior to this work, an $\tilde{O}(\epsilon^{-2} + \log(n))$ upper bound for F_p estimation was only known in the restricted *random-order* model, where it is assumed that the stream updates are in a uniformly random ordering [13]. Our techniques are based on novel analysis of the behavior of the p-stable random variables used in the $O(\epsilon^{-2} \log(n))$ upper bound of [35], and also give rise to a space optimal algorithm for entropy estimation.

We remark that F_p estimation in the range $p \in (0, 1)$ is useful for several reasons. Firstly, for p near 1, F_p estimation is often used as a subroutine for estimating the empirical entropy of a stream, which itself is useful for network anomaly detection ([47], also see [31] and the references therein). Moment estimation is also used in weighted sampling algorithms for data streams [50, 42, 38] (see [23] for a survey of such samplers and their applications). Here, the goal is to sample an index $i \in [n]$ with probability $|\mathcal{X}_i|^p / F_p$. These samplers can be used to find heavy-hitters in the stream, estimate cascaded norms [2, 50], and design representative histograms of $\mathcal{X}$ on which more complicated algorithms are run [28, 27, 55, 29, 33, 24]. Furthermore, moment estimation for fractional p, such as $p = .5$ and $p = .25$, has been shown to be useful for data mining [22].

---

[1] the $\tilde{O}$ here suppresses a single $(\log\log n + \log 1/\epsilon)$ factor, and in general we use $\tilde{O}$ and $\tilde{\Omega}$ to hide $\log\log n$ and $\log 1/\epsilon$ terms.

[2] This space complexity is measured *between updates*. To read and process the $\Theta(\log(n))$-bit identity of an update, the algorithm will use an additional $O(\log(n))$-bit working memory tape during an update. Note that all lower bounds only apply to the space complexity between updates, and allow arbitrary space to process updates.

For the range of $p \in (1, 2]$, we prove an $\tilde{O}(\epsilon^{-2})$-bits of max-communication upper bound in the distributed models most frequently used to prove *lower bounds* for streaming. This result rules out a large and very commonly used class of approaches for proving lower bounds against the space complexity of streaming algorithms for F_p estimation. Our approach is based on a randomized rounding scheme for p-stable sketches. We show that our rounding scheme can be additionally applied to design improved protocols for the distributed heavy hitters and approximate matrix product problems. We now introduce the model in which all the aforementioned results hold.

1.1 Multi-Party Communication

In this work, we study a more general model than streaming, known as the message passing multi-party communication model. All of our upper bounds apply to this model, and our streaming algorithms are just the result of special cases of our communication protocols. In the message passing model, there are m players, each positioned at a unique vertex in a graph $G = (V, E)$. The i-th player is given as input an integer vector $X_i \in \mathbb{Z}^n$. The goal of the players is to work together to jointly approximate some function $f : \mathbb{R}^n \to \mathbb{R}$ of the aggregate vector $\mathcal{X} = \sum_{i=1}^n X_i$, such as the p-th moment $f(\mathcal{X}) = F_p = \|\mathcal{X}\|_p^p = \sum_{i=1}^n |\mathcal{X}_i|^p$. In the message passing model, as opposed to the *broadcast* model of communication, the players are only allowed to communicate with each other over the edges of G. Thus player i can send a message to player j only if $(i, j) \in E$, and this message will only be received by player j (and no other). At the end of the protocol, it is assumed that at least one player holds the approximation to $f(\mathcal{X})$. The goal of multi-party communication is to solve the approximation problem using small total communication between all the players over the course of the execution. More specifically, the goal is to design protocols that use small *max-communication*, which is the total number of bits sent over any edge of G. Our protocols hold in an even more restricted setting, known as the *one-shot* setting, where each player is allowed to communicate exactly once over the course of the entire protocol.

We now observe that data streams can be modeled as a special case of one-shot multi-party communication. Here, the graph G in question is the line graph on m vertices. If the updates to the data stream vector are $(i_1, \Delta_1), \ldots, (i_m, \Delta_m)$, then the t-th player has input $X_t \in \mathbb{Z}^n$, where $(X_t)_{i_t} = \Delta_t$ and $(X_t)_j = 0$ for $j \neq i_t$. The aggregate vector $\mathcal{X} = \sum_{i=1}^m X_i$ is just the frequency vector at the end of the stream, and the space complexity of any algorithm is just the max-communication used over any edge of the corresponding communication protocol. Since we are primarily interested in insertion only streams, in this work we will consider the *non-negative data* model, where $X_i \in \{0, 1, \ldots, M\}^n$ for all input vectors X_i, for some $M > 0$ (as in streaming, we assume $M = \text{poly}(n, m)$ for simplicity). Note that an equivalent condition is that each $X_i \in \mathbb{R}^n_{\geq 0}$ such that the entries of X_i can be stored in $O(\log M)$-bits.

We are now ready to introduce our results for moment estimation in the message passing model. Let d be the *diameter* of the communication graph G. Our first result is a protocol for F_p estimation when $p \in (1, 2]$ which uses a max communication of $\tilde{O}(\epsilon^{-2} \log d)$ bits. Using similar techniques, we also obtain a (optimal for $d = \Theta(1)$) bound of $\tilde{O}(\epsilon^{-2} \log n \log d)$ for the heavy hitters problem, which is to find the coordinates of $\mathcal{X}$ which contribute at least an ϵ fraction of the total $\sqrt{F_2} = \|\mathcal{X}\|_2$ of $\mathcal{X}$. For $p \in (0, 1)$, we give an $\tilde{O}(\epsilon^{-2})$ upper bound for F_p estimation. Notice that this is independent of the graph topology, and thus holds for the line graph, where we derive our $\tilde{O}(\epsilon^{-2})$ upper bound for F_p estimation in the random oracle streaming model. We then show how the streaming algorithm can be derandomized to not require a random oracle, now using an optimal $\tilde{O}(\epsilon^{-2} + \log(n))$-bits of space. Our techniques also result in an $\tilde{O}(\epsilon^{-2})$ upper bound for additively approximating the empirical entropy of the vector $\mathcal{X}$.

Our results for $p \in (1, 2]$ have interesting implications for any attempts to prove *lower-bounds* for streaming algorithms that estimate F_p, which we now describe. The link between streaming and communication complexity is perhaps one of the most fruitful sources of space lower bounds for algorithms in computer science. Namely, nearly all lower bounds for the space complexity of randomized streaming algorithms are derived via reductions from communication problems. For an incomplete list of such reductions, see [58, 61, 45, 42, 46, 10, 16, 57, 48, 49, 40] and the references therein. Now nearly all such lower bounds (and all of the ones that were just cited) hold in either the 2-party setting (G has 2 vertices), the coordinator model, or the black-board model. In the coordinator model there are m players, each with a single edge to a central coordinator (i.e., G is a star graph on $m + 1$ vertices). Note that the diameter d of the coordinator graph is 2. In the multi-player black-board model, every message that is sent is written to a shared blackboard that can be read by all players. Observe that any one-way protocol for the coordinator model immediately results in a protocol with the same communication for the blackboard model. Namely, each player simply writes what it would have sent to the coordinator on the blackboard, and at the end of the protocol the blackboard contains all the information that the coordinator would have had. For these three settings, our protocol gives an $\tilde{O}(\epsilon^{-2})$ max-communication upper bound for F_p estimation, $p \in (1, 2]$. This completely rules out the approach for proving lower bounds against F_p estimation in a stream via any of these three techniques. In particular, it appears that any lower bound for F_p estimation via communication complexity in this regime of p will need to use a graph with $\Omega(n)$ diameter, such as the line graph, without a black-board.

The coordinator and black-board models have also been studied in many other settings than for proving lower bounds against streaming. For instance, in the *Distributed Functional Monitoring* literature [25, 63, 60, 34, 56, 37], each player is receiving a continuous stream of updates to their inputs X_i, and the coordinator must continuously update its approximation to $f(\mathcal{X})$. The black-board model is also considered frequently for designing communication upper bounds, such as those for set disjointness [6, 16, 30]. Finally, there is substantial literature which considers numerical linear algebra and clustering problems in the coordinator model [61, 20, 5, 62]. Thus, our upper bounds can be seen as a new and useful contribution to these bodies of literature as well.

1.2 Our Contributions

As noted, the upper bounds in this paper all hold in the general multi-party message passing model, over an arbitrary topology G. Our algorithms also have the additional property that they are *one-shot*, meaning that each player is allowed to communicate exactly once. Our protocols pre-specify a central vertex $\mathcal{C} \in V$ of G. Specifically, $\mathcal{C}$ will be a *center* of G, which is a vertex with minimal max-distance to any other vertex. Our protocols then proceed in d rounds, where d is the diameter of G. Upon termination of the protocols, the central vertex $\mathcal{C}$ will hold the estimate of the protocol. We note that $\mathcal{C}$ can be replaced by any other vertex v, and d will then be replaced by the max distance of any other vertex to v. A summary of our results is given in Table 1.

We first formally state our general result for F_p estimation, $1 < p \leq 2$. Note that, while we state all our results for constant probability of success, by repeating $\log(1/\delta)$ times and taking the median of the estimates, this is boosted to $1 - \delta$ in the standard way.

▶ **Theorem 12.** *For $p \in (1, 2]$, there is a protocol for $(1 \pm \epsilon)$ approximating F_p which succeeds with probability $3/4$ in the message passing model. The protocol uses a max communication of $O(\frac{1}{\epsilon^2}(\log\log n + \log d + \log 1/\epsilon))$ bits, where d is the diameter of G.*

Table 1 For the communication problems above, the bounds are for the max-communication (in bits) across any edge. For the streaming problems, the bounds are for the space requirements of the algorithm. Here, d is the diameter of the communication network G. For all problems except point estimation, there is a matching $\Omega(\epsilon^{-2})$ lower bound. The problem of point estimation itself has a matching $\Omega(\epsilon^{-2}\log n)$ lower bound for graphs with constant d.

Problem	Prior best upper bound	Upper Bound (this work)	Notes
F_p, $1 < p \leq 2$	$O(\epsilon^{-2}\log(n))$ [45]	$\tilde{O}(\epsilon^{-2}\log(d))$	
F_p, $p < 1$	$O(\epsilon^{-2}\log(n))$[45]	$\tilde{O}(\epsilon^{-2})$	
F_p Streaming, $p < 1$	$O(\epsilon^{-2}\log(n))$[45]	$\tilde{O}(\epsilon^{-2})$	
Entropy	–	$\tilde{O}(\epsilon^{-2})$	
Entropy Streaming	$O(\epsilon^{-2}\log^2(n))$ [21]	$\tilde{O}(\epsilon^{-2})$	random oracle
Point Estimation	$O(\epsilon^{-2}\log^2(n))$ [18]	$\tilde{O}(\epsilon^{-2}\log(d)\log(n))$	
Approx Matrix Prod.	–	$\tilde{O}(1)$	per coordinate of sketch

For graphs with constant diameter, such as the coordinator model, our max communication bound of $\tilde{O}(\epsilon^{-2})$ matches the $\Omega(\epsilon^{-2})$ lower bound [58, 17], which follows from a 2-player reduction from the Gap-Hamming Distance problem. For $p = 2$, our *total communication* in the coordinator model matches the $\Omega(m^{p-1}/\epsilon^2)$ total communication lower bound (up to $\log\log(n)$ and $\log(1/\epsilon)$ terms) for non-one shot protocols [60]. For one shot protocols, we remark that there is an $\Omega(m/\epsilon^2)$ total communication lower bound for any $p \in (0, 2] \setminus \{1\}$ (see Appendix A). As discussed previously, our result also has strong implications for streaming algorithms, demonstrating that no $\Omega(\epsilon^{-2}\log n)$ lower bound for F_p estimation, $p \in (1, 2]$, can be derived via the common settings of 2-party, coordinator, or blackboard communication complexity.

Our main technique used to obtain Theorem 12 is a new randomized rounding scheme for p-stable sketches. We next show that this randomized rounding protocol can be applied to give improved communication upper bounds for the *point-estimation* problem. Here, the goal is to output a vector $\tilde{X} \in \mathbb{R}^n$ that approximates $\mathcal{X}$ well coordinate-wise. The result is formally given below in Theorem 14.

▶ **Theorem 14.** *Consider a message passing topology $G = (V, E)$ with diameter d, where the i-th player is given as input $X^i \in \mathbb{Z}^n_{\geq 0}$ and $\mathcal{X} = \sum_{i=1}^m X^i$. Then there is a communication protocol which outputs an estimate $\tilde{\mathcal{X}} \in \mathbb{R}^n$ of $\mathcal{X}$ such that $\|\tilde{\mathcal{X}} - \mathcal{X}\|_\infty \leq \epsilon\|\mathcal{X}_{tail(\epsilon^{-2})}\|_2$ with probability $1 - 1/n^c$ for any constant $c \geq 1$. Here $\mathcal{X}_{tail(\epsilon^{-2})}$ is $\mathcal{X}$ with the ϵ^{-2} largest (in absolute value) coordinates set equal to 0. The protocol uses a max communication of $O(\frac{1}{\epsilon^2}\log(n)(\log\log n + \log d + \log 1/\epsilon))$.*

For graphs with small diameter, our protocols demonstrate an improvement over the previously best known sketching algorithms, which use space $O(\epsilon^{-2}\log^2(n))$ to solve the point estimation problem [18]. Note that there is an $\Omega(\epsilon^{-2}\log n)$-max communication lower bound for the problem. This follows from the fact that point-estimation also solves the L_2 *heavy-hitters* problem. Here the goal is to output a set $S \subset [n]$ of size at most $|S| = O(\epsilon^{-2})$ which contains all $i \in [n]$ with $|\mathcal{X}_i| \geq \epsilon\|\mathcal{X}\|_2$ (such coordinates are called heavy hitters). The lower bound for heavy hitters is simply the result of the space required to store the $\log(n)$-bit identities of all possible ϵ^{-2} heavy hitters. Note that for the heavy hitters problem alone, there is an optimal streaming $O(\epsilon^{-2}\log(n))$-bits of space upper bound called BPTree [9]. However, BPTree cannot be used in the general distributed setting, since it crucially relies on the sequential natural of a stream.

Next, we demonstrate that F_p estimation for $p < 1$ is in fact possible with max communication independent of the graph topology. After derandomizing our protocol, this results in a optimal streaming algorithm for F_p estimation, $p < 1$, which closes a long line of research on the problem for this particular range of p [58, 35, 45, 44, 15, 13].

▶ **Theorem 21.** *For $p \in (0, 1)$, there is a protocol for F_p estimation in the message passing model which succeeds with probability $2/3$ and has max-communication of $O(\frac{1}{\epsilon^2}(\log\log n + \log 1/\epsilon))$.*

▶ **Theorem 22.** *There is a streaming algorithm for F_p estimation, $p \in (0, 1)$, which outputs a value $\tilde{R}$ such that with probability at least $2/3$, we have that $|\tilde{R} - \|X\|_p| \leq \epsilon\|X\|_p$. The algorithm uses $O((\frac{1}{\epsilon^2}(\log\log n + \log 1/\epsilon) + \frac{\log 1/\epsilon}{\log\log 1/\epsilon}\log n)$-bits of space. In the random oracle model, the space is $O(\frac{1}{\epsilon^2}(\log\log n + \log 1/\epsilon))$.*

The above bound matches the $\Omega(\epsilon^{-2})$ max communication lower bound of [58] in the shared randomness model, which comes from 2-party communication complexity. Moreover, our streaming algorithm matches the $\Omega(\log n)$ lower bound for streaming when a random oracle is not allowed. As an application of our protocol for F_p estimation, $p < 1$, we demonstrate a communication optimal protocol for additive approximation of the empirical *Shannon entropy* $H(\mathcal{X})$ of the aggregate vector $\mathcal{X}$. Here, $H = H(\mathcal{X})$ is defined by $H = \sum_{i=1}^{n} p_i \log(1/p_i)$ where $p_i = |\mathcal{X}_i|/\|\mathcal{X}\|_1$ for $i \in [n]$. The goal of our protocols is to produce an estimate $\tilde{H} \in \mathbb{R}$ of H such that $|\tilde{H} - H| \leq \epsilon$. Our result is as follows.

▶ **Theorem 26.** *There is a multi-party communication protocol in the message passing model that outputs a ϵ-additive error of the Shannon entropy H. The protocol uses a max-communication of $O(\frac{1}{\epsilon^2}(\log\log(n) + \log(1/\epsilon))$-bits.*

Note that for a *multiplicative* approximation of the Shannon entropy, there is a $\tilde{\Omega}(\epsilon^{-2})$ lower bound [14]. For additive estimation, [43] gives a $\Omega(\epsilon^{-2}\log(n))$ lower bound in the turnstile model. Using a similar reduction, we prove a matching $\Omega(\epsilon^{-2})$ lower bound for additive ϵ approximation in the insertion only model (see Appendix B for the proof). Furthermore, our protocol directly results in an $\tilde{O}(\epsilon^{-2})$-bits of space, insertion only *streaming* algorithm for entropy estimation in the random oracle model. Here, the random oracle model means that the algorithm is given query access to an arbitrarily long string of random bits. We note that many lower bounds in communication complexity (and all of the bounds discussed in this paper except for the $\Omega(\log n)$ term in the lower bound for F_p estimation) also apply to the random oracle model. Previously, the best known algorithm for the insertion only random oracle model used $O(\epsilon^{-2}\log(n))$-bits [47, 21], whereas the best known algorithm for the non-random oracle model uses $O(\epsilon^{-2}\log^2(n))$-bits (the extra factor of $\log(n)$ comes from a standard application of Nisan's pseudo-random generator [53]).

▶ **Theorem 27.** *There is a streaming algorithm for ϵ-additive approximation of the empirical Shannon entropy of an insertion only stream in the random oracle model, which succeeds with probability $3/4$. The space required by the algorithm is $O(\frac{1}{\epsilon^2}(\log\log(n) + \log(1/\epsilon))$ bits.*

Finally, we show how our techniques can be applied to the important numerical linear algebraic primitive of *approximate matrix product*, which we now define.

▶ **Definition 1.** *The multi-party approximate matrix product problem is defined as follows. Instead of vector valued inputs, each player is given $X_i \in \{0, 1, \ldots, M\}^{n\times t_1}$ and $Y_i \in \{0, 1, \ldots, M\}^{n\times t_2}$, where $\mathcal{X} = \sum_i X_i$ and $\mathcal{Y} = \sum_i Y_i$. Here, it is generally assumed that $n >> t_1, t_2$ (but not required). The players must work together to jointly compute a matrix $R \in \mathbb{R}^{t_1\times t_2}$ such that $\|R - \mathcal{X}^T\mathcal{Y}\|_F \leq \epsilon\|\mathcal{X}\|_F\|\mathcal{Y}\|_F$, where for a matrix $A \in \mathbb{R}^{n\times m}$, $\|A\|_F = (\sum_{i=1}^{n}\sum_{j=1}^{m} A_{i,j}^2)^{1/2}$ is the Frobenius norm of A.*

▶ **Theorem 29.** *There is a protocol which outputs, at the central vertex $\mathcal{C}$, a matrix $R \in \mathbb{R}^{t_1 \times t_2}$ which solves the approximate communication protocol with probability $3/4$ [3]. The max communication required by the protocol is $O\left(\epsilon^{-2}(t_1+t_2)(\log\log n + \log 1/\epsilon + \log d)\right)$, where d is the diameter of the communication topology G.*

We remark that an upper bound of $O\left(\epsilon^{-2}(t_1+t_2)\log n\right)$ was already well-known from sketching theory [59], and our main improvement is removing the $\log(n)$ factor for small diameter graphs, such as the coordinator model where distributed numerical linear algebra is usually considered.

1.3 Other Related Work

As mentioned, a closely related line of work is in the *distributed functional monitoring model*. Here, there are m machines connected to a central coordinator (the coordinator topology). Each machine then receives a stream of updates, and the coordinator must maintain at all time steps an approximation of some function, such as a moment estimation or a uniform sample, of the union of all streams. We note that there are two slightly different models here. One model is where the items (coordinates) being updated in the separate streams are considered disjoint, and each time an insertion is seen it is to a unique item. This model is considered especially for the problem of maintaining a uniform sample of the items in the streams [25, 34, 56, 37]. The other model, which is more related to ours, is where each player is receiving a stream of updates to a *shared* overall data vector $\mathcal{X} \in \mathbb{R}^n$. This can be seen as a distributed streaming setting, where the updates to a centralized stream are split over m servers, and is considered in [60, 25, 3]. For the restricted setting of *one-way* algorithms, which only transmit messages from the sites to the coordinators, any such algorithm can be made into a one-shot protocol for the multi-party message passing model. Here, each machine just simulates a stream on their fixed input vectors X_i, and sends all the messages that would have been sent by the functional monitoring protocol.

Perhaps the most directly related result to our upper bound for for F_p estimation, $p \in (1, 2]$, is in the distributed functional monitoring model, where Woodruff and Zhang [60] show a $O(m^{p-1}\text{poly}(\log(n), 1/\epsilon) + m\epsilon^{-1}\log(n)\log(\log(n)/\epsilon))$[4] *total communication* upper bound. We remark here, however, that the result of [60] is incomparable to ours for several reasons. Firstly, their bounds are only for total communication, whereas their max communication can be substantially larger than $O(1/\epsilon^2)$. Secondly, while it is claimed in the introduction that the protocols are one way (i.e., only the players speak to the coordinator, and not vice versa), this is for their threshold problem and not for F_p estimation[5]. As remarked before, there is an $\Omega(m/\epsilon^2)$ total communication lower bound for one-way protocols, which demonstrates that their complexity could not hold in our setting (we sketch a proof of this in Appendix A).

[3] We remark that there are standard techniques to boost the probability of the matrix sketching results to $1-\delta$, using a blow-up of $\log(\delta)$ in the communication. See e.g. Section 2.3 of [59]

[4] We remark that the $\text{poly}(\log(n), 1/\epsilon)$ terms here are rather large, and not specified in the analysis of [60].

[5] The reason for this is as follows. Their algorithm reduces F_p estimation to the threshold problem, where for a threshold τ, the coordinator outputs 1 when the F_p first exceeds $\tau(1+\epsilon)$, and outputs 0 whenever the F_p is below $\tau(1-\epsilon)$. To solve F_p estimation, one then runs this threshold procedure for the $\log(mMn)/\epsilon$ thresholds $\tau = (1+\epsilon), (1+\epsilon)^2, \ldots, (mMn)^2$ in parallel. However, the analysis from [60] only demonstrates a total communication of $O(k^{1-p}\text{poly}(\log(n), \epsilon^{-1}))$ for the time steps *before* the threshold τ is reached. Once the threshold is reached, the communication would increase significantly, thus the coordinator must inform all players when a threshold τ is reached so that they stop sending messages for τ, violating the one-way property. This step also requires an additive k messages for each of the $O(\epsilon^{-1}\log(n))$ thresholds, which results in the $O(m\epsilon^{-1}\log(n)\log(\log(n)\epsilon)))$ term.

The message passing model itself has been the subject of significant research interest over the past two decades. The majority of this work is concerned with *exact* computation of Boolean functions of the inputs. Perhaps the canonical multi-party problem, and one which has strong applications to streaming, is set disjointness, where each player has a subset $S_i \subset [n]$ and the players want to know if $\cap_{i=1}^m S_i$ is empty. Bar-Yossef et al. [6] demonstrated strong bounds for this problem in the black-board model. This lower bound resulted in improved (polynomially sized) lower bounds for streaming F_p estimation for $p > 2$. These results for disjointness have since been generalized and improved using new techniques [16, 30, 41, 8]. Finally, we remark that while most results in the multi-party message passing model are not topology dependent, Chattopadhyay, Radhakrishnan, and Rudra have demonstrated that tighter topology-dependent lower bounds are indeed possible in the message passing model [19].

2 Preliminaries

Let f be a function $f : \mathbb{R}^n \to \mathbb{R}$. Let $G = (V, E)$ be a connected undirected graph with m vertices, i.e. $V = \{1, \ldots, m\}$. In the message passing model on the graph topology G, there are m players, each placed at a unique vertex of G, with unbounded computational power. Player i is given as input only a vector $X_i \in \mathbb{Z}^n$, which is known as the Number in Hand (NIH) model of communication. Let $\mathcal{X} = \sum_{i=1}^n X_i$ be the aggregate vector of the players inputs. The goal of the players is to jointly compute or approximate the function $f(\mathcal{X})$ by carrying out some previously unanimously agreed upon communication protocol. It is assumed that the graph topology of G is known to all players.

In this paper, we are concerned with the non-negative input model. Namely, the inputs X_i satisfy $X_i \in \{0, 1, \ldots, M\}^n$ for all players i. Note an equivalent assumption to is that $(X_i)_j \geq 0$ for all i, and that the $(X_i)_j$'s can be specified in $O(\log(M))$ bits.

▶ Remark 2. For ease of presentation, we assume that $m, M = O(n^c)$ for some constant c. This allows us to simplify complexity bounds and write $\log(nmM) = O(\log n)$. This is a common assumption in the streaming literature, where m corresponds to the length of the stream. We remark, however, that all our results hold for general m, n, M, by replacing each occurrence of n in the communication complexity with (mnM).

During execution of the protocol, a player $i \in V$ is only allowed to send a message to a player j if $(i, j) \in E$. Thus, players may only communicate directly with their neighbors in the graph G. In contrast to the *broadcast* and *blackboard* models of communication, in the message passing model the message sent by player i to player j is only received by player j, and no other player. Upon termination of the protocol, at least one player must hold an approximation of the value $f(\mathcal{X})$. For the protocols considered in this paper, this player will be fixed and specified by the protocol beforehand. We use $\mathcal{C} \in V$ to denote the distinguished player specified by the protocol to store the approximation at the end of the execution.

Every such communication protocol in this model can be divided into rounds, where on the j-th round some subset $S_j \subseteq V$ of the players simultaneously send a message across one of their edges. Although it is not a restriction in the message passing model, our protocols satisfy the additional property that each player communicates *exactly once*, across one of its edges, and that each player will receive messages from its neighbors in exactly one round. Specifically, for each player i, there will be exactly one round j where some subset of its neighbors send player i a message, and then player i will send a single message in round $j + 1$, and never again communicate. Such protocols are called *one-shot* protocols.

The *total communication* cost of a protocol is the total number of bits sent in all the messages during its execution. The *max-communication* of a protocol is the maximum number of bits sent across any edge over the execution of the protocol. Communication protocols can be either deterministic or randomized. In this paper we consider the standard *public-coin* model of communication, where each player is given shared access to an arbitrarily long string of random bits. This allows players to jointly utilize the same source of randomness without having to communicate it.

Our protocols for F_p estimation will utilize the p-stable distribution, D_p, which we will now introduce. For $p = 2$, the distribution D_2 is the just standard Gaussian distribution. Note for $p < 2$, the distributions have heavy tails – they decay like x^{-p}. Thus, for $p < 2$, the variance is infinite, and for $p \leq 1$, the expectation is undefined.

▶ **Definition 3.** *For $0 < p \leq 2$, there exists a probability distribution D_p called the p-stable distribution. If $Z \sim D_p$, $p < 2$, then the characteristic function of D_p is given by $\mathbb{E}[e^{itZ}] = e^{-|t|^p}$. For $p = 2$, D_2 is the standard Gaussian distribution. Moreover, for any n, and any $x \in \mathbb{R}^n$, if $Z_1, \ldots, Z_n \sim D_p$ are independent, then $\sum_{i=1}^n Z_i x_i \sim \|x\|_p Z$, where $Z \sim D_p$, and $\sim$ means distributed identically to.*

Standard methods for generating p-stable random variables are discussed in [54]. Note that all protocols in this paper will generate these variables only to precision $1/\text{poly}(n)$. For a distribution D_p, we write D_p^n to denote the product distribution of D_p. Thus $Z \sim D_p^n$ means $Z \in \mathbb{R}^n$ and $Z_1, \ldots, Z_n$ are drawn i.i.d. from D_p. For reals $a, b \in \mathbb{R}$, we write $a = (1 \pm \epsilon)b$ to denote the containment $a \in [(1-\epsilon)b, (1+\epsilon)b]$. For an integer $t \geq 0$, we write $[t]$ to denote the set $\{1, 2, \ldots, t\}$.

3 Message Passing F_p Estimation, $p > 1$

In this section, we provide our algorithm for F_p estimation, $1 \leq p \leq 2$, in the message passing model. We begin by specifying the distinguished vertex $\mathcal{C} \in V$ which will hold and output the F_p approximation at the end of the protocol. For a vertex $v \in G$, define its eccentricity $\texttt{ecc}(v) = \max_{u \in V} d(v, u)$, where $d(v, u)$ is the graph distance between v, u. We then set $\mathcal{C} \in V$ to be any vertex with minimal eccentricity. Such a vertex is known as a center of G. We now fix a shortest path spanning tree T for G, rooted at the distinguished player $\mathcal{C}$. The spanning tree T has the property that the path between $\mathcal{C}$ and any vertex $v \in V$ in the tree T is also a shortest path between $\mathcal{C}$ and v in G. Thus the distance between $\mathcal{C}$ and any vertex $v \in V$ is the same in T as it is in G. The fact that the depth of T is at most d, where d is the diameter of G, now follows naturally. Such a shortest path spanning tree T can be easily obtained via a breath first search. First, we will need a technical Lemma about the behavior of p-stables. To prove it, we first use the following fact about the tails of p stables, which can be found in [54].

▶ **Proposition 4.** *If $Z \sim D_p$ for $0 < p < 2$, then $\Pr[|Z| \geq \lambda] \leq O(\frac{1}{\lambda^p})$.*

Also, we use the straightforward fact that $\|X_i\|_p^p \leq \|\sum_{i=1}^m X_i\|_p^p$ for non-negative vectors X_i and $p \geq 1$.

▶ **Fact 5.** *If $X_1, \ldots, X_m \in \mathbb{R}^n$ are entry-wise non-negative vectors and $1 \leq p \leq 2$, then $\sum_{i=1}^m \|X_i\|_p^p \leq \|\sum_{i=1}^m X_i\|_p^p$.*

▶ **Lemma 6.** *Fix $1 \le p \le q \le 2$, and let $Z = (Z_1, Z_2, \ldots, Z_n) \sim D_p^m$. Suppose $X_1, \ldots, X_m \in \mathbb{R}^n$ are non-negative vectors, with $\mathcal{X} = \sum_j X_j$. Then for any $\lambda \ge 1$, if either $q - p \ge c > 0$ for some constant c independent of m, or if $p = 2$, we have*

$$\Pr\left[\sum_{j=1}^{m} |\langle Z, X_j\rangle|^q \ge C\lambda^q \|\mathcal{X}\|_p^q\right] \le \frac{1}{\lambda^p}$$

Otherwise, we have $\Pr[\sum_{j=1}^m |\langle Z, X_j\rangle|^q \ge C\log(\lambda m)\lambda^q \|\mathcal{X}\|_p^q] \le \frac{1}{\lambda^p}$, where C is some constant (depending only on c in the first case).

▶ **Corollary 7.** *Suppose $Z = (Z_1, \ldots, Z_m)$ where the Z_i's are uniform over $\{1, -1\}$ and pairwise independent, and let $X_1, \ldots, X_m$ be non-negative vectors with $\mathcal{X} = \sum_j X_j$. Then for any $\lambda \ge 1$, we have $\Pr[\sum_{j=1}^m |\langle Z, X_j\rangle|^2 \ge \lambda \|\mathcal{X}\|_2^2] \le \frac{1}{\lambda}$*

▶ **Corollary 8.** *Let $Z = (Z_1, Z_2, \ldots, Z_n) \sim D_2^m$ be i.i.d. Gaussian. Suppose $X_1, \ldots, X_m \in \mathbb{R}^n$ are non-negative vectors, with $\mathcal{X} = \sum_j X_j$. Then for any $\lambda \ge c\log(m)$ for some sufficiently large constant c, we have $\Pr[\sum_{j=1}^m |\langle Z, X_j\rangle| \ge \lambda \|\mathcal{X}\|_2^2] \le \exp(-C\lambda)$, where C is some universal constant.*

3.1 Randomized Rounding of Sketches

We now introduce our randomized rounding protocol. Consider non-negative integral vectors $X_1, X_2, \ldots, X_m \in \mathbb{Z}_{\ge 0}^n$, with $\mathcal{X} = \sum_{i=1}^n X_i$. Fix a message passing topology $G = (V, E)$, where each player $i \in V$ is given as input X_i. Fix any vertex $\mathcal{C}$ that is a center of G, and let T be a shortest path spanning tree of G rooted at $\mathcal{C}$ as described at the beginning of the section. Let d be the depth of T. The players use shared randomness to choose a random vector $Z \in \mathbb{R}^n$, and their goal is to approximately compute $\langle Z, \mathcal{X}\rangle = \langle Z, \sum_{i=1}^m X_i\rangle$. The goal of this section is to develop a d-round randomized rounding protocol, so that at the end of the protocol the approximation to $\langle Z, \mathcal{X}\rangle$ is stored at the vertex $\mathcal{C}$.

We begin by introducing the rounding primitive which we use in the protocol. Fix $\epsilon > 0$, and let $\gamma = (\epsilon\delta/\log(nm))^C$, for a sufficiently large constant $C > 1$. For any real value $r \in \mathbb{R}$, let $i_r \in \mathbb{Z}$ and $\alpha_i \in \{1, -1\}$ be such that $(1+\gamma)^{i_r} \le \alpha_i r \le (1+\gamma)^{i_r+1}$. Now fix p_r such that: $\alpha_i r = p_r(1+\gamma)^{i_r+1} + (1-p_r)(1+\gamma)^{i_r}$. We then define the rounding random variable $\Gamma(r)$ by

$$\Gamma(r) = \begin{cases} 0 & \text{if } r = 0 \\ \alpha_i(1+\gamma)^{i_r+1} & \text{with probability } p_r \\ \alpha_i(1+\gamma)^{i_r} & \text{with probability } 1 - p_r \end{cases}$$

The following proposition is clear from the construction of p_r and the fact that the error is deterministically bounded by $\gamma|r|$.

▶ **Proposition 9.** *For any $r \in R$, We have $\mathbb{E}[\Gamma(r)] = r$ and $\boldsymbol{Var}[\Gamma(r)] \le r^2\gamma^2$*

We partition T into d layers, so that all nodes at distance $d - t$ from $\mathcal{C}$ in T are put in layer t. Define $L_t \subset [n]$ to be the set of players at layer t in the tree. For any vertex $u \in G$, let T_u be the subtree of T rooted at u (including the vertex u). For any player i, let $C_i \subset [n]$ be the set of children of i in the tree T. The procedure for all players $j \in V$ is then given as Algorithm 1.

Algorithm 1 Recursive Randomized Rounding.

Procedure for node j in layer i:

1. Choose random vector $Z \in \mathbb{R}^n$ using shared randomness.
2. Receive rounded sketches $r_{j_1}, r_{j_2}, \ldots, r_{j_{t_j}} \in \mathbb{R}$ from the t_j children of node j in the prior layer (if any such children exist).
3. Compute $x_j = \langle X_j, Z\rangle + r_{j_1} + r_{j_2} + \cdots + r_{j_t} \in \mathbb{R}$.
4. Compute $r_j = \Gamma(x_j)$. If player $j \neq \mathcal{C}$, then send r_j it to the parent node of j in T. If $j = \mathcal{C}$, then output r_j as the approximation to $\langle Z, \mathcal{X}\rangle$.

For each player i in layer 0, they take their input X_i, and compute $\langle Z, X_i\rangle$. They then round their values as $r_i = \Gamma(\langle Z, X_i\rangle)$, where the randomness used for the rounding function Γ is drawn independently for each call to Γ. Then player i sends r_i to their parent in T. In general, consider any player i at depth $j > 0$ of T. At the end of the j-th round, player i will receive a rounded value r_ℓ for every child vertex $\ell \in C_i$. They then compute $x_i = \langle Z, X_i\rangle + \sum_{\ell \in C_i} r_\ell$, and $r_i = \Gamma(x_i)$, and send r_i to their parent in T. This continues until, on round d, the center vertex $\mathcal{C}$ receives r_ℓ for all children $\ell \in C_{\mathcal{C}}$. The center $\mathcal{C}$ then outputs $r_{\mathcal{C}} = \langle Z, X_{\mathcal{C}}\rangle + \sum_{\ell \in C_{\mathcal{C}}} r_\ell$ as the approximation.

For any player i, let $Q_i = \sum_{u \in T_i} X_u$, and $y_i = \langle Z, Q_i\rangle$. Then define the error e_i at player i as $e_i = y_i - r_i$. We first prove a proposition that states the expectation of the error e_i for any player i is zero, and then the main lemma which bounds the variance of e_i. The error bound of the protocol at $\mathcal{C}$ then results from an application of Chebyshev's inequality.

▶ **Proposition 10.** *For any player i, we have $\mathbb{E}[e_i] = 0$. Moreover, for any players i, j such that $i \notin T_j$ and $j \notin T_i$, the variables e_i and e_j are statistically independent.*

▶ **Lemma 11.** *Fix $p \in [1, 2]$, and let $Z = (Z_1, Z_2, \ldots, Z_n) \sim D_p^n$. Then the above procedure when run on $\gamma = (\epsilon\delta/(d\log(nm)))^C$ for a sufficiently large constant C, produces an estimate $r_{\mathcal{C}}$ of $\langle Z, \mathcal{X}\rangle$, held at the center vertex $\mathcal{C}$, such that $\mathbb{E}[r_{\mathcal{C}}] = \langle Z, \mathcal{X}\rangle$. Moreover, over the randomness used to draw Z, with probability $1 - \delta$ for $p < 2$, and with probability $1 - e^{-1/\delta}$ for Gaussian Z, we have $\mathbb{E}[(r_{\mathcal{C}} - \langle Z, \mathcal{X}\rangle)^2] \leq (\epsilon/\delta)^2 \|\mathcal{X}\|_p$. Thus, with probability at least $1 - O(\delta)$, we have $|r_{\mathcal{C}} - \langle Z, \mathcal{X}\rangle| \leq \epsilon\|\mathcal{X}\|_p$. Moreover, if $Z = (Z_1, Z_2, \ldots, Z_n) \in \mathbb{R}^n$ where each $Z_i \in \{1, -1\}$ is a 4-wise independent Rademacher variable, then the above bound holds with $p = 2$ (and with probability $1 - \delta$).*

▶ **Theorem 12.** *For $p \in (1, 2]$, there is a protocol for F_p estimation which succeeds with probability $3/4$ in the message passing model, which uses a total of $O(\frac{m}{\epsilon^2}(\log(\log(n)) + \log(d) + \log(1/\epsilon)))$ communication, and a max communication of $O(\frac{1}{\epsilon^2}(\log(\log(n)) + \log(d) + \log(1/\epsilon)))$, where d is the diameter of the communication network.*

3.2 Heavy Hitters and Point Estimation

In this section, we show how our randomized rounding protocol can be used to solve the L_2 heavy hitters problem. For a vector $\mathcal{X} \in \mathbb{R}^n$, let $\mathcal{X}_{\text{tail}(k)}$ be $\mathcal{X}$ with the k largest (in absolute value) entries set equal to 0. Formally, given a vector $\mathcal{X} \in \mathbb{R}^n$, the heavy hitters problem is to output a set of coordinates $H \subset [n]$ of size at most $|H| = O(\epsilon^{-2})$ that contains all $i \in [n]$ with $|\mathcal{X}_i| \geq \epsilon\|\mathcal{X}_{tail(1/\epsilon^2)}\|_2$. Our protocols solve the strictly harder problem of *point-estimation*. The point estimation problem is to output a $\tilde{\mathcal{X}} \in \mathbb{R}^n$ such that $\|\tilde{\mathcal{X}} - \mathcal{X}\|_\infty \leq \epsilon\|\mathcal{X}_{tail(1/\epsilon^2)}\|_2$. Our protocol uses the well-known *count-sketch* matrix S [18], which we now introduce.

▶ **Definition 13.** *Given a precision parameter ϵ and an input vector $\mathcal{X} \in \mathbb{R}^n$, count-sketch stores a table $A \in \mathbb{R}^{\ell \times 6/\epsilon^2}$, where $\ell = \Theta(\log(n))$. Count-sketch first selects pairwise independent hash functions $h_j : [n] \to [6/\epsilon^2]$ and 4-wise independent $g_j : [n] \to \{1, -1\}$, for $j = 1, 2, \ldots, \ell$. Then for all $i \in [\ell]$, $j \in [6/\epsilon^2]$, it computes the following linear function $A_{i,j} = \sum_{k \in [n], h_i(k)=j} g_i(k)\mathcal{X}_k$, and outputs an approximation $\tilde{\mathcal{X}}$ of $\mathcal{X}$ given by $\tilde{\mathcal{X}}_k = median_{i \in [\ell]}\{g_i(k)A_{i,h_i(k)}\}$*

Observe that the table $A \in \mathbb{R}^{\ell \times 6/\epsilon^2}$ can be flattened into a vector $A \in \mathbb{R}^{6\ell/\epsilon^2}$. Given this, A can be represented as $A = S\mathcal{X}$ for a matrix $S \in \mathbb{R}^{6\ell/\epsilon^2 \times n}$. For any $i \in [\ell]$, $j \in [6/\epsilon^2]$, and $\ell \in [n]$, the matrix S is given by $S_{(i-1)(6/\epsilon^2)+j,\ell} = \delta_{i,j,\ell} g_j(\ell)$, where $\delta_{i,j,\ell}$ indicates the event that $h_i(\ell) = j$. Given $S\mathcal{X}$, one can solve the point-estimation problem as described in Definition 13 [18]. In order to reduce the communication from sending each coordinate of $S\mathcal{X}$ exactly, we can use our rounding procedure to approximately compute the sketch $S\mathcal{X}$, which will give us the following theorem.

▶ **Theorem 14.** *Consider a message passing topology $G = (V, E)$ with diameter d, where the i-th player is given as input $X_i \in \mathbb{Z}^n_{\geq 0}$ and $\mathcal{X} = \sum_{i=1}^m X_i$. Then there is a communication protocol which outputs an estimate $\tilde{\mathcal{X}} \in \mathbb{R}^n$ of $\mathcal{X}$ such that $\|\tilde{\mathcal{X}} - \mathcal{X}\|_\infty \leq \epsilon \|\mathcal{X}_{tail(1/\epsilon^2)}\|_2$ with probability $1 - 1/n^c$ for any constant $c \geq 1$. The protocol uses $O(\frac{m}{\epsilon^2}\log(n)(\log(\log(n)) + \log(d) + \log(1/\epsilon)))$ total communication, and a max communication of $O(\frac{1}{\epsilon^2}\log(n)(\log(\log(n)) + \log(d) + \log(1/\epsilon)))$.*

4 F_p Estimation for $p < 1$

In this section, we develop algorithms for F_p estimation for $p < 1$ in the message passing model, and in the process obtain improved algorithms for entropy estimation. We begin by reviewing the fundamental sketching procedure used in our estimation protocol. The algorithm is known as a Morris counter [51, 26]. The algorithm first picks a base $1 < b \leq 2$, and initializes a counter $C \leftarrow 0$. Then, every time it sees an insertion, it increments the counter $C \leftarrow C + \delta$, where $\delta = 1$ with probability b^{-C}, and $\delta = 0$ otherwise (in which case the counter remains unchanged). After n insertions, the value n can be estimated by $\tilde{n} = (b^C - b)/(b - 1) + 1$.

▶ **Definition 15.** *The approximate counting problem is defined as follows. Each player i is given a positive integer value $x_i \in \mathbb{Z}_{\geq 0}$, and the goal is for some player at the end to hold an estimate of $x = \sum_i x_i$.*

▶ **Proposition 16** (Proposition 5 [26]). *If C_n is the value of the Morris counter after n updates, then $\mathbb{E}[\tilde{n}] = n$, and $\boldsymbol{Var}[\tilde{n}] = (b-1)n(n+1)/2$.*

▶ **Corollary 17.** *If C_n is the value of a Morris counter run on a stream of n insertions with base $b = (1 + (\epsilon\delta)^2)$, then with probability at least $1 - \delta$, we have $\tilde{n} = (1 \pm \epsilon)n$ with probability at least $1 - \delta$. Moreover, with probability at least $1 - \delta$, the counter C_n requires $O(\log\log(n) + \log(1/\epsilon) + \log(1/\delta))$-bits to store.*

▶ **Lemma 18.** *Given Morris counters X, Y run on streams of length n_1, n_2 respectively, There is a merging procedure that produces a Morris counter Z which is distributed identically to a Morris counter that was run on a stream of $n_1 + n_2$ insertions.*

▶ **Corollary 19.** *There is a protocol for F_1 estimation of non-negative vectors, equivalently for the approximate counting problem, in the message passing model which succeeds with probability $1 - \delta$ and uses a max-communication of $O((\log\log(n) + \log(1/\epsilon) + \log(1/\delta))$-bits.*

We now note that Morris counters can easily used as approximate counters for streams with both insertions and deletions (positive and negative updates), by just storing a separate Morris counter for the insertions and deletions, and subtracting the estimate given by one from the other at the end.

▶ **Corollary 20.** *Using two Morris counters separately for insertions and deletions, on a stream of I insertions and D deletions, there is an algorithm, called a signed Morris counter, which produces $\tilde{n}$ with $|\tilde{n} - n| \leq \epsilon(I + D)$, where $n = I - D$, with probability $1 - \delta$, using space $O(\log\log(I + D) + \log(1/\epsilon) + \log(1/\delta))$.*

Hereafter, when we refer to a Morris counter that is run on a stream which contains both positive and negative updates as a *signed* Morris counter. Therefore, the guarantee of Corollary 20 apply to such signed Morris counters, and moreover such signed Morris counters can be Merged as in Lemma 18 with the same guarantee.

Algorithm 2 Multi-party F_p estimation protocol, $p < 1$.

Procedure for player j

$k \leftarrow \Theta(1/\epsilon^2)$, $\epsilon' \leftarrow \Theta(\epsilon \frac{\delta^{1/p}}{\log(n/\delta)})$, $\delta \leftarrow 1/(200k)$

1. Using shared randomness, choose sketching matrix $S \in \mathbb{R}^{k \times n}$ of i.i.d. p-stable random variables, with $k = \Theta(1/\epsilon)$. Generate S up to precision $\eta = \text{poly}(1/(n, m, M))$, so that $\eta^{-1}S$ has integral entries.
2. For each $i \in [k]$, receive signed Morris counters $y_{j_1,i}, y_{j_2,i}, \ldots, y_{j_t,i}$ from the $t \in \{0, \ldots, m\}$ children of node j in the prior layer.
3. Compute $\eta^{-1}\langle S_i, X_j\rangle \in \mathbb{Z}$, where S_i is the i-th row of S, and run a new signed Morris counter C on $\eta^{-1}\langle S_i, X_j\rangle$ with parameters (ϵ', δ').
4. Merge the signed Morris counters $y_{j_1,i}, y_{j_2,i}, \ldots, y_{j_t,i}, C$ into a counter $y_{j,i}$.
5. Send the merged signed Morris counter $y_{j,i}$ to the parent of player j. If player j is the root node $\mathcal{C}$, then set C_i to be the estimate of the signed Morris counter $y_{j,i}$, and return the estimate $\eta \cdot \text{median}\,\{\frac{|C_1|}{\theta_p}, \ldots, \frac{|C_k|}{\theta_p}\}$, where θ_p is the median of the distribution $\mathcal{D}_p$.

We now provide our algorithm for F_p estimation in the message passing model with $p \leq 1$. Our protocol is similar to our algorithm for $p \geq 1$. We fix a vertex $\mathcal{C}$ which is a center of the communication topology. We then consider the shortest path tree T rooted at $\mathcal{C}$, which has depth at most d, where d is the diameter of G. The players then choose random vectors $S_i \in \mathbb{R}^n$ for $i \in [k]$, and the j-th player computes $\langle S_i, X_j\rangle$, and adds this value to a Morris counter. Each player receives Morris counters from their children in T, and thereafter merges these Morris counters with its own. Finally, it sends this merged Morris counter, containing updates from all players in the subtree rooted at j, to the parent of j in T. At the end, the center $\mathcal{C}$ holds a Morris counter C_i which approximates $\sum_j \langle S_i, X_j\rangle$. The main algorithm for each player j is given formally as Algorithm 2.

▶ **Theorem 21.** *For $p \in (0, 1)$, there is a protocol for F_p estimation in the message passing model which succeeds with probability $2/3$ and uses a total communication of $O(\frac{m}{\epsilon^2}(\log\log(n) + \log(1/\epsilon))$-bits, and a max-communication of $O(\frac{1}{\epsilon^2}(\log\log(n) + \log(1/\epsilon))$-bits. The protocol requires a total of at most d rounds, where d is the diameter of the communication topology G.*

4.1 The Streaming Algorithm for F_p Estimation, $p < 1$

As discussed earlier, the insertion-only streaming model of computation is a special case of the above communication setting, where the graph in question is the line graph, and each player receives vector $X_i \in \mathbb{R}^n$ which is the standard basis vector $e_j \in \mathbb{R}^n$ for some $j \in [n]$. The only step remaining to fully generalize the result to the streaming setting is an adequate derandomization of the randomness required to generate the matrix S. Our derandomization will follow from the results of [45], which demonstrate that, using a slightly different estimator known as the log-cosine estimator, the entries of each row S_i can be generated with only $\Theta(\log(1/\epsilon)/\log\log(1/\epsilon))$-wise independence, and the seeds used to generate separate rows of S_i need only be pairwise independent. Thus, storing the randomness used to generate S requires only $O(\frac{\log(1/\epsilon)}{\log\log(1/\epsilon)}\log(n))$-bits of space.

We now discuss the estimator of [45] precisely. The algorithm generates a matrix $S \in \mathbb{R}^{k\times n}$ and $S' \in \mathbb{R}^{k'\times n}$ with $k = \Theta(1/\epsilon^2)$ and $k' = \Theta(1)$, where each entry of S, S' is drawn from $\mathcal{D}_p$. For a given row i of S, the entries $S_{i,j}$ are $\Theta(\log(1/\epsilon)/\log\log(1/\epsilon))$-wise independent, and for $i \neq i'$, the seeds used to generate $\{S_{i,j}\}_{j=1}^n$ and $\{S_{i',j}\}_{j=1}^n$ are pairwise independent. S' is generated with only $\Theta(1)$-wise independence between the entries in a given row in S', and pairwise independence between rows. The algorithm then maintains the vectors $y = S\mathcal{X}$ and $y' = S'\mathcal{X}$ throughout the stream, where $\mathcal{X} \in \mathbb{Z}_{\geq 0}^n$ is the stream vector. Define $y'_{med} = \text{median}\{|y'_i|\}_{i=1}^{k'}/\theta_p$, where θ_p is the median of the distribution $\mathcal{D}_p$ ([45] discusses how this can be approximated to $(1 \pm \epsilon)$ efficiently). The log-cosine estimator R of $\|\mathcal{X}\|_p$ is then given by $R = y'_{med} \cdot \left(-\ln\left(\frac{1}{k}\sum_{i=1}^k \cos\left(\frac{y_i}{y'_{med}}\right)\right)\right)$

▶ **Theorem 22.** *There is a streaming algorithm for insertion only F_p estimation, $p \in (0,1)$, outputs a value $\tilde{R}$ such that with probability at least $2/3$, we have that $|\tilde{R} - \|\mathcal{X}\|_p| \leq \epsilon\|\mathcal{X}\|_p$ where $\mathcal{X} \in \mathbb{R}^n$ is the state of the stream vector at the end of the stream. The algorithm uses $O((\frac{1}{\epsilon^2}(\log\log(n) + \log(1/\epsilon)) + \frac{\log(1/\epsilon)}{\log\log(1/\epsilon)}\log(n))$-bits of space.*

5 Entropy Estimation

In this section, we show how our results imply improved algorithms for entropy estimation in the message-passing model. Here, for a vector $\mathcal{X} \in \mathbb{R}^n$, the Shannon entropy is given by $H = \sum_{i=1}^n \frac{|\mathcal{X}_i|}{\|\mathcal{X}\|_1}\log(\frac{\|\mathcal{X}\|_1}{|\mathcal{X}_i|})$. We follow the approach taken by [21, 47, 31, 32] for entropy estimation in data streams, which is to use sketched of independent *maximally-skewed stable random variables*. While we introduced p-stable random variables in Definition 3 as the distribution with characteristic function $\mathbb{E}[e^{itZ}] = e^{-|t|^p}$, we remark now that the p-stable distribution is also parameterized by an additional *skewness* parameter $\beta \in [-1, 1]$. Up until this point, we have assumed $\beta = 0$. In this section, however, we will be using maximally skewed, meaning $\beta = -1$, $p = 1$-stable random variables. We introduce these now

▶ **Definition 23** (Stable distribution, general)**.** *There is a distribution $F(p, \beta, \gamma, \delta)$ called the p-stable distribution with skewness parameter $\beta \in [-1, 1]$, scale γ, and position δ. The characteristic function of a $Z \sim F(p, \beta, \gamma, \delta)$ variable Z is given by:*

$$\mathbb{E}[e^{-itZ}] = \begin{cases} \exp\left(-\gamma^p|t|^p\left[1 - i\beta\tan(\frac{\pi p}{2})\text{sign}(t)\right] + i\delta t\right) & \text{if } p \in (0,2] \setminus \{1\} \\ \exp\left(-\gamma|t|\left[1 + i\beta\frac{2}{\pi}\text{sign}(t)\log(|t|)\right] + i\delta t\right) & \text{if } p = 1 \end{cases}$$

where sign$(t) \in \{1, -1\}$ is the sign of a real $t \in \mathbb{R}$. Moreover, if $Z \sim F(p, \beta, \gamma, 0)$ for any $\beta \in [-1, 1]$ and $0 < p < 2$, for any $\lambda > 0$ we have $\Pr[|Z| > C\lambda] \leq (\frac{\gamma}{\lambda})^p$, where C is some universal constant. We refer the reader to [54] for a further discussion on the parameterization and behavior of p-stable distributions with varying rates.

Algorithm 3 Entropy Estimation algorithm of [21].

Sketching algorithm for Entropy Estimation

`Input`: $\mathcal{X} \in \mathbb{R}^n$

1. Generate $S \in \mathbb{R}^{k \times n}$ for $k = \Theta(1/\epsilon^2)$ of i.i.d. $F(1, -1, \pi/2, 0)$ random variables to precision $\eta = 1/\text{poly}(M, n)$.
2. Compute $S\mathcal{X} \in \mathbb{R}^k$.
3. Set $y_i \leftarrow (S\mathcal{X})_i / \|\mathcal{X}\|_1$ for $i \in [k]$
4. Return $\tilde{H} = -\log\left(\frac{1}{k}\sum_{i=1}^{k} e^{y_i}\right)$

The algorithm of [21] is given formally as Algorithm 3. The guarantee of the algorithm is given in Theorem 24.

▶ **Theorem 24** ([21]). *The above estimate $\tilde{H}$ satisfies $|\tilde{H} - H| < \epsilon$ with probability at least $9/10$.*

▶ **Lemma 25.** *Fix $0 < \epsilon_0 < \epsilon$. Let $S \in \mathbb{R}^{k \times n}$ with $k = \Theta(1/\epsilon^2)$ be a matrix of i.i.d. $F(1, -1, \pi/2, 0)$ random variables to precision $\eta = 1/poly(M, n)$. Then there is a protocol in the message passing model that outputs $Y \in \mathbb{R}^k$ at a centralized vertex with $\|Y - S\mathcal{X}\|_\infty \leq \epsilon_0 \|\mathcal{X}\|_1$ with probability $9/10$. The protocol uses a total communication of $O(\frac{m}{\epsilon^2}(\log\log(n) + \log(1/\epsilon_0))$-bits, and a max-communication of $O(\frac{1}{\epsilon^2}(\log\log(n) + \log(1/\epsilon_0)))$-bits.*

▶ **Theorem 26.** *There is a multi-party communication protocol in the message passing model that outputs a ϵ-additive error of the Shannon entropy H. The protocol uses a max-communication of $O(\frac{1}{\epsilon^2}(\log\log(n) + \log(1/\epsilon))$-bits.*

Since our protocol does not depend on the topology of G, a direct corollary is that we obtain a $\tilde{O}(\epsilon^{-2})$-bits of space *streaming* algorithm for entropy estimation in the random oracle model. Recall that the random oracle model allows the streaming algorithm query access to an arbitrarily long tape of random bits. This fact is used to store the random sketching matrix S.

▶ **Theorem 27.** *There is a streaming algorithm for ϵ-additive approximation of the empirical Shannon entropy of an insertion only stream in the random oracle model, which succeeds with probability $3/4$. The space required by the algorithm is $O(\frac{1}{\epsilon^2}(\log\log(n) + \log(1/\epsilon))$ bits.*

6 Approximate Matrix Product in the Message Passing Model

In this section, we consider the approximate regression problem in the message passing model over a topology $G = (V, E)$. Here, instead of vector valued inputs, each player is given as input two integral matrices $X_i \in \{0, 1, 2, \ldots, M\}^{n \times t_1}$, $Y_i \in \{0, 1, 2, \ldots, M\}^{n \times t_2}$. It is generally assumed that $n >> t_1, t_2$, so the matrices X_i, Y_i are rectangular. Let $\mathcal{X} = \sum_{i=1}^{m} X_i$ and $\mathcal{Y} = \sum_i Y_i$. The goal of the players is to approximate the matrix product $\mathcal{X}^T\mathcal{Y} \in \mathbb{R}^{t_1 \times t_2}$. Specifically, at the end of the protocol one player must output a matrix $R \in \mathbb{R}^{t_1 \times t_2}$ such that $\|R - \mathcal{X}^T\mathcal{Y}\|_F \leq \epsilon \|\mathcal{X}\|_F \|\mathcal{Y}\|_F$, where for a matrix A, $\|A\|_F = (\sum_{i,j} A_{i,j}^2)^{1/2}$ is the Frobenius norm of A.

We now describe a classic sketching algorithm which can be used to solve the approximate regression problem. The algorithm picks a $S \in \mathbb{R}^{k \times n}$ of i.i.d. Gaussian variables with variance $1/k$. It then computes $S\mathcal{X}$ and $S\mathcal{Y}$, and outputs $(S\mathcal{X})^T S\mathcal{Y}$. The following fact about such sketches will demonstrate correctness.

▶ **Lemma 28** ([43]). *Fix matrices $\mathcal{X} \in \mathbb{R}^{n\times t_1}, \mathcal{Y} \in \mathbb{R}^{n\times t_2}$ and $0 < \epsilon_0$. Let $S \in \mathbb{R}^{k\times n}$ be a matrix of i.i.d. Gaussian random variables with variance $1/k$, for $k = \Theta(1/(\delta\epsilon_0^2))$. Then we have $\Pr[\|\mathcal{X}^T S^T S\mathcal{Y} - \mathcal{X}^T\mathcal{Y}\|_F \le \epsilon_0\|\mathcal{X}\|_F\|\mathcal{Y}\|_F] \ge 1-\delta$. Moreover, with the same probability we have $\|S\mathcal{X}\|_F = (1\pm\epsilon_0)\|\mathcal{X}\|_F$ and $\|S\mathcal{Y}\|_F = (1\pm\epsilon_0)\|\mathcal{Y}\|_F$*

Now by Lemma 11, the central vertex $\mathcal{C}$ can recover a value $r_{\mathcal{C}}^{i,j}$ such that $\mathbb{E}[r_{\mathcal{C}}^{i,j}] = (S\mathcal{X})_{i,j}$ and $\mathbf{Var}[r_{\mathcal{C}}^{i,j}] \le \epsilon^2\|\mathcal{X}_{*,j}\|_2$ (after setting δ sufficiently small), where $\mathcal{X}_{*,j}$ is the j-th *column* of $\mathcal{X}$. Thus, the central vertex can obtain a random matrix $R^{\mathcal{X}} \in \mathbb{R}^{k\times t_1}$ such that $\mathbb{E}[R^{\mathcal{X}}] = (S\mathcal{X})$ and $\mathbb{E}[\|R^{\mathcal{X}} - S\mathcal{X}\|_F^2] \le k\epsilon^2\sum_{j=1}^{t_1}\|\mathcal{X}_{*,j}\|_2$. Setting $\epsilon = \mathrm{poly}(1/k) = \mathrm{poly}(1/\epsilon_0)$ small enough, we obtain $\mathbb{E}[\|R^{\mathcal{X}} - S\mathcal{X}\|_F^2] \le \epsilon_0^2\|\mathcal{X}\|_F$. Similarly, we can obtain a $R^{\mathcal{Y}}$ at the central vertex $\mathcal{C}$, and output the estimate $R = (R^{\mathcal{X}})^T R^{\mathcal{Y}}$. Utilizing the error guarantees of Lemma 11 as well as Lemma 28, we obtain the following theorem.

▶ **Theorem 29.** *Given inputs $\mathcal{X} = \sum_{i=1}^m X_i, \mathcal{Y} = \sum_{i=1}^m Y_i$ as described above, there is a protocol which outputs, at the central vertex $\mathcal{C}$, a matrix $R \in \mathbb{R}^{t_1\times t_2}$ such that with probability $3/4$ we have $\|R - \mathcal{X}^T\mathcal{Y}\|_F \le \epsilon\|\mathcal{X}\|_F\|\mathcal{Y}\|_F$ The max communication required by the protocol is $O\left(\epsilon^{-2}(t_1+t_2)(\log\log n + \log 1/\epsilon + \log d)\right)$, where d is the diameter of the communication topology G.*

References

1. Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 20–29. ACM, 1996.
2. Alexandr Andoni, Robert Krauthgamer, and Krzysztof Onak. Streaming algorithms from precision sampling. *arXiv preprint*, 2010. `arXiv:1011.1263`.
3. Chrisil Arackaparambil, Joshua Brody, and Amit Chakrabarti. Functional monitoring without monotonicity. In *International Colloquium on Automata, Languages, and Programming*, pages 95–106. Springer, 2009.
4. Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom. Models and issues in data stream systems. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 1–16. ACM, 2002.
5. Maria Florina Balcan, Yingyu Liang, Le Song, David Woodruff, and Bo Xie. Communication efficient distributed kernel principal component analysis. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 725–734. ACM, 2016.
6. Ziv Bar-Yossef, Thathachar S Jayram, Ravi Kumar, and D Sivakumar. An information statistics approach to data stream and communication complexity. *Journal of Computer and System Sciences*, 68(4):702–732, 2004.
7. Jarosław Błasiok, Jian Ding, and Jelani Nelson. Continuous monitoring of lp norms in data streams. *arXiv preprint*, 2017. `arXiv:1704.06710`.
8. Mark Braverman, Faith Ellen, Rotem Oshman, Toniann Pitassi, and Vinod Vaikuntanathan. A tight bound for set disjointness in the message-passing model. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 668–677. IEEE, 2013.
9. Vladimir Braverman, Stephen R Chestnut, Nikita Ivkin, Jelani Nelson, Zhengyu Wang, and David P Woodruff. BPTree: an L2 heavy hitters algorithm using constant memory. *arXiv preprint*, 2016. `arXiv:1603.00759`.
10. 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 *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 261–276. ACM, 2016.

11 Vladimir Braverman, Jonathan Katzman, Charles Seidell, and Gregory Vorsanger. An Optimal Algorithm for Large Frequency Moments Using O (nˆ(1-2/k)) Bits. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2014)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2014.

12 Vladimir Braverman and Rafail Ostrovsky. Recursive sketching for frequency moments. *arXiv preprint*, 2010. `arXiv:1011.2571`.

13 Vladimir Braverman, Emanuele Viola, David Woodruff, and Lin F Yang. Revisiting frequency moment estimation in random order streams. *arXiv preprint*, 2018. `arXiv:1803.02270`.

14 Amit Chakrabarti, Graham Cormode, and Andrew McGregor. A near-optimal algorithm for estimating the entropy of a stream. *ACM Transactions on Algorithms (TALG)*, 6(3):51, 2010.

15 Amit Chakrabarti, Graham Cormode, and Andrew McGregor. Robust Lower Bounds for Communication and Stream Computation. *Theory of Computing*, 12(1):1–35, 2016.

16 Amit Chakrabarti, Subhash Khot, and Xiaodong Sun. Near-optimal lower bounds on the multi-party communication complexity of set disjointness. In *18th IEEE Annual Conference on Computational Complexity, 2003. Proceedings.*, pages 107–117. IEEE, 2003.

17 Amit Chakrabarti and Oded Regev. An optimal lower bound on the communication complexity of gap-hamming-distance. *SIAM Journal on Computing*, 41(5):1299–1317, 2012.

18 Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. *Automata, languages and programming*, pages 784–784, 2002.

19 Arkadev Chattopadhyay, Jaikumar Radhakrishnan, and Atri Rudra. Topology matters in communication. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 631–640. IEEE, 2014.

20 Jiecao Chen, He Sun, David Woodruff, and Qin Zhang. Communication-optimal distributed clustering. In *Advances in Neural Information Processing Systems*, pages 3727–3735, 2016.

21 Peter Clifford and Ioana Cosma. A simple sketching algorithm for entropy estimation over streaming data. In *Artificial Intelligence and Statistics*, pages 196–206, 2013.

22 Graham Cormode, Piotr Indyk, Nick Koudas, and S Muthukrishnan. Fast mining of massive tabular data via approximate distance computations. In *Proceedings 18th International Conference on Data Engineering*, pages 605–614. IEEE, 2002.

23 Graham Cormode and Hossein Jowhari. L p Samplers and Their Applications: A Survey. *ACM Computing Surveys (CSUR)*, 52(1):16, 2019.

24 Graham Cormode, S Muthukrishnan, and Irina Rozenbaum. Summarizing and mining inverse distributions on data streams via dynamic inverse sampling. In *Proceedings of the 31st international conference on Very large data bases*, pages 25–36. VLDB Endowment, 2005.

25 Graham Cormode, S Muthukrishnan, and Ke Yi. Algorithms for distributed functional monitoring. *ACM Transactions on Algorithms (TALG)*, 7(2):21, 2011.

26 Philippe Flajolet. Approximate counting: a detailed analysis. *BIT Numerical Mathematics*, 25(1):113–134, 1985.

27 Phillip B Gibbons and Yossi Matias. New sampling-based summary statistics for improving approximate query answers. In *ACM SIGMOD Record*, volume 27, pages 331–342. ACM, 1998.

28 Phillip B Gibbons, Yossi Matias, and Viswanath Poosala. Fast Incremental Maintenance of Approximate Histograms. In *Proceedings of the 23rd International Conference on Very Large Data Bases*, pages 466–475. Morgan Kaufmann Publishers Inc., 1997.

29 Anna C Gilbert, Yannis Kotidis, S Muthukrishnan, and Martin J Strauss. How to summarize the universe: Dynamic maintenance of quantiles. In *VLDB'02: Proceedings of the 28th International Conference on Very Large Databases*, pages 454–465. Elsevier, 2002.

30 Andre Gronemeier. Asymptotically optimal lower bounds on the nih-multi-party information. *arXiv preprint*, 2009. `arXiv:0902.1609`.

31 Nicholas JA Harvey, Jelani Nelson, and Krzysztof Onak. Sketching and streaming entropy via approximation theory. In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 489–498. IEEE, 2008.

32 Nicholas JA Harvey, Jelani Nelson, and Krzysztof Onak. Streaming algorithms for estimating entropy. In *2008 IEEE Information Theory Workshop*, pages 227–231. IEEE, 2008.

33 Ling Huang, XuanLong Nguyen, Minos Garofalakis, Joseph M Hellerstein, Michael I Jordan, Anthony D Joseph, and Nina Taft. Communication-efficient online detection of network-wide anomalies. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 134–142. IEEE, 2007.

34 Zengfeng Huang, Ke Yi, and Qin Zhang. Randomized algorithms for tracking distributed count, frequencies, and ranks. In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGAI symposium on Principles of Database Systems*, pages 295–306. ACM, 2012.

35 Piotr Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *Journal of the ACM (JACM)*, 53(3):307–323, 2006.

36 Piotr Indyk and David Woodruff. Optimal approximations of the frequency moments of data streams. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 202–208. ACM, 2005.

37 Rajesh Jayaram, Gokarna Sharma, Srikanta Tirthapura, and David P. Woodruff. Weighted Reservoir Sampling from Distributed Streams. In *Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, SIGMOD/PODS '19, 2019.

38 Rajesh Jayaram and David P Woodruff. Perfect lp sampling in a data stream. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 544–555. IEEE, 2018.

39 Thathachar S Jayram, Ravi Kumar, and D Sivakumar. The One-Way Communication Complexity of Hamming Distance. *Theory of Computing*, 4(1):129–135, 2008.

40 Thathachar S Jayram and David P Woodruff. The data stream space complexity of cascaded norms. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 765–774. IEEE, 2009.

41 TS Jayram. Hellinger strikes back: A note on the multi-party information complexity of AND. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 562–573. Springer, 2009.

42 Hossein Jowhari, Mert Sağlam, and Gábor Tardos. Tight Bounds for Lp Samplers, Finding Duplicates in Streams, and Related Problems. In *Proceedings of the Thirtieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '11, pages 49–58, New York, NY, USA, 2011. ACM. `doi:10.1145/1989284.1989289`.

43 Daniel M Kane and Jelani Nelson. Sparser johnson-lindenstrauss transforms. *Journal of the ACM (JACM)*, 61(1):4, 2014.

44 Daniel M Kane, Jelani Nelson, Ely Porat, and David P Woodruff. Fast moment estimation in data streams in optimal space. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 745–754. ACM, 2011.

45 Daniel M Kane, Jelani Nelson, and David P Woodruff. On the exact space complexity of sketching and streaming small norms. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 1161–1178. SIAM, 2010.

46 Michael Kapralov, Jelani Nelson, Jakub Pachocki, Zhengyu Wang, David P Woodruff, and Mobin Yahyazadeh. Optimal lower bounds for universal relation, and for samplers and finding duplicates in streams. *arXiv preprint*, 2017. `arXiv:1704.00633`.

47 Ping Li and Cun-Hui Zhang. A new algorithm for compressed counting with applications in shannon entropy estimation in dynamic data. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 477–496, 2011.

48 Yi Li and David P Woodruff. A tight lower bound for high frequency moment estimation with small error. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 623–638. Springer, 2013.

49 Andrew McGregor, A Pavan, Srikanta Tirthapura, and David P Woodruff. Space-Efficient Estimation of Statistics Over Sub-Sampled Streams. *Algorithmica*, 74(2):787–811, 2016.

50 Morteza Monemizadeh and David P Woodruff. 1-pass relative-error lp-sampling with applications. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 1143–1160. SIAM, 2010.

51 Robert Morris. Counting large numbers of events in small registers. *Communications of the ACM*, 21(10):840–842, 1978.

52 Shanmugavelayutham Muthukrishnan et al. Data streams: Algorithms and applications. *Foundations and Trends® in Theoretical Computer Science*, 1(2):117–236, 2005.

53 Noam Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449–461, 1992.

54 J. P. Nolan. *Stable Distributions - Models for Heavy Tailed Data*. Birkhauser, Boston, 2018. In progress, Chapter 1 online at http://fs2.american.edu/jpnolan/www/stable/stable.html.

55 Frank Olken. *Random sampling from databases*. PhD thesis, University of California, Berkeley, 1993.

56 Srikanta Tirthapura and David P Woodruff. Optimal random sampling from distributed streams revisited. In *International Symposium on Distributed Computing*, pages 283–297. Springer, 2011.

57 Omri Weinstein and David P Woodruff. The simultaneous communication of disjointness with applications to data streams. In *International Colloquium on Automata, Languages, and Programming*, pages 1082–1093. Springer, 2015.

58 David Woodruff. Optimal space lower bounds for all frequency moments. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 167–175. Society for Industrial and Applied Mathematics, 2004.

59 David P Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science*, 10(1–2):1–157, 2014.

60 David P Woodruff and Qin Zhang. Tight bounds for distributed functional monitoring. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 941–960. ACM, 2012.

61 David P Woodruff and Qin Zhang. Distributed Statistical Estimation of Matrix Products with Applications. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 383–394. ACM, 2018.

62 David P Woodruff and Peilin Zhong. Distributed low rank approximation of implicit functions of a matrix. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pages 847–858. IEEE, 2016.

63 Ke Yi and Qin Zhang. Optimal tracking of distributed heavy hitters and quantiles. *Algorithmica*, 65(1):206–223, 2013.

A Proof Sketch of $\Omega(m/\epsilon^2)$ Lower Bound for F_p estimation in the One-Way Coordinator Model

We now sketch the proof of the $\Omega(m/\epsilon^2)$ lower bound that was remarked upon in the introduction. First, consider the following problem Alice is given a vector $x \in \mathbb{R}^t$, and bob $y \in \mathbb{R}^t$, such that $x_i \geq 0, y_i \geq 0$ for all $i \in [t]$. Alice and Bob both send a message to Eve, who must then output a $(1 \pm \epsilon)$ approximation to $\|x + y\|_p$, for $p \in (0, 2] \setminus \{1\}$. Via a reduction from the Gap-Hamming communication problem, there is a $\Omega(1/\epsilon^2)$-bit communication lower bound for this problem [58]. More specifically, there is a distribution $\mathcal{D}$ over inputs $(x, y) \in \mathbb{R}^t \times \mathbb{R}^t$, such that any communication protocol that solves the above problem on these inputs correctly with probability 3/4 must send $\Omega(1/\epsilon^2)$ bits.

Now consider the one-way coordinator model, where there are m players connected via an edge to a central coordinator. They are given inputs $x_1, \ldots, x_m$, and must each send a single message to the coordinator, who them must estimate $\|x\|_p = \|x_1 + x_2 + \cdots + x_m\|_p$. Consider

two distribution, P_1, P_2 over the inputs $(x_1, \ldots, x_m)$. In the first, two players i, j are chosen uniformly at random, and given as inputs $(x, y) \sim \mathcal{D}$, and the rest of the players are given the 0 vector. In P_2, we draw $(x, y) \sim \mathcal{D}$, and every player is given either x or y at random. The players are then either given input from P_1 or P_2, with probability 1/2 for each. In the first case, if the two players with the input do not send $\Omega(1/\epsilon^2)$ bits, then they will not be able to solve the estimation problem via the 2-party lower bound. However, given only their input, the distributions P_1 and P_2 are indistinguishable to a given player. So the players cannot tell if the input is from P_1 or P_2, so any player that gets an non-zero input must assume they are in case P_1 if they want to solve the communication problem with sufficiently high constant probability, and send $\Omega(1/\epsilon^2)$ bits of communication. This results in $\Omega(m/\epsilon^2)$ total communication when the input is from P_2, which is the desired lower bound.

B $\Omega(1/\epsilon^2)$ Lower Bound for additive approximation of Entropy in Insertion-Only Streams

We now prove the $\Omega(1/\epsilon^2)$-bits of space lower bound for any streaming algorithm that produces an approximation $\tilde{H}$ such that $|\tilde{H} - H| < \epsilon$ with probability 3/4. Here H is the empirical entropy of the stream vector $\mathcal{X}$, namely $H = H(\mathcal{X}) = -\sum_{i=1}^{n} \frac{|\mathcal{X}_i|}{F_1} \log \frac{|\mathcal{X}_i|}{F_1}$. To prove the lower bound, we must first introduce the GAP-HAMDIST problem. Here, there are two players, Alice and Bob. Alice is given $x \in \{0,1\}^t$ and Bob receives $y \in \{0,1\}^t$. Let $\Delta(x, y) = |\{i \mid x_i \neq y_i\}|$ be the Hamming distance between two binary strings x, y. Bob is promised that either $\Delta(x, y) \leq t/2 - \sqrt{t}$ (NO instance) or $\Delta(x, y) \geq t/2 + \sqrt{t}$ (YES instance), and must decide which holds. Alice must send a single mesage to Bob, from which he must decide which case the inputs are in. It is known that any protocol which solves this problem with constant probability must send $\Omega(t)$-bits in the worst case (i.e. the maximum number of bits sent, taken over all inputs and random bits used by the protocol).

▶ **Proposition 30** ([58, 39]). *Any protocol which solves the* GAP-HAMDIST *problem with probability at least* 2/3 *must send* $\Omega(t)$*-bits of communication in the worst case.*

We remark that while a $\Omega(1/\epsilon^2)$ lower bound is known for *multiplicative-approximation* of the entropy, to the best of our knowledge there is no similar lower bound written in the literature for additive approximation.

▶ **Theorem 31.** *Any algorithm for* ϵ*-additive approximation of the entropy* H *of a stream, in the insertion-only model, which succeeds with probability at least* 2/3*, requires space* $\Omega(\epsilon^{-2})$

Proof. Given a $x, y \in \{0,1\}^t$ instance of GAP-HAMDIST, for $t = \Theta(1/\epsilon^2)$, Alice constructs a stream on $2t$ items. Let x' be the result of flipping all the bits of x, and let $x'' = x \circ 0^t + 0^t \circ x' \in \{0,1\}^{2t}$ where $\circ$ denotes concatenation. Define y', y'' similarly. Alice then inserts updates so that the stream vector $\mathcal{X} = x''$, and then sends the state of the streaming algorithm to Bob, who inserts his vector, so that now $\mathcal{X} = x'' + y''$. We demonstrate that the entropy of H differs by an additive term of at least ϵ between the two cases. In all cases case, we have

$$\begin{aligned} H &= \frac{t-\Delta}{t}\log(t) + \frac{\Delta}{2t}\log(2t) \\ &= \log(t) + \Delta\left(\frac{2\log(t) - \log 2t}{2t}\right) \end{aligned} \tag{1}$$

We can assume $t \geq 4$, and then $2\log(t) - \log(2t) = C > 0$, where C is some fixed value known to both players that is bounded away from 0. So as Δ increases, the entropy increases. Thus in a YES instance, the entropy is at least

$$\begin{aligned} H &\geq \log(t) + (t/2 + \sqrt{t})\frac{C}{2t} \\ &= \log(t) + (1/4 + 1/2\sqrt{t})C \\ &= \log(t) + C/4 + \Theta(\epsilon) \end{aligned} \tag{2}$$

In addition, in the NO instance, the entropy is maximized when $\Delta = t/2 - \sqrt{T}$. so we have

$$\begin{aligned} H &\leq \log(t) + (t/2 - \sqrt{t})\frac{C}{2t} \\ &= \log(t) + C/4 - \Theta(\epsilon) \end{aligned} \tag{3}$$

Therefore, the entropy differs between YES and NO instances by at least an additive $\Theta(\epsilon)$ term. After sufficient rescaling of ϵ by a constant, we obtain our $\Omega(t) = \Omega(1/\epsilon^2)$ lower bound for additive entropy estimation via the linear lower bound for GAP-HAMDIST from Proposition 30. ◀