Sketching and Streaming Entropy via Approximation Theory*

Nicholas J. A. Harvey† MIT nickh@mit.edu

Jelani Nelson‡ MIT minilek@mit.edu

Krzysztof Onak§ MIT konak@mit.edu

Abstract

We give near-optimal sketching and streaming algorithms for estimating Shannon entropy in the most general streaming model, with arbitrary insertions and deletions. This improves on prior results that obtain suboptimal space bounds in the general model, and near-optimal bounds in the insertion-only model without sketching. Our high-level approach is simple: we give algorithms to estimate Tsallis entropy, and use them to extrapolate an estimate of Shannon entropy. The accuracy of our estimates is proven using approximation theory arguments and extremal properties of Chebyshev polynomials. Our work also yields the best-known and near-optimal additive approximations for entropy, and hence also for conditional entropy and mutual information.

1 Introduction

Streaming algorithms have attracted much attention in several computer science communities, notably theory, databases, and networking. Many algorithmic problems in this model are now well-understood, for example, the problem of estimating frequency moments [1, 2, 12, 19, 34, 37]. More recently, several researchers have studied the problem of estimating the empirical entropy of a stream [4, 7, 8, 14, 15, 39].

Motivation. Entropy is a fundamentally important quantity that can be used to measure information content, the uncertainty of a random variable, or the compressibility of a text. It also finds several practical applications in computer networking, such as network anomaly detection. Let us consider a concrete example. One form of malicious activity on the internet is *port scanning*, in which attackers probe target machines, trying to find open ports which could be leveraged for further attacks. In contrast, typical internet traffic is directed to a small number of heavily used ports for web traffic, email delivery, etc. Consequently, when a port scanning attack is underway, there is a significant change in the distribution of port numbers in the packets being delivered. It has been shown that measuring the entropy of the distribution of port numbers provides an effective means to detect such attacks. See Lakhina et al. [20] and Xu et al. [38] for further information about such problems and methods for their solution.

Our Techniques. In this paper, we give an algorithm for estimating empirical Shannon entropy while using space nearly optimal in terms of the desired estimation accuracy. Our algorithm is actually a sketching algorithm, not just a streaming algorithm, and it applies to general streams which allow insertions and deletions of elements. One attractive aspect of our work is its clean high-level approach: we reduce the entropy estimation problem to the well-studied frequency moment problem. More concretely, we give algorithms for estimating Tsallis entropy, which is closely related to frequency moments. The link to Shannon entropy is established by proving bounds on the rate at which Tsallis entropy converges to Shannon entropy.

The full version of this paper establishes similar results for the convergence of Rényi entropy to Shannon entropy. Remarkably, it seems that such an analysis was not previously known.

There are several technical obstacles that arise with this approach. Unfortunately, it does not seem that the optimal amount of space can be obtained while using just a single estimate of Tsallis entropy. We overcome this obstacle by using several estimates, together with approximation theory arguments and certain extremal properties of Chebyshev polynomials. To our knowledge, this is the first use of such techniques in the context of streaming algorithms, and it seems likely that these techniques could be applicable to many other problems.

Such arguments yield good algorithms for additively estimating entropy, but obtaining a good multiplicative approximation is more difficult when the entropy is very

*An early version of this work that presented a simpler algorithm appeared in the IEEE Information Theory Workshop [16].

†Supported in part by a Natural Sciences and Engineering Research Council of Canada PGS Scholarship, by NSF contract CCF-0515221 and by ONR grant N00014-05-1-0148.

‡Supported by a National Defense Science and Engineering Graduate (NDSEG) Fellowship.

§Supported in part by NSF grant 0514771.

small. In such a scenario, there is necessarily a very heavy element, and the task that one must solve is to estimate the moment of all elements *excluding* this heavy element. This task has become known as the *residual moment* estimation problem, and it is emerging as a useful building block for other streaming problems [4, 6, 12]. To estimate the α^{th} residual moment for $\alpha \in (0, 2]$, we show that $\tilde{O}(\varepsilon^{-2} \log m)$ bits[1,2] of space suffice with a random oracle and $\tilde{O}(\varepsilon^{-2} \log^2 m)$ bits without. Here we use the notation $f(m, \varepsilon) = \tilde{O}(g(m, \varepsilon))$ if $f(m, \varepsilon) = O(g(m, \varepsilon)(\log\log m + \log(1/\varepsilon))^{O(1)})$. In comparison, existing algorithms use $O(\varepsilon^{-2} \log^2 m)$ bits for $\alpha = 2$ [13], and $O(\varepsilon^{-2} \log m)$ for $\alpha = 1$ [12]. No non-trivial algorithms were previously known for $\alpha \notin \{1, 2\}$. That said, the previously known algorithms were more general in ways irrelevant to our work: they can remove the k heaviest elements without requiring that they are sufficiently heavy.

Multiplicative Entropy Estimation. Let us now state the performance of the entropy estimation algorithms more explicitly. We focus exclusively on single-pass algorithms unless otherwise noted. The first algorithms for approximating entropy in the streaming model are due to Guha et al. [15]; they achieved $O(\varepsilon^{-2} + \log m)$ words of space[3] but assumed a randomly ordered stream. Chakrabarti, Do Ba and Muthukrishnan [8] then gave an algorithm for worst-case ordered streams using $O(\varepsilon^{-2} \log^2 m)$ words of space, but required two passes over the input. The algorithm of Chakrabarti, Cormode and McGregor [7] uses $O(\varepsilon^{-2} \log m)$ words of space to give a multiplicative $1 + \varepsilon$ approximation, although their algorithm cannot produce sketches and only applies to insertion-only streams. In contrast, the algorithm of Bhuvanagiri and Ganguly [4] provides a sketch and can handle deletions but requires roughly $\tilde{O}(\varepsilon^{-3} \log^4 m)$ words[4].

Our work focuses primarily in the *strict turnstile model* (defined in Section 2), which allows deletions. Our algorithm for multiplicatively estimating Shannon entropy uses $\tilde{O}(\varepsilon^{-2} \log m)$ words of space. These bounds are nearly-optimal in terms of the dependence on ε, since there is a lower bound of $\tilde{\Omega}(\varepsilon^{-2})$ bits even for insertion-only streams [7]. Our algorithms assume access to a random oracle. This assumption can be removed through the use of Nisan's pseudorandom generator [23], increasing the space bounds by a factor of $O(\log(m/\varepsilon))$.

[1]When giving bounds, we often use the following tilde notation: we say $f(m, \varepsilon) = \tilde{O}(g(m, \varepsilon))$ if $f(m, \varepsilon) = O(g(m, \varepsilon)(\log\log m + \log(1/\varepsilon))^{O(1)})$.

[2]The length of the stream is denoted m and the approximation accuracy is $1 + \varepsilon$. For precise definitions, see Section 2.

[3]A word is a string of $\lceil \log m \rceil$ bits.

[4]A recent, yet unpublished improvement by the same authors [5] improves this to $\tilde{O}(\varepsilon^{-3} \log^3 m)$ words.

Additive Entropy Estimation. Additive approximations of entropy are also useful, as they directly yield additive approximations of conditional entropy and mutual information, which cannot be approximated multiplicatively in small space [18]. Chakrabarti et al. [7] noted that since Shannon entropy is bounded above by $\log m$, a multiplicative $(1 + (\varepsilon/\log m))$ approximation yields an additive ε-approximation. In this way, the work of Chakrabarti et al. [7] and Bhuvanagiri and Ganguly [4] yield additive ε approximations using $O(\varepsilon^{-2} \log^3 m)$ and $\tilde{O}(\varepsilon^{-3} \log^7 m)$ words of space respectively. Our algorithm yields an additive ε approximation using only $\tilde{O}(\varepsilon^{-2} \log m)$ words of space. In particular, our space bounds for multiplicative and additive approximation differ by only $\log\log m$ factors. Zhao et al. [39] give practical methods for additively estimating the so-called entropy norm of a stream. Their algorithm can be viewed as a special case of ours since it interpolates Shannon entropy using two estimates of Tsallis entropy, although this interpretation was seemingly unknown to those authors.

Other Information Statistics. We also give algorithms for approximating Rényi [30] and Tsallis [35] entropy. Rényi entropy plays an important role in expanders, pseudorandom generators, quantum computation, and ecology. Tsallis entropy is an important quantity in physics that generalizes Boltzmann-Gibbs entropy, and also plays a role in quantum physics. Rényi and Tsallis entropy are both parameterized by a scalar $\alpha \geq 0$. The efficiency of our estimation algorithms depends on α, and is stated precisely in Section 7.

Approximating Entropy from Samples. One conceivable approach to design a streaming algorithm for approximating entropy is to leverage the existing work on approximating the entropy of a discrete probability distribution from independent random samples [3, 25, 26, 29, 36]. It can easily be shown that in this model it is not possible to obtain a multiplicative entropy approximation with $o(m)$ samples. It is also known [3, 29, 36] that additive approximation of entropy requires $n^{\Omega(1)}$ samples. Therefore, it seems unlikely that the sampling approach yields space-efficient streaming algorithms.

2 Preliminaries

Let $A = (A_1, \ldots, A_n) \in \mathbb{Z}^n$ be a vector initialized as $\vec{0}$ which is modified by a stream of m updates. Each update is of the form (i, v), where $i \in [n]$ and $v \in \{-M, \ldots, M\}$, and causes the change $A_i \leftarrow A_i + v$. For simplicity in stating bounds, we henceforth assume $m \geq n$ and $M = 1$; the latter can be simulated by increasing m by a factor of M and representing an update (i, v) with $|v|$ separate updates (though in actuality our algorithm can perform all $|v|$ updates simultaneously in the time it takes to do one update). The vector A gives rise to a probability distribution

$x = (x_1, \ldots, x_n)$ with $x_i = |A_i| / \|A\|_1$. Thus for each i either $x_i = 0$ or $x_i \geq 1/m$.

In the *strict turnstile model*, we assume $A_i \geq 0$ for all $i \in [n]$ at the end of the stream. In the *general update model* we make no such assumption. For the remainder of this paper, we assume the strict turnstile model and assume access to a random oracle, unless stated otherwise. Our algorithms also extend to the general update model, typically increasing bounds by a factor of $O(\log m)$. As remarked above, the random oracle can be removed, using [23], while increasing the space by another $O(\log(m/\varepsilon))$ factor.

We now define some notation. For real $\alpha > 0$, the α^{th} norm of a vector $x \in \mathbb{R}^n$ is defined $\|x\|_\alpha = (\sum_{i=1}^n |x_i|^\alpha)^{1/\alpha}$; also, $\|x\|_0 = |\{\, i : x_i \neq 0 \,\}|$. (For $\alpha \in [0, 1)$, $\|\cdot\|_\alpha$ is not actually a norm in the usual sense.) We define the α^{th} *moment* of the stream as $F_\alpha = \sum_{i=1}^n |A_i|^\alpha = \|A\|_\alpha^\alpha$. For x a probability distribution, we define the α^{th} *Rényi entropy* as $H_\alpha = \log(\|x\|_\alpha^\alpha)/(1-\alpha)$ and the α^{th} *Tsallis entropy* as $T_\alpha = (1 - \|x\|_\alpha^\alpha)/(\alpha - 1)$. Shannon entropy H is defined by $H = -\sum_{i=1}^n x_i \log x_i$. A straightforward application of l'Hôpital's rule shows that $H = \lim_{\alpha \to 1} H_\alpha = \lim_{\alpha \to 1} T_\alpha$. It is often convenient to focus on the quantity $\alpha - 1$ instead of α, so we define $H(a) = H_{1+a}$ and $T(a) = T_{1+a}$.

We will often need to approximate frequency moments, for which we use the following:

Fact 2.1 (Indyk [17], Li [21], [22]). There is an algorithm to compute a multiplicative $(1+\varepsilon)$-approximation of F_α for any $\alpha \in (0, 2]$. The algorithm succeeds with constant probability. It uses $O(\varepsilon^{-2} \log m)$ bits of space in the general update model, and $O\left(\left(\frac{|\alpha-1|}{\varepsilon^2} + \frac{1}{\varepsilon}\right) \log m\right)$ bits of space in the strict turnstile model.

For any function $a \mapsto f(a)$, we denote its k^{th} derivative with respect to a by $f^{(k)}(a)$.

3 A Simple Algorithm

As a precursor to our full approach, consider estimating Shannon entropy H by estimating Tsallis entropy $T(y) = T_{1+y}$ for $y \approx 0$. To do so, we can use Fact 2.1 to compute $\tilde{F}_{1+y}$, a $(1 \pm \tilde{\varepsilon})$-approximation to F_{1+y}. To be concrete, we choose $y = -\Theta(\varepsilon/(\log n \log m))$ and $\tilde{\varepsilon} = \varepsilon \cdot y$. The space required is $O(\varepsilon^{-3} \log n \log m)$ words. The following argument shows this gives an additive $O(\varepsilon)$ approximation. With constant probability, $\tilde{F}_{1+y} = (1 \pm \tilde{\varepsilon}) F_{1+y}$. Then our estimate is

$$\begin{aligned} \tilde{T}(y) &= \frac{1}{y}\left(1 - \frac{\tilde{F}_{1+y}}{||A||_1^{1+y}}\right) \qquad (3.1) \\ &= \frac{1}{y}\left(1 - \frac{F_{1+y}}{||A||_1^{1+y}}\right) + \frac{\tilde{\varepsilon}}{y} \sum_{i=1}^{n} \left(\frac{A_i}{||A||_1}\right)^{1+y} \\ &= T(y) \pm O\left(\frac{\tilde{\varepsilon}}{y}\right) = H \pm O(\varepsilon). \end{aligned}$$

The third equality holds by choice of y since $1/m \leq A_i/||A||_1 \leq 1$. The last equality follows by the mean value theorem and a bound on the absolute value of the derivative of T near y. We prove such a bound in Section 5.2.1. Specifically, Lemma 5.1 with $\varepsilon = \tilde{\varepsilon}$ and $k = 1$ shows that the derivative $T^{(1)}(z)$ is $O(\log n \log m)$ for $y \leq z < 0$.

In Section 5, we improve on this simple algorithm by estimating Tsallis entropy at multiple points. This scheme is analyzed using certain approximation theory arguments, which we discuss in the next section.

4 Noisy Extrapolation

In this section, we describe an extrapolation technique that lies at the heart of our main streaming algorithms for Shannon entropy. Let $f : \mathbb{R} \to \mathbb{R}$ be a continuous function that we can evaluate approximately at every point except 0. Further, suppose that evaluating $f(y)$ becomes increasingly expensive as y goes to 0. We want to approximate $f(0)$. Therefore, we approximate f at a few carefully chosen points $y_0, \ldots, y_k$ far from 0 and use the achieved values to extrapolate the value of f at 0. Let $z_i = f(y_i) + \Delta_i$ be the approximation to $f(y_i)$ that we compute. Δ_i is the error on approximating $f(y_i)$. We then compute the only polynomial p of degree at most k such that $p(y_i) = z_i$, and hope that $p(0)$ is a good approximation to $f(0)$.

The polynomial p can be decomposed into two polynomials p_f and p_Δ of degree at most k such that $p = p_f + p_\Delta$, and for each i, $p_f(y_i) = f(y_i)$ and $p_\Delta(y_i) = \Delta_i$. We have $|p(0) - f(0)| \leq |p_f(0) - f(0)| + |p_\Delta(0)|$. We analyze and bound each of the last two terms separately. A standard result on approximation of functions by polynomials can be used to bound the first term, provided f is sufficiently smooth. Bounding the second term is one of the main contributions of the paper. It requires a careful choice of y_i and employs extremal properties of Chebyshev polynomials. An application of the technique is described in more detail in Section 5.2.

4.1 Bounding the First Error Term

The following standard result on approximation of functions by polynomials can be used to bound the error due to use of extrapolation. Recall that the notation $f^{(k)}$ denotes the k^{th} derivative of f.

Fact 4.1 (Phillips and Taylor [28], Theorem 4.2). Let $y_0, y_1, \ldots, y_k$ be points in the interval $[a, b]$. Let $f : \mathbb{R} \to \mathbb{R}$ be such that $f^{(1)}, \ldots, f^{(k)}$ exist and are continuous on $[a, b]$, and $f^{(k+1)}$ exists on (a, b). Then, for every $y \in [a, b]$, there exists $\xi_y \in (a, b)$ such that

$$f(y) - p_f(y) = \left(\prod_{i=0}^{k} (y - y_i)\right) \frac{f^{(k+1)}(\xi_y)}{(k+1)!},$$

where $p_f(y)$ is the degree-k polynomial obtained by interpolating the points $(y_i, f(y_i))$, $0 \le i \le k$.

As long as f is sufficiently smooth and has bounded derivatives, and y is not too far from each y_i, the above fact immediately yields a good bound on the extrapolation error.

4.2 Bounding the Second Error Term

We now show how to bound $|p_\triangle(0)|$, the error due to the fact that we learn each $f(y_i)$ only approximately. The careful choice of $y_0, y_1, \ldots, y_k$ and extremal properties of Chebyshev polynomials are used to limit $|p_\triangle(0)|$. We first describe properties of Chebyshev polynomials that are important to us, then explain how we pick our points y_0 through y_k, and eventually sketch how the absolute value of $p_\triangle$ can be bounded at 0.

4.2.1 Review of Chebyshev Polynomials

Our technique exploits certain extremal properties of Chebyshev polynomials. For a basic introduction to Chebyshev polynomials we refer the reader to [27, 28, 31]. A thorough treatment of these objects can be found in [32]. We now present the background relevant for our purposes.

Definition 4.2. The set $\mathcal{P}_k$ consists of all polynomials of degree at most k with real coefficients. The Chebyshev polynomial of degree k, $P_k(x)$, is defined by the recurrence

$$P_k(x) \;=\; \begin{cases} 1, & (k=0) \\ x, & (k=1) \\ 2xP_{k-1}(x) - P_{k-2}(x), & (k \ge 2) \end{cases} \qquad (4.1)$$

and satisfies $|P_k(x)| \le 1$ for all $x \in [-1,1]$. The value $|P_k(x)|$ equals 1 for exactly $k+1$ values of x in $[-1,1]$; specifically, $P_k(\eta_{j,k}) = (-1)^j$ for $0 \le j \le k$, where $\eta_{j,k} = \cos(j\pi/k)$. The set $\mathcal{C}_k$ is defined as the set of all polynomials $p \in \mathcal{P}_k$ satisfying $\max_{0\le j\le k} |p(\eta_{j,k})| \le 1$.

Fact 4.3 (Extremal Growth Property). If $p \in \mathcal{C}_k$ and $|t| \ge 1$, then $|p(t)| \le |P_k(t)|$.

Proof. See [32, Ex. 1.5.11] or Rogosinski [33]. ∎

Fact 4.3 states that all polynomials which are bounded on certain "critical points" of the interval $I = [-1,1]$ cannot grow faster than Chebyshev polynomials once leaving I.

4.2.2 The Choice of y_i

We will use Fact 4.3 to bound $p_\triangle(0)$. Since this fact provides no guarantees at $t = 0$, we produce a new polynomial from $p_\triangle$ by applying an affine map to its domain, then bound this new polynomial at a point t slightly larger than 1. Fact 4.3 requires that this new polynomial is bounded on the points $\eta_{i,k}$, so we will chose the y_i's accordingly. The affine map is also parameterized by a value $\ell > 0$ which is irrelevant for now, but is used in Section 5 to control how close the y_i's are to 0, and consequently the efficiency of estimating $f(y_i)$.

Formally, define

$$g_\ell(t) \;:=\; \left(\frac{\ell}{2k^2+1}\right)\left(k^2 \cdot t \;-\; (k^2+1)\right), \text{ and} \qquad (4.2)$$

$$y_i \;:=\; g_\ell(\eta_{i,k}). \qquad (4.3)$$

Note that $g_\ell(1+\frac{1}{k^2}) = 0$ and that

$$-\ell \;=\; g_\ell(-1) \;\le\; y_i \;\le\; g_\ell(1) \;=\; -\frac{\ell}{2k^2+1}. \qquad (4.4)$$

Let Δ be an upper bound on the error with which we can approximate f, i.e., $|\Delta_i| = |p_\triangle(y_i)| \le \Delta$. Then the polynomial $\tilde{p}_\triangle(t) := p_\triangle(g_\ell(t))/\Delta$ has the property $|\tilde{p}_\triangle(\eta_{i,k})| \le 1$, that is, $\tilde{p}_\triangle(t)$ belongs to $\mathcal{C}_k$. Furthermore, $p_\triangle(0) = \Delta \cdot \tilde{p}_\triangle(g_\ell^{-1}(0)) = \Delta \cdot \tilde{p}_\triangle(1+\frac{1}{k^2})$. By Fact 4.3, we then have $|p_\triangle(0)| = \Delta \cdot |\tilde{p}_\triangle(1+\frac{1}{k^2})| \le \Delta \cdot |P_k(1+\frac{1}{k^2})|$. To finish bounding $|p_\triangle(0)|$, we use the following lemma.

Lemma 4.4. Let P_k be the k^{th} Chebyshev polynomial, where $k \ge 1$. Then

$$|P_k(1+k^{-c})| \;\le\; \prod_{j=1}^{k}\left(1+\frac{2j}{k^c}\right) \;\le\; e^{2k^{2-c}}.$$

Proof. By induction and Eq. (4.1). ∎

Therefore, $|p_\triangle(0)| \le \Delta \cdot e^2$. Summarizing, our error in estimating $f(0)$ is not significantly worsened by having only approximate knowledge of each $f(y_i)$.

We note that for our analysis, we needed an upper bound on how fast P_k grows once leaving $[-1,1]$. In the past however, lower bounds on the growth of P_k outside of $[-1,1]$ had been used, for example, to obtain a low-degree polynomial approximating the **OR** function on n variables [24].

One can show that our choice of y_i is optimal in the following sense. Consider points $t_0, \ldots, t_k \in [-1,1]$, and consider the class $\mathcal{C}'$ of polynomials p of degree at most k such that $\max_{0\le i\le k} |p(t_i)| \le 1$. The maximum absolute value of a polynomial in $\mathcal{C}'$ for any x is exactly $\sum_{i=0}^{k} |l_i(x)|$, where l_i is the i^{th} Lagrange basis polynomial

$$l_i(x) = \prod_{j\ne i} \frac{x-t_j}{t_i-t_j}.$$

It can be shown that for each $x \in (-\infty,-1) \cup (1,\infty)$, $\sum_{i=0}^{k} |l_i(x)|$ is minimized when $t_i = \eta_{i,k}$. Our choice of y_i corresponds to the optimal choice of t_i, and therefore, minimizes the maximum possible $|p_\triangle(0)|$, given that each Δ_i is bounded in absolute value by the same $\Delta > 0$. We omit details in this version of the paper.

Algorithm 1. Algorithm for additively approximating empirical Shannon entropy. Recall that we define $T(\alpha) = T_{1+\alpha}$.

Choose error parameter $\tilde{\varepsilon}$ and k points $\{y_0, \ldots, y_k\}$
Process the entire stream:
  For each $i \in \{0, \ldots, k\}$, compute $\tilde{F}_{1+y_i}$, a $(1+\tilde{\varepsilon})$-approximation of the frequency moment F_{1+y_i}
For each i, compute $\tilde{T}(y_i) = \left(1 - \tilde{F}_{1+y_i}/||A||_1^{1+y_i}\right)/y_i$
Return an estimate of $T(0)$ by polynomial interpolation using the points $\tilde{T}(y_i)$

5 Estimating Shannon Entropy

5.1 Overview

We begin by describing Algorithm 1, a general algorithm for computing an additive approximation to Shannon entropy. The remainder of this paper describes and analyzes various details and incarnations of this algorithm, including extensions to give a multiplicative approximation in Section 5.3. We assume that m, the length of the stream, is known in advance, though in fact our algorithm works with only a constant factor increase in space as long as a value m' satisfying $m \le m' \le m^{O(1)}$ is known. Computing $\|A\|_1$ is trivial since we assume the strict turnstile model at present.

5.2 Multi-point Interpolation

The algorithm of Section 3 is limited by the following trade-off: if we choose the point y_0 to be close to 0, the accuracy increases, but the space usage also increases. In this section, we mitigate that problem by applying the noisy extrapolation technique of Section 4 and interpolating with multiple points. This allows us to obtain good accuracy without taking the points too close to 0.

The algorithm estimates Tsallis entropy with error parameter $\tilde{\varepsilon} = \varepsilon/(200(k+1)^3 \log m)$, where $k = \log(1/\varepsilon) + \log\log m$. We define the points $y_0, y_1, \ldots, y_k$ by setting $\ell = 1/(2(k+1)\log m)$ and $y_i = g_\ell(\eta_{i,k})$, as in Eq. (4.3).

The correctness of the algorithm is proven in Section 5.2.1. Let us now analyze the space requirements. To compute the estimate $\tilde{F}_{1+y_i}$, the number of words of space required is at most

$$O\Big(\frac{|y_i|}{\tilde{\varepsilon}^2} + \frac{1}{\tilde{\varepsilon}}\Big) \;=\; O\Big(\frac{|y_i|}{\tilde{\varepsilon}^2}\Big) \;=\; O\Big(\tilde{\varepsilon}^{-2}/\log m\Big).$$

The first bound follows from Fact 2.1, the second since Eq. (4.4) shows that $|y_i| \ge \ell/(2k^2+1) > \tilde{\varepsilon}$, and the third since Eq. (4.4) shows that $|y_i| \le \ell = 1/(2(k+1)\log m)$. By our choice of $k = \tilde{O}(1)$ and $\tilde{\varepsilon}$, the total space required is $\tilde{O}(\varepsilon^{-2}\log m)$ words.

5.2.1 Correctness

To prove correctness of our algorithm, it suffices to bound the two error terms defined in Section 4. We now show that both are at most $\varepsilon/2$.

To bound the first error term, we use Fact 4.1. To apply this fact, a bound on $|T^{(k+1)}(y)|$ is needed. It suffices to consider the interval $[-\ell, 0)$, since Eq. (4.4) ensures that $y_i \in [-\ell, 0)$ for all i. Since $\ell = 1/(2(k+1)\log m)$, Lemma 5.1 (proof omitted) implies that

$$|T^{(k+1)}(\xi)| \;\le\; \frac{4\,\log^{k+1}(m)\,H}{k+2} \quad \forall \xi \in [-\ell, 0). \tag{5.1}$$

Lemma 5.1. Let ε be in $(0, 1/2]$. Then, $|T^{(k)}(-\frac{\varepsilon}{(k+1)\log m})| \;\le\; 4\,\log^k(m)\,H/(k+1)$.

Thus, by Fact 4.1 and Eq. (5.1), we have

$$\begin{aligned} |T(0) - p_T(0)| &\le |\ell|^{k+1} \cdot \frac{4\,\log^{k+1}(m)\,H}{(k+1)!\,(k+2)} \\ &\le \frac{1}{2^{k+1}\log^{k+1}(m)} \cdot \frac{4\,\log^{k+1}(m)\,H}{(k+2)!} \\ &\le \frac{2\varepsilon}{(k+2)!} \;\le\; \frac{\varepsilon}{2}, \end{aligned} \tag{5.2}$$

since $2^k = (\log m)/\varepsilon$ and $H \le \log m$.

We now have to bound the second error term $|p_\Delta|$, since Algorithm 1 does not compute the exact values $T(y_i)$, it only computes approximations. The accuracy of these approximations can be determined as follows.

$$\tilde{T}(y_i) \;=\; \frac{1 - \tilde{F}_{1+y_i}/||A||_1^{1+y_i}}{y_i} \;\le\; T(y_i) - \tilde{\varepsilon} \cdot \frac{\sum_{j=1}^n x_j^{1+y_i}}{y_i}. \tag{5.3}$$

To analyze the last term, recall that $x_j \ge 1/m$ for each i and $y_i \ge -\ell$, so that $x_i^{y_i} \le m^\ell = m^{1/2(k+1)\log m} < 2$. Thus $\sum_{j=1}^n x_j^{1+y_i} \le 2\sum_{j=1}^n x_j = 2$. By Eq. (4.4),

$$-\frac{2\tilde{\varepsilon}}{y_i} \;\le\; \frac{2(2k^2+1)\tilde{\varepsilon}}{\ell} \;\le\; \frac{(2k^2+1)\varepsilon}{50(k+1)^2} \;\le\; \frac{\varepsilon}{25}.$$

Thus, we have

$$T(y_i) \;\le\; \tilde{T}(y_i) \;\le\; T(y_i) + \frac{\varepsilon}{25}. \tag{5.4}$$

Hence, the additive error $|p_\Delta(y_i)| = |\Delta_i|$ on each $T(y_i)$ is bounded by $\Delta := \varepsilon/25$. In Section 4, we showed that $|p_\Delta(0)| \le e^2 \cdot \Delta \le \frac{\varepsilon}{2}$. This completes the analysis.

5.3 Multiplicative Approximation of Shannon Entropy

We now discuss how to extend the multi-point interpolation algorithm to obtain a multiplicative approximation of Shannon entropy. The main tool that we require is a multiplicative estimate of Tsallis entropy, rather than the additive estimates used above. Section 7 shows that the required multiplicative estimates can be efficiently computed, using tools provided in Section 6.

The modifications to the multi-point interpolation algorithm are as follows. We set $k = \log(1/\varepsilon)$ and $\tilde{\varepsilon} = \varepsilon/8$. We then use Algorithm 1, but instead of computing an additive estimate of $T(y_i)$ as above, we let $\tilde{T}(y_i)$ be a $(1+\tilde{\varepsilon})$-multiplicative estimate, computed using Theorem 7.3. Then

$$T(y_i) \leq \tilde{T}(y_i) \leq T(y_i) + \tilde{\varepsilon}T(y_i) \leq T(y_i) + 4\tilde{\varepsilon}H,$$

the last inequality by Lemma 5.1 with $k = 0$. Next, as in Eq. (5.2), $|T(0) - p_T(0)| \leq \varepsilon H/2$, since $2^k = 1/\varepsilon$, so we obtain a $(1+\varepsilon)$-multiplicative approximation to H.

6 Estimating Residual Moments

To multiplicatively approximate Shannon entropy, the algorithm of Section 5.3 requires a multiplicative approximation of Tsallis entropy. Section 7 shows that the required quantities can be computed. The main tool needed is an efficient algorithm for estimating *residual moments*. That is the topic of the present section.

Define the residual α^{th} moment to be $F_\alpha^{\text{res}} := \sum_{i=2}^{n} |A_i|^\alpha = F_\alpha - |A_1|^\alpha$, where we reorder the items such that $|A_1| \geq |A_2| \geq \ldots \geq |A_n|$. In this section, we present two efficient algorithms to compute a $1+\varepsilon$ multiplicative approximation to F_α^{res} for $\alpha \in (0,2]$. These algorithms succeed with constant probability under the assumption that a heavy hitter exists, say $|A_1| \geq \frac{4}{5}\|A\|_1$. The algorithm of Section 6.1 is valid only in the strict turnstile model. Its space usage has a complicated dependence on α; for the primary range of interest, $\alpha \in [1/3, 1)$, the bound is $O((\varepsilon^{-1/\alpha} + \varepsilon^{-2}(1-\alpha) + \log n)\log m)$ bits. The algorithm of Section 6.2 is valid in the general update model and uses $\tilde{O}(\varepsilon^{-2}\log m)$ bits of space.

A subroutine that is needed for both our algorithms in this section is to detect whether a heavy element exists ($|A_i| \geq \frac{4}{5}\|A\|_1$) and to find the identity of that element. To accomplish this, we use the following result, which essentially follows from the Count-Min Sketch data structure of Cormode and Muthukrishnan [11].

Fact 6.1 ([11]). There exists a family $\mathcal{H}$ of hash functions mapping the n elements to $O(1/\varepsilon)$ bins with $|\mathcal{H}| = n^{O(1)}$ such that a random $h \in \mathcal{H}$ satisfies the following two properties. Let $w \in \mathbb{R}_+^n$ be a weight vector with $\sum_i w_i = 1$.

(1) For any element s, with probability at least $15/16$, the weight of elements that collide with element s is at most $\varepsilon \cdot \sum_{i \neq s} w_i$.

(2) If $\max_i w_i < 4/5$ then, with probability at least $1/20$, every bin has at most a $7/8$ fraction of the weight.

To find a heavy element, we proceed as follows. First, define $w_i = |A_i| / \|A\|_1$ and $\varepsilon = 1/10$. Select a hash function from $\mathcal{H}$ and use it to partition the elements into bins. For each bin, we maintain a counter of the net L_1-weight that hashes to it. If there is a bin of weight at least $7/8$, then we declare that there is a heavy element (an element of weight at least $4/5$). Otherwise, we declare that all elements have weight at most $7/8$. By Fact 6.1, this test has (one-sided) error at most $19/20$. Repeated independent trials can reduce the failure probability as desired.

If a heavy element exists, we can determine its identity via a group-testing type of argument: we maintain $\lceil \log_2 n \rceil$ counters, of which the i^{th} counts the number of elements which have their i^{th} bit set. If there is heavy element, we can determine its i^{th} bit by checking whether the fraction of elements with their i^{th} bit set is at least $3/5$. The space required is $O(\log n \cdot \log m)$ bits.

6.1 Bucketing Algorithm

In this section, we describe an algorithm for estimating F_α^{res} that works only in the strict turnstile model (i.e., $A \geq 0$). The algorithm has two cases, depending on the value of α. The second case is handled in part by artificially inserting deletions into the stream, an idea used by [12] for residual L_1 estimation.

Case 1: $\alpha = (0, \frac{1}{3}) \cup [1, 2]$. We use the hash function from Fact 6.1 to partition the elements into bins. For each bin, we maintain a count of the number of elements that hash to it and a sketch of the α^{th} moment using Fact 2.1. (These are identical if $\alpha = 1$.) If there is a bin whose counter is more than $7/8$ of the total then there is a heavy element, whose identity can be determined as above. Our estimate is the sum of the moment sketches for all bins except the one containing the heavy element. The approximation guarantee follows from Fact 6.1, property (1), with weights $w_i = A_i^\alpha / \|A\|_\alpha^\alpha$. By Fact 2.1, the space usage is

$$O\left((\tfrac{1}{\varepsilon} + \log n)\log m + \tfrac{1}{\varepsilon} \cdot \left(\tfrac{|\alpha-1|}{\varepsilon^2} + \tfrac{1}{\varepsilon} \right) \log m \right)$$
$$= O\left(\left(\tfrac{|\alpha-1|}{\varepsilon^3} + \tfrac{1}{\varepsilon^2} + \log n \right) \log m \right) \text{ bits.}$$

If $\alpha = 1$, the space is only $O\big((\frac{1}{\varepsilon} + \log n)\log m \big)$ bits.

Case 2: $\alpha = [\frac{1}{3}, 1)$. This idea is to keep just one sketch of the α^{th} moment for the entire stream. At the end, we estimate F_α^{res} by artificially appending deletions to the stream which almost entirely remove the heavy element from the sketch.

The algorithm computes four quantities in parallel. First, $\tilde{F}_1^{\text{res}} = (1 \pm \varepsilon')F_1^{\text{res}}$ with error parameter $\varepsilon' = \varepsilon^{1/\alpha}$, using the above algorithm with $\alpha = 1$. Second, $\tilde{F}_\alpha = (1 \pm \varepsilon)F_\alpha$ using Fact 2.1. Third, F_1, which is trivial in the strict turnstile model. Lastly, we determine the identity of the heavy element as in Fact 6.1.

Now we explain how to estimate F_α^{res}. Without loss of generality, element 1 is heavy. The key observation is that $F_1 - \tilde{F}_1^{\text{res}}$ is a very good approximation to A_1. So if we delete the heavy element $(F_1 - \tilde{F}_1^{\text{res}})$ times, the number of remaining occurrences is non-negative and at most $\varepsilon' F_1^{\text{res}}$. Define $\hat{F}_\alpha^{\text{res}}$ to be the value of $\tilde{F}_\alpha$ after processing these deletions. Clearly $F_\alpha^{\text{res}} \le \hat{F}_\alpha^{\text{res}}$. On the other hand, the remaining occurrences of the heavy element contribute at most $(\varepsilon' F_1^{\text{res}})^\alpha$ to $\hat{F}_\alpha^{\text{res}}$. Thus

$$\hat{F}_\alpha^{\text{res}} \;\le\; F_\alpha^{\text{res}} + (\varepsilon')^\alpha (F_1^{\text{res}})^\alpha \;\le\; F_\alpha^{\text{res}} + (\varepsilon')^\alpha F_\alpha^{\text{res}},$$

the last inequality by concavity of the function $y \mapsto y^\alpha$. This shows that $\hat{F}_\alpha^{\text{res}} = (1 \pm \varepsilon)F_\alpha^{\text{res}}$, since $(\varepsilon')^\alpha = \varepsilon$. The space used by this algorithm is at most

$$\begin{aligned} & O\left(\tfrac{1}{\varepsilon'}\log m + \left(\tfrac{1-\alpha}{\varepsilon^2} + \tfrac{1}{\varepsilon}\right)\log m + \log n \log m\right) \\ = \; & O\left(\left(\tfrac{1}{\varepsilon^{1/\alpha}} + \tfrac{1-\alpha}{\varepsilon^2} + \log n\right)\log m\right) \text{ bits.} \end{aligned}$$

6.2 Geometric Mean Algorithm

This section describes an algorithm for estimating F_α^{res} in the general update model. At a high level, the algorithm uses a hash function to partition the stream elements into two substreams, then separately estimates the moment F_α for the substreams. The estimate for the substream which does not contain the heavy hitter yields a good estimate of F_α^{res}. We improve accuracy of this estimator by averaging many independent trials. Detailed description and analysis follow.

We use Li's *geometric mean estimator* [22] for estimating F_α since it is unbiased. (This property will be useful later.) The geometric mean estimator is defined as follows. Let k and α be parameters. We let $y = R \cdot A$, where A is the vector representing the stream and R is a $k \times n$ matrix whose entries are i.i.d. samples from an α-stable distribution. Define

$$\tilde{F}_\alpha \;=\; \frac{\prod_{j=1}^k |y_j|^{\alpha/k}}{[\frac{2}{\pi}\Gamma(\frac{\alpha}{k})\Gamma(1-\frac{1}{k})\sin(\frac{\pi\alpha}{2k})]^k}.$$

Li analyzed the variance of $\tilde{F}_\alpha$ as $k \to \infty$, however for our purposes we are only interested in the case $k = 3$ and henceforth restrict to only this case. (One can show $\tilde{F}_\alpha$ has unbounded variance for $k < 3$.) In this case, the estimator can be computed using $O(\log m)$ bits of space. Building on Li's analysis, we show the following.

Lemma 6.2. There exists an absolute constant C_{GM} such that $\operatorname{Var}\left[\tilde{F}_\alpha\right] \le C_{GM} \cdot \operatorname{E}\left[\tilde{F}_\alpha\right]^2$.

Proof (sketch). Define the function $V : \mathbb{R}^+ \to \mathbb{R}$ by

$$V(\alpha) = \frac{\left[\frac{2}{\pi}\Gamma(\frac{2\alpha}{3})\Gamma(\frac{1}{3})\sin(\frac{\pi\alpha}{3})\right]^3}{\left[\frac{2}{\pi}\Gamma(\frac{\alpha}{3})\Gamma(\frac{2}{3})\sin(\frac{\pi\alpha}{6})\right]^6} - 1$$

In the full version of the paper we show

$$\lim_{\alpha \to 0} V(\alpha) = \frac{\Gamma\left(\frac{1}{3}\right)^3}{\Gamma\left(\frac{2}{3}\right)^6} - 1$$

Li shows in [22] that the variance of the geometric mean estimator with $k = 3$ is $V(\alpha)F_\alpha^2$. As $\Gamma(z)$ and $\sin(z)$ are continuous for $z \in \mathbb{R}_+$, so is $V(\alpha)$. Furthermore, since $\lim_{\alpha\to 0} V(\alpha)$ exists, we define $V(0)$ to be this limit. Thus $V(\alpha)$ is continuous on $[0, 2]$, and the extreme value theorem implies there exists a constant C_{GM} such that $V(\alpha) \le C_{GM}$ on $[0, 2]$. ■

Let r denote the number of independent trials. For each $j \in [r]$, the algorithm picks a function $h_j : [n] \to \{0, 1\}$ uniformly at random. For $j \in [r]$ and $l \in \{0, 1\}$, define $F_{\alpha,j,l} = \sum_{i:h_j(i)=l} |A_i|^\alpha$. This is the α^{th} moment for the l^{th} substream during the j^{th} trial.

For each j and l, our algorithm computes an estimate $\tilde{F}_{\alpha,j,l}$ of $F_{\alpha,j,l}$ using the geometric mean estimator. We also run in parallel the algorithm of Fact 6.1 to discover which $i \in [n]$ is the heavy hitter; henceforth assume $i = 1$. Our overall estimate for F_α^{res} is then

$$\tilde{F}_\alpha^{\text{res}} \;=\; \frac{2}{r}\sum_{j=1}^r \tilde{F}_{\alpha,j,1-h_j(1)}.$$

The space used by our algorithm is simply the space required for r geometric mean estimators and the one heavy hitter algorithm. The latter uses $O\big((\varepsilon^{-1} + \log n)\log m\big)$ bits of space. Thus the total space required is $O\big((r + \varepsilon^{-1} + \log n)\log m\big)$ bits.

To analyze the algorithm, define $R = \left\lceil \log_{1+\frac{\varepsilon}{c_1}} m \right\rceil$. Define $I_z = \left\{ i : (1 + \frac{\varepsilon}{c_1})^z \le |A_i| < (1 + \frac{\varepsilon}{c_1})^{z+1} \right\}$ for $0 \le z \le R$. Let z^* satisfy $(1 + \frac{\varepsilon}{c_1})^{z^*} \le |A_1| < (1 + \frac{\varepsilon}{c_1})^{z^*+1}$. For $1 \le j \le r$ and $0 \le z \le R$, define $X_{j,z} = \sum_{i \in I_z} \mathbf{1}_{h_j(i) \ne h_j(1)}$. To analyze the j^{th} trial, we need the following simple claim.

Claim 6.3. $\operatorname{E}\left[2 \cdot F_{\alpha,j,1-h_j(1)}\right] = \big(1 + O(\varepsilon)\big) \cdot F_\alpha^{\text{res}}$.

We now show concentration for $X_z := \frac{1}{r}\sum_{1 \le j \le r} X_{j,z}$. By independence of the h_j's, Chernoff bounds show that $X_z = (1 \pm \varepsilon)\operatorname{E}[X_z]$ with probability at least $1 - \exp(-\Theta(\varepsilon^2 r))$. This quantity is at least $1 - \frac{1}{8(R+1)}$ if we choose $r = c_2 \left\lceil \varepsilon^{-2}(\log\log ||A||_1 + \log(c_3/\varepsilon)) \right\rceil$. The *good event* is the event that, for all z, $X_z = (1 \pm \varepsilon)\operatorname{E}[X_z]$; a union bound shows that this occurs with probability at least $7/8$. So suppose that the good event occurs. Then a calculation analogous to Claim 6.3 shows that

$$\sum_j \frac{2}{r} \cdot F_{\alpha,j,1-h_j(1)} \;=\; \big(1 \pm O(\varepsilon)\big) \cdot F_\alpha^{\mathrm{res}}. \qquad (6.1)$$

Recall that $\tilde{F}_\alpha^{\mathrm{res}} = \sum_{j=1}^r \frac{2}{r} \tilde{F}_{\alpha,j,1-h_j(1)}$. Since the geometric mean estimator is unbiased, we also have that

$$\mathrm{E}\left[\tilde{F}_\alpha^{\mathrm{res}}\right] \;=\; \mathrm{E}\left[\sum_j \frac{2}{r} F_{\alpha,j,1-h_j(1)}\right]. \qquad (6.2)$$

We conclude the analysis by showing that the random variable $\tilde{F}_\alpha^{\mathrm{res}}$ is concentrated. By Lemma 6.2 applied to each substream, and properties of variance, we have

$$\begin{aligned} \mathrm{Var}\left[\tilde{F}_\alpha^{\mathrm{res}}\right] &= \frac{4}{r^2} \sum_{j=1}^r \mathrm{Var}\left[\tilde{F}_{\alpha,j,1-h_j(1)}\right] \\ &\le \frac{4\,C_{GM}}{r} \cdot \mathrm{E}\left[\tilde{F}_{\alpha,j,1-h_j(1)}\right]^2 \\ &\le \frac{C_{GM}}{r} \cdot \mathrm{E}\left[\tilde{F}_\alpha^{\mathrm{res}}\right]^2. \end{aligned}$$

Chebyshev's inequality therefore shows that

$$\Pr\left[\ \tilde{F}_\alpha^{\mathrm{res}} = (1 \pm \varepsilon)\,\mathrm{E}\left[\tilde{F}_\alpha^{\mathrm{res}}\right]\ \right] \;\ge\; 1 - \frac{C_{GM}}{\varepsilon^2\, r} \;>\; 6/7,$$

by appropriate choice of constants. This event and the good event both occur with probability at least $3/4$. When this holds, we have

$$\begin{aligned} \tilde{F}_\alpha^{\mathrm{res}} &= (1 \pm \varepsilon)\,\mathrm{E}\left[\tilde{F}_\alpha^{\mathrm{res}}\right] \\ &= (1 \pm \varepsilon)\,\mathrm{E}\left[\sum_j \frac{2}{r} F_{\alpha,j,1-h_j(1)}\right] \\ &= \big(1 \pm O(\varepsilon)\big) \cdot F_\alpha^{\mathrm{res}}, \end{aligned}$$

by Eq. (6.2) and Eq. (6.1).

7 Estimation of Rényi and Tsallis Entropy

This section summarizes our algorithms for estimating Tsallis entropy. These algorithms are used as subroutines for estimating Shannon entropy in Section 5. Our techniques can also be used to estimate Rényi entropy. The algorithms presented here may be of independent interest.

The techniques we use for both the entropies are almost identical. In particular, to compute an additive approximation of T_α or H_α, for $\alpha \in (0,1) \cup (1,2]$, it suffices to compute a sufficiently precise multiplicative approximation of the α-th moment. Due to space constraints, we only sketch the proofs for Tsallis entropy estimation. The complete proofs for both Rényi and Tsallis entropy appear in the full version.

7.1 Additive Approximation

Theorem 7.1. There is a streaming algorithm that computes an additive ε-approximation of Rényi entropy in $O\left(\frac{\log m}{|1-\alpha|\cdot\varepsilon^2}\right)$ bits of space for any $\alpha \in (0,1) \cup (1,2]$.

Theorem 7.2. There is a streaming algorithm for additive approximation of Tsallis entropy T_α using

- $O\left(\frac{n^{2(1-\alpha)}\log m}{(1-\alpha)\varepsilon^2}\right)$ bits, for $\alpha \in (0,1)$.
- $O\left(\frac{\log m}{(\alpha-1)\varepsilon^2}\right)$ bits, for $\alpha \in (1,2]$.

Proof. If $\alpha \in (0,1)$, then because the function x^α is concave, we get by Jensen's inequality

$$\sum_{i=1}^n {x_i}^\alpha \le n \cdot \left(\frac{1}{n}\right)^\alpha = n^{1-\alpha}.$$

If we compute a multiplicative $(1 + (1-\alpha)\cdot\varepsilon\cdot n^{\alpha-1})$-approximation to the α^{th} moment, we obtain an additive $(1-\alpha)\cdot\varepsilon$-approximation to $(\sum_{i=1}^n x_i^\alpha) - 1$. This in turn gives an additive ε-approximation to T_α. By Fact 2.1,

$$O\left(\left(\frac{1-\alpha}{((1-\alpha)\cdot\varepsilon\cdot n^{\alpha-1})^2} + \frac{1}{(1-\alpha)\cdot\varepsilon\cdot n^{\alpha-1}}\right)\log m\right)$$

bits of space suffice to achieve the required approximation to the α^{th} moment. This bound simplifies to $O(n^{2(1-\alpha)}\log m/((1-\alpha)\varepsilon^2))$ bits.

For $\alpha > 1$, the value $F_\alpha/||A||_1^\alpha$ is at most 1, so it suffices to approximate F_α to within a factor of $1 + (\alpha-1)\cdot\varepsilon$. For $\alpha \in (1,2]$, again using Fact 2.1, we can achieve this using $O(\log m/((\alpha-1)\varepsilon^2))$ bits of space. ■

7.2 Multiplicative Approximation

Multiplicative approximation of Tsallis entropy is more difficult than additive approximation, because a single moment approximation is not sufficient. Nevertheless, we prove the following theorem.

Theorem 7.3. There is a streaming algorithm for multiplicative $(1+\varepsilon)$-approximation of Tsallis entropy for any $\alpha \in (0,1) \cup (1,2]$ using $\tilde{O}\left(\log m/(|1-\alpha|\varepsilon^2)\right)$ bits of space.

Proof (sketch). We show that if all $x_i \le 5/6$, then $T_\alpha(x) \ge C$, for some absolute constant C. Therefore, if all $x_i \le 5/6$, then $T_\alpha(x)$ is sufficiently large, and additive approximation with error $\varepsilon \cdot C$ gives a good multiplicative approximation of $T_\alpha(x)$.

Assume now that there is $x_j \ge 2/3$. Let $A = x_j^\alpha - 1$ and $B = \sum_{i \ne j} x_i^\alpha$. We show that the sum of sufficiently good multiplicative approximations to A and B gives a multiplicative approximation to $A + B = \sum x_i^\alpha - 1$, which is a value that immediately yields a multiplicative approximation to Tsallis entropy. This follows from the fact that if there is a heavy element, then $A \leq 0$ and $B \geq 0$ are values of different magnitudes, and small errors on them are still relatively small with respect to $|A+B|$. We prove that to multiplicatively approximate A in this case, it suffices to have a multiplicative approximation to $F_1^{\text{res}} = 1 - x_j$. Further, $B = F_\alpha^{\text{res}}$. Hence, we can use the algorithms of Section 6.1 to approximate A and B. We also use these algorithms to check if there is a heavy element, and to decide which of the two cases, $\max_i x_i \leq 5/6$ and $\max_i x_i \geq 2/3$ holds. ■

Tsallis entropy can be efficiently approximated both multiplicatively and additively also for $\alpha > 2$, but we omit a proof of that fact in this version of the paper.

Using similar techniques, one can also obtain an algorithm for Rényi entropy.

Theorem 7.4. There is a streaming algorithm for multiplicative $(1+\varepsilon)$-approximation of Rényi entropy for any $\alpha \in (0,1) \cup (1,2]$. The algorithm uses $\tilde{O}\left(\log m/(|1-\alpha|\varepsilon^2)\right)$ bits of space.

Theorem 7.4 is in fact tight in the sense that $(1+\varepsilon)$-multiplicative approximation of H_α for $\alpha > 2$ requires polynomial space, as seen in the following theorem. Since the proof follows very closely along the lines of Theorem 3.1 of [2] but using the stronger multiparty disjointness lower bound of [9], we omit the proof here.

Theorem 7.5. For any constant $\alpha > 2$, any randomized constant-pass streaming algorithm which $(1+\varepsilon)$-approximates $H_\alpha(X)$ requires $\Omega(n^{1-2/\alpha-2\varepsilon}/\log n)$ bits of space. If the algorithm is allowed only one pass, the lower bound increases to $\Omega(n^{1-2/\alpha-2\varepsilon})$ bits.

8 Modifications for General Update Streams

The algorithms described in Section 5 and Section 7 are for the strict turnstile model. They can be extended to work in the general update model with a few modifications.

First, we cannot efficiently and exactly compute $\|A\|_1 = F_1$ in the general update model. However, a $(1+\varepsilon)$-multiplicative approximation can be computed in $O(\varepsilon^{-2}\log m)$ bits of space by Fact 2.1. In Section 3 and Section 5.2, the value of $\|A\|_1$ is used as a normalization factor to scale the estimate of F_α to an estimate of $\sum_{i=1}^n x_i^\alpha$. (See, e.g., Eq. (3.1) and Eq. (5.3).) However,

$$\frac{\tilde{F}_\alpha}{(\tilde{F}_1)^\alpha} = \frac{(1\pm\varepsilon)\cdot F_\alpha}{\left((1\pm\varepsilon)\cdot F_1\right)^\alpha} = \left(1\pm O(\varepsilon)\right)\cdot\frac{F_\alpha}{F_1^\alpha},$$

so the fact that F_1 can only be approximated in the general update model affects the analysis only by increasing the constant factor that multiplies ε. A similar modification must also be applied to all algorithms in Section 7; we omit the details.

Next, the multiplicative algorithm in Section 5.3 needs to compute a multiplicative estimate of $T(y_i)$ using Theorem 7.3. In the general update model, a weaker result than Theorem 7.3 holds: we obtain a multiplicative $(1+\varepsilon)$-approximation of Tsallis entropy for any $\alpha \in (0,1)\cup(1,2]$ using $\tilde{O}\left(\log m/(|1-\alpha|\cdot\varepsilon)^2\right)$ bits of space. The proof is nearly identical to that of Theorem 7.3, except that the moment estimator of Fact 2.1 uses more space, and we must use the residual moment algorithm of Section 6.2 instead of Section 6.1. Similar modifications must be made to Theorem 7.1, Theorem 7.2 and Theorem 7.4, with a commensurate increase in the space bounds.

9 Conclusion

We hope that the techniques from approximation theory that we introduce may be useful for streaming and sketching other functions. For example, Cormode et al. [10] observe that $F_{\varepsilon/\log m} = (1 \pm O(\varepsilon))L_0$, where L_0, or the "Hamming norm", is the number of non-zero frequencies in the vector A. They thus reduce L_0 estimation to estimating a single moment with p near 0. Using our techniques from Section 4 together with bounds on the higher order derivatives of F_p as a function of p, one can easily show that a $(1\pm\varepsilon)$-approximation of L_0 can be obtained using $\tilde{O}(1)$ moment estimations F_{p_i} for $p_i = \tilde{\Theta}(1/\log m)$ for each i, which can be useful when moment estimations near $p=0$ are expensive (though better methods are already known for L_0 estimation).

Also, consider the following function $G_{\alpha,k}(x) = \sum_i x_i^\alpha(\log n)^k$, where $k \in \mathbb{N}$ and $\alpha \in [0,\infty)$. One can show that

$$\lim_{\beta\to\alpha}\frac{G_{\alpha,k}(x) - G_{\beta,k}(x)}{\alpha-\beta} = G_{\beta,k+1}(x).$$

Note that $G_{\alpha,0}(x)$ is the α^{th} moment of x, and one can attempt to estimate $G_{\alpha,k+1}$ by computing $G_{\beta,k}$ for $\beta=\alpha$ and β close to α. It is not unlikely that our techniques can be generalized to estimation of functions $G_{\alpha,k}$ for $\alpha \in (0,2]$. Can one also use our techniques for approximation of other classes of functions?

Acknowledgements

We thank Piotr Indyk and Ping Li for many helpful discussions. We thank Henryk Woźniakowski for explaining how the additional extrapolation error due to noise can be bounded by Lagrange polynomials. We thank Swastik Kopparty for pointing out the work of Nisan and Szegedy [24]. We also thank Jonathan Kelner for some pointers to the approximation theory literature.

References

[1] N. Alon, Y. Matias, and M. Szegedy. The Space Complexity of Approximating the Frequency Moments. *Journal of Computer and System Sciences*, 58(1):137–147, 1999.

[2] Z. Bar-Yossef, T. S. Jayram, R. Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *JCSS*, 68(4), 2004.

[3] T. Batu, S. Dasgupta, R. Kumar, and R. Rubinfeld. The complexity of approximating the entropy. *SIAM J. Comput.*, 35(1):132–150, 2005.

[4] L. Bhuvanagiri and S. Ganguly. Estimating entropy over data streams. In *Proceedings of ESA*, 2006.

[5] L. Bhuvanagiri and S. Ganguly. Hierarchical Sampling from Sketches: Estimating Functions over Data Streams, 2008. Manuscript.

[6] L. Bhuvanagiri, S. Ganguly, D. Kesh, and C. Saha. Simpler algorithm for estimating frequency moments of data streams. In *Proceedings of SODA*, 2006.

[7] A. Chakrabarti, G. Cormode, and A. McGregor. A near-optimal algorithm for computing the entropy of a stream. In *Proceedings of SODA*, 2007.

[8] A. Chakrabarti, K. Do Ba, and S. Muthukrishnan. Estimating Entropy and Entropy Norm on Data Streams. In *Proceedings of STACS*, 2006.

[9] A. Chakrabarti, S. Khot, and X. Sun. Near-optimal lower bounds on the multi-party communication complexity of set disjointness. In *Proceedings of Complexity*, 2003.

[10] G. Cormode, M. Datar, P. Indyk, and S. Muthukrishnan. Comparing data streams using hamming norms (how to zero in). *IEEE Trans. Knowl. Data Eng.*, 15(3):529–540, 2003.

[11] G. Cormode and S. Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *J. Algorithms*, 55(1):58–75, 2005.

[12] S. Ganguly and G. Cormode. On estimating frequency moments of data streams. In *APPROX-RANDOM*, pages 479–493, 2007.

[13] S. Ganguly, D. Kesh, and C. Saha. Practical algorithms for tracking database join sizes. In *FSTTCS*, 2005.

[14] Y. Gu, A. McCallum, and D. F. Towsley. Detecting anomalies in network traffic using maximum entropy estimation. In *Internet Measurment Conference*, pages 345–350, 2005.

[15] S. Guha, A. McGregor, and S. Venkatasubramanian. Streaming and sublinear approximation of entropy and information distances. In *Proceedings of SODA*, 2006.

[16] N. J. A. Harvey, J. Nelson, and K. Onak. Streaming algorithms for estimating entropy. In *Proceedings of IEEE Information Theory Workshop*, 2008.

[17] P. Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *J. ACM*, 53(3):307–323, 2006.

[18] P. Indyk and A. McGregor. Declaring independence via the sketching of sketches. In *Proceedings of SODA*, 2008.

[19] P. Indyk and D. P. Woodruff. Optimal approximations of the frequency moments of data streams. In *STOC*, 2005.

[20] A. Lakhina, M. Crovella, and C. Diot. Mining anomalies using traffic feature distributions. In *SIGCOMM*, 2005.

[21] P. Li. Compressed counting. CoRR abs/0802.2305v2, 2008.

[22] P. Li. Estimators and tail bounds for dimension reduction in l_p $(0 < p \leq 2)$ using stable random projections. In *Proceedings of SODA*, 2008.

[23] N. Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 4:301–313, 1994.

[24] N. Nisan and M. Szegedy. On the degree of boolean functions as real polynomials. *Computational Complexity*, 12(4):449–461, 1992.

[25] L. Paninski. Estimation of entropy and mutual information. *Neural Computation*, 15(6):1191–1253, 2003.

[26] L. Paninski. Estimating entropy on m bins given fewer than m samples. *IEEE Transactions on Information Theory*, 50(9):2200–2203, 2004.

[27] G. M. Phillips. *Interpolation and Approximation by Polynomials*. Springer-Verlag, New York, 2003.

[28] G. M. Phillips and P. J. Taylor. *Theory and Applications of Numerical Analysis*. Academic Press, 2nd edition, 1996.

[29] S. Raskhodnikova, D. Ron, A. Shpilka, and A. Smith. Strong lower bounds for approximating distribution support size and the distinct elements problem. In *FOCS*, 2007.

[30] A. Rényi. On measures of entropy and information. In *Proc. Fourth Berkeley Symp. Math. Stat. and Probability*, volume 1, pages 547–561, 1961.

[31] T. J. Rivlin. *An Introduction to the Approximation of Functions*. Dover Publications, New York, 1981.

[32] T. J. Rivlin. *Chebyshev Polynomials: From Approximation Theory to Algebra and Number Theory*. Wiley, 1990.

[33] W. W. Rogosinski. Some elementary inequalities for polynomials. *The Mathematical Gazette*, 39(327):7–12, 1955.

[34] M. E. Saks and X. Sun. Space lower bounds for distance approximation in the data stream model. In *STOC*, 2002.

[35] C. Tsallis. Possible generalization of boltzmann-gibbs statistics. *Journal of Statistical Physics*, 52:479–487, 1988.

[36] P. Valiant. Testing symmetric properties of distributions. In *Proceedings of STOC*, 2008.

[37] D. Woodruff. *Efficient and Private Distance Approximation in the Communication and Streaming Models*. PhD thesis, Massachusetts Institute of Technology, 2007.

[38] K. Xu, Z.-L. Zhang, and S. Bhattacharyya. Profiling internet backbone traffic: behavior models and applications. In *Proceedings of SIGCOMM*, 2005.

[39] H. Zhao, A. Lall, M. Ogihara, O. Spatscheck, J. Wang, and J. Xu. A Data Streaming Algorithm for Estimating Entropies of OD Flows. In *Proceedings of IMC*, 2007.