Sketching and Streaming High-Dimensional Vectors

by

Jelani Nelson

S.B., Massachusetts Institute of Technology (2005)
M.Eng., Massachusetts Institute of Technology (2006)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2011

Author . . . .
Department of Electrical Engineering and Computer Science
May 20, 2011

Certified by . . . .
Erik D. Demaine
Professor
Thesis Supervisor

Certified by . . . .
Piotr Indyk
Professor
Thesis Supervisor

Accepted by . . . .
Leslie A. Kolodziejski
Chairman, Department Committee on Graduate Students

Sketching and Streaming High-Dimensional Vectors

by

Jelani Nelson

Submitted to the Department of Electrical Engineering and Computer Science
on May 20, 2011, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computer Science and Engineering

Abstract

A *sketch* of a dataset is a small-space data structure supporting some prespecified set of queries (and possibly updates) while consuming space substantially sublinear in the space required to actually store all the data. Furthermore, it is often desirable, or required by the application, that the sketch itself be computable by a small-space algorithm given just one pass over the data, a so-called *streaming algorithm*. Sketching and streaming have found numerous applications in network traffic monitoring, data mining, trend detection, sensor networks, and databases.

In this thesis, I describe several new contributions in the area of sketching and streaming algorithms.

- The first space-optimal streaming algorithm for the distinct elements problem. Our algorithm also achieves $O(1)$ update and reporting times.

- A streaming algorithm for Hamming norm estimation in the turnstile model which achieves the best known space complexity.

- The first space-optimal algorithm for pth moment estimation in turnstile streams for $0 < p < 2$, with matching lower bounds, and another space-optimal algorithm which also has a fast $O(\log^2(1/\varepsilon)\log\log(1/\varepsilon))$ update time for $(1 \pm \varepsilon)$-approximation.

- A general reduction from empirical entropy estimation in turnstile streams to moment estimation, providing the only known near-optimal space-complexity upper bound for this problem.

- A proof of the Johnson-Lindenstrauss lemma where every matrix in the support of the embedding distribution is much sparser than previous known constructions. In particular, to achieve distortion $(1 \pm \varepsilon)$ with probability $1 - \delta$, we embed into optimal dimension $O(\varepsilon^{-2}\log(1/\delta))$ and such that every matrix in the support of the distribution has $O(\varepsilon^{-1}\log(1/\delta))$ non-zero entries per column.

Thesis Supervisor: Erik D. Demaine
Title: Professor

Thesis Supervisor: Piotr Indyk
Title: Professor

Acknowledgments

Including my years as an undergraduate I have been at MIT for a full decade, and I have come across many friends, colleagues, research advisors, and teachers who have made the experience very enjoyable. For this, I am very grateful.

I thank my advisors Prof. Erik Demaine and Prof. Piotr Indyk for their guidance in my PhD studies. I was primarily interested in external memory data structures when I entered the program, an area within Erik's research interests, and Erik was very supportive while I explored questions in this realm. After taking a course on sketching and streaming algorithms, my research interests migrated toward that area at which point Piotr began giving me much guidance and became an unofficial (and eventually official) co-advisor, and then even officemate if one considers the 6th floor lounge to be an office. I have received a great amount of influence and guidance from Erik and Piotr, and I am grateful to have been their student.

I would like to thank my coauthors: Tim Abbott, Miki Ajtai, Michael Bender, Michael Burr, Timothy Chan, Erik Demaine, Marty Demaine, Ilias Diakonikolas, Martin Farach-Colton, Vitaly Feldman, Jeremy Fineman, Yonatan Fogel, Nick Harvey, Avinatan Hassidim, John Hugg, Daniel Kane, Bradley Kuszmaul, Stefan Langerman, Raghu Meka, Krzysztof Onak, Ely Porat, Eynat Rafalin, Kathryn Seyboth, David Woodruff, and Vincent Yeung. I have been fortunate to have enjoyed every collaboration I have been a part of with these colleagues, and I have learned much from them. I in particular collaborated quite frequently with Daniel and David, both of whom I knew even before beginning my PhD studies, and it has been a pleasure to work with them so often. I have learned a great deal by doing so.

While on the topic of learning from others, aside from advisors and coauthors I have been fortunate to have been surrounded by several colleagues and friends who I have learned a great deal from concerning theoretical computer science and other relevant areas of mathematics, at MIT, at the IBM Almaden Research Center and Microsoft Research New England during summer internships, at the Technion in the summer prior to beginning my PhD, and elsewhere.

Last but not least, I would like to thank God and my parents for creating me and enabling me to have an enjoyable life thus far.

In memory of my high school mathematics teacher José Enrique Carrillo.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In 1975, Morris discovered the following surprising result: one can count up to m using exponentially fewer than $\log m$ bits. For example, he showed how to approximately count up to numbers as large as $130,000$ using 8-bit registers, with 95% confidence that his returned estimate was within 24% of the true count. In general terms, the method allows one to approximately count the number of items in a sequence of length m using only $O(\log \log m)$ bits, given only one pass over the sequence and such that the estimated sequence length has small relative error with large probability. The method was simple: maintain a value ν in memory which one imagines represents $\log(1 + m)$, then after seeing a new item in the sequence, increment ν with probability exponentially small in ν. Morris' algorithm was the first non-trivial low-space *streaming algorithm*: an algorithm that computes a function of its input given only one or a few passes over it. His method was published three years later in [93].

Two more surprising streaming algorithms were discovered soon afterward. In 1978, Munro and Paterson showed how to compute the *median* of an array of m numbers by a two-pass deterministic streaming algorithm storing only $O(\sqrt{m} \log m)$ numbers [95]. Then in 1983, Flajolet and Martin showed that given just one pass over an array with elements in $\{1, \ldots, d\}$, one can approximate the number of *distinct elements* in the array with low relative error and with high probability, using only $O(\log d)$ bits of space [49].

Beyond those discoveries, little attention was given to streaming algorithms until 1996, when Alon, Matias, and Szegedy tackled the problem of approximating the frequency moments of a vector being updated in a stream [8]. By the late 1990s and early 2000s, the popularity of the Internet made the need for space-efficient streaming algorithms clear. Internet services such as search engines needed algorithms which could, on-the-fly, make sense of massive amounts of data that were constantly being collected. Furthermore, algorithms using memory proportional to the total amount of data collected were infeasible. Major ISPs were also processing massive amounts of network traffic at their routers and needed succinct ways to summarize traffic for later processing.

In this thesis, we investigate several fundamental streaming problems and provide solutions which are more space-efficient, and in some cases also more time-efficient, than those previously known.

Problem Name	Desired Output	Our Contribution
distinct elements	$F_0 = \|\{i : x_i \neq 0\}\|$	optimal time and space
moment estimation	$F_p = \sum_i \|x_i\|^p$	optimal space, fast time
empirical entropy estimation	$H = -\sum_i q_i \lg(q_i)$ ($q_i = \|x_i\|/(\sum_i \|x_i\|)$)	near-optimal space in terms of ε
ℓ_2-dimensionality reduction	$\\|y\\|_2 = (1 \pm \varepsilon)\\|x\\|_2$ (with probability $1 - \delta$)	optimal target dimension, sparse embedding matrices

Table 1.1: Catalog of problems studied in this thesis, and our contributions.

To be a bit more formal, a *sketch* [19] is a small-space data structure which allows some pre-specified set of queries and (possibly) updates. Often we seek sketches which are much smaller, even exponentially so, than the information theoretic minimum required to store all the data explicitly. A *streaming algorithm* is an algorithm which can maintain a sketch of the data given only one or few passes over it, though we will only be concerned with one-pass streaming algorithms. Furthermore, almost all the algorithms we will describe (except for F_0) produce *linear sketches* of data that can be represented as a high-dimensional vector. That is, we consider the case where the data is represented by some vector $x \in \mathbb{R}^d$, where d is very large so that storing x explicitly in memory is either undesirable or infeasible. Initially x starts as $\vec{0}$, and each update adds some (possibly negative) amount to a specified coordinate in x. The sketch of x is obtained by some, possibly randomized, linear map $x \mapsto Sx$. This model of vector updates in streaming algorithms is known as the *turnstile model* [96], and it arises naturally in several contexts. For example, in a stream of queries to a search engine, x_i can be the number of search queries containing the ith word in some lexicon. Or in a stream of network traffic, x_i may be the total number of bytes sent to IP address i.

Aside from the above two scenarios, sketching and streaming algorithms in general have numerous applications. A sketch naturally reduces the storage required to store a dataset [19]. Furthermore, since a sketch requires little space it can be communicated across machines efficiently. In the case of linear sketches, sketches at different machines can then be added to obtain a sketch for the aggregate data, or subtracted to obtain a sketch for the difference vector if one then wants to query some distance measure. Even sketches stored at the same machine, but perhaps representing different datasets or data collected over different time intervals, can be efficiently compared to compute various similarity or distance measures.

In this thesis we study several problems, described in Table 1.1. For all these problems, it was known from previous work [8] that one must settle for an algorithm which is both *randomized* and *approximate*. That is, we are only guaranteed to compute a $(1 \pm \varepsilon)$-approximation to our desired function, and our estimate is only guaranteed to be this good with probability $1 - \delta$.

In Section 1.1 we give the preliminary definitions and notation that are used consistently throughout the thesis. In the following sections we discuss prior work and our contributions for each of the problems in Table 1.1.

1.1 Notation and Preliminaries

Throughout this thesis, $x \in \mathbb{R}^d$ represents a vector being updated in a data stream, and $y = Sx$ represents the linear sketch of x we maintain. Unless explicitly stated otherwise, such as when discussing the distinct elements problem, we always assume x is updated in the turnstile model. That is, there is a sequence of m updates $(i, v) \in [d] \times \{-M, \ldots, M\}$ each causing the change $x_i \leftarrow x_i + v$. Sometimes we discuss algorithms that operate in the *strict turnstile model*, which is the turnstile model with the additional promise that $\forall i\ x_i \geq 0$ at the end of the stream. The *insertion-only model* is that for which all updates (i, v) have $v = 1$.

In all our algorithms, we will want to compute some quantity with probability $1 - \delta$ for some $0 < \delta < 1/2$ known up front to the algorithm. Unless explicitly stated otherwise, when discussing all our algorithms we assume a constant error probability $\delta = 1/3$. For all the problems we consider other than ℓ_2-dimensionality reduction, arbitrarily small δ can be achieved by running $O(\log(1/\delta))$ instances of the algorithm in parallel with independent randomness then computing the median estimate. All our algorithms also either only produce an approximate answer (when computing a function), or incur some distortion (in our ℓ_2-dimensionality algorithm). The distortion we incur is always written as $1 \pm \varepsilon$ for some $0 < \varepsilon < 1/2$; i.e. we produce an embedding y of x such that $\|y\|_2^2 \in [(1-\varepsilon)\|x\|_2^2, (1+\varepsilon)\|x\|_2^2]$. In problems where our aim is to produce an estimate of some function $f(x)$, we settle for computing a value $\tilde{f}$ such that $\tilde{f} \in [(1-\varepsilon)f(x), (1+\varepsilon)f(x)]$ for some $0 < \varepsilon < 1/2$ known up front to the algorithm.

When analyzing the efficiency of our algorithms, we measure space in bits. When measuring running time, we measure the number of standard machine word operations in the RAM model (arithmetic, bitwise operations, and bitshifts on machine words each take constant time). We assume a word size of $\Omega(\log dmM)$ so that x can be indexed in constant time, and so that each $|x_i|$ fits in a word. In stating running times we distinguish between *update time*, the time it takes to process an update (i, v) in the stream, and *reporting time*, the time it takes to produce an estimate $\tilde{f}$ of $f(x)$ when queried.

We use $\mathbf{1}_{\mathcal{E}}$ to denote the indicator random variable for event $\mathcal{E}$. All logarithms are base-2 unless otherwise stated, and we often use the notation $A \approx_\varepsilon B$ to mean that $|A - B| = O(\varepsilon)$. For a functions $f : \mathbb{R} \to \mathbb{R}$, we use $f^{(\ell)}$ to denote the ℓth derivative of f ($f^{(0)}$ denotes f itself). We use $[n]$ to denote $\{1, \ldots, n\}$. When we say $f = \tilde{O}(g)$, we mean $f = O(g \cdot \mathrm{polylog}(g))$. Also, $f = \tilde{\Omega(g)}$ means $f = \Omega(g/\mathrm{polylog}(g))$. Lastly, we use $f = \tilde{\Theta}(g)$ to mean $f = \tilde{O}(g)$ and $f = \tilde{\Omega}(g)$.

We frequently throughout this thesis hide dependence on d in upper and lower bounds. In all discussions of upper and lower bounds in this thesis, other than in Chapter 5, it may be assumed that $d = O(m^2)$ at the cost of an additive $O(\log\log d)$ in the space upper bound. Furthermore, $\Omega(\log\log d)$ is also a lower bound for several problems discussed in this thesis; see Section 3.5 for details.

Reference	Space	Update Time	Notes
[49]	$O(\varepsilon^{-2}\log d)$	—	Random oracle
[8]	$O(\log d)$	$O(\log d)$	constant ε
[58]	$O(\varepsilon^{-2}\log d)$	$O(\varepsilon^{-2})$	
[13]	$O(\varepsilon^{-3}\log d)$	$O(\varepsilon^{-3})$	
[12]	$O(\varepsilon^{-2}\log d)$	$O(\log(\varepsilon^{-1}))$	
[12]	$O(\varepsilon^{-2}\log\log d$ $+\mathrm{poly}(\log(\varepsilon^{-1}\log d))\log d)$	$\varepsilon^{-2}\mathrm{poly}(\log(\varepsilon^{-1}\log d))$	
[12]	$O(\varepsilon^{-2}(\log(\varepsilon^{-1}\log d)+\log d))$	$O(\varepsilon^{-2}(\log(\varepsilon^{-1}\log d)))$	
[41]	$O(\varepsilon^{-2}\log\log d+\log d)$	—	Random oracle, additive error
[44]	$O(\varepsilon^{-2}\log d)$	—	Random oracle
[15]	$O(\varepsilon^{-2}\log d)$	$O(\log(\varepsilon^{-1}))$	
[48]	$O(\varepsilon^{-2}\log\log d+\log d)$	—	Random oracle, additive error
This thesis	$O(\varepsilon^{-2}+\log d)$	$O(1)$	Optimal

Table 1.2: Comparison of our algorithm to previous algorithms on estimating the number of distinct elements in a data stream. Update time not listed for algorithms assuming random oracle.

1.2 Distinct Elements

In this problem, we are given a sequence of integers $i_1,\ldots,i_m\in[d]$ and would like to estimate the number of distinct integers. In turnstile model notation, every update is of the form $(i,1)$ for $i\in[d]$, and we would like to estimate $F_0\stackrel{\mathrm{def}}{=}|\{i : x_i\neq 0\}|$. Estimating the number of distinct elements in a data stream is a fundamental problem in network traffic monitoring, query optimization, data mining, and several other database areas.

In network traffic monitoring, routers with limited memory track statistics such as distinct destination IPs, requested URLs, and source-destination pairs on a link [44]. Distinct elements estimation is also useful in detecting Denial of Service attacks [6]. In such applications the data is too large to fit at once in main memory or too massive to be stored, being a continuous flow of data packets. This makes small-space algorithms necessary. Furthermore, the algorithm should process each stream update (i.e., packet) quickly to keep up with network speeds. For example, [23] reported packet header information being produced at .5GB per hour while estimating the spread of the Code Red worm, for which they needed to estimate the number of distinct Code Red sources passing through a link.

Another application is to data mining: for example, estimating the number of distinct queries made to a search engine, or distinct users clicking on a link or visiting a website. Distinct item estimation was also used in estimating connectivity properties of the Internet graph [104].

Other applications include selecting a minimum-cost query plan [114], database design [47], OLAP [102, 115], data integration [22, 37], and data warehousing [1].

The problem of space-efficient F_0 estimation is well-studied, beginning with the work of Flajolet and Martin [49], and continuing with a long line of research, [8, 12, 13, 15, 21, 31, 41, 44, 48, 57, 58, 69, 124].

Our Contribution: We settle both the space- and time-complexities of F_0 estimation by giving an algorithm using $O(\varepsilon^{-2} + \log d)$ space, with $O(1)$ worst-case update and reporting times. Our space upper bound matches known lower bounds [8, 69, 124] up to a constant factor, and the $O(1)$ update and reporting times are clearly optimal. This contribution was obtained in joint work with Kane and Woodruff [79].

A detailed comparison of our results to those in previous work is given in Table 1.2. There is a wide spectrum of time/space tradeoffs but the key points are that none of the previous algorithms achieved our optimal $O(\varepsilon^{-2} + \log d)$ space, and the only ones to achieve optimal $O(1)$ update and/or reporting time had various restrictions, e.g., the assumption of access to a random oracle (that is, a truly random hash function) and/or a small constant additive error in the estimate. The best previous algorithms without any assumptions are due to Bar-Yossef *et al.* [12], who provide algorithms with various tradeoffs (see Table 1.2).

We also give a new algorithm for estimating ℓ_0, also known as the *Hamming norm* of a vector [32], with optimal running times and near-optimal space. We sometimes refer to the Hamming norm of x as $\|x\|_0$. This problem is simply a generalization of F_0 estimation to turnstile streams; in particular, we define $\ell_0 \stackrel{\text{def}}{=} |\{i : x_i \neq 0\}|$. While F_0 estimation is useful for a single stream or for taking unions of streams if there are no deletions, ℓ_0 estimation can be applied to a pair of streams to measure the number of unequal item counts. This makes it more flexible than F_0, and can be used in applications such as maintaining ad-hoc communication networks amongst cheap sensors [66]. It also has applications to data cleaning to find columns that are mostly similar [37]. Even if the rows in the two columns are in different orders, streaming algorithms for ℓ_0 can quickly identify similar columns. As with F_0, ℓ_0 estimation is also useful for packet tracing and database auditing [32]. It was also used for estimating distance to periodicity [43].

We give an ℓ_0 estimation algorithm with $O(1)$ update and reporting times, using a total of $O(\varepsilon^{-2}(\log d)(\log(1/\varepsilon) + \log\log mM))$ bits of space, both of which improve upon the previously best known algorithm of Ganguly [54], which had $O(\log(1/\varepsilon))$ update time and required $O(\varepsilon^{-2} \log d \log mM)$ space. Our update and reporting times are optimal, and the space is optimal up to the $\log(1/\varepsilon) + \log\log mM$ term due to known lower bounds [8, 78]. Furthermore, the algorithm of [54] only works in the strict turnstile model.

1.3 Moment Estimation

In the moment estimation problem, we would like to compute a $(1 \pm \varepsilon)$-approximation to the value $F_p \stackrel{\text{def}}{=} \|x\|_p^p = \sum_{i=1}^d |x_i|^p$. For constant p bounded away from 0, up to changing ε by a constant factor this problem is equivalent to $(1 \pm \varepsilon)$-approximation of the ℓ_p norm of x, i.e. $\|x\|_p$.

Reference	Space	Update Time	Model	Which p
[8]	$O(\varepsilon^{-2} \log mM)$	$O(\varepsilon^{-2})$	unrestricted updates	$p = 2$
[29, 118]	$O(\varepsilon^{-2} \log mM)$	$O(1)$	unrestricted updates	$p = 2$
[45]	$O(\varepsilon^{-2} \log mM)$	$O(\varepsilon^{-2})$	≤ 2 updates per coordinate	$p = 1$
[67, 87]	$O(\varepsilon^{-2} \log d \log mM)$	$O(\varepsilon^{-2})$	unrestricted updates	$p \in (0, 2)$
[78]	$O(\varepsilon^{-2} \log mM)$	$O(\varepsilon^{-2})$	unrestricted updates	$p \in (0, 2)$
[55]	$\varepsilon^{-2-p} \cdot \mathrm{polylog}(mM)$	$\mathrm{polylog}(mM)$	unrestricted updates	$p \in (0, 2)$
[97]	$O(\varepsilon^{-2} \log mM \log(1/\varepsilon))$	$O(\log^2 mM)$	≤ 2 updates per coordinate	$p = 1$
This thesis	$O(\varepsilon^{-2} \log mM)$	$\tilde{O}(\log^2(1/\varepsilon))$	unrestricted updates	$p \in (0, 2)$

Table 1.3: Comparison of our contribution to previous works on F_p estimation in data streams.

It is known that not all ℓ_p norms can be efficiently approximated in a data stream. In particular, [11, 27] show that polynomial space in d, m is required for $p > 2$, whereas space polylogarithmic in these parameters is achievable for $0 < p \leq 2$ [8, 67].[1] In this thesis, we focus on this feasible regime for p. We remark that streaming approximations to ℓ_p in this range area interesting for several reasons. ℓ_1 estimation is used as a subroutine for dynamic earthmover distance approximation [66], approximate linear regression and best rank-k approximation of a matrix (with respect to the ℓ_1 norm) [46], cascaded norm estimation of a matrix [71], and network traffic monitoring [45]. Estimation of the ℓ_2-norm is useful for database query optimization [7] and network traffic anomaly detection [83]. Both ℓ_1 and ℓ_2 estimation subroutines are used in approximate histogram maintenance [59]. Norm estimation for fractional p was shown useful for mining tabular data in [33] ($p = 0.5$ and $p = 0.25$ were specifically suggested), and we use ℓ_p estimation for fractional p near 1 as a subroutine for estimating empirical entropy in Chapter 4, which in turn is again useful for network traffic anomaly detection (see [63] and the references therein). Also, ℓ_p estimation for all $0 < p \leq 2$ is used as a subroutine for weighted sampling in turnstile streams [92].

Previously known upper bounds for moment estimation are listed in Table 1.3. All space bounds listed hide an additive $O(\log \log d)$ term. The works listed with the restriction of "≤ 2 updates per coordinate" operate in the turnstile model given the guarantee that each coordinate is updated at most twice: at most once positively, and at most once negatively. In terms of lower bounds, [8] showed a lower bound of $\Omega(\log \min\{d, m\})$, and [124] showed an $\Omega(1/\varepsilon^2)$ lower bound.

Our Contributions: In Chapter 3 we resolve the space complexity of ℓ_p estimation for $0 < p \leq 2$ up to constant factors. In particular, the space complexity is $\Theta(\varepsilon^{-2} \log(mM) + \log \log d)$ bits, as long as this bound is at most $\min\{d, m\}$. For p strictly less than 2, our upper bound is new, and our lower bound is new for all $0 < p \leq 2$. We also show that this optimal space bound is achievable by an algorithm with update time only $O(\log^2(1/\varepsilon) \log \log(1/\varepsilon))$. These contributions were obtained in joint works with Kane and Woodruff [78], Woodruff [98], and Kane, Porat, and

[1] When $0 < p < 1$, ℓ_p is not a norm since it does not satisfy the triangle inequality, though it is still a well-defined function.

Woodruff [77].

Our space-optimal upper bound is given in Section 3.2, with the lower bound given in Section 3.4. The space-optimal upper bound which also has fast update time is given in Section 3.3. Subsequent to our work, our lower bound was further improved to take success probability into account: Jayram and Woodruff [72] showed that success probability $1 - \delta$ requires space $\Omega(\varepsilon^{-2} \log(mM) \log(1/\delta))$, which is optimal since one can output the median estimate of $O(\log(1/\delta))$ parallel instantiations of our algorithm.

1.4 Entropy Estimation

In this problem we would like to estimate $H \stackrel{\text{def}}{=} -\sum_{i=1}^{d} q_i \ln q_i$, where $q_i \stackrel{\text{def}}{=} x_i/\|x\|_1$. That is, we consider the empirical probability distribution defined by x and would like to estimate its entropy. The primary motivation for empirical entropy estimation is for applications in computer networking, such as network anomaly detection. Let us consider a concrete example. One form of malicious activity on the internet is *distributed denial of service attacks* (DDoS attacks), in which a large number of machines overwhelm a target server by flooding it with a large number of packets, rendering it unable to attend to legitimate traffic. Thus when a DDoS attack is underway, if one takes x to be indexed by destination IP, then x becomes much more concentrated than usual. This causes the entropy of the empirical distribution defined by x to drop. See Lakhina et al. [85] and Xu et al. [126] for more discussion of empirical entropy estimation in data streams in the context of network anomaly detection.

The first algorithms for approximating entropy in the streaming model are due to Guha et al. [60]; they achieved $O((\varepsilon^{-2}+\log d) \log^2 d)$ space in the insertion-only model, assuming that the stream is randomly ordered. Chakrabarti, Do Ba and Muthukrishnan [26] then gave an algorithm for worst-case ordered insertion-only streams using $O(\varepsilon^{-2} \log^2 m \log d)$ space, but required two passes over the input. The algorithm of Chakrabarti, Cormode and McGregor [25] uses $O(\varepsilon^{-2} \log^2 m)$ space to give a multiplicative $1 \pm \varepsilon$ approximation in insertion-only streams. In contrast, the algorithm of Bhuvanagiri and Ganguly [16] operates in the most general turnstile model but requires roughly $\tilde{O}(\varepsilon^{-3} \log^3 mM)$ space.

Our Contribution: We give a general reduction from entropy estimation to moment estimation which works in the turnstile model. Plugging in our optimal moment estimation algorithms then yields an entropy estimation algorithm using $\widetilde{O}(\varepsilon^{-2}) \cdot \mathrm{polylog}(mM)$ space. This is the first turnstile entropy estimation algorithm which gives nearly-optimal dependence on ε, as there is an $\tilde{\Omega}(\varepsilon^{-2})$ lower bound even when negative updates are not allowed [25]. This contribution was obtained in joint work with Harvey and Onak [63].

1.5 Johnson-Lindenstrauss Transforms

The randomized Johnson-Lindenstrauss lemma states:

Lemma 1 (JL Lemma [73]). *For any integer $d > 0$, and any $0 < \varepsilon, \delta < 1/2$, there exists a probability distribution $\mathcal{D}$ on $\mathbb{R}^{k\times d}$ for $k = \Theta(\varepsilon^{-2}\log(1/\delta))$ such that for any $x \in \mathbb{R}^d$ with $\|x\|_2 = 1$, $\mathbf{Pr}_{S\sim\mathcal{D}}[|\|Sx\|_2^2 - 1| > \varepsilon] < \delta$.*

Definition 2. *We call any distribution $\mathcal{D}$ satisfying the JL lemma a* JL-distribution*. If $\mathcal{S}$ is a family of matrices such that the uniform distribution over $\mathcal{S}$ is a JL-distribution, then we call $\mathcal{S}$ a* JL-family*.*

Proofs of the JL lemma can be found in [2, 9, 18, 36, 50, 68, 73, 75, 89]. The value of k in the JL lemma is known to be optimal [72] (also see a later proof in [75].

The JL lemma is a key ingredient in the JL flattening theorem, which states that any n points in Euclidean space can be embedded into $O(\varepsilon^{-2}\log n)$ dimensions so that all pairwise Euclidean distances are preserved up to $1 \pm \varepsilon$. The JL lemma is a fundamental tool for speeding up solutions to many high-dimensional problems: closest pair, nearest neighbor, diameter, minimum spanning tree, etc. It also speeds up some clustering and string processing algorithms, and can further be used to reduce the amount of storage required to store a dataset, e.g. in streaming algorithms. Recently it has also found applications in approximate numerical algebra problems such as linear regression and low-rank approximation [30, 113]. See [65, 119] for discussions of these and other applications.

1.5.1 Sparse Johnson-Lindenstrauss Transforms

Standard proofs of the JL lemma take a distribution over dense matrices (e.g. i.i.d. Gaussian or Bernoulli entries), and thus performing the embedding naïvely takes $O(k \cdot \|x\|_0)$ time where x has $\|x\|_0$ non-zero entries. Several works have devised other distributions which give faster embedding times [3, 4, 5, 64, 82, 121]. The two best of these which are incomparable are [4], which requires $O(d \log k + k^{2+\gamma})$ time to embed into optimal dimension $k = O(\varepsilon^{-2}\log(1/\delta))$ for any $\gamma > 0$, and [5, 82], which require $O(d \log d)$ time to embed into suboptimal dimension $k = O(\varepsilon^{-2}\log(1/\delta)\log^4 d)$. All these methods however require $\Omega(\min\{d, k \cdot \|x\|_0\})$ time to embed any vector, even if $\|x\|_0 = 1$. This feature is particularly unfortunate in streaming applications, where a vector x receives coordinate-wise updates of the form $x \leftarrow x + v \cdot e_i$ in a data stream, so that to maintain some linear embedding Sx of x we should repeatedly calculate Se_i during updates. Since $\|e_i\|_0 = 1$, these approaches do not improve upon the $O(k \cdot \|e_i\|_0)$ embedding time coming from dense matrix-vector multiplication.

Even aside from streaming applications, several practical situations give rise to vectors with $\|x\|_0 \ll d$. For example, a common similarity measure for comparing text documents in data mining and information retrieval is cosine similarity [105], which is approximately preserved under any JL embedding. Here, a document is represented as a bag of words with the dimensionality d being the size of the lexicon, and we usually would not expect any single document to contain anywhere near d distinct words (i.e., we expect sparse vectors). In networking applications, if $x_{i,j}$ counts bytes sent from source i to destination j in some time interval, then d is the total number of IP pairs, whereas most pairs of IPs do not communicate with each other.

One way to speed up embedding time in the JL lemma for sparse vectors is to devise a distribution over sparse embedding matrices. This was first investigated in [2], which gave a JL distribution where only one third of the entries of each matrix in its support was non-zero, without increasing the number of rows k from dense constructions even by a constant factor. Later, the works [29, 118] gave a distribution over matrices with only $O(\log(1/\delta))$ non-zero entries per column, but the algorithm for estimating $\|x\|_2$ given the linear sketch then relied on a median calculation, and thus these schemes did not provide an embedding into ℓ_2. In several applications, such as nearest-neighbor search [68] and some approximate numerical linear algebra problems [30, 113], an embedding into a normed space or even ℓ_2 itself is required, and thus median estimators cannot be used. Recently Dasgupta, Kumar, and Sarlós [35], building upon work in [122], gave a JL distribution over matrices where each column has at most $s = \tilde{O}(\varepsilon^{-1}\log^3(1/\delta))$ non-zero entries, thus speeding up the embedding time to $O(s \cdot \|x\|_0)$. This "DKS construction" requires $O(ds\log k)$ bits of random seed to sample a matrix from their distribution. The work of [35] left open two main directions: (1) understand the sparsity parameter s that can be achieved in a JL distribution, and (2) devise a sparse JL transform distribution which requires few random bits to sample from, for streaming applications where storing a long random seed requires prohibitively large memory.

The work [75] made progress on both these questions by showing $\tilde{O}(\varepsilon^{-1}\log^2(1/\delta))$ sparsity was achievable by giving an alternative analysis of the scheme of [35] which also only required $O(\log(1/(\varepsilon\delta))\log d)$ seed length. The work of [18] later gave a tighter analysis under the assumption $\varepsilon < 1/\log^2(1/\delta)$, improving the sparsity and seed length further by $\log(1/\varepsilon)$ and $\log\log(1/\delta)$ factors in this case. In this thesis (in Section 5.1.4) we show that the DKS scheme *requires* $s = \tilde{\Omega}(\varepsilon^{-1}\log^2(1/\delta))$, and thus a departure from their construction is required to obtain better sparsity.

Our Contribution: In Section 5.1 we give two new constructions achieving sparsity $s = \Theta(\varepsilon^{-1}\log(1/\delta))$ for ℓ_2 embedding into optimal dimension $k = O(\varepsilon^{-2}\log(1/\delta))$. This is the first sparsity bound which is always asymptotically smaller than k, regardless of how ε and δ are related. One of our distributions requires $O(\log(1/\delta)\log d)$ uniform random bits to sample from. This contribution was obtained in joint work with Kane [76].

1.5.2 Derandomizing the Johnson-Lindenstrauss Transform

Another question related to the Johnson-Lindenstrauss transform is that of *derandomizing* it; that is, devising a JL-distribution $\mathcal{D}$ over linear mappings such that one can efficiently sample a random mapping from $\mathcal{D}$ using few uniformly random bits. If the distribution is the uniform distribution over some set of linear mappings, i.e. a JL family, then it is clear that one needs $\Omega(\log(1/\delta) + \log d/k)$ random bits. That is, the set size must be at least $\max\{1/\delta, d/k\}$. The JL family size must be at least

$1/\delta$ to achieve a non-zero error probability which is at most δ, and it must be at least d/k since otherwise the intersection of the kernels of all the linear mappings in the set would be non-empty.

Existentially it is known that there exists a JL-distribution which can be sampled using $O(\log(d/\delta))$ bits. This is because derandomizing the JL lemma is connected to pseudorandom generators (PRGs) against degree-2 polynomial threshold functions (PTFs) over the hypercube [38, 90]. A degree-t PTF is a function $f : \{-1,1\}^d \to \{-1,1\}$ which can be represented as the sign of a degree-t d-variate polynomial. A PRG that δ-fools degree-t PTFs is a function $F : \{-1,1\}^s \to \{-1,1\}^d$ such that for any degree-t PTF f,

$$|\mathbf{E}_{z\in\mathcal{U}^s}[f(F(z))] - \mathbf{E}_{x\in\mathcal{U}^d}[f(x)]| < \delta,$$

where $\mathcal{U}^m$ is the uniform distribution on $\{-1,1\}^m$.

Note that the conclusion of the JL lemma can be rewritten as

$$\mathbf{E}_S[I_{[1-\varepsilon,1+\varepsilon]}(\|Sx\|_2^2)] \geq 1-\delta,$$

where $I_{[a,b]}$ is the indicator function of the interval $[a,b]$, and furthermore S can be taken to have random $\pm 1/\sqrt{k}$ entries [2]. Noting that $I_{[a,b]}(z) = (\mathrm{sign}(z-a) - \mathrm{sign}(z-b))/2$ and using linearity of expectation, we see that any PRG which δ-fools $\mathrm{sign}(p(x))$ for degree-t polynomials p must also δ-fool $I_{[a,b]}(p(x))$. Now, for fixed x, $\|Sx\|_2^2$ is a degree-2 polynomial over the boolean hypercube in the variables $S_{i,j}$ and thus a PRG which δ-fools degree-2 PTFs also gives a JL family with the same seed length[2]. It was shown via the probabilistic method that there exist PRGs for degree-2 PTFs with seed length $O(\log(1/\delta) + \log d)$ (see Section B of the full version of [90] for a proof).

Our Contribution: In Section 5.4, we give a JL-distribution which can be sampled from using a uniform random seed of length $s = O(\log d + \log(1/\varepsilon)\log(1/\delta) + \log(1/\delta)\log\log(1/\delta))$. Furthermore, given the seed and a vector $x \in \mathbb{R}^d$, one can compute the embedding in $d^{O(1)}$ time. This contribution was obtained in joint work with Kane and Meka [74].

Regarding previous works, the ℓ_2-streaming algorithm of Alon, Matias, and Szegedy [8] implies a JL family with seed length $O(\log d)$ and with $k = O(1/(\varepsilon^2\delta))$. Clarkson and Woodruff [30] showed that a scaled random Bernoulli matrix with $\Omega(\log(1/\delta))$-wise independent entries gives a JL distribution. Their work thus implies seed length $O(\log(1/\delta)\log d)$ with optimal target dimension k. Karnin, Rabani, and Shpilka [81] gave a family with seed length $(1+o(1))\log d + O(\log^2(1/(\varepsilon\delta)))$ with optimal k.

Other derandomizations of the JL lemma include the works [42] and [117]. A common application of the JL lemma is the case where there are n vectors $y^1,\ldots,y^n \in \mathbb{R}^d$ and one wants to find a matrix $S \in \mathbb{R}^{k\times d}$ to preserve $\|y^i - y^j\|_2$ to within relative error ε simultaneously for all i,j. In this case, one can set $\delta = 1/n^2$ and apply

[2]By "seed", we mean the string of uniform random bits that is fed into the PRG.

Lemma 1, then perform a union bound over all i, j pairs. The works of [42, 117] do not give JL families, but rather give deterministic algorithms for finding such a matrix S in the case that the vectors $y^1, \ldots, y^n$ are known up front.

Chapter 2

Distinct Elements

In this chapter, we describe our algorithms for F_0 and ℓ_0 estimation in data streams. We would like to $(1 \pm \varepsilon)$-approximate

$$F_0, \ell_0 \stackrel{\text{def}}{=} |\{i : x_i \neq 0\}|.$$

The difference between ℓ_0 and F_0 estimation is in the update model. In the case of F_0 estimation, every update in the data stream adds 1 to some coordinate in x, whereas ℓ_0 estimation implies the turnstile model, which can have negative updates.

Our algorithms build upon several techniques given in previous works, with added twists to achieve our desired bounds. In for example [12], it was observed that if one somehow knows ahead of time a value $R = \Theta(F_0)$, refining to $(1 \pm \varepsilon)$-approximation becomes easier. For example, [12] suggested a "balls and bins" approach to estimating F_0 given such an R. The key intuition is that when hashing A balls randomly into K bins, the number of bins hit by at least one ball is highly concentrated about its expectation, and treating this expectation as a function of A then inverting provides a good approximation to A with high probability for $A = \Theta(K)$. Then if one subsamples each index in $[d]$ with probability $2^{-\log(R/K)}$ and only processes the substream consisting of subsampled indices, in expectation the number of distinct items surviving is $\Theta(K)$, at which point the balls-and-bins approach can be simulated by hashing indices (the "balls") into entries of a bitvector (the "bins").

Following the above scheme, an estimate of F_0 can be obtained by running a constant-factor approximation in parallel to obtain such an R at the end of the stream, and meanwhile performing the above scheme for geometrically increasing guesses of R, one of which must be correct to within a constant factor. Thus, the bits tracked can be viewed as a bitmatrix: rows corresponding to $\log d$ levels of subsampling, and columns corresponding to entries in the bitvector. At the end of the stream, upon knowing R, the estimate from the appropriate level of subsampling is used. Such a scheme with $K = \Theta(1/\varepsilon^2)$ works, and gives $O(\varepsilon^{-2} \log d)$ space since there are $\log d$ levels of subsampling.

It was then first observed in [41] that in fact an estimator can be obtained without maintaining the full bitmatrix above. Specifically, for each column [41] gives an estimator that for each column only required keeping track of the deepest row with its bit set to 1. This allowed the algorithm to collapse the bitmatrix above to $O(\varepsilon^{-2} \log\log d)$ bits. Though, their analysis of their estimator required access to a purely random hash function.

Our F_0 algorithm is inspired by the above two algorithms of [12, 41]. We give a subroutine RoughEstimator using $O(\log d)$ space which, with high probability, provides a constant-factor approximation to F_0 at all times in the stream simultaneously. Previous subroutines gave a constant factor approximation to F_0 at any *particular* point in the stream with probability $1 - \delta$ using $O(\log d \log(1/\delta))$ space; a good approximation at all times then required setting $\delta = 1/m$ to apply a union bound, thus requiring $O(\log d \log m)$ space. The next observation is that if $R = \Theta(F_0)$, the largest row index with a 1 bit for any given column is heavily concentrated around the value $\log(F_0/K)$. Thus, if we *bitpack* the K counters and store their offsets from $\log(R/K)$, we expect to only use $O(K)$ space for all counters combined. Whenever R changes, we update all our offsets.

There are of course obvious obstacles in obtaining worst-case $O(1)$ running times, such as the occasional need to decrement all K counters (when R increases), or to locate the starting position of a counter in a bitpacked array when reading and writing entries. For the former, we use an approach inspired by the technique of deamortization of global rebuilding (see [101, Ch. 5]), and for the latter we use a "variable-bit-length array" data structure [17]. Furthermore, we analyze our algorithm without assuming a truly random hash function, and show that a combination of fast r-wise independent hash functions [116] and uniform hashing [103] suffice to have sufficient concentration in all probabilistic events we consider.

Our ℓ_0 estimation algorithm also uses subsampling and a balls-and-bins approach, but needs a different subroutine for obtaining the value R, and for representing the bitmatrix. Specifically, if one maintains each bit as a counter and tests for the counter being non-zero, frequencies of opposite sign may cancel to produce 0 and give false negatives. We instead store the dot product of frequencies in each counter with a random vector over a suitably large finite field. We remark that Ganguly's algorithm [54] is also based on a balls-and-bins approach, but on viewing the number of bins hit by *exactly* one ball (and not at least one ball), and the source of his algorithm's higher complexity stems from implementing this different viewpoint.

Notation: In this section, we use $\mathcal{H}_r(S, T)$ to denote any r-wise independent hash family mapping S onto T. We use $\mathrm{lsb}(x)$ to denote the least significant bit of x when written in binary (0-indexed). For a function f of the stream, e.g. F_0, we let $f(t)$ denote f after only processing the first t updates in the data stream. Thus for example, $F_0(t)$ is the number of distinct items amongst the first t in the stream.

Bibliographic remark: The algorithms described in this chapter were developed jointly with Kane and Woodruff [79].

2.1 Balls and Bins with Limited Independence

In the analysis of the correctness of our algorithms, we require some understanding of the balls and bins random process with limited independence. We note that [12] also required a similar analysis, but was only concerned with approximately preserving the expectation under bounded independence whereas we are also concerned with approximately preserving the variance. Specifically, consider throwing a set of A balls into K bins at random and wishing to understand the random variable X being the number of bins receiving at least one ball. This balls-and-bins random process can be modeled by choosing a random hash function $h \in \mathcal{H}_A([A],[K])$, i.e., h acts fully independently on the A balls, and letting $X = |i \in [K] : h^{-1}(i) \neq \emptyset|$. When analyzing our F_0 algorithm, we require an understanding of how X behaves when $h \in \mathcal{H}_k([A],[K])$ for $k \ll A$.

Henceforth, we let X_i denote the indicator random variable for the event that at least one ball lands in bin i under a truly random hash function h, so that $X = \sum_{i=1}^{K} X_i$.

The following fact is standard. For completeness' sake, we provide a proof in Section 2.4.1.

Fact 3. $\mathbf{Var}[X] = K(K-1)\left(1-\frac{2}{K}\right)^A + K\left(1-\frac{1}{K}\right)^A - K^2\left(1-\frac{1}{K}\right)^{2A}$, *and* $\mathbf{E}[X] = K\left(1-\left(1-\frac{1}{K}\right)^A\right)$

The proof of the following lemma can also be found in Section 2.4.1.

Lemma 4. *If* $1 \le A \le K/20$*, then* $\mathbf{Var}[X] < 4A^2/K$*.*

We now state a lemma that r-wise independence for small r suffices to preserve $\mathbf{E}[X]$ to within $1 \pm \varepsilon$, and to preserve $\mathbf{Var}[X]$ to within an additive ε^2. We note that item (1) in the following lemma was already shown in [12, Lemma 1] but with a stated requirement of $r = \Omega(\log(1/\varepsilon))$, though their proof actually seems to only require $r = \Omega(\log(1/\varepsilon)/\log\log(1/\varepsilon))$. Our proof of item (1) also only requires this r, but we require dependence on K in our proof of item (2).

Lemma 5. *There exists some constant ε_0 such that the following holds for $\varepsilon \le \varepsilon_0$. Let A balls be mapped into K bins using a random $h \in \mathcal{H}_{2(r+1)}([A],[K])$, where $r = c\log(K/\varepsilon)/\log\log(K/\varepsilon)$ for a sufficiently large constant $c > 0$. Suppose $1 \le A \le K$. For $i \in [K]$, let X'_i be an indicator variable which is 1 if and only if there exists at least one ball mapped to bin i by h. Let $X' = \sum_{i=1}^{K} X'_i$. Then the following hold:*

(1). $|\mathbf{E}[X'] - \mathbf{E}[X]| \le \varepsilon\mathbf{E}[X]$

(2). $\mathbf{Var}[X'] - \mathbf{Var}[X] \le \varepsilon^2$

Proof. Let A_i be the random variable number counting the number of balls in bin i when picking $h \in \mathcal{H}_{2(r+1)}([A],[K])$. Define the function:

$$f_r(n) = \sum_{i=0}^{r} (-1)^i \binom{n}{i}$$

We note that $f_r(0) = 1$, $f_r(n) = 0$ for $1 \le n \le r$ and $|f_r(n)| \le \binom{n}{r+1}$ otherwise. Let $f(n) = 1$ if $n = 0$ and 0 otherwise. We now approximate X_i as $1 - f_r(A_i)$. We note that this value is determined entirely by r-wise independence of h. We note that this is also

$$1 - f(A_i) \pm O\left(\binom{A_i}{r+1}\right) = X_i \pm O\left(\binom{A_i}{r+1}\right).$$

The same expression holds for the X'_i, and thus both $\mathbf{E}[X'_i]$ and $\mathbf{E}[X_i]$ are sandwiched inside an interval of size bounded by twice the expected error. To bound the expected error we can use $(r+1)$-independence. We have that the expected value of $\binom{A_i}{r+1}$ is $\binom{A}{r+1}$ ways of choosing $r+1$ of the balls times the product of the probabilities that each ball is in bin i. This is

$$\binom{A}{r+1} K^{-(r+1)} \le \left(\frac{eA}{K(r+1)}\right)^{r+1} \le \frac{A}{K} \cdot \left(\frac{e}{r+1}\right)^{r+1},$$

with the last inequality using that $A \le K$. Thus, $|\mathbf{E}[X_i] - \mathbf{E}[X'_i]| \le \varepsilon^2 A/K$ for $r = c\log(1/\varepsilon)/\log\log(1/\varepsilon)$ for sufficiently large constant c. In this case $|\mathbf{E}[X] - \mathbf{E}[X']| \le \varepsilon^2 A \le \varepsilon \mathbf{E}[X]$ for ε smaller than some constant since $\mathbf{E}[X] = \Omega(A)$ for $A \le K$.

We now analyze $\mathbf{Var}[X']$. We approximate $X_i X_j$ as $(1 - f_r(A_i))(1 - f_r(A_j))$. This is determined by $2r$-independence of h and is equal to

$$\begin{aligned}\left(1 - f(A_i) \pm O\left(\binom{A_i}{r+1}\right)\right)&\left(1 - f(A_j) \pm O\left(\binom{A_j}{r+1}\right)\right)\\ = X_i X_j \pm O\left(\binom{A_i}{r+1} + \binom{A_j}{r+1} + \binom{A_i}{r+1}\binom{A_j}{r+1}\right)&\end{aligned}$$

We can now analyze the error using $2(r+1)$-wise independence. The expectation of each term in the error is calculated as before, except for products of the form

$$\binom{A_i}{r+1}\binom{A_j}{r+1}.$$

The expected value of this is

$$\binom{A}{r+1, r+1} K^{-2(r+1)} \le \binom{A}{r+1}^2 K^{-2(r+1)} \le \left(\frac{eA}{K(r+1)}\right)^{2(r+1)}.$$

Thus, for $r = c' \log(K/\varepsilon)/\log\log(K/\varepsilon)$ for sufficiently large $c' > 0$, each summand in the error above is bounded by $\varepsilon^3/(6K^2)$, in which case $|\mathbf{E}[X_i X_j] - \mathbf{E}[X_i X_j]| \le \varepsilon^3/K^2$.

We can also make c' sufficiently large so that $|\mathbf{E}[X]-\mathbf{E}[X']| \le \varepsilon^3/K^2$. Now, we have

$$\begin{aligned}
\mathbf{Var}[X'] - \mathbf{Var}[X] &\le |(\mathbf{E}[X] - \mathbf{E}[X']) \\
&\qquad + 2\sum_{i<j}(\mathbf{E}[X_iX_j] - \mathbf{E}[X_i'X_j']) - (\mathbf{E}^2[X] - \mathbf{E}^2[X'])| \\
&\le |\mathbf{E}[X] - \mathbf{E}[X']| + K(K-1) \\
&\qquad \times \cdot \max_{i<j} |\mathbf{E}[X_iX_j] - \mathbf{E}[X_i'X_j']| + |\mathbf{E}^2[X] - \mathbf{E}^2[X']| \\
&\le \varepsilon^3/K^2 + \varepsilon^3 + \mathbf{E}^2[X](2\varepsilon^3/K^2 + (\varepsilon^3/K^2)^2) \\
&\le 5\varepsilon^3
\end{aligned}$$

which is at most ε^2 for ε sufficiently small. ■

We now give a consequence of the above lemma.

Lemma 6. *There exists a constant ε_0 such that the following holds. Let X' be as in Lemma 5, and also assume $1 \le A \le K/20$ with $K = 1/\varepsilon^2$ and $\varepsilon \le \varepsilon_0$. Then*

$$\mathbf{Pr}[|X' - \mathbf{E}[X]| \le 8\varepsilon\mathbf{E}[X]] \ge 4/5$$

Proof. Observe that

$$\begin{aligned}
\mathbf{E}[X] &\ge (1/\varepsilon^2)\left(1 - \left(1 - A\varepsilon^2 + \binom{A}{2}\varepsilon^4\right)\right) \\
&= (1/\varepsilon^2)\left(A\varepsilon^2 - \binom{A}{2}\varepsilon^4\right) \\
&\ge (39/40)A,
\end{aligned}$$

since $A \le 1/(20\varepsilon^2)$.

By Lemma 5 we have $\mathbf{E}[X'] \ge (1-\varepsilon)\mathbf{E}[X] > (9/10)A$, and additionally using Lemma 4 we have that $\mathbf{Var}[X'] \le \mathbf{Var}[X] + \varepsilon^2 \le 5\varepsilon^2A^2$. Set $\varepsilon' = 7\varepsilon$. Applying Chebyshev's inequality,

$$\begin{aligned}
\mathbf{Pr}[|X' - \mathbf{E}[X']| \ge (10/11)\varepsilon'\mathbf{E}[X']] &\le \mathbf{Var}[X']/((10/11)^2(\varepsilon')^2\mathbf{E}^2[X']) \\
&\le 5 \cdot A^2\varepsilon^2/((10/11)^2(\varepsilon')^2(9/10)^2A^2) \\
&< (13/2)\varepsilon^2/(10\varepsilon'/11)^2 \\
&< 1/5
\end{aligned}$$

Thus, with probability at least 1/5, by the triangle inequality and Lemma 5 we have $|X' - \mathbf{E}[X]| \le |X' - \mathbf{E}[X']| + |\mathbf{E}[X'] - \mathbf{E}[X]| \le 8\varepsilon\mathbf{E}[X]$. ■

2.2 F_0 estimation algorithm

In this section we describe our F_0 estimation algorithm. Our algorithm requires, in part, a constant-factor approximation to F_0 at every point in the stream in which F_0 is

1. Set $K_{\mathrm{RE}} = \max\{8, \log(d)/\log\log(d)\}$.
2. Initialize $3K_{\mathrm{RE}}$ counters $C_1^j, \ldots, C_{K_{\mathrm{RE}}}^j$ to -1 for $j \in [3]$.
3. Pick random $h_1^j \in \mathcal{H}_2([d], [0, d-1])$, $h_2^j \in \mathcal{H}_2([d], [{K_{\mathrm{RE}}}^3])$,
   $h_3^j \in \mathcal{H}_{2K_{\mathrm{RE}}}([{K_{\mathrm{RE}}}^3], [K_{\mathrm{RE}}])$ for $j \in [3]$.
4. **Update(i):** For each $j \in [3]$, set $C^j_{h_3^j(h_2^j(i))} \leftarrow \max\left\{C^j_{h_3^j(h_2^j(i))}, \mathrm{lsb}(h_1^j(i))\right\}$.
5. **Estimator:** For integer $\ell \geq 0$, define $T_\ell^j = |\{i : C_i^j \geq \ell\}|$.
   For the largest $\ell = \ell^*$ with $T_\ell^j \geq \rho K_{\mathrm{RE}}$, set $\widetilde{F}_0^j = 2^{\ell^*} K_{\mathrm{RE}}$.
   If no such ℓ exists, $\widetilde{F}_0^j = -1$.
   Output $\widetilde{F}_0 = \mathrm{median}\{\widetilde{F}_0^1, \widetilde{F}_0^2, \widetilde{F}_0^3\}$.

Figure 2-1: RoughEstimator pseudocode. With probability $1 - o(1)$, $\widetilde{F}_0(t) = \Theta(F_0(t))$ at every point t in the stream for which $F_0(t) \geq K_{\mathrm{RE}}$. The value ρ is $.99 \cdot (1 - e^{-1/3})$.

sufficiently large. We describe a subroutine RoughEstimator in Section 2.2.1 which provides this, using $O(\log d)$ space, then we give our full algorithm in Section 2.2.2.

We remark that the algorithm we give in Section 2.2.2 is space-optimal, but is not described in a way that achieves $O(1)$ worst-case update and reporting times. In Section 2.2.4, we describe modifications to achieve optimal running times while preserving space-optimality.

We note that several previous algorithms could give a constant-factor approximation to F_0 with success probability $2/3$ using $O(\log d)$ space. To understand why our guarantees from RoughEstimator are different, one should pay particular attention to the quantifiers. In previous algorithms, it was guaranteed that there exists a constant $c > 0$ such that at any *particular* point t in the stream, with probability at least $1 - \delta$ the output $\widetilde{F}_0(t)$ is in $[F_0(t), cF_0(t)]$, with the space used being $O(\log d \log(1/\delta))$. To then guarantee $\widetilde{F}_0(t) \in [F_0(t), cF_0(t)]$ *for all* $t \in [m]$ with probability $2/3$, one should set $\delta = 1/(3m)$ to then union bound over all t, giving an overall space bound of $O(\log d \log m)$. Meanwhile, in our subroutine RoughEstimator, we ensure that with probability $2/3$, $\widetilde{F}_0(t) \in [F_0(t), cF_0(t)]$ for all $t \in [m]$ simultaneously, and the overall space used is $O(\log d)$.

2.2.1 RoughEstimator

We now show that RoughEstimator (Figure 2-1) with probability $1 - o(1)$ (as $d \to \infty$) outputs a constant-factor approximation to $F_0(t)$ for every t in which $F_0(t)$

is sufficiently large. That is, if our estimate of $F_0(t)$ is $\widetilde{F}_0(t)$,

$$\mathbf{Pr}[\forall t \in [m] \text{ s.t. } F_0 \geq K_{\mathrm{RE}},\ \widetilde{F}_0(t) = \Theta(F_0(t))] = 1 - o(1),$$

where K_{RE} is as in Figure 2-1.

Theorem 7. *With probability $1-o(1)$, the output $\widetilde{F}_0$ of* RoughEstimator *satisfies $F_0(t) \leq \widetilde{F}_0(t) \leq 8F_0(t)$ for every $t \in [m]$ with $F_0(t) \geq K_{\mathrm{RE}}$ simultaneously. The space used is $O(\log d)$.*

Proof. We first analyze space. The counters in total take $O(K_{\mathrm{RE}} \log\log d) = O(\log d)$ bits. The hash functions h_1^j, h_2^j each take $O(\log d)$ bits. The hash functions h_3^j take $O(K_{\mathrm{RE}} \log(K_{\mathrm{RE}})) = O(\log d)$ bits.

We now analyze correctness.

Lemma 8. *For any fixed point t in the stream with $F_0(t) \geq K_{\mathrm{RE}}$, and fixed $j \in [3]$, with probability $1 - O(1/K_{\mathrm{RE}})$ we have $F_0(t) \leq \widetilde{F}_0^j(t) \leq 4F_0(t)$.*

Proof. The algorithm RoughEstimator of Figure 2-1 can be seen as taking the median output of three instantiations of a subroutine, where each subroutine has K_{RE} counters $C_1, \ldots, C_{K_{\mathrm{RE}}}$, hash functions h_1, h_2, h_3, and defined quantities $T_\ell(t) = |\{i : C_i(t) \geq \ell\}|$, where $C_i(t)$ is the state of counter C_i at time t. We show that this subroutine outputs a value $\widetilde{F}_0(t) \in [F_0(t), 4F_0(t)]$ with probability $1 - O(1/K_{\mathrm{RE}})$.

Define $I_\ell(t) \subseteq I(t)$ as the set of $i \in I(t)$ with $\mathrm{lsb}(h_1(i)) \geq \ell$. Note $|I_\ell(t)|$ is a random variable, and

$$\mathbf{E}[|I_\ell(t)|] = \frac{F_0(t)}{2^\ell},\ \mathbf{Var}[|I_\ell(t)|] = \frac{F_0(t)}{2^\ell} - \frac{F_0(t)}{2^{2\ell}} \leq \mathbf{E}[|I_\ell(t)|],$$

with the latter using 2-wise independence of h_1. Then by Chebyshev's inequality,

$$\mathbf{Pr}\left[\left||I_\ell(t)| - \frac{F_0(t)}{2^\ell}\right| \geq q \cdot \frac{F_0(t)}{2^\ell}\right] \leq \frac{1}{q^2 \cdot \mathbf{E}[|I_\ell(t)|]}. \tag{2.1}$$

Since $F_0(t) \geq K_{\mathrm{RE}}$, there exists an $\ell' \in [0, \log d]$ such that $K_{\mathrm{RE}}/2 \leq \mathbf{E}[|I_{\ell'}(t)|] < K_{\mathrm{RE}}$. We condition on the event $\mathcal{E}$ that $K_{\mathrm{RE}}/3 \leq I_{\ell'}(t) \leq 4K_{\mathrm{RE}}/3$, and note

$$\mathbf{Pr}[\mathcal{E}] = 1 - O(1/K_{\mathrm{RE}})$$

by Eq. (2.1). We also condition on the event $\mathcal{E}'$ that for all $\ell'' > \ell' + 1$, $I_{\ell''}(t) \leq 7K_{\mathrm{RE}}/24$. Applying Eq. (2.1) with $\ell = \ell' + 2$ and using that $I_{\ell+1}(t) \subseteq I_\ell(t)$,

$$\mathbf{Pr}[\mathcal{E}'] \geq 1 - O(1/K_{\mathrm{RE}}).$$

We now define two more events. The first is the event $\mathcal{E}''$ that $T_{\ell'}(t) \geq \rho K_{\mathrm{RE}}$. The second is the event $\mathcal{E}'''$ that $T_{\ell''}(t) < \rho K_{\mathrm{RE}}$ for all $\ell'' > \ell' + 1$. Note that if $\mathcal{E}'' \wedge \mathcal{E}'''$ holds, then $F_0(t) \leq \widetilde{F}_0(t) \leq 4F_0(t)$. We now show that these events hold with large probability.

Define the event $\mathcal{A}$ that the indices in $I_{\ell'}(t)$ are perfectly hashed under h_2, and the event $\mathcal{A}'$ that the indices in $I_{\ell'+2}(t)$ are perfectly hashed under h_2. Then

$$\mathbf{Pr}[\mathcal{A} \mid \mathcal{E}] \geq 1 - O(1/K_{\mathrm{RE}}).$$

and similarly for $\mathbf{Pr}[\mathcal{A}' \mid \mathcal{E}']$.

Note that, conditioned on $\mathcal{E} \wedge \mathcal{A}$, $T_{\ell'}(t)$ is distributed exactly as the number of non-empty bins when throwing $|I_{\ell'}(t)|$ balls uniformly at random into K_{RE} bins. This is because, conditioned on $\mathcal{E} \wedge \mathcal{A}$, there are no collisions of members of $I_{\ell'}(t)$ under h_2, and the independence of h_3 is larger than $|I_{\ell'}(t)|$. Thus,

$$\mathbf{E}[T_{\ell'}(t) \mid \mathcal{E} \wedge \mathcal{A}] = \left(1 - \left(1 - \frac{1}{K_{\mathrm{RE}}}\right)^{|I_{\ell'}(t)|}\right) K_{\mathrm{RE}}.$$

The same argument applies for the conditional expectation $\mathbf{E}[T_{\ell''}(t) \| \mathcal{E}' \wedge \mathcal{A}']$ for $\ell'' > \ell' + 1$. Call these conditional expectations E_ℓ. Then since $(1 - 1/n)/e \leq (1 - 1/n)^n \leq 1/e$ for all real $n \geq 1$ (see for example Proposition B.3 of [94]), we have that E_ℓ/K_{RE} lies in the interval

$$\left[\left(1 - e^{-\frac{|I_\ell(t)|}{K_{\mathrm{RE}}}}\right), \left(1 - e^{-\frac{|I_\ell(t)|}{K_{\mathrm{RE}}}}\left(1 - \frac{1}{K_{\mathrm{RE}}}\right)^{\frac{|I_\ell(t)|}{K_{\mathrm{RE}}}}\right)\right]$$

Thus for $\ell'' > \ell' + 1$,

$$E_{\ell''} \leq \left(1 - e^{-7/24}\left(1 - \frac{1}{K_{\mathrm{RE}}}\right)^{7/24}\right) K_{\mathrm{RE}}, \text{ and } E_{\ell'} \geq \left(1 - e^{-1/3}\right) K_{\mathrm{RE}}.$$

A calculation shows that $E_{\ell''} < .99 E_{\ell'}$ since $K_{\mathrm{RE}} \geq 8$.

By negative dependence in the balls and bins random process (see [40]), the Chernoff bound applies to $T_\ell(t)$ and thus

$$\mathbf{Pr}\left[|T_{\ell'}(t) - E_{\ell'}| \geq \epsilon E_{\ell'} \mid \mathcal{E} \wedge \mathcal{A}\right] \leq 2e^{-\epsilon^2 E_{\ell'}/3}$$

for any $\epsilon > 0$, and thus by taking ϵ a small enough constant,

$$\mathbf{Pr}[\mathcal{E}'' \mid \mathcal{E} \wedge \mathcal{A}] \geq 1 - e^{-\Omega(K_{\mathrm{RE}})}.$$

We also have, for $\ell'' > \ell' + 1$,

$$\mathbf{Pr}[\mathcal{E}''' \mid \mathcal{E}' \wedge \mathcal{A}'] = 1 - \mathbf{Pr}\left[T_{\ell''}(t) \geq \rho K_{\mathrm{RE}} \mid \mathcal{E}' \wedge \mathcal{A}'\right] \geq 1 - e^{-\Omega(K_{\mathrm{RE}})}.$$

Thus, overall,

$$\begin{aligned}
\mathbf{Pr}[\mathcal{E}'' \wedge \mathcal{E}'''] &\geq \mathbf{Pr}[\mathcal{E}'' \wedge \mathcal{E}''' \wedge \mathcal{E} \wedge \mathcal{E}' \wedge \mathcal{A} \wedge \mathcal{A}'] \\
&\geq \mathbf{Pr}[\mathcal{E}'' \wedge \mathcal{E} \wedge \mathcal{A}] + \mathbf{Pr}[\mathcal{E}''' \wedge \mathcal{E}' \wedge \mathcal{A}'] - 1 \\
&= \mathbf{Pr}[\mathcal{E}'' \mid \mathcal{E} \wedge \mathcal{A}] \cdot \mathbf{Pr}[\mathcal{A} \mid \mathcal{E}] \cdot \mathbf{Pr}[\mathcal{E}] \\
&\quad + \mathbf{Pr}[\mathcal{E}''' \mid \mathcal{E}' \wedge \mathcal{A}'] \cdot \mathbf{Pr}[\mathcal{A}' \mid \mathcal{E}'] \cdot \mathbf{Pr}[\mathcal{E}'] - 1 \\
&\geq 1 - O(1/K_{\mathrm{RE}})
\end{aligned}$$

■

Now, note that for any $t \in [m]$, if $\widetilde{F}_0^j(t)$ is a 4-approximation to $F_0(t)$ for at least two values of j, then $\widetilde{F}_0(t) \in [F_0(t), 4F_0(t)]$. Thus by Lemma 8, $\widetilde{F}_0(t) \in [F_0(t), 4F_0(t)]$ with probability $1-O(1/K_{\mathrm{RE}}{}^2)$. Let t_ℓ be the first time in the stream when $F_0(t_\ell) = 2^\ell$ (if no such time exists, let $t_\ell = \infty$). Then by a union bound, our estimate of $\widetilde{F}_0(t_\ell)$ is in $[F_0(t_\ell), 4F_0(t_\ell)]$ at all t_ℓ for $\ell \in [0, \log d]$ with probability $1 - O((\log d)/K_{\mathrm{RE}}{}^2) = 1 - o(1)$. Now, observe that our estimates $\widetilde{F}_0(t)$ can only ever increase with t. Thus, if our estimate is in $[F_0(t_\ell), 4F_0(t_\ell)]$ at all points t_ℓ, then it is in $[F_0(t), 8F_0(t)]$ for all $t \in [m]$. This concludes our proof. ■

2.2.2 Full algorithm

In this section we analyze our main algorithm, described in Figure 2-2, which $(1 \pm O(\varepsilon))$-approximates F_0 with 11/20 probability. We again point out that the implementation described in Figure 2-2 is not our final algorithm which achieves $O(1)$ update and reporting times; the final optimal algorithm is a modification of Figure 2-2, described in Section 2.2.4. We assume throughout that $F_0 \geq K/32$ and deal with the case of small F_0 in Section 2.2.3. The space used is $O(\varepsilon^{-2} + \log d)$ bits. Note that the 11/20 can be boosted to $1 - \delta$ for arbitrary $\delta > 0$ by running $O(\log(1/\delta))$ instantiations of our algorithm in parallel and returning the median estimate of F_0. Also, the $O(\varepsilon)$ term in the error guarantee can be made ε by running the algorithm with $\varepsilon' = \varepsilon/C$ for a sufficiently large constant C. Throughout this section we without loss of generality assume d is larger than some constant d_0, and $1/\varepsilon^2 \geq C \log d$ for a constant C of our choice, and is a power of 2. If one desires a $(1 \pm \varepsilon)$-approximation for $\varepsilon > 1/\sqrt{C \log d}$, we simply run our algorithm with $\varepsilon = 1/\sqrt{C \log d}$, which worsens our promised space bound by at most a constant factor.

The algorithm of Figure 2-2 works as follows. We maintain $K = 1/\varepsilon^2$ counters $C_1, \ldots, C_K$ as well as three values A, b, est. Each index is hashed to some level between 0 and $\log d$, based on the least significant bit of its hashed value, and is also hashed to one of the counters. Each counter maintains the deepest level of an item that was hashed to it. Up until this point, this information being kept is identical as in the LogLog [41] and HyperLogLog [48] algorithms (though our analysis will not require that the hash functions be truly random). The value A keeps track of the amount of storage required to store all the C_i, and our algorithm fails if this value ever becomes much larger than a constant times K (which we show does not happen with large probability). The value est is such that 2^{est} is a $\Theta(1)$-approximation to F_0 at all times

1. Set $K = 1/\varepsilon^2$.
2. Initialize K counters $C_1, \ldots, C_K$ to -1.
3. Pick random $h_1 \in \mathcal{H}_2([d], [0, d-1])$, $h_2 \in \mathcal{H}_2([d], [K^3])$, $h_3 \in \mathcal{H}_r([K^3], [K])$ for $r = \Omega(\log(1/\varepsilon)/\log\log(1/\varepsilon))$.
4. Initialize $A, b, \mathrm{est} = 0$.
5. Run an instantiation RE of RoughEstimator.
6. **Update(i):** Set $x \leftarrow \max\{C_{h_3(h_2(i))}, \mathrm{lsb}(h_1(i)) - b\}$.

   Set $A \leftarrow A - \left\lceil \log(2 + C_{h_3(h_2(i))}) \right\rceil + \lceil \log(2+x) \rceil$.

   If $A > 3K$, Output FAIL.

   Set $C_{h_3(h_2(i))} \leftarrow x$. Also feed i to RE.

   Let R be the output of RE.

   if $R > 2^{\mathrm{est}}$:

   (a) $\mathrm{est} \leftarrow \log(R)$, $b_{\mathrm{new}} \leftarrow \max\{0, \mathrm{est} - \log(K/32)\}$.

   (b) For each $j \in [K]$, set $C_j \leftarrow \max\{-1, C_j + b - b_{\mathrm{new}}\}$

   (c) $b \leftarrow b_{\mathrm{new}}$, $A \leftarrow \sum_{j=1}^{K} \lceil \log(C_j + 2) \rceil$.
7. **Estimator:** Define $T = |\{j : C_j \geq 0\}|$. Output $\widetilde{F}_0 = 2^b \cdot \frac{\ln\left(1 - \frac{T}{K}\right)}{\ln\left(1 - \frac{1}{K}\right)}$.

Figure 2-2: F_0 algorithm pseudocode. With probability 11/20, $\widetilde{F}_0 = (1 \pm O(\varepsilon))F_0$.

and is obtained via RoughEstimator, and b is such that we expect $F_0(t)/2^b$ to be $\Theta(K)$ at all points in the stream. Each C_i then actually holds the *offset* (from b) of the deepest level of an item that was hashed to it; if no item of level b or deeper hashed to C_i, then C_i stores -1. Furthermore, the counters are bitpacked so that C_i only requires $O(1 + \log(C_i))$ bits of storage (Section 2.2.4 states a known data structure which allows the bitpacked C_i to be stored in a way that supports efficient reads and writes).

Theorem 9. *The algorithm of Figure 2-2 uses* $O(\varepsilon^{-2} + \log d)$ *space.*

Proof. The hash functions h_1, h_2 each require $O(\log d)$ bits to store. The hash function h_3 takes $O(k \log(K)) = O(\log^2(1/\varepsilon))$ bits. The value b takes $O(\log\log d)$ bits. The value A never exceeds the total number of bits to store all counters, which is $O(\varepsilon^{-2} \log d)$, and thus A can be represented in $O(\log(1/\varepsilon) + \log\log d)$ bits. The counters C_j never in total consume more than $O(1/\varepsilon^2)$ bits by construction, since we output FAIL if they ever would. ■

Theorem 10. *The algorithm of Figure 2-2 outputs a value which is* $(1 \pm O(\varepsilon))F_0$ *with probability at least* $11/20$ *as long as* $F_0 \geq K/32$.

Proof. Let $\widetilde{F}_0^{\mathrm{RE}}(t)$ be the estimate of F_0 offered by RE at time t. Throughout this proof we condition on the event $\mathcal{E}$ that $F_0(t) \leq \widetilde{F}_0^{\mathrm{RE}}(t) \leq 8F_0(t)$ for all $t \in [m]$, which occurs with probability $1 - o(1)$ by Theorem 7.

We first show that the algorithm does not output FAIL with large probability. Note A is always $\sum_{i=1}^{K} \lceil \log(C_i + 2) \rceil$, and we must thus show that with large probability this quantity is at most $3K$ at all points in the stream. Let $A(t)$ be the value of A at time t (before running steps (a)-(c)), and similarly define $C_j(t)$. We condition on the randomness used by RE, which is independent from the remaining parts of the algorithm. Let $t_1, \ldots, t_{\ell-1}$ be the points in the stream where the output of RE changes, i.e $\widetilde{F}_0^{\mathrm{RE}}(t_j - 1) \neq \widetilde{F}_0^{\mathrm{RE}}(t_j)$ for all $j \in [\ell - 1]$, and define $t_\ell = m$. We note that $A(t)$ takes on its maximum value for $t = t_j$ for some $j \in [\ell]$, and thus it suffices to show that $A(t_j) \leq 3K$ for all $j \in [\ell]$. We furthermore note that $\ell \leq \log d + 3$ since $\widetilde{F}_0^{\mathrm{RE}}(t)$ is weakly increasing, only increases in powers of 2, and is always between 1 and $8F_0 \leq 8d$ given that $\mathcal{E}$ occurs. Now,

$$A(t) \leq K + \sum_{i=1}^{K} \log(C_i(t) + 2) \leq K + K \cdot \log\left(\frac{\sum_{i=1}^{K} C_i(t)}{K} + 2\right)$$

with the last inequality using concavity of the logarithm and Jensen's inequality. It thus suffices to show that, with large probability, $\sum_{i=1}^{K} C_i(t_j) \leq 2K$ for all $j \in [\ell]$.

Fix some $t = t_j$ for $j \in [\ell]$. For $i \in I(t)$, let $X_i(t)$ be the random variable $\max\{-1, \mathrm{lsb}(h_1(i)) - b\}$, and let $X(t) = \sum_{i \in I(t)} X_i(t)$. Note $\sum_{i=1}^{K} C_i(t) \leq X(t)$, and thus it suffices to lower bound $\mathbf{Pr}[X(t) \leq 2K]$.

We have that $X_i(t)$ equals s with probability $1/2^{b+s+1}$ for $0 \leq s < \log(d) - b$, equals $\log(d) - b$ with probability $1/d$, and equals -1 with the remaining probability mass. Thus

$$\mathbf{E}[X(t)] \le F_0(t) \cdot \left(1/d + \sum_{s=0}^{\log(d)-b-1} 2^{-(b+s+1)}\right) = F_0(t)/2^b.$$

Furthermore, by choice of h_1 the $X_i(t)$ are pairwise independent, and thus $\mathbf{Var}[X(t)] \le \mathbf{E}[X(t)]$ since $\mathbf{Var}[X_i(t)] \le \mathbf{E}[X_i(t)]$. Then by Chebyshev's inequality,

$$\mathbf{Pr}[X(t) > 2K] < \frac{F_0(t)}{2^b \cdot (2K - F_0(t)/2^b)^2}$$

Conditioned on $\mathcal{E}$, $K/256 \le F_0(t)/2^b \le K/32$, implying the above probability is at most $1/(32K)$. Then by a union bound over all t_j for $j \in [\ell]$, we have that $X(t) \le 2K$ for all $j \in [\ell]$ with probability at least $1 - \ell/(32K) \ge 1 - 1/32$ by our assumed upper bound on ε, implying we output FAIL with probability at most $1/32$.

We now show that the output from Step 7 in Figure 2-2 is $(1 \pm O(\varepsilon))F_0$ with probability $11/16$. Let $\mathcal{A}$ be the algorithm in Figure 2-2, and let $\mathcal{A}'$ be the same algorithm, but without the third line in the update rule (i.e., $\mathcal{A}'$ never outputs FAIL). We first show that the output $\widetilde{F_0}$ of $\mathcal{A}'$ is $(1 \pm O(\varepsilon))F_0$ with probability $5/8$. Let I_b be the set of indices $i \in I$ such that $\mathrm{lsb}(i) \ge b$. Then $\mathbf{E}[|I_b|] = F_0/2^b$, and $\mathbf{Var}[|I_b|] \le \mathbf{E}[|I_b|]$, with the last inequality in part using pairwise independence of h_1. We note that conditioned on $\mathcal{E}$, we have

$$K/256 \le \mathbf{E}[|I_b|] \le K/32$$

Let $\mathcal{E}'$ be the event that $K/300 \le |I_b| \le K/20$. Then by Chebyshev's inequality,

$$\mathbf{Pr}[\mathcal{E}' \mid \mathcal{E}] \ge 1 - O(1/K) = 1 - o(1).$$

Also, if we let $\mathcal{E}''$ be the event that I_b is perfectly hashed under h_2, then pairwise independence of h_2 gives

$$\mathbf{Pr}[\mathcal{E}'' \mid \mathcal{E}'] \ge 1 - O(1/K) = 1 - o(1).$$

Now, conditioned on $\mathcal{E}' \wedge \mathcal{E}''$, we have that T is a random variable counting the number of bins hit by at least one ball under a r-wise independent hash function, where there are $B = |I_b|$ balls, K bins, and $r = \Omega(\log(K/\varepsilon)/\log\log(K/\varepsilon))$. Then by Lemma 6, $T = (1 \pm 8\varepsilon)(1 - (1 - 1/K)^B)K$ with 4/5 probability, in which case

$$\ln(1 - T/K) = \ln((1 - 1/K)^B \pm 8\varepsilon(1 - (1 - 1/K)^B))$$

Conditioned on $\mathcal{E}'$, $(1 - 1/K)^B = \Theta(1)$, and thus the above is $\ln((1 \pm O(\varepsilon))(1 - 1/K)^B) = B\ln(1 - 1/K) \pm O(\varepsilon)$ since $\ln(1 + x) = O(|x|)$ for $|x| < 1/2$, and thus

$$\widetilde{F_0} = B \cdot 2^b \pm O(\varepsilon \cdot 2^b K). \tag{2.2}$$

Conditioned on $\mathcal{E}$, we have that $2^b \leq 256F_0/K$, and thus the error term in Eq. (2.2) is $O(\varepsilon F_0)$. Also, $\mathbf{E}[B] = F_0/2^b$, which is at least $K/256$ conditioned on $\mathcal{E}$. Thus by pairwise independence of h_1, Chebyshev's inequality implies

$$\mathbf{Pr}\left[|B - \mathbf{E}[B]| \geq c/\sqrt{K}\right] \leq \frac{\mathbf{E}[B]}{(c^2/K) \cdot \mathbf{E}[B]^2} \leq \left(\frac{16}{c}\right)^2$$

since $\mathbf{Var}[B] \leq \mathbf{E}[B]$, which we can make an arbitrarily small constant by setting c to be a large constant. Note that $1/\sqrt{K}$ is just ε, and thus we have that $B = (1 \pm O(\varepsilon))F_0/2^b$ with arbitrarily large constant probability.

Putting everything together, we have that, conditioned on $\mathcal{E} \wedge \mathcal{E}' \wedge \mathcal{E}''$, $\widetilde{F}_0 = (1 \pm O(\varepsilon))F_0$ with probability at least $4/5 - \delta$ for any constant $\delta > 0$ of our choice, e.g. $\delta = 1/5$. Since $\mathbf{Pr}[\mathcal{E} \wedge \mathcal{E}' \wedge \mathcal{E}''] \geq 1 - o(1)$, we thus have

$$\mathbf{Pr}[\widetilde{F}_0 = (1 \pm O(\varepsilon))F_0] \geq 3/5 - o(1).$$

Note our algorithm in Figure 2-2 succeeds as long as (1) we do not output FAIL, and (2) $\widetilde{F}_0 = (1 \pm O(\varepsilon))F_0$, and thus overall we succeed with probability at least $1 - 2/5 - o(1) - 1/32 > 11/20$. ■

2.2.3 Handling small F_0

In Section 2.2.2, we assumed that $F_0 = \Omega(K)$ for $K = 1/\varepsilon^2$ (specifically, $F_0 \geq K/32$). In this subsection, we show how to deal with the case that F_0 is small, by running a similar (but simpler) algorithm to that of Figure 2-2 in parallel.

We can apply Lemma 6 as was done in the proof of Theorem 10. We maintain $K' = 2K$ bits $B_1, \ldots, B_{K'}$ in parallel, initialized to 0. When seeing an index i in the stream, in addition to carrying out Step 6 of Figure 2-2, we also set $B_{h_3(h_2(i))}$ to 1 (h_3 can be taken to have range $K' = 2K$, and its evaluation can be taken modulo K when used in Figure 2-2 to have a size-K range). Let t_0 be the smallest $t \in [m]$ with $F_0(t) = K'/64$, and t_1 be the smallest $t \in [m]$ with $F_0(t) = K'/32$ (if no such t_i exist, set them to ∞). Define $T_B(t) = |\{i : B_i(t) = 1\}|$, and define $\widetilde{F}_0^B(t) = \ln(1 - T_B(t)/K')/\ln(1 - 1/K')$. Then by similar calculations as in Theorem 10 and a union bound over t_0, t_1, $\mathbf{Pr}[\widetilde{F}_0^B(t_i) = (1 \pm O(\varepsilon))F_0(t_i)$ for $i \in \{0, 1\}]$ is at least $1 - 2 \cdot (1/5) - o(1) = 3/5 - o(1)$. Noting that $\widetilde{F}_0^B(t)$ monotonically increases with t, we can do the following: for t with $\widetilde{F}_0^B(t) \geq K'/32 = K/16$, we output the estimator from Figure 2-2; else, we output $\widetilde{F}_0^B(t)$. We summarize this section with the following theorem.

Theorem 11. *Let $\eta > 0$ be any fixed constant, and $\varepsilon > 0$ be given. There is a subroutine requiring $O(\varepsilon^{-2} + \log d)$ space which with probability $1 - \eta$ satisfies the property that there is some $t' \in [m]$ satisfying: (1) for any fixed $t < t'$, $(1 \pm O(\varepsilon))F_0$ is output, and (2) for any $t \geq t'$ the subroutine outputs* LARGE*, and we are guaranteed $F_0(t) \geq 1/(16\varepsilon^2)$.'*

2.2.4 Running time

In this subsection we discuss an implementation of our F_0 algorithm in Figure 2-2 with $O(1)$ update and reporting times. We first state a few theorems from previous works.

Theorem 12 (Brodnik [20], Fredman and Willard [52])**.** *The least and most significant bits of an integer fitting in a machine word can be computed in constant time.*

The next two theorems give hash families which have strong independence properties while only requiring $O(1)$ evaluation time (recall that the r-wise independent hash functions of Carter and Wegman require $\Theta(r)$ evaluation time).

Theorem 13 (Pagh and Pagh [103, Theorem 1.1])**.** *Let $S \subseteq U = [u]$ be a set of $z > 1$ elements, and let $V = [v]$, with $1 < v \leq u$. Suppose the machine word size is $\Omega(\log u)$. For any constant $c > 0$ there is word RAM algorithm that, using time $\log z \log^{O(1)} v$ and $O(\log z + \log\log u)$ bits of space, selects a family $\mathcal{H}$ of functions from U to V (independent of S) such that:*

1. *With probability $1 - O(1/z^c)$, $\mathcal{H}$ is z-wise independent when restricted to S.*
2. *Any $h \in \mathcal{H}$ can be represented by a RAM data structure using $O(z \log(v))$ bits of space, and h can be evaluated in constant time after an initialization step taking $O(z)$ time.*

The following is a corollary of Theorem 2.16 in [116].

Theorem 14 (Siegel [116])**.** *Let $U = [u]$ and $V = [v]$ with $u = v^c$ for some constant $c \geq 1$, where the machine word size is $\Omega(\log v)$. Suppose one wants an $r(v)$-wise independent hash family $\mathcal{H}$ of functions mapping U to V for $k(v) = v^{o(1)}$. For any constant $\epsilon > 0$ there is a randomized procedure for constructing such an $\mathcal{H}$ which succeeds with probability $1 - 1/v^\epsilon$, taking v^ϵ bits of space. A random $h \in \mathcal{H}$ can be selected using v^ϵ bits of random seed, and h can be evaluated in $O(1)$ time.*

We now describe a fast version of RoughEstimator.

Lemma 15. RoughEstimator *can be implemented with $O(1)$ worst-case update and reporting times, at the expense of only giving a 16-approximation to $F_0(t)$ for every $t \in [m]$ with $F_0(t) \geq K_{\mathrm{RE}}$, for K_{RE} as in Figure 2-1.*

Proof. We first discuss update time. We replace each h_3^j with a random function from the hash family $\mathcal{H}$ of Theorem 13 with $z = 2K_{\mathrm{RE}}$, $u = {K_{\mathrm{RE}}}^3$, $v = K_{\mathrm{RE}}$. The constant c in Item 1 of Theorem 13 is chosen to be 1, so that each h_3^j is uniform on any given subset of z items of $[u]$ with probability $1 - O(1/K_{\mathrm{RE}})$. Note that the proof of correctness of RoughEstimator (Theorem 7) only relied on the h_3^j being uniform on some unknown set of $4K_{\mathrm{RE}}/3 < 2K_{\mathrm{RE}}$ indices with probability $1-O(1/K_{\mathrm{RE}})$ (namely, those indices in $I_{r'}(t)$). The space required to store any $h \in \mathcal{H}$ is $z \log v = O(\log d)$, which does not increase our space bound for RoughEstimator. Updates then require computing a least significant bit, and computing the h_1^j, h_2^j, h_3^j, all taking constant time.

For reporting time, in addition to the information maintained in Figure 2-1, we also maintain three sets of counters $A_0^j, A_1^j, A_2^j, A_3^j, A_4^j$ for $j \in [3]$. For a fixed j, the A_i^j store $T_{\ell+i}^j$ for an ℓ we now specify. Roughly speaking, for the values of t where $F_0(t) \geq K_{\mathrm{RE}}$, ℓ will be such that, conditioned on RoughEstimator working correctly, 2^ℓ will always be in $[F_0(t)/2, 8F_0(t)]$. We then alter the estimator of RoughEstimator to being $2^{\ell+1}$.

Note that, due to Theorem 11, the output of RoughEstimator does not figure into our final F_0 estimator until $F_0(t) \geq (1-O(\varepsilon))/(32\varepsilon^2)$, and thus the output of the algorithm is irrelevant before this time. We start off with $\ell = \log(1/(32\varepsilon^2))$. Note that $A_0^j, A_1^j, A_2^j, A_3^j, A_4^j$ can be maintained in constant time during updates. At some point t_1, the estimator from Section 2.2.3 will declare that $F_0(t_1) = (1 \pm O(\varepsilon))/(32\varepsilon^2)$, at which point we are assured $F_0(t_1) \geq 1/(64\varepsilon^2) \geq \log d$ (assuming ε is smaller than some constant, and assuming that $1/\varepsilon^2 \geq 64 \log d$). Similarly, we also have $F_0(t_1) \leq 1/(16\varepsilon^2) \leq 4 \log d$. Thus, by our choice of ℓ and conditioned on the event that RoughEstimator of Figure 2-1 succeeds (i.e., outputs a value in $[F_0(t), 8F_0(t)]$ for all t with $F_0(t) \geq K_{\mathrm{RE}}$), we can determine the median across the j of the largest ℓ^* such that $T_\ell^j \geq \rho K_{\mathrm{RE}}$ from the A_i^j and set $\ell(t_1) = \ell^*$ so that $2^{\ell(t_1)}$ is in $[F_0(t_1), 8F_0(t_1)]$.

Our argument henceforth is inductive: conditioned on the output of RoughEstimator from Figure 2-1 being correct (always in $[F_0(t), 8F_0(t)]$), $2^{\ell(t)}$ will always be in $[F_0(t)/2, 8F_0(t)]$ for all $t \geq t_1$, which we just saw is true for $t = t_1$. Note that conditioned on RoughEstimator being correct, its estimate of F_0 cannot jump by a factor more than 8 at any given point in the stream. Furthermore, if this happens, we will detect it since we store up to A_4^j. Thus, whenever we find that the estimate from RoughEstimator changed (say from $2^{\ell'}$ to $2^{\ell''}$), we increment ℓ by $\ell'' - \ell'$ and set each A_i^j to $A_{i+\ell''-\ell'}^j$ for $i \leq 4 + \ell' - \ell''$ For $4 + \ell' - \ell'' < i \leq 4$, we recompute A_i^j from scratch, by looping over the K_{RE} counters C_i. This requires $O(K_{\mathrm{RE}})$ work, but note that since $t \geq t_1$, there must be at least K_{RE} updates before $F_0(t)$ doubles, and thus we can afford to do $O(1)$ work toward this looping per update. In the meantime 2^ℓ cannot fall below $F_0/2$. ■

We will use the following "variable-bit-length array" data structure to implement the array C of counters in Figure 2-2, which has entries whose binary representations may have unequal lengths. Specifically, in Figure 2-2, the bit representation of C_i requires $O(1 + \log(C_i + 2))$ bits.

Definition 16 (Blandford, Blelloch [17]). *A* variable-bit-length array (VLA) *is a data structure implementing an array $C_1, \ldots, C_n$ supporting the following operations: (1)* **update**(i, x) *sets the value of C_i to x, and (2)* **read**(i) *returns C_i. Unlike in standard arrays, the C_i are allowed to have bit-representations of varying lengths, and we use* $\mathrm{len}(C_i)$ *to represent the length of the bit-representation of C_i.*

Theorem 17 (Blandford and Blelloch [17]). *There is a VLA data structure using $O(n + \sum_i \mathrm{len}(C_i))$ space to store n elements, supporting worst-case $O(1)$ updates and reads, under the assumptions that (1) $\mathrm{len}(C_i) \leq w$ for all i, and (2) $w \geq \log(\mathcal{M})$.*

Here w is the machine word size, and $\mathcal{M}$ is the amount of memory available to the VLA.

We now give a time-optimal version of Figure 2-2.

Theorem 18. *The algorithm of Figure 2-2 can be implemented with $O(1)$ worst-case update and reporting times.*

Proof. For update time, we select h_3 from the hash family of Theorem 14, which requires $O(1/\varepsilon^{\epsilon})$ space for arbitrarily small $\epsilon > 0$ of our choosing (say, $\epsilon = 1$), and thus this space is dominated by other parts of the algorithm. We then can evaluate h_1, h_2, h_3 in constant time, as well as compute the required least significant bit in constant time. Updating A requires computing the ceiling of a base-2 logarithm, but this is just a most significant bit computation which we can do in $O(1)$ time by Theorem 12. We can also read and write the C_j in constant time whilst using the same asymptotic space by Theorem 17.

What remains is to handle the **if** statement for when $R > 2^{\mathrm{est}}$. Note that a naïve implementation would require $O(K)$ time. Though this **if** statement occurs infrequently enough that one could show $O(1)$ amortized update time, we instead show the stronger statement that an implementation is possible with $O(1)$ worst case update time. The idea is similar to that in the proof of Lemma 15: when b_{new} changes, it cannot change more than a constant number of times again in the next $O(K)$ updates, and so we can spread the $O(K)$ required work over the next $O(K)$ stream updates, doing a constant amount of work each update.

Specifically, note that b_{new} only ever changes for times t when $R(t) > 2^{\mathrm{est(t)}} \geq K/16$, conditioned on the subroutine of Theorem 11 succeeding, implying that $F_0(t) \geq K/256$, and thus there must be at least $K/256$ updates for $F_0(t)$ to double. Since RoughEstimator always provides an 8-approximation, est can only increase by at most 3 in the next $K/256$ stream updates. We will maintain a primary and secondary instantiation of our algorithm, and only the primary receives updates. Then in cases where $R > 2^{\mathrm{est}}$ and b_{new} changes from b, we copy a sufficiently large constant number of the C_j (specifically, $3 \cdot 256$) for each of the next $K/256$ updates, from the primary to secondary structure, performing the update $C_j \leftarrow \max\{-1, C_j + b - b_{\mathrm{new}}\}$ in the secondary structure. If RoughEstimator fails and est changes by more than 3 in the next $K/256$ updates, we output FAIL. Meanwhile, during this copy phase, we process new stream updates in both the primary and secondary structures, and we answer updates from the primary structure. The analysis of correctness remains virtually unchanged, since the value 2^b corresponding to the primary structure still remains a constant-factor approximation to F_0 during this copy phase.

For reporting time, note we can maintain $T = |\{i : C_i \geq 0\}|$ during updates, and thus the reporting time is the time to compute a natural logarithm, which can be made $O(1)$ via a small lookup table (see Section 2.4.2). ■

1. Set $K = 1/\varepsilon^2$.
2. Instantiate a $\log d \times K$ bit-matrix A, initializing each $A_{i,j}$ to 0.
3. Pick random $h_1 \in \mathcal{H}_2([d],[0,d-1])$, $h_2 \in \mathcal{H}_2([d],[K^3])$, $h_3 \in \mathcal{H}_r([K^3],[K])$ for $r = \Omega(\log(1/\varepsilon)/\log\log(1/\varepsilon))$.
4. Obtain a value $R \in [F_0, cF_0]$ from some oracle, for some constant $c \geq 1$.
5. **Update(i):** Set $A_{\mathrm{lsb}(h_1(i)),h_3(h_2(i))} \leftarrow 1$.
6. **Estimator:** Set $T = |\{j \in [K] : A_{\log(16R/K),j} = 1\}|$. Output $\widetilde{F}_0 = \frac{32R}{K} \cdot \frac{\ln\left(1-\frac{T}{K}\right)}{\ln\left(1-\frac{1}{K}\right)}$.

Figure 2-3: An algorithm skeleton for F_0 estimation.

2.3 ℓ_0 estimation algorithm

Our ℓ_0 algorithm is based on the approach to F_0 estimation in Figure 2-3. In this approach, we maintain a $\log d \times K$ bit-matrix A, and upon receiving an update i, we subsample i to the row determined by the lsb of a hash evaluation, then evaluate another hash function to tell us a column and set the corresponding bit of A to 1. Note that our algorithm from Section 2.2 is just a space-optimized implementation of this approach. Specifically, in Figure 2-2 we obtained a c-approximation R to F_0 via RoughEstimator for $c = 8$. The value b we maintained was just $\max\{0, \log(32R/K)\}$. Then rather than explicitly maintaining A, we instead maintained counters C_j which allowed us to deduce whether $A_{b,j} = 1$ (specifically, $A_{b,j} = 1$ iff $C_j = 0$).

The proof of correctness of the approach in Figure 2-3 is thus essentially identical to that of Theorem 10 (in fact simpler, since we do not have to upper bound the case of outputting FAIL), so we do not repeat it here. Thus, we need only show that the approach in Figure 2-3 can be implemented for some constant $c \geq 1$ in the context of ℓ_0 estimation. Specifically, we must show that (a) the bit-matrix A can be maintained (with large probability), and (b) we can implement the oracle in Step 4 of Figure 2-3 to give a c-approximation to ℓ_0 for some constant $c \geq 1$.

We first show (a), that we can maintain the bit-matrix A with large probability. In fact, note our estimate of ℓ_0 only depends on one particular row $i^* = \log(16R/K)$ of A, so we need only ensure that we maintain row i^* with large constant probability. We first give two facts.

Fact 19. *Let $t, r > 0$ be integers. Pick $h \in \mathcal{H}_2([r],[t])$. For any $S \subset [r]$,*

$$\mathbf{E}\left[\sum_{i=1}^{s} \binom{|h^{-1}(i) \cap S|}{2}\right] \leq |S|^2/(2t).$$

Proof. Write $|S| = s$. Let $X_{i,j}$ indicate $h(i) = j$. By linearity of expectation, the desired expectation is then

$$t \sum_{i<i'} \mathbf{E}[X_{i,1}]\mathbf{E}[X_{i',1}] = t\binom{s}{2}\frac{1}{t^2} \le \frac{s^2}{2t}.$$

■

Fact 20. *Let $\mathbb{F}_q$ be a finite field and $v \in \mathbb{F}_q^d$ be a non-zero vector. Then, a random $w \in \mathbb{F}_q^d$ gives $\mathbf{Pr}_w[v \cdot w = 0] = 1/q$, where $v \cdot w$ is the inner product over $\mathbb{F}_q$.*

Proof. The set of vectors orthogonal to v is a linear subspace $V \subset \mathbb{F}_q^d$ of dimension $d-1$ and thus has q^{d-1} points. Thus, $\mathbf{Pr}[w \in V] = 1/q$. ■

Lemma 21. *There is a scheme representing each $A_{i,j}$ using $O(\log(1/\varepsilon)+\log\log(mM))$ bits such that, for $i^* = \log(16R/K)$, the (i^*)th row of A can be recovered with probability $2/3$. Furthermore, the update time and time to recover any $A_{i,j}$ are both $O(1)$.*

Proof. We represent each $A_{i,j}$ as a counter $B_{i,j}$ of $O(\log(K)+\log\log(mM))$ bits. We interpret $A_{i,j}$ as being the bit "1" if $B_{i,j}$ is non-zero; else we interpret $A_{i,j}$ as 0. The details are as follows. We choose a prime p randomly in $[D, D^3]$ for $D = 100K\log(mM)$. Notice that for mM larger than some constant, by standard results on the density of primes there are at least $K^2\log^2(mM)$ primes in the interval $[D, D^3]$. Since every frequency x_i is at most mM in magnitude and thus has at most $\log(mM)$ prime factors, non-zero frequencies remain non-zero modulo p with probability $1 - O(1/K^2)$, which we condition on occurring. We also randomly pick a vector $\mathbf{u} \in \mathbb{F}_p^K$ and $h_4 \in \mathcal{H}_2([K^3],[K])$. Upon receiving an update (i, v), we increment $B_{\mathrm{lsb}(h_1(i)),h_3(h_2(i))}$ by $v \cdot \mathbf{u}_{h_4(h_2(i))}$, then reduce modulo p.

Define $I_{i^*} = \{i \in I : \mathrm{lsb}(i) = i^*\}$. Note that conditioned on $R \in [\|x\|_0, c\|x\|_0]$, we have $\mathbf{E}[I_{i^*}] \le K/32$, and thus $\mathbf{Pr}[|I_{i^*}| \le K/20] = 1 - O(1/K) = 1 - o(1)$ by Chebyshev's inequality. We condition on this event occurring. Also, since the range of h_2 is of size K^3, the indices in I_{i^*} are perfectly hashed with probability $1 - O(1/K) = 1 - o(1)$, which we also condition on occurring.

Let $\mathcal{Q}$ be the event that p does not divide any $|x_j|$ for $j \in I_{i^*}$. Then by a union bound, $\mathbf{Pr}[\mathcal{Q}] = 1 - O(1/K)$.

Let $\mathcal{Q}'$ be the event that $h_4(h_2(j)) \ne h_4(h_2(j'))$ for distinct $j, j' \in I_{i^*}$ with $h_3(h_2(j)) = h_3(h_2(j'))$.

Henceforth, we also condition on both $\mathcal{Q}$ and $\mathcal{Q}'$ occurring, which we later show holds with good probability. Define J as the set of $j \in [K]$ such that $h_3(h_2(i)) = j$ for at least one $i \in I_{i^*}$, so that to properly represent the $A_{i^*,j}$ we should have $B_{i^*,j}$ non-zero iff $j \in J$. For each $j \in J$, $B_{i^*,j}$ can be viewed as maintaining the dot product of a non-zero vector $\mathbf{v}$, the frequency vector x restricted to coordinates in I_{i^*} which hashed to j, with a random vector $\mathbf{w}$, namely, the projection of $\mathbf{u}$ onto coordinates in I_{i^*} that hashed to j. The vector $\mathbf{v}$ is non-zero since we condition on $\mathcal{Q}$, and $\mathbf{w}$ is random since we condition on $\mathcal{Q}'$.

Now, let $X_{i,j}$ be a random variable indicating that $h_3(h_2(j)) = h_3(h_2(j'))$ for distinct $j, j' \in I_{i^*}$. Let $X = \sum_{j<j'} X_{j,j'}$. By Fact 19 with $r = K^3$, $t = K$, and

$s = |I_{i^*}| < K/20$, we have that $\mathbf{E}[X] \le K/800$. Let $Z = \{\{j, j'\} \in \binom{I_{i^*}}{2} : h_3(h_2(j)) = h_3(h_2(j'))\}$. For $(j, j') \in Z$ let $Y_{j,j'}$ be a random variable indicating $h_4(h_2(j)) = h_4(h_2(j'))$, and let $Y = \sum_{(j,j') \in Z} Y_{j,j'}$. Then by pairwise independence of h_4, and the fact that we conditioned on I_{i^*} being perfectly hashed under h_2, we have

$$\mathbf{E}[Y] = \sum_{(j,j') \in Z} \mathbf{Pr}[h_4(h_2(j)) = h_4(h_2(j'))] = |Z|/K.$$

Note $|Z| = X$. Conditioned on $X \le 20\mathbf{E}[X] \le K/40$, which happens with probability at least $19/20$ by Markov's inequality, we have that $\mathbf{E}[Y] \le |Z|/K \le 1/40$, so that $\mathbf{Pr}[Y \ge 1] \le 1/40$. Thus, $\mathcal{Q}'$ holds with probability at least $(19/20) \cdot (39/40) > 7/8$.

Finally, by Fact 20 with $q = p$, and union bounding over all K counters $B_{i^*,j}$, no $B_{i^*,j}$ for $j \in J$ is 0 with probability $1 - K/p \ge 99/100$. Thus, our scheme overall succeeds with probability $(7/8) \cdot (99/100) - o(1) > 2/3$. ■

We next show (b) in Section 2.3.1, i.e. give an algorithm providing an $O(1)$-approximation to ℓ_0 with $O(1)$ update and reporting times. The space used is $O(\log d \log\log mM)$.

Note that, as with our F_0 algorithm, we also need to have an algorithm which provides a $(1 \pm \varepsilon)$-approximation when $\ell_0 \ll 1/\varepsilon^2$. Just as in Section 2.2.3, this is done by using the same scheme as in Section 2.2.3, but using Lemma 21 to represent our bit array.

Putting everything together, we have the following.

Theorem 22. *There is an algorithm using space $O(\varepsilon^{-2} \log d(\log(1/\varepsilon) + \log\log(mM)))$ for $(1 \pm \varepsilon)$-approximating ℓ_0 with $2/3$ success probability, and with $O(1)$ worst-case update and reporting times.*

2.3.1 A Rough Estimator for ℓ_0 estimation

We describe here a subroutine RoughL0Estimator which gives a constant-factor approximation to ℓ_0 with probability $9/16$. First, we need the following lemma which states that when ℓ_0 is at most some constant c, it can be computed exactly in small space. The lemma follows by picking a random prime $p = \Theta(\log(mM) \log\log(mM))$ and pairwise independently hashing the universe into $[\Theta(c^2)]$ buckets. Each bucket is a counter which tracks the sum of frequencies modulo p of updates to universe items landing in that bucket. The estimate of ℓ_0 is then the total number of non-zero counters, and the maximum estimate after $O(\log(1/\eta))$ trials is finally output. This gives the following.

Lemma 23. *Given $\eta \in (0, 1/2)$, there is an algorithm using $O(c^2 \log\log(mM) \log(1/\eta))$ space, in addition to needing to store $O(\log(1/\eta))$ independently chosen pairwise independent hash functions mapping $[d]$ into $[c^2]$, which when given the promise that $\ell_0 \le c$ can output ℓ_0 exactly with probability at least $1 - \eta$. The worst-case update and reporting times are $O(1)$.*

Now we describe RoughL0Estimator. We pick a function $h : [d] \to [d]$ at random from a pairwise independent family. For each $0 \le j \le \log d$ we create a substream $\mathcal{S}^j$ consisting of those $x \in [d]$ with $\mathrm{lsb}(h(x)) = j$. Let $\ell_0(\mathcal{S})$ denote ℓ_0 of the substream $\mathcal{S}$. For each $\mathcal{S}^j$ we run an instantiation B_j of Lemma 23 with $c = 141$ and $\eta = 1/16$. All instantiations share the same $O(\log(1/\eta))$ hash functions $h^1, \ldots, h^{O(\log(1/\eta))}$.

To obtain our final estimate of ℓ_0 for the entire stream, we find the largest value of j for which B^j declares $\ell_0(\mathcal{S}^j) > 8$. Our estimate of ℓ_0 is $\tilde{\ell}_0 = 2^j$. If no such j exists, we estimate $\tilde{\ell}_0 = 1$.

Theorem 24. RoughL0Estimator *with probability at least* $9/16$ *outputs a value* $\tilde{\ell}_0$ *satisfying* $\ell_0 \le \tilde{\ell}_0 \le 110\ell_0$. *The space used is* $O(\log d \log\log mM)$, *and the update and reporting times are* $O(1)$.

Proof. The space to store h is $O(\log d)$. The $\Theta(\log(1/\eta))$ hash functions h^i in total require $O(\log(1/\eta)\log d) = O(\log d)$ bits to store since $1/\eta = O(1)$. The remaining space to store a single B^j for a level is $O(\log\log mM)$ by Lemma 23, and thus storing all B^j across all levels requires $O(\log d \log\log mM)$ space.

As for running time, upon receiving a stream update (i, v), we first hash i using h, taking time $O(1)$. Then, we compute $\mathrm{lsb}(h(i))$, also in constant time. Now, given our choice of η for B^j, we can update B^j in $O(1)$ time by Lemma 23.

To obtain $O(1)$ reporting time, we again use the fact that we can compute the least significant bit of a machine word in constant time. We maintain a single machine word z of at least $\log d$ bits and treat it as a bit vector. We maintain that the jth bit of z is 1 iff $\ell_0(\mathcal{S}^j)$ is reported to be greater than 8 by B^j. This property can be maintained in constant time during updates. Constant reporting time then follows since finding the deepest level j with greater than 8 reported elements is equivalent to computing $\mathrm{lsb}(z)$.

Now we prove correctness. Observe that $\mathbf{E}[\ell_0(\mathcal{S}^j)] = \ell_0/2^{j+1}$ when $j < \log d$ and $\mathbf{E}[\ell_0(\mathcal{S}^j)] = \ell_0/2^j = \ell_0/d$ when $j = \log d$. Let j^* be the largest j satisfying $\mathbf{E}[\ell_0(\mathcal{S}^j)] \ge 1$ and note that $1 \le \mathbf{E}[\ell_0(\mathcal{S}^{j^*})] \le 2$. For any $j > j^*$, $\mathbf{Pr}[\ell_0(\mathcal{S}^j) > 8] < 1/(8 \cdot 2^{j-j^*-1})$ by Markov's inequality. Thus, by a union bound, the probability that any $j > j^*$ has $\ell_0(\mathcal{S}^j) > 8$ is at most $(1/8) \cdot \sum_{j-j^*=1}^{\infty} 2^{-(j-j^*-1)} = 1/4$. Now, let $j^{**} < j^*$ be the largest j such that $\mathbf{E}[\ell_0(\mathcal{S}^j)] \ge 55$, if such a j exists. Since we increase the j by powers of 2, we have $55 \le \mathbf{E}[\ell_0(\mathcal{S}^{j^{**}})] < 110$. Note that h is pairwise independent, so $\mathbf{Var}[\ell_0(\mathcal{S}^{j^{**}})] \le \mathbf{E}[\ell_0(\mathcal{S}^{j^{**}})]$. For this range of $\mathbf{E}[\ell_0(\mathcal{S}^{j^{**}})]$, we then have by Chebyshev's inequality that

$$\mathbf{Pr}\left[|\ell_0(\mathcal{S}^{j^{**}}) - \mathbf{E}[\ell_0(\mathcal{S}^{j^{**}})]| \ge 3\sqrt{\mathbf{E}[\ell_0(\mathcal{S}^{j^{**}})]}\right] \le 1/9.$$

If $|\ell_0(\mathcal{S}^{j^{**}}) - \mathbf{E}[\ell_0(\mathcal{S}^{j^{**}})]| < 3\sqrt{\mathbf{E}[\ell_0(\mathcal{S}^{j^{**}})]}$, then

$$32 < 55 - 3\sqrt{55} < \ell_0(\mathcal{S}^{j^{**}}) < 110 + 3\sqrt{110} < 142$$

since $55 \le \mathbf{E}[\ell_0(\mathcal{S}^{j^{**}})] < 110$.

So far we have shown that with probability at least $3/4$, $\ell_0(\mathcal{S}^j) \le 8$ for all $j > j^*$. Thus, for these j the B^j will estimate ℓ_0 of the corresponding substreams to be at most 8, and we will not output $\tilde{\ell}_0 = 2^j$ for $j > j^*$. On the other hand, we know for

j^{**} (if it exists) that with probability at least 8/9, $\mathcal{S}^{j^{**}}$ will have $32 < \ell_0(\mathcal{S}_i^{j^{**}}) < 142$. By our choice of $c = 141$ and $\eta = 1/16$ in the B^j, $B^{j^{**}}$ will output a value $\tilde{\ell}_0(\mathcal{S}_i^{j^{**}}) \geq \ell_0(\mathcal{S}_i^{j^{**}})/4 > 8$ with probability at least $1 - (1/9 + 1/16) > 13/16$ by Lemma 23. Thus, with probability at least $1 - (3/16 + 1/4) = 9/16$, we output $\tilde{\ell}_0 = 2^j$ for some $j^{**} \leq j \leq j^*$, which satisfies $110 \cdot 2^j < \ell_0 \leq 2^j$. If such a j^{**} does not exist, then $\ell_0 < 55$, and 1 is a 55-approximation in this case. ■

2.4 Further required proofs

2.4.1 Various calculations for balls and bins

Fact 3 (restatement). $\mathbf{Var}[X] = K\,(K-1)\left(1 - \frac{2}{K}\right)^A + K\left(1 - \frac{1}{K}\right)^A - K^2\left(1 - \frac{1}{K}\right)^{2A}$ *and* $\mathbf{E}[X] = K\left(1 - \left(1 - \frac{1}{K}\right)^A\right)$

Proof. The computation for $\mathbf{E}[X]$ follows by linearity of expectation.

For $\mathbf{Var}[X]$, we have

$$\begin{aligned}
\mathbf{Var}[X] &= \mathbf{E}[X^2] - \mathbf{E}^2[X] \\
&= \sum_i \mathbf{E}[X_i^2] + 2\sum_{i<j} \mathbf{E}[X_i X_j] - \mathbf{E}^2[X]
\end{aligned}$$

We have $\mathbf{E}[X_i^2] = \mathbf{E}[X_i]$, so the first sum is simply $\mathbf{E}[X]$. We now calculate $\mathbf{E}[X_i X_j]$ for $i \neq j$. Let Y_i indicate that at least one good ball landed in bin i, and let Z_i indicate that at least one bad ball landed in bin i. Then,

$$\begin{aligned}
\mathbf{E}[X_i X_j] &= \mathbf{Pr}[Y_i \wedge Y_j \wedge \bar{Z}_i \wedge \bar{Z}_j] \\
&= \mathbf{Pr}[\bar{Z}_i \wedge \bar{Z}_j] \cdot \mathbf{Pr}[Y_i \wedge Y_j | \bar{Z}_i \wedge \bar{Z}_j] \\
&= \mathbf{Pr}[\bar{Z}_i \wedge \bar{Z}_j] \cdot \mathbf{Pr}[Y_i \wedge Y_j] \\
&= \left(1 - \frac{2}{K}\right)^B \cdot (1 - \mathbf{Pr}[\bar{Y}_i \wedge \bar{Y}_j] - \mathbf{Pr}[Y_i \wedge \bar{Y}_j] \\
&\qquad - \mathbf{Pr}[\bar{Y}_i \wedge Y_j]) \\
&= \left(1 - \frac{2}{K}\right)^B \cdot \left(1 - \mathbf{Pr}[\bar{Y}_i \wedge \bar{Y}_j] - 2 \cdot \mathbf{Pr}[Y_i \wedge \bar{Y}_j]\right) \\
&= \left(1 - \frac{2}{K}\right)^B \cdot (1 - \mathbf{Pr}[\bar{Y}_i \wedge \bar{Y}_j] \\
&\qquad - 2 \cdot \mathbf{Pr}[\bar{Y}_j] \cdot \mathbf{Pr}[Y_i | \bar{Y}_j]) \\
&= \left(1 - \frac{2}{K}\right)^B \cdot \left(1 - \left(1 - \frac{2}{K}\right)^A - 2\left(1 - \frac{1}{K}\right)^A \right. \\
&\qquad \left. \times \left(1 - \left(1 - \frac{1}{K-1}\right)^A\right)\right)
\end{aligned}$$

$$
\begin{aligned}
&= \left(1-\frac{2}{K}\right)^B \cdot \left(1-\left(1-\frac{2}{K}\right)^A - 2\left(1-\frac{1}{K}\right)^A \right. \\
&\qquad \left. + 2\left(1-\frac{2}{K}\right)^A\right) \\
&= \left(1-\frac{2}{K}\right)^B \cdot \left(1-2\left(1-\frac{1}{K}\right)^A + \left(1-\frac{2}{K}\right)^A\right)
\end{aligned}
$$

The variance calculation then follows by noting $2\sum_{i<j} \mathbf{E}[X_i X_j] = K(K-1)\mathbf{E}[X_1 X_2]$ then expanding out $\mathbf{E}[X] + K(K-1)\mathbf{E}[X_1 X_2] - \mathbf{E}^2[X]$. ■

The following is a proof of Lemma 4. Recall that there are A balls being tossed into K bins independently at random, and X is the random variable which is the number of bins receiving at least one ball.

Lemma 4 (restatement). *If* $1 \leq A \leq K/20$, *then* $\mathbf{Var}[X] < 4A^2/K$.

Proof. By Fact 3,

$$
\begin{aligned}
\mathbf{Var}[X] &= K\left(K-1\right)\left(1-\frac{2}{K}\right)^A + K\left(1-\frac{1}{K}\right)^A \\
&\qquad - K^2\left(1-\frac{1}{K}\right)^{2A} \\
&= K^2\left[\left(1-\frac{2}{K}\right)^A - \left(1-\frac{1}{K}\right)^{2A}\right] \\
&\qquad + K\left[\left(1-\frac{1}{K}\right)^A - \left(1-\frac{2}{K}\right)^A\right] \\
&= K^2\left(1-\frac{2}{K}\right)^A\left[1-\left(\frac{1-\frac{2}{K}+\frac{1}{K^2}}{1-\frac{2}{K}}\right)^A\right] \\
&\qquad + K\left[\left(1-\frac{1}{K}\right)^A - \left(1-\frac{2}{K}\right)^A\right] \\
&= K^2\left(1-\frac{2}{K}\right)^A\left[1-\left(1+\frac{1}{K^2\left(1-\frac{2}{K}\right)}\right)^A\right] \\
&\qquad + K\left[\left(1-\frac{1}{K}\right)^A - \left(1-\frac{2}{K}\right)^A\right] \\
&= K^2\left(1-\frac{2}{K}\right)^A\left[1-\left(1+\frac{A}{K^2\left(1-\frac{2}{K}\right)}+E_1\right)\right] \\
&\qquad + K\left[\left(1-\frac{A}{K}+E_2\right) - \left(1-\frac{2A}{K}+E_3\right)\right],
\end{aligned}
$$

where E_1, E_2, and E_3 are the sum of quadratic and higher terms of the binomial expansions for $(1+1/(K^2(1-2/K)))^A$, $(1-1/K)^A$, and $(1-2/K)^A$, respectively. Continuing the expansion,

$$\begin{aligned}
\mathbf{Var}[X] &= -K^2\left(1-\frac{2}{K}\right)^A\left(\frac{A}{K^2\left(1-\frac{2}{K}\right)}+E_1\right) \\
&\qquad + A + K(E_2-E_3) \\
&= -A\left(1-\frac{2}{K}\right)^{A-1} - K^2E_1\left(1-\frac{2}{K}\right)^A \\
&\qquad + A + K(E_2-E_3) \\
&= -A\left(1-\frac{2(A-1)}{K}+E_4\right) - K^2E_1\left(1-\frac{2}{K}\right)^A \\
&\qquad + A + K(E_2-E_3) \\
&= -A + \frac{2A(A-1)}{K} - AE_4 - K^2E_1\left(1-\frac{2}{K}\right)^A \\
&\qquad + A + K(E_2-E_3) \\
&= \frac{2A(A-1)}{K} - AE_4 - K^2E_1\left(1-\frac{2}{K}\right)^A \\
&\qquad + K(E_2-E_3),
\end{aligned}$$

where E_4 is the sum of quadratic and higher terms of the binomial expansion of $(1-2/K)^{A-1}$.

We claim that E_1, E_2, E_3, and E_4 are non-negative. We use Bernoulli's inequality, which is that $(1+x)^i \geq 1+ix$ for integers $i \geq 1$ and any real number $x \geq -1$. This inequality implies the sum of quadratic and higher terms in the expansion of $(1+x)^i$ is non-negative, since $1+ix$ corresponds to the remaining terms. Since $1 \leq A \leq K/20$, we have that $K \geq 20$, and so $1/(K^2(1-2/K)), -1/K$, and $-2/K$ are greater than -1. Bernoulli's inequality thus implies E_1, E_2, and E_3 are non-negative. Moreover, since we can in fact assume that $A \geq 2$, as otherwise there is 1 ball and so $\mathbf{Var}[X] = 0$, we have that $A-1 \geq 1$ and so Bernoulli's inequality implies E_4 is also non-negative.

Hence,

$$\mathbf{Var}[X] \leq \frac{2A(A-1)}{K} + K(E_2-E_3).$$

We also have,

$$\begin{aligned}
E_2 - E_3 &= \left(\frac{\binom{A}{2}}{K^2} - \frac{\binom{A}{3}}{K^3} + \cdots\right) - \left(\frac{4\binom{A}{2}}{K^2} - \frac{8\binom{A}{3}}{K^3} + \cdots\right) \\
&= -\frac{3\binom{A}{2}}{K^2} + \frac{7\binom{A}{3}}{K^3} - \cdots
\end{aligned}$$

This series can be upper bounded by the series

$$\sum_{i=2}^{\infty} \frac{(2^i - 1)(A/K)^i}{i!},$$

and lower bounded by the series

$$-\sum_{i=2}^{\infty} \frac{(2^i - 1)(A/K)^i}{i!}.$$

This series, in absolute value, is a geometric series with starting term $3A^2/(2K^2)$ and ratio at most $A/K \leq 1/20$. Thus, $|E_2 - E_3| \leq (20/19) \cdot (3A^2/(2K^2)) = (30/19) \cdot (A/K)^2$, and so $|K(E_2 - E_3)| \leq (30/19) \cdot A^2/K$.

Hence, $\mathbf{Var}[X] \leq \frac{2A^2}{K} + \frac{30A^2}{19K} < \frac{4A^2}{K}$, as desired. ■

2.4.2 A compact lookup table for the natural logarithm

Lemma 25. *Let $K > 4$ be a positive integer, and write $\gamma = 1/\sqrt{K}$. It is possible to construct a lookup table requiring $O(\gamma^{-1}\log(1/\gamma))$ bits such that $\ln(1-c/K)$ can then be computed with relative accuracy γ in constant time for all integers $c \in [4K/5]$.*

Proof. We set $\gamma' = \gamma/15$ and discretize the interval $[1, 4K/5]$ geometrically by powers of $(1+\gamma')$. We precompute the natural algorithm evaluated at $1 - \rho/K$ for all discretization points ρ, with relative error $\gamma/3$, creating a table A taking space $O(\gamma^{-1}\log(1/\gamma))$. We answer a query $\ln(1-c/K)$ by outputting the natural logarithm of the closest discretization point in A. First, we argue that the error from this output is small enough. Next, we argue that the closest discretization point can be found in constant time.

For the error, the output is up to $(1 \pm \gamma/3)$,

$$\begin{aligned}\ln(1 - (1 \pm \gamma')c/K) &= \ln(1 - c/K \pm \gamma' c/K) \\ &= \ln(1 - c/K) \pm 5\gamma' c/K \\ &= \ln(1 - c/K) \pm \gamma c/(3K).\end{aligned}$$

Using the fact that $|\ln(1-z)| \geq z/(1-z)$ for $0 < z < 1$, we have that $|\ln(1-c/K)| \geq c/(K-c) \geq c/K$. Thus,

$$\begin{aligned}(1 \pm \gamma/3)(\ln(1 - c/(3K)) \pm \gamma c/K) &= (1 \pm \gamma/3)^2 \ln(1 - c/K) \\ &= (1 \pm \gamma)\ln(1 - c/K).\end{aligned}$$

Now, for finding the discretization point, note we need to look up $A[\lceil \log_{1+\gamma'}(c) \rceil] = A[\lceil \log(c)/(a\gamma') \rceil]$, where $a\gamma' = \log(1+\gamma')$ (note, we can compute $\log(1+\gamma') = a\gamma'$ in preprocessing). Now, write $c = c' \cdot 2^k$ where $k = \lfloor \log(c) \rfloor$ and thus $1 \leq c' < 2$. We can compute k in $O(1)$ time since it is the most significant bit of c. We know $\log_{1+\gamma'}(c) = \log(c' \cdot 2^k)/(a\gamma') = k/(a\gamma') + \log(c')/(a\gamma')$. Now, the derivative of the log function in the range $[1, 2)$ is sandwiched between two constants. Thus, if we discretize $[1, 2)$ evenly into $O(\gamma'^{-1})$ buckets and store the log of a representative of each bucket in a lookup table B, we can additively $O(\gamma')$-approximate $\log c'$ by table lookup of $B[\lfloor (c' - 1)/\gamma' \rfloor]$. So now we have computed

$$\begin{aligned} k/(a\gamma') + &(\log c' + O(\gamma'))/(a\gamma') \\ &= k/(a\gamma') + \log(c')/(a\gamma') \pm O(1). \end{aligned}$$

This $O(1)$ can be taken to be arbitrarily small, say at most $1/3$, by tuning the constant in the discretization. Thus we know the correct index to look at in our index table A up to $\pm 1/3$; since indices are integers, we are done. ■

Chapter 3

Moment Estimation

In this chapter we describe our contributions for the problem of moment estimation. We would like to $(1 \pm \varepsilon)$-approximate

$$F_p \stackrel{\text{def}}{=} \|x\|_p^p = \sum_{i=1}^d |x_i|^p.$$

For p a constant bounded away from 0, this is equivalent to estimating $\ell_p \stackrel{\text{def}}{=} \|x\|_p$ up to changing ε by a constant factor.

In Section 3.2 we describe an improved derandomization of Indyk's median algorithm [67], which is sufficient to yield the first optimal space algorithm for this problem. The update time of this algorithm, however, is $\Theta(1/\varepsilon^2)$. In Section 3.3 we give a more involved algorithm which simultaneously achieves optimal space and a faster update time of $O(\log^2(1/\varepsilon)\log\log(1/\varepsilon))$.

The analysis of both our algorithms requires an analytical tool we dub *FT-mollification*, which we describe in Section 3.1. Essentially, we use FT-mollification to show that the CDF of a random variable $A = \sum_i A_i x_i$ is pointwise close everywhere to the CDF of a random variable $B = \sum_i B_i x_i$ as long as the B_i and A_i are identically distributed, and where the A_i are fully independent and the B_i are only r-wise independent for sufficiently large r. This statement is equivalent to the statement

$$\forall t \in \mathbb{R}, |\mathbf{E}[I_{[t,\infty)}(A) - \mathbf{E}[I_{[t,\infty)}(B)]| < \gamma$$

for some small $\gamma > 0$. One might think to show such a statement by approximating $I_{[t,\infty)}$ by a low-degree polynomial, obtained for example by Taylor expansion, then bounding the difference in expectations by bounding the expectation of the Taylor error. Unfortunately, Taylor's theorem requires the function to be sufficiently smooth, whereas $I_{[t,\infty)}$ is not even continuous. Instead, we devise a smooth approximation $\tilde{I}^c_{[t,\infty)}$ to $I_{[a,b]}$ then show the chain of inequalities

$$\mathbf{E}[I_{[t,\infty)}(A)] \approx_\varepsilon \mathbf{E}[\tilde{I}^c_{[t,\infty)}(A)] \approx_\varepsilon \mathbf{E}[\tilde{I}^c_{[t,\infty)}(B)] \approx_\varepsilon \mathbf{E}[I_{[t,\infty)}(B)].$$

This smooth function $\tilde{I}^c_{[t,\infty)}$ is obtained via our FT-mollification procedure.

Bibliographic remarks: FT-mollification was initially investigated in joint work with Kane and Woodruff [78], though it was later refined to the form seen in Section 3.1 in subsequent joint work with Diakonikolas and Kane [38]. Section 3.2 and Section 3.4 are based on joint work with Kane and Woodruff [78]. Section 3.3 is based on joint work with Kane, Porat, and Woodruff [77], which itself built upon joint work with Woodruff [98].

3.1 FT-Mollification

In this section, we describe a technical procedure called *FT-mollification* for smoothing real functions. The main theorem of this section is the following.

Theorem 26. *Let $f : \mathbb{R} \to \mathbb{R}$ be bounded and $c > 1$ be arbitrary. Then, there exists a function $\tilde{f}^c : \mathbb{R} \to \mathbb{R}$ satisfying the following properties.*

i. $\|(\tilde{f}^c)^{(\ell)}\|_\infty \le \|f\|_\infty \cdot (2c)^\ell$ *for all* $\ell \ge 0$.

ii. Fix some $x \in \mathbb{R}$. *Then if* $|f(x) - f(y)| \le \varepsilon$ *whenever* $x - y \le \delta$ *for some* $\varepsilon \ge 0$, $\delta > 0$, *then* $|\tilde{F}^c(x) - F(x)| \le \varepsilon + \|f\|_\infty \cdot 5/(2c^2\delta^2)$.

iii. If f is nonnegative everywhere, then so is $\tilde{f}^c$.

We prove Theorem 26 by a procedure we dub *FT-mollification*. This procedure works as follows. We define the function $b : \mathbb{R} \to \mathbb{R}$ by

$$b(x) = \frac{\sqrt{15}}{4} \cdot \begin{cases} 1 - x^2 & \text{for } |x| < 1 \\ 0 & \text{otherwise} \end{cases}.$$

The value $\sqrt{15}/4$ was chosen so that $\|b\|_2^2 = 1$.

We let $\hat{b} : \mathbb{R}^d \to \mathbb{R}$ denote the Fourier transform of b, i.e.

$$\hat{b}(t) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} b(x) e^{-itx} dx.$$

Finally, $B : \mathbb{R} \to \mathbb{R}$ denotes the function $\hat{b}^2$, and we define $B_c : \mathbb{R} \to \mathbb{R}$ by

$$B_c(x) = c \cdot B(cx).$$

Definition 27 (FT-mollification)**.** *For $f : \mathbb{R} \to \mathbb{R}$ and given $c > 0$, we define the* FT-mollification $\tilde{f}^c : \mathbb{R} \to \mathbb{R}$ *by*

$$\tilde{f}^c(x) = (B_c * f)(x) = \int_{\mathbb{R}} B_c(y) f(x - y) dy.$$

FT-mollification provides the function $\tilde{f}^c$ guaranteed in Theorem 26. Before we prove Theorem 26, we show a few lemmas.

Lemma 28. *For any $c > 0$,*

$$\int_{\mathbb{R}} B_c(x)dx = 1.$$

Proof. Since $B = \hat{b}^2$, the stated integral when $c = 1$ is $\|\hat{b}\|_2^2$, which is $\|b\|_2^2 = 1$ by Plancherel's theorem. For general c, make the change of variables $u = cx$ then integrate over u. ∎

Lemma 29. *For any $\ell \geq 0$, $\|B^{(\ell)}\|_1 \leq 2^\ell$.*

Proof. By the product rule for derivatives,

$$B^{(\ell)} = \sum_{j=0}^{\ell} \binom{\ell}{j} \cdot \hat{b}^{(j)} \cdot \hat{b}^{(\ell-j)}$$

Thus,

$$\begin{aligned}
\|B^{(\ell)}\|_1 &= \left\| \sum_{j=0}^{\ell} \binom{\ell}{j} \cdot \hat{b}^{(j)} \cdot \hat{b}^{(\ell-j)} \right\|_1 \\
&\leq \sum_{j=0}^{\ell} \binom{\ell}{j} \cdot \left\|\hat{b}^{(j)}\right\|_2 \cdot \left\|\hat{b}^{(\ell-j)}\right\|_2 && (3.1) \\
&= \sum_{j=0}^{\ell} \binom{\ell}{j} \left\|x^j \cdot b\right\|_2 \cdot \left\|x^{\ell-j} \cdot b\right\|_2 && (3.2) \\
&\leq \sum_{j=0}^{\ell} \binom{\ell}{j} && (3.3) \\
&= 2^\ell
\end{aligned}$$

Eq. (3.1) follows by Cauchy-Schwarz. Eq. (3.2) follows from Plancherel's theorem, since the Fourier transform of $\hat{b}^{(j)}$ is $x^j \cdot b$, up to factors of i. Eq. (3.3) follows since $\|x^{(j)} \cdot b\|_2 \leq \|b\|_2 = 1$. ∎

Lemma 30. *Let $z > 0$ be arbitrary. Then*

$$\int_{|x|>z} B(x)dx = 5/(2z^2).$$

Proof. Consider the integral

$$S = \int_{\mathbb{R}} x^2 \cdot B(x)dx.$$

Recalling that $B = \hat{b}^2$, the Fourier transform of B is $(2\pi)^{-1/2}(b*b)$. The above integral is $(2\pi)^{1/2}$ times the Fourier transform of $x^2 \cdot B$, evaluated at 0. Since multiplying a function by ix corresponds to differentiation in the Fourier domain,

$$S = \left(\frac{d^2}{dx^2}(b * b)\right)(0) = \left(\left(\frac{d}{dx}b\right) * \left(\frac{d}{dx}b\right)\right)(0) = \left\|\frac{d}{dx}b\right\|_2^2$$

with the last equality using that the derivative of b is an odd function.

We have, for $|x| < 1$,

$$\left(\frac{\partial}{\partial x_i}b\right)(x) = -2x,$$

and thus,

$$\left\|\frac{d}{dx}b\right\|_2^2 = \frac{5}{2}.$$

We now finish the proof of the lemma. Since B has unit integral on $\mathbb{R}$ (Lemma 28) and is nonnegative everywhere, we can view B as the density function of a probability distribution on $\mathbb{R}$. Then S can be viewed as $\mathbf{E}_{x\sim B}[x^2]$. Then by Markov's inequality,

$$\mathbf{Pr}_{x\sim B}\left[x^2 \geq z^2\right] \leq \mathbf{E}[x^2]/z^2,$$

which is equivalent to

$$\mathbf{Pr}_{x\sim B}\left[|x| \geq z\right] \leq \mathbf{E}[x^2]/z^2.$$

We conclude by observing that the above probability is simply

$$\int_{|x|\geq z} B(x)dx.$$

■

We are now ready to prove Theorem 26, the main theorem of this section.

Proof (of Theorem 26). We first prove (i).

$$\begin{aligned}
\left|(\tilde{f}^c)^{(\ell)}(x)\right| &= \left|(B_c * f)^{(\ell)}(x)\right| \\
&= \left|(B_c^{(\ell)} * f)(x)\right| \\
&= \left|\int_{\mathbb{R}} B_c^{(\ell)}(y)f(x-y)dy\right| \\
&\leq \|f\|_\infty \cdot \left\|B_c^{(\ell)}\right\|_1 \\
&= \|f\|_\infty \cdot c^\ell \cdot \left\|B^{(\ell)}\right\|_1 \\
&\leq \|f\|_\infty \cdot (2c)^\ell
\end{aligned} \tag{3.4}$$

with the last inequality holding by Lemma 29.

We now prove (ii).

$$\begin{aligned}
\tilde{f}^c(x) &= (B_c * f)(x) \\
&= \int_{\mathbb{R}} B_c(x-y)f(y)dy
\end{aligned}$$

$$
\begin{aligned}
&= f(x) + \int_{\mathbb{R}} (f(y) - f(x))B_c(x-y)dy && (3.5)\\
&= f(x) + \int_{|x-y|<\delta} (f(y) - f(x))B_c(x-y) + \int_{|x-y|\geq\delta} (f(y) - f(x))B_c(x-y)\\
&= f(x) \pm \varepsilon \cdot \int_{|x-y|<\delta} |B_c(x-y)| + \int_{|x-y|\geq\delta} (f(y) - f(x))B_c(x-y)\\
&= f(x) \pm \varepsilon \cdot \int_{\mathbb{R}} B_c(x-y) + \int_{|x-y|\geq\delta} (f(y) - f(x))B_c(x-y)\\
&= f(x) \pm \varepsilon \pm \|f\|_\infty \cdot \int_{|x-y|\geq\delta} B_c(x-y)dy\\
&= f(x) \pm \varepsilon \pm \|f\|_\infty \cdot \int_{|u|\geq c\delta} B(u)du\\
&= f(x) \pm \varepsilon \pm \|f\|_\infty \cdot 5/(2c^2\delta^2)
\end{aligned}
$$

where Eq. (3.5) uses Lemma 28.

Item (iii) follows since if f is nonnegative everywhere, then $\tilde{f}^c$ is the convolution of two nonnegative functions and is thus itself nonnegative. ■

We then naturally have the following corollary. By $d(x, \partial S)$ below, we simply mean the distance from x to the complement of S if $x \in S$, and the distance from x to S if $x \notin S$.

Corollary 31. *For any subset $S \subseteq \mathbb{R}$ and $x \in \mathbb{R}$,*

$$|I_S(x) - \tilde{I}_S^c(x)| \leq \min\left\{1, \frac{5}{2c^2 \cdot d(x, \partial S)^2}\right\}.$$

Proof. We have $|I_S(x) - \tilde{I}_S^c(x)| \leq 1$ always since $\tilde{I}_S^c$ is nonnegative and is never larger than $\|I_S\|_\infty = 1$. The other bound is obtained by applying Theorem 26 to $f = I_S$ with $\varepsilon = 0$, $\delta = d(x, \partial S)$. ■

3.2 Slow Moment Estimation in Optimal Space

Here we describe a proof that Indyk's ℓ_p estimation algorithm [67] can be more efficiently derandomized (Figure 3-1) to produce a space-optimal algorithm. Remarks on approximating median($|\mathcal{D}_p|$) are in Section 3.2.2. This space-optimal variant can be implemented to have update time $O(1/\varepsilon^2)$. In Section 3.3 we give a more sophisticated algorithm, building on some of the ideas in this section, which simultaneously achieves optimal space and update time $\tilde{O}(\log^2(1/\varepsilon))$.

We assume $p \in (0, 2)$ is a fixed constant bounded away from $\{0, 2\}$. Some constants in our asymptotic notation are functions of p. We also assume $\|x\|_p > 0$; $\|x\|_p = 0$ is detected when $y = 0$ in Figure 3-1. Finally, we assume $\varepsilon \geq 1/\sqrt{m}$. Otherwise, the trivial solution of keeping the entire stream in memory requires $O(m \log(dM)) =$

1. Pick a random matrix $S \in \mathbb{R}^{k\times d}$ as follows for $k = \Theta(1/\varepsilon^2)$. Each $S_{i,j}$ is distributed according to $\mathcal{D}_p$. For fixed i, the $S_{i,j}$ are r-wise independent with $r = \Theta(1/\varepsilon^p)$. For $i \neq i'$, the seeds used to generate the $\{S_{i,j}\}_{j=1}^d$ and $\{S_{i',j}\}_{j=1}^d$ are pairwise independent.
2. Maintain the vector $y = Sx$ throughout the stream.
3. Output $\mathrm{median}\{|y_i|\}_{i=1}^k/\mathrm{median}(|\mathcal{D}_p|)$.

Figure 3-1: Indyk's derandomized ℓ_p estimation algorithm pseudocode, $0 < p < 2$, assuming infinite precision.

$O(\varepsilon^{-2}\log(dM))$ space, which can be made $O(\varepsilon^{-2}\log(mM))$ by the argument in Section 3.5.1. Our main theorem in this section is the following.

Theorem 32. *For all $p \in (0, 2)$, the algorithm of Figure 3-1 can be implemented with limited precision to use space $O(\varepsilon^{-2}\log(mM))$ and output $(1\pm\varepsilon)\|x\|_p$ with probability at least $7/8$. The update time is $O(1/\varepsilon^2)$.*

To understand the first step of Figure 3-1, we recall the definition of a p-stable distribution, initially studied by Lévy and Khintchine (a thorough treatment of these distributions can be found in a book by Zolotarev [128]).

Definition 33. *For $0 < p \leq 2$, there exists a probability distribution $\mathcal{D}_p$ called the* p-stable distribution *with $\mathbf{E}[e^{itZ}] = e^{-|t|^p}$ for $Z \sim \mathcal{D}_p$. For any d and vector $x \in \mathbb{R}^d$, if $Z, Z_1, \ldots, Z_d \sim \mathcal{D}_p$ are independent, then $\sum_{j=1}^d Z_j x_j \sim \|x\|_p Z$.*

We also state a lemma giving the decay of the density function of $\mathcal{D}_p$, which will be useful later. See Theorem 42 in Section 3.2.2 for a proof.

Lemma 34. *For $0 < p < 2$, the density function φ_p of the p-stable distribution satisfies $\varphi_p(x) = O(1/(1 + |x|)^{p+1})$ and is an even function. The cumulative distribution function thus satisfies $\Phi_p(x) = O(1/(1 + |x|)^{-p})$.*

We now prove the following technical lemma, which plays a role in our later analyses.

Lemma 35. *There exists an $\varepsilon_0 > 0$ such that the following holds. Let n be a positive integer and $0 < \varepsilon < \varepsilon_0$, $0 < p < 2$ be given. Let $f : \mathbb{R} \to \mathbb{R}$ satisfy $\|f^{(\ell)}\|_\infty = O(\alpha^\ell)$ for all $\ell \geq 0$, for some α satisfying $\alpha^p \geq \log(1/\varepsilon)$. Let $r = \alpha^p$. Let $a \in \mathbb{R}^n$ satisfy $\|a\|_p = O(1)$. Let X_i be a $3Cr$-independent family of p-stable random variables for C a suitably large even constant. Let Y_i be a fully independent family of p-stable random variables. Let $X = \sum_i a_i X_i$ and $Y = \sum_i a_i Y_i$. Then $\mathbf{E}[f(X)] = \mathbf{E}[f(Y)] + O(\varepsilon)$.*

Proof. The basic idea of the proof will be to show that the expectation can be computed to within $O(\varepsilon)$ just by knowing that the X_i's are $O(r)$-wise independent.

Our main idea is to approximate f by a Taylor series and use our knowledge of the moments of the X_i. The problem is that the tails of p-stable distributions for $p < 2$ are wide, and hence the expectations of the moments are infinite. We circumvent this difficulty via a combination of truncation and approximate inclusion-exclusion.

Define the random variables

$$U_i = \begin{cases} 1 & \text{if } |a_i X_i| > \lambda \\ 0 & \text{otherwise} \end{cases}$$

and

$$X_i' = (1 - U_i)X_i = \begin{cases} 0 & \text{if } |a_i X_i| > \lambda \\ X_i & \text{otherwise} \end{cases},$$

where we set $\lambda = 1/\alpha$. We note a couple of properties of these. First,

$$\mathbf{E}[U_i] = O\left(\int_{|a_i|^{-1}\lambda}^{\infty} x^{-1-p} dx\right) = O\left(|a_i|^p \lambda^{-p}\right)$$

by Lemma 34. We would also like to bound the moments of X_i'. We note that $\mathbf{E}[(a_i X_i')^\ell]$ is 1 for $\ell = 0$, by symmetry is 0 when ℓ is odd, and otherwise is

$$\begin{aligned} O\left(\int_0^{|a_i|^{-1}\lambda} (a_i x)^\ell x^{-p-1}\right) &= O\left(|a_i|^\ell (|a_i|^{-1}\lambda)^{\ell-p}\right) \\ &= O\left(|a_i|^p \lambda^{\ell-p}\right) \end{aligned} \tag{3.6}$$

where the implied constant above can be chosen to hold independently of ℓ.

For $S \subseteq [n]$, let $\mathbf{1}_S$ be the indicator random variable for the event

$$S = \{i : U_i = 1\}.$$

Then since $\sum_{S\subseteq[n]} \mathbf{1}_S = 1$ for any point in the probability space,

$$\mathbf{E}[f(X)] = \mathbf{E}\left[\sum_{S\subseteq[n]} \mathbf{1}_S \cdot f(X)\right] = \sum_{S\subseteq[n]} \mathbf{E}[\mathbf{1}_S \cdot f(X)]. \tag{3.7}$$

Now let $\mathbf{1}_S'$ be the indicator random variable for the event

$$S \subseteq \{i : U_i = 1\}$$

so that

$$\mathbf{1}_S = \mathbf{1}_S' \cdot \left(\prod_{i\notin S} (1 - \mathbf{1}_{\{i\}}')\right) = \sum_{T\subset[n]\backslash S} (-1)^{|T|} \mathbf{1}_{S\cup T}',$$

and define

$$\begin{aligned} & F_{S,T}\left(\overrightarrow{X}\right) \\ & = (-1)^{|T|}\left(\prod_{i\in S\cup T} U_i\right) f\left(\sum_{i\in S} a_i X_i + \sum_{i\notin S} a_i X_i'\right). \end{aligned}$$

Then, by definition of U_i and Eq. (3.7),

$$\mathbf{E}[f(X)] = \mathbf{E}\left[\sum_{S\subseteq[n]}\sum_{T\subseteq[n]\backslash S} F_{S,T}\left(\overrightarrow{X}\right)\right].$$

We will approximate $\mathbf{E}[f(X)]$ as

$$\mathbf{E}\left[\sum_{\substack{S\subseteq[n]\\ |S|\le Cr}}\sum_{\substack{T\subseteq[n]\backslash S\\ |T|\le Cr}} F_{S,T}\left(\overrightarrow{X}\right)\right]. \tag{3.8}$$

That is, we approximate $\mathbf{E}[f(X)]$ using approximate inclusion-exclusion, by truncating the summation to not include large S, T. Call the function inside the expectation in Eq. (3.8) $F\left(\overrightarrow{X}\right)$. We would like to bound the error in approximating $f(X)$ by $F\left(\overrightarrow{X}\right)$. Fix values of the X_i, and let O be the set of i with $U_i = 1$. We note that

$$F\left(\overrightarrow{X}\right) = \sum_{\substack{S\subseteq O\\ |S|\le Cr}}\sum_{\substack{T\subseteq O\backslash S\\ |T|\le Cr}} (-1)^{|T|} f\left(\sum_{i\in S} a_i X_i + \sum_{i\notin S} a_i X_i'\right).$$

Notice that other than the $(-1)^{|T|}$ term, the expression inside the sum does not depend on T. This means that if $0 < |O\backslash S| \le Cr$ then the inner sum is 0, since $O\backslash S$ will have exactly as many even subsets as odd ones. Hence if $|O| \le Cr$, we have that

$$\begin{aligned} F\left(\overrightarrow{X}\right) &= \sum_{S=O} f\left(\sum_{i\in S} a_i X_i + \sum_{i\notin S} a_i X_i'\right) \\ &= f\left(\sum_{i\in O} a_i X_i + \sum_{i\notin O} a_i X_i'\right) \\ &= f(X). \end{aligned}$$

Otherwise, after fixing O and S, we can sum over possible values of $t = |T|$ and obtain:

$$\sum_{\substack{T \subseteq O \setminus S \\ |T| \leq Cr}} (-1)^{|T|} = \sum_{t=0}^{Cr} (-1)^t \binom{|O \setminus S|}{t}.$$

In order to bound this we use the following lemma.

Lemma 36. *For integers $A \geq B+1 > 0$ we have that $\sum_{i=0}^{B} (-1)^i \binom{A}{i}$ and $\sum_{i=0}^{B+1} (-1)^i \binom{A}{i}$ have different signs, with the latter sum being 0 if $A = B+1$.*

Proof. First suppose that $B < A/2$. We note that since the terms in each sum are increasing in i, each sum has the same sign as its last term, proving our result in this case. For $B \geq A/2$ we note that $\sum_{i=0}^{A} (-1)^i \binom{A}{i} = 0$, and hence letting $j = A - i$, we can replace the sums by $(-1)^{A+1} \sum_{j=0}^{A-B-1} (-1)^j \binom{A}{j}$ and $(-1)^{A+1} \sum_{j=0}^{A-B-2} (-1)^j \binom{A}{j}$, reducing to the case of $B' = A - B - 1 < A/2$. ■

Using Lemma 36, we note that $\sum_{t=0}^{Cr} (-1)^t \binom{|O \setminus S|}{t}$ and $\sum_{t=0}^{Cr+1} (-1)^t \binom{|O \setminus S|}{t}$ have different signs. Therefore we have that

$$\left| \sum_{t=0}^{Cr} (-1)^t \binom{|O \setminus S|}{t} \right| \leq \binom{|O \setminus S|}{Cr+1} = \binom{|O| - |S|}{Cr+1}.$$

Recalling that $\|f\|_\infty$ is bounded, we are now ready to bound $\left| F\left(\overrightarrow{X}\right) - f(X) \right|$. Recall that if $|O| \leq Cr$, this difference is 0, and otherwise we have that

$$\begin{aligned}
\left| F\left(\overrightarrow{X}\right) - f(X) \right| &\leq \|f\|_\infty \cdot \sum_{\substack{S \subseteq O \\ |S| \leq Cr}} \binom{|O| - |S|}{Cr+1} \\
&= O\left(\sum_{s=0}^{Cr} \binom{|O|}{s} \binom{|O| - s}{Cr+1} \right) \\
&= O\left(\sum_{s=0}^{Cr} \binom{|O|}{Cr+s+1} \binom{Cr+s+1}{s} \right) \\
&= O\left(\sum_{s=0}^{Cr} 2^{Cr+s+1} \binom{|O|}{Cr+s+1} \right).
\end{aligned}$$

Therefore we can bound the error as

$$\begin{aligned}
&\left| \mathbf{E}\left[F\left(\overrightarrow{X}\right) \right] - \mathbf{E}[f(X)] \right| \\
&= O\left(\sum_{s=0}^{Cr} 2^{Cr+s+1} \mathbf{E}\left[\binom{|O|}{Cr+s+1} \right] \right).
\end{aligned} \tag{3.9}$$

We note that
$$\binom{|O|}{Cr+s+1} = \sum_{\substack{I\subseteq[n]\\|I|=Cr+s+1}} \prod_{i\in I} U_i.$$
Hence by linearity of expectation, $(2Cr+1)$-wise independence, and the fact that $s \leq Cr$,
$$\begin{aligned}
\mathbf{E}\left[\binom{|O|}{Cr+s+1}\right] &= \sum_{\substack{I\subseteq[n]\\|I|=Cr+s+1}} \mathbf{E}\left[\prod_{i\in I} U_i\right] \\
&= \sum_{\substack{I\subseteq[n]\\|I|=Cr+s+1}} \prod_{i\in I} O(|a_i|^p\lambda^{-p}) \\
&= e^{O(Cr)} \sum_{\substack{I\subseteq[n]\\|I|=Cr+s+1}} \left(\prod_{i\in I} |a_i|^p\lambda^{-p}\right).
\end{aligned}$$
We note when this sum is multiplied by $(Cr+s+1)!$, these terms all show up in the expansion of $\left(\|a\|_p^p\lambda^{-p}\right)^{Cr+s+1}$. In fact, for any integer $0 \leq t \leq n$,
$$\sum_{\substack{I\subseteq[n]\\|I|=t}} \left(\prod_{i\in I} |a_i|^p\lambda^{-p}\right) \leq \frac{(\|a\|_p^p\lambda^{-p})^t}{t!}. \tag{3.10}$$
Hence, since $\|a\|_p = O(1)$,
$$\begin{aligned}
\mathbf{E}\left[\binom{|O|}{Cr+s+1}\right] &= \frac{e^{O(Cr)}\lambda^{-p(Cr+s+1)}}{(Cr+s+1)!} \\
&= e^{O(Cr)}\left(\frac{\lambda^{-p}}{Cr}\right)^{(Cr+s+1)}.
\end{aligned}$$
Therefore, by choice of λ and Eq. (3.9),
$$\begin{aligned}
\left|\mathbf{E}\left[F\left(\overrightarrow{X}\right)\right] - \mathbf{E}[f(X)]\right| &= e^{O(Cr)}\sum_{s=0}^{Cr}\left(\frac{\lambda^{-p}}{Cr}\right)^{(Cr+s+1)} \\
&= C^{-O(Cr)} = O(\varepsilon). \qquad (3.11)
\end{aligned}$$
Hence it suffices to approximate $\mathbf{E}\left[F\left(\overrightarrow{X}\right)\right]$.

Recall
$$F\left(\overrightarrow{X}\right) = \sum_{\substack{S,T\subseteq[n]\\|S|,|T|\leq Cr\\S\cap T=\emptyset}} F_{S,T}\left(\overrightarrow{X}\right).$$

We will attempt to compute the conditional expectation of $F_{S,T}\left(\overrightarrow{X}\right)$, conditioned on the X_i for $i \in S \cup T$. Note the independence on the X_i's is sufficient that the values of the X_i for $i \in S \cup T$ are fully independent of one another, and that even having fixed these values, the remaining X_i are still Cr-wise independent.

We begin by making some definitions. Let $R = [n]\backslash(S \cup T)$. Having fixed S, T, and the values of X_i for $i \in S \cup T$, set $c = \sum_{i\in S} a_i X_i$ and $X' = \sum_{i \in R} a_i X_i'$. We note that $F_{S,T}\left(\overrightarrow{X}\right) = 0$ unless $U_i = 1$ for all $i \in S \cup T$, and otherwise that $F_{S,T}\left(\overrightarrow{X}\right) = f(c + X')$. This is because if $U_i = 1$ for some $i \in T$, then $X_i' = 0$. Let $p_c(x)$ be the Taylor series for $f(c + x)$ about 0, truncated so that its highest degree term is degree $Cr - 1$. We approximate $\mathbf{E}[f(c + X')]$ by $\mathbf{E}[p_c(X')]$.

Lemma 37. $|\mathbf{E}[f(c + X')] - \mathbf{E}[p_c(X')]| < e^{-Cr}$.

Proof. By Taylor's theorem, the fact that C is even, and our given bounds on $\|f^{(Cr)}\|_\infty$,

$$|p_c(x) - f(c+x)| \leq \frac{|x|^{Cr}\alpha^{Cr}}{(Cr)!} = \frac{x^{Cr}\alpha^{Cr}}{(Cr)!}.$$

We note that $\mathbf{E}[p_c(X')]$ is determined simply by the independence properties of the X_i since it is a low-degree polynomial in functions of the X_i.

We now attempt to bound the error in approximating $f(c+x)$ by $p_c(x)$. In order to do so we will wish to bound $\mathbf{E}[(X')^{Cr}]$. Let $\ell = Cr$. We have that $\mathbf{E}[(X')^\ell] = \mathbf{E}\left[\left(\sum_{i\in R} a_i X_i'\right)^\ell\right]$. Expanding this out and using linearity of expectation yields a sum of terms of the form $\mathbf{E}\left[\prod_{i \in R}(a_i X_i')^{\ell_i}\right]$, for some non-negative integers ℓ_i summing to ℓ. Let L be the set of i so that $\ell_i > 0$. Since $|L| \leq \ell$ which is at most the degree of independence, Eq. (3.6) implies that the above expectation is $\left(\prod_{i\in L}|a_i|^p \lambda^{-p}\right)\lambda^\ell e^{O(|L|)}$. Notice that the sum of the coefficients in front of such terms with a given L is at most $|L|^\ell$. This is because for each term in the product, we need to select an $i \in L$. Eq. (3.10) implies that summing $\prod_{i \in L}|a_i|^p\lambda^{-p}$ over all subsets L of size s, gives at most $\frac{(\|a\|_p^p\lambda^{-p})^s}{s!}$. Putting everything together:

$$\begin{aligned}\mathbf{E}\left[(X')^\ell\right] &\leq \sum_{s=1}^{\ell} \frac{s^\ell \lambda^{\ell - sp} e^{O(s)}}{s!} \\ &= \sum_{s=1}^{\ell} \exp(\ell \log s - s\log s \\ &\qquad - (\ell - sp)\log(1/\lambda) + O(s)).\end{aligned} \tag{3.12}$$

The summand (ignoring the $O(s)$) is maximized when

$$\frac{\ell}{s} + \log(1/\lambda^p) = \log(s) + 1.$$

Rearranging terms and setting $u = \ell \cdot \lambda^p$, this happens when $\ell = s\log(\lambda^p \cdot s) + s$, which occurs for

$$s = \left(1 + O\left(\frac{\log\log(u)}{\log(u)}\right)\right) \cdot \frac{\ell}{\log(u)}.$$

Since the sum is at most ℓ times the biggest term,

$$\begin{aligned}\mathbf{E}\left[(X')^{\ell}\right] \leq \exp\Bigg(\ell \cdot \Bigg(&\log(\ell) - \log\log(u) - \frac{\log(\ell)}{\log(u)} \\ &- \log(1/\lambda) + \frac{\log(1/\lambda^p)}{\log(u)} + O(1)\Bigg)\Bigg).\end{aligned}$$

Therefore we have that

$$\begin{aligned}|\mathbf{E}[f(c+X')] - \mathbf{E}[p_c(X')]| &\leq \mathbf{E}\left[\frac{(X')^{\ell}\alpha^{\ell}}{\ell!}\right] \\ &\leq \exp\Bigg(\ell \cdot \Bigg(\log(\alpha) - \log\log(u) - \frac{\log(\ell)}{\log(u)} \\ &\qquad - \left(1 - \frac{p}{\log(u)}\right)\log(1/\lambda) + O(1)\Bigg)\Bigg) \\ &= \exp\Bigg(\ell \cdot \Bigg(\log(\alpha) - \log\log(u) - \frac{\log(\ell)}{\log(u)} \\ &\qquad - \left(\frac{1}{p} - \frac{1}{\log(u)}\right)\log(\ell/u) + O(1)\Bigg)\Bigg) \\ &= \exp\Big(\ell \cdot \Big(\log(\alpha) - \log\log(u) - \log(\ell^{1/p}) \\ &\qquad + \log(u^{1/p}) + O(1)\Big)\Big) \\ &= \exp\left(-\ell \cdot \left(\log\left(\frac{1}{\alpha \cdot \lambda}\right) + \log\log(u) - O(1)\right)\right) \\ &< e^{-\ell}\end{aligned}$$

with the last inequality holding for C (and hence u) a sufficiently large constant. ■

So to summarize:

$$\mathbf{E}[f(X)] = \mathbf{E}\left[F\left(\overrightarrow{X}\right)\right] + O(\varepsilon).$$

Now,

$$\mathbf{E}\left[F\left(\overrightarrow{X}\right)\right] = \sum_{\substack{S,T \subseteq [n] \\ |S|,|T| \leq Cr \\ S \cap T = \emptyset}} \mathbf{E}\left[F_{S,T}\left(\overrightarrow{X}\right)\right]$$

$$= \sum_{\substack{S,T\subseteq[n]\\|S|,|T|\le Cr\\S\cap T=\emptyset}} (-1)^{|T|} \int_{\{x_i\}_{i\in S\cup T}} \left(\left(\prod_{i\in S\cup T} U_i \right) \times \mathbf{E}[f(c+X')] \right) dX_i(x_i)$$

$$= \sum_{\substack{S,T\subseteq[n]\\|S|,|T|\le Cr\\S\cap T=\emptyset}} (-1)^{|T|} \int_{\{x_i\}_{i\in S\cup T}} \left(\left(\prod_{i\in S\cup T} U_i \right) \times \left(\mathbf{E}[p_c(X')] \pm e^{-Cr}\right) \right) dX_i(x_i).$$

We recall that the term involving $\mathbf{E}[p_c(X')]$ is entirely determined by the $3Cr$-independence of the X_i's. We are left with an error of magnitude

$$e^{-Cr} \cdot \left(\sum_{\substack{S,T\subseteq[n]\\|S|,|T|\le Cr\\S\cap T=\emptyset}} (-1)^{|T|} \int_{\{x_i\}_{i\in S\cup T}} \left(\prod_{i\in S\cup T} U_i \right) dX_i(x_i) \right)$$

$$\le e^{-Cr} \cdot \left(\sum_{\substack{S,T\subseteq[n]\\|S|,|T|\le Cr\\S\cap T=\emptyset}} \mathbf{E}\left[\prod_{i\in S\cup T} U_i \right] \right)$$

$$\le e^{-Cr} \cdot \left(\sum_{\substack{S,T\subseteq[n]\\|S|,|T|\le Cr\\S\cap T=\emptyset}} \left(\prod_{i\in S\cup T} |a_i|^p \lambda^{-p} \right) e^{O(|S|+|T|)} \right).$$

Letting $s = |S| + |T|$, we change this into a sum over s. We use Eq. (3.10) to upper bound the product. We also note that given $S \cup T$, there are at most 2^s ways to pick S and T. Putting this together and recalling the choice of λ and that $\|a\|_p = O(1)$, the above is at most

$$e^{-Cr} \left(\sum_{s=0}^{2Cr} 2^s \left(\frac{\|a\|_p^{ps} \lambda^{-ps}}{s!} \right) e^{O(s)} \right) = e^{-Cr} \left(\sum_{s=0}^{2Cr} \frac{O\left(\frac{1}{\lambda^p}\right)^s}{s!} \right) < e^{-Cr} \cdot e^{O\left(\frac{1}{\lambda^p}\right)} = O(\varepsilon). \tag{3.13}$$

Hence $\mathbf{E}[f(X)]$ is determined up to $O(\varepsilon)$. ■

Remark 38. *For constant α in the statement of Lemma 35, one can slightly optimize the proof to show that $r = \log(1/\varepsilon)/\log\log(1/\varepsilon)$ suffices. We describe here the necessary changes in the proof. First, we instead set $\lambda = (Cr)^{-1/10}$. In the first inequality where the value of λ is used, namely Eq. (3.11), the desired difference is then $(Cr)^{-O(Cr)} = O(\varepsilon)$. Next, in the proof of Lemma 37, the summand in Eq. (3.12) is maximized when $s = O(\ell/\log\ell)$, and straightforward calculations show that the desired difference in the statement of Lemma 37 is $O(\varepsilon^2)$. The left hand side of Eq. (3.13) is then at most $O(\varepsilon^2)\cdot e^{O(1/\lambda^p)}$, which is still $O(\varepsilon)$.*

We now prove Theorem 32. We defer analysis of the required precision (and hence the required space) to Section 3.2.1, and here just argue correctness.

Proof (of Theorem 32). Consider first the following argument that Indyk's median estimator provides a $(1\pm\varepsilon)$-approximation when $k = \Theta(1/\varepsilon^2)$ and we use a sketch matrix B such that the $B_{i,j}$ are *fully* independent from $\mathcal{D}_p$. The following argument is only slightly different from Indyk's original argument, but is presented in such a way that adapts well to the entries of the sketch matrix having limited independence. Let $z = Bx$ be the sketch when using the fully independent matrix B. Since we scale our final output by median$(|\mathcal{D}_p|)$, we henceforth argue as if median$(|\mathcal{D}_p|) = 1$ (remarks on computing median$(|\mathcal{D}_p|)$ with relative error $1\pm\varepsilon$ are in Section 3.2.2. The value 1 being the median is equivalent to the statement $\mathbf{E}[I_{[-1,1]}(z_i/\|x\|_p)] = 1/2$, since $\mathcal{D}_p$ is symmetric. Let φ_p be the density function of $\mathcal{D}_p$. Then by compactness, φ_p takes on some minimum value η_p in the interval $[-2,2]$ (a strictly positive lower bound on η_p can be efficiently computed using Theorem 42). Then

$$\mathbf{E}\left[I_{[-1+\varepsilon,1-\varepsilon]}\left(\frac{z_i}{\|x\|_p}\right)\right] \le \frac{1}{2} - \eta_p\varepsilon = \frac{1}{2} - \Theta(\varepsilon) \tag{3.14}$$

and

$$\mathbf{E}\left[I_{[-1-\varepsilon,1+\varepsilon]}\left(\frac{z_i}{\|x\|_p}\right)\right] \ge \frac{1}{2} + \eta_p\varepsilon = \frac{1}{2} + \Theta(\varepsilon). \tag{3.15}$$

Now if we let

$$Z = \frac{1}{k}\sum_{i=1}^{k} I_{[-1+\varepsilon,1-\varepsilon]}\left(\frac{z_i}{\|x\|_p}\right)$$

and

$$Z' = \frac{1}{k}\sum_{i=1}^{k} I_{[-1-\varepsilon,1+\varepsilon]}\left(\frac{z_i}{\|x\|_p}\right)$$

then $\mathbf{E}[Z] = 1/2 - \Theta(\varepsilon)$, $\mathbf{E}[Z'] = 1/2 + \Theta(\varepsilon)$, and $\mathbf{Var}[Z], \mathbf{Var}[Z'] \le 1/k = \Theta(\varepsilon^2)$. Writing $k = c'/\varepsilon^2$, Chebyshev's inequality and a union bound imply $Z = 1/2 - \Theta(\varepsilon)$ and $Z' = 1/2 + \Theta(\varepsilon)$ with probability 7/8 for large c'. We conclude by noting median$\{|z_i|\}_{i=1}^k = (1\pm\varepsilon)\|x\|_p$ when both these events occur.

We now modify the above argument to handle our case where we use the sketch $y = Sx$ with the $S_{i,j}$ only r-wise independent for fixed i, and the seeds used to generate different rows of S being pairwise independent. Note that once we established bounds on $\mathbf{E}[Z]$ and $\mathbf{E}[Z']$ above, concentration of Z, Z' was shown via Chebyshev's inequality, which only required pairwise independence between rows of S. Thus, we need only show that $\mathbf{E}[I_{[a,b]}(z_i/\|x\|_p)] \approx_\varepsilon \mathbf{E}[I_{[a,b]}(y_i/\|x\|_p)]$. Ideally we would just apply Lemma 35, but we cannot do this directly: the function $I_{[a,b]}$ does not have bounded high-order derivatives. We instead argue indirectly via FT-mollification. Let c be some value in $\Theta(1/\varepsilon)$, and let $Z_i = z_i/\|x\|_p$ and $Y_i = y_i/\|x\|_p$. We argue the following chain of inequalities:

$$\mathbf{E}[I_{[a,b]}(Z_i)] \approx_\varepsilon \mathbf{E}[\tilde{I}^c_{[a,b]}(Z_i)] \approx_\varepsilon \mathbf{E}[\tilde{I}^c_{[a,b]}(Y_i)] \approx_\varepsilon \mathbf{E}[I_{[a,b]}(Y_i)]. \tag{3.16}$$

Noting that $I_{[a,b]}(y) = I_{[a,\infty)}(y) - I_{[b,\infty)}(y)$, it suffices to show Eq. (3.16) with $I_{[a,b]}, \tilde{I}^c_{[a,b]}$ replaced by $I_{[\theta,\infty)}, \tilde{I}_{[\theta,\infty)}$ for arbitrary $\theta \in \mathbb{R}$.

$\mathbf{E}[\mathbf{I}_{[\theta,\infty)}(\mathbf{Z_i})] \approx_\varepsilon \mathbf{E}[\tilde{\mathbf{I}}^{\mathbf{c}}_{[\theta,\infty)}(\mathbf{Z_i})]$**:** We have

$$\begin{aligned}
|\mathbf{E}[I_{[\theta,\infty)}(Z_i)] - \mathbf{E}[\tilde{I}^c_{[\theta,\infty)}(Z_i)]| &\le \mathbf{E}[|I_{[\theta,\infty)}(Z_i) - \tilde{I}^c_{[\theta,\infty)}(Z_i)|] \\
&\le \mathbf{Pr}[|Z_i - \theta| < \varepsilon] + \sum_{s=0}^{\infty} \mathbf{Pr}[2^s\varepsilon \le |Z_i - \theta| < 2^{s+1}\varepsilon] \cdot O(c^{-2}2^{-2s}\varepsilon^{-2}) \\
&\le O(\varepsilon) + O(c^{-2}\varepsilon^{-2}) \cdot \sum_{s=0}^{\infty} 2^{-2s} \cdot \mathbf{Pr}[|Z_i - \theta| < 2^{s+1}\varepsilon] \\
&= O(\varepsilon) + O(c^{-2}\varepsilon^{-2}) \cdot O(\varepsilon)
\end{aligned}$$

The third inequality holds because the p-stable distribution is anticoncentrated (the probability that Z_i is in any interval of length t is $O(t)$).

$\mathbf{E}[\tilde{\mathbf{I}}^{\mathbf{c}}_{[\mathbf{a},\mathbf{b}]}(\mathbf{Z_i})] \approx_\varepsilon \mathbf{E}[\tilde{\mathbf{I}}^{\mathbf{c}}_{[\mathbf{a},\mathbf{b}]}(\mathbf{Y_i})]$**:** Apply Lemma 35 with $\alpha = \Theta(1/\varepsilon)$ and vector $x/\|x\|_p$.

$\mathbf{E}[\tilde{\mathbf{I}}^{\mathbf{c}}_{[\mathbf{a},\mathbf{b}]}(\mathbf{Y_i})] \approx_\varepsilon \mathbf{E}[\mathbf{I}_{[\mathbf{a},\mathbf{b}]}(\mathbf{Y_i})]$**:** We argue as in the first inequality, but now we must show anticoncentration of the Y_i. That is, we must show that for any $t, \gamma \in \mathbb{R}$ with $\gamma \ge \varepsilon$, $\mathbf{Pr}[|Y_i - t| < \gamma] = O(\gamma)$. Suppose there exists nonnegative $f_{t,\gamma} : \mathbb{R} \to \mathbb{R}$ with

i. $\|f^{(\ell)}_{t,\gamma}\|_\infty = O(\alpha^\ell)$ for all $\ell \ge 0$, with $\alpha = O(1/\gamma)$

ii. $\mathbf{E}[f_{t,\gamma}(D)] = O(\gamma)$ for $D \sim \mathcal{D}_p$

iii. $f_{t,\gamma} \ge I_{[t-\gamma,t+\gamma]}$ on $\mathbb{R}$

By (i), (ii) and Lemma 35 we would have $\mathbf{E}[f_{t,\gamma}(Y_i)] \approx_\gamma \mathbf{E}[f_{t,\gamma}(D)] = O(\gamma)$. Then (iii) implies $\mathbf{E}[I_{[t-\gamma,t+\gamma]}(Y_i)] \le \mathbf{E}[f_{t,\gamma}(Y_i)] = O(\gamma)$, as desired. It only remains to exhibit such a function $f_{t,\gamma}$. We take $f_{t,\gamma} = 2\tilde{I}^{c'}_{[t-2\gamma,t+2\gamma]}$ for $c' = \sqrt{5}/\gamma$. Item (i) is immediate from Theorem 26(i). Item (ii) follows by applying the argument from the first two steps of Eq. (3.16) applied to the function $I_{[t-2\gamma,t+2\gamma]}$. For item (iii), first consider y with $|y - t| > \gamma$. In this case, $I_{[t-\gamma,t+\gamma]}(y) = 0$, whereas $f_{t,\gamma}(y) \ge 0$. Now consider the case $|y - t| < \gamma$. Then we have that for $S = [t - 2\gamma, t + 2\gamma]$, $\partial S = \{t - 2\gamma, t + 2\gamma\}$. Thus, $d(y, \partial S) > \gamma$. Now, applying Corollary 31 with our choice of c', we have $|I_{[t-2\gamma,t+2\gamma]}(y) - \tilde{I}^{c'}_{[t-2\gamma,t+2\gamma]}(y)| < 1/2$. Since $I_{[t-2\gamma,t+2\gamma]} = 1$, this implies $f_{t,\gamma}(y) \geq 1 = I_{[t-\gamma,t+\gamma]}(y)$.

Naïvely the update time would be $O(1/\varepsilon^{2+p})$, since during an update we would have to evaluate an r-wise independent hash function k times for $r = O(1/\varepsilon^p)$ and $k = O(1/\varepsilon^2)$. We discuss in Remark 39 though how in fact the algorithm can be implemented to have update time $O(k + r)$. ■

Remark 39. To achieve the desired update time, observe that each row of S needs a random hash function from an r-wise independent family mapping $[d]$ into $[q]$, where $q = \mathrm{poly}(mM/\varepsilon)$ fits in a machine word (this is due to the precision considerations in Section 3.2.1). This can be achieved by picking a random polynomial over $\mathbb{F}_q$ of degree $r-1$. Across rows, these polynomials only need to be pairwise-independent. Thus, if the polynomial corresponding to row i is P_i, we can set $P_i = Ai + B$, where A, B are independent random polynomials of degree $r-1$ over $\mathbb{F}_q$. It's easy to check the P_i are now pairwise independent (view the coefficient vectors of A, B as elements of the field $\mathbb{F}_{q^r}$, then note we're just evaluating a random degree-1 polynomial over this field to get the P_i). Thus, to evaluate our k different hash functions, rather than spending $\Theta(r)$ time for each of k rows, we just evaluate A, B each in $\Theta(r)$ time, then spend an additional $O(1)$ time per row for a total of $O(k + r)$ time

3.2.1 Alterations for handling limited precision

In this section, we deal with the precision issues mentioned in the proof of Theorem 32.

We deal with rounding errors first. We will pick some number $\beta = \Theta(\varepsilon/m)$. We round each $S_{i,j}$ to the nearest multiple of β. This means that we only need to store the y_i to a precision of β. This does produce an error in these values of size at most $\|x\|_1\beta \leq \|x\|_0\|x\|_\infty\beta \leq m\|x\|_p\beta = \Theta(\varepsilon\|x\|_p)$. Changing all entries in a vector by $\varepsilon\|x\|_p$ cannot change the median by more than $\varepsilon\|x\|_p$.

Next we need to determine how to sample from these continuous distributions. It was shown by [28], and also used in [67], that a p-stable random variable X can be generated by taking θ uniform in $[-\pi/2, \pi/2]$, t uniform in $[0, 1]$ and letting

$$X = f(t, \theta) = \frac{\sin(p\theta)}{\cos^{1/p}(\theta)} \cdot \left(\frac{\cos(\theta(1-p))}{\log(1/t)}\right)^{(1-p)/p}.$$

We would like to know how much of an error is introduced by using values of t and θ only accurate to within β'. This error is at most β' times the derivative of f. This derivative is not large except when θ or $(1-p)\theta$ is close to $\pm\pi/2$, or when t is close to 0 or 1. Since we only ever need mk different values of $S_{i,j}$, we can assume that with large constant probability we never get a t or θ closer to these values than $O(\varepsilon^2/m)$. In such a case the derivative will be bounded by $(m/\varepsilon)^{O(1)}$. Therefore, if we choose t and θ with a precision of $(\varepsilon/m)^{O(1)}$, we introduce error at most β in producing X.

3.2.2 Approximating the median of $|\mathcal{D}_p|$

The purpose of this section is to show that there is an efficient uniform algorithm for approximating median($|\mathcal{D}_p|$) to within relative error $1 \pm \varepsilon$, given $0 < \varepsilon < 1/2$ and $0 < p < 2$ as input. The method described here is certainly not the most efficient way of performing this task, but our goal is just to show that the task is possible. For remarks on efficiency, see Remark 40.

Let φ_p be the density of the p-stable distribution $\mathcal{D}_p$ with characteristic function $e^{-|t|^p}$. That is, if $X \sim \mathcal{D}_p$, then[1]

$$\hat{\varphi}_p(t) = \mathbf{E}[e^{itX}] = e^{-|t|^p}, \qquad \varphi_p(x) = \frac{1}{2\pi} \cdot \int_{\mathbb{R}} e^{-|t|^p} e^{-itx} dt.$$

We provide an explicit and efficiently computable function φ_p^- which is strictly positive for sufficiently large inputs such that for $X \sim \mathcal{D}_p$,

$$\forall x \in \mathbb{R},\ \varphi_p^-(x) \leq \varphi_p(x). \tag{3.17}$$

Computing φ_p^- and other quantities in this section requires approximating the Γ function, which can be done using Lanczos approximation [86]. Theorem 42 gives a good φ_p^-, which is $\Theta(\varphi_p)$ as $x \to \infty$.

With such φ_p^- in hand, there is a simple algorithm to approximate the median of $|\mathcal{D}_p|$ to within relative error ε with success probability $3/4$. First, since φ_p is unimodal with mode zero (see [127]), we have that

$$\|\varphi_p\|_\infty = \varphi_p(0) = \frac{1}{2\pi p}\Gamma(1/p).$$

This implies a lower bound on the median of

$$x_{\text{med}}^- = \frac{1}{4} \cdot \frac{1}{\varphi_p(0)} = \frac{\pi p}{2\Gamma(1/p)}.$$

Thus, an ε'-additive approximation to the median x_{med} of $|\mathcal{D}_p|$ for $\varepsilon' = \varepsilon \cdot x_{\text{med}}^-$ is a relative-error $(1 \pm \varepsilon)$ approximation.

We now discuss how to obtain such an additive approximation. Note x_{med} is such that the p-stable distribution contains $1/4$ probability mass in the interval $[0, x_{\text{med}}]$. We first try to identify an interval containing strictly more than $1/4$ probability mass. We do this iteratively by guessing i as the right endpoint of the interval in iteration i, and verify the guess with a simple randomized algorithm having success probability $1/2^{i+3}$: we take $\Theta(\epsilon^{-2} \cdot i)$ samples for any constant $\epsilon > 1/3 - 1/4$ and check whether a $1/3 \pm \epsilon$ fraction landed in the interval. This works with the desired success probability by the Chernoff-Hoeffding bound. Now after having identified an i^* such that $[0, i^*]$ contains strictly larger than $1/4$ probability mass, we know that x_{med} lies in the

[1] For convenience, in this section we define $\hat{f}(t)$ as $\int_{\mathbb{R}} f(x) e^{-itx} dx$, unlike in Section 3.1 which had an extra factor of $(\sqrt{2\pi})^{-1}$.

interval $[x^-_{\text{med}}, i^*]$. Also, due to unimodality, we know

$$\varphi_p(x) \geq \varphi_p(i^*) \geq \varphi_p^-(i^*)$$

in this interval, and where we can compute the last term efficiently. Thus, by Chebyshev's equality it suffices to take the median of $8 \cdot (\varepsilon' \cdot \varphi_p^-(i^*))^{-2}$ samples. Our total error probability is then at most $\sum_{i=1}^{\infty} 2^{-(i+3)} + 1/8 \leq 1/4$.

Remark 40. The running time of the above algorithm is $O(1/\varepsilon^2)$ (with a hidden constant depending on p) to obtain a $(1 \pm \varepsilon)$-approximation of $\mathrm{median}(|\mathcal{D}_p|)$, but it should be possible to accomplish this task using $\mathrm{polylog}(1/\varepsilon)$ time. The above algorithm can be used with constant ε to first obtain a constant-factor approximation to the median, which can then be used as an initial guess for an iterative numerical method. In each iteration one would then have to numerically approximate an integral to approximate the cumulative distribution function of $\mathcal{D}_p$. We do not delve into the details since our algorithm must already spend $O(1/\varepsilon^2)$ pre-processing time initializing counters, and thus the approach described above does not dominate the pre-processing cost by more than a constant factor.

We now give the function φ_p^-. Our approach contains no new ideas and essentially follows an approach of Oleszkiewicz in [100, Lemma 1], which he used to derive asymptotic behavior of the p-pseudostable distribution for $p > 2$. In fact, we were made aware that this approach should be extendable to $p < 2$ in discussions with Oleszkiewicz, and we thank him for this suggestion. This section also benefited from discussions with Daniel Kane.

We first need the following lemma (see [100, Lemma 2] for a proof).

Lemma 41. *For any $p > 0$,*

$$\int_{\mathbb{R}} |t|^p e^{-|t|} e^{-itx} dt = \frac{2\Gamma(p+1)}{(1+x^2)^{\frac{p+1}{2}}} \cdot \cos\left(\frac{p+1}{2}\pi - (p+1)\arcsin\frac{1}{\sqrt{1+x^2}}\right)$$

The proof of following theorem then closely follows that of [100, Lemma 1]. We make no attempt whatsoever to optimize constants.

Theorem 42. *For $0 < p < 2$, define φ_p^+ and φ_p^- as follows:*

$$\varphi_p^-(x) \stackrel{def}{=} \frac{\Gamma(p+1)}{\pi \cdot x^{p+1}} \cdot \sin\left(\frac{\pi}{2}p\right) - \sum_{j=2}^{\lceil 3/p \rceil} \frac{2^{(jp+1)/2}}{j!} \cdot \frac{\Gamma(jp+1)}{\pi \cdot x^{jp+1}} - \sum_{j=1}^{\lceil 3/p \rceil} \frac{2^{(jp+2)/2}}{j!} \cdot \frac{\Gamma(jp+2)}{\pi \cdot x^{jp+2}}$$
$$- \frac{1}{2}\sum_{j=1}^{\lceil 3/p \rceil} \frac{2^{(jp+3)/2}}{j!} \cdot \frac{\Gamma(jp+3)}{\pi \cdot x^{jp+3}} \pm \left|\frac{a_{1,p} + a_{2,p}}{x^3}\right|$$

and

$$\varphi_p^+(x) \stackrel{def}{=} 2^{(p+1)/2} \cdot \frac{\Gamma(p+1)}{\pi \cdot x^{p+1}} \cdot \sin\left(\frac{\pi}{2}p\right) + \sum_{j=2}^{\lceil 3/p \rceil} \frac{2^{(jp+1)/2}}{j!} \cdot \frac{\Gamma(jp+1)}{\pi \cdot x^{jp+1}}$$
$$+ \sum_{j=1}^{\lceil 3/p \rceil} \frac{2^{(jp+2)/2}}{j!} \cdot \frac{\Gamma(jp+2)}{\pi \cdot x^{jp+2}} + \frac{1}{2} \sum_{j=1}^{\lceil 3/p \rceil} \frac{2^{(jp+3)/2}}{j!} \cdot \frac{\Gamma(jp+3)}{\pi \cdot x^{jp+3}} \pm \left| \frac{a_{1,p} + a_{2,p}}{x^3} \right|$$

for

$$a_{1,p} \stackrel{def}{=} \frac{80}{p} \cdot \Gamma(6 + 1/p) + 926 \cdot \left\lceil \frac{3}{p} \right\rceil \cdot 6!, \quad a_{2,p} \stackrel{def}{=} 60e + \left\lceil \frac{3}{p} \right\rceil \cdot \left[7^3 \cdot \frac{5}{2} + 8^3 \cdot \frac{3e}{2} \right]$$

Then for any $x \geq 1$, $\varphi_p^-(x) \leq \varphi_p(x) \leq \varphi_p^+(x)$.

Proof. Define

$$h_p(t) = e^{-|t|^p} - \sum_{j=1}^{\lceil 3/p \rceil} \frac{(-1)^j}{j!} \cdot \left(|t|^{jp} e^{-|t|} + |t|^{jp+1} e^{-|t|} + \frac{1}{2} \cdot |t|^{jp+2} e^{-|t|} \right).$$

By expanding the above exponentials as series, we find that h_p is thrice-differentiable. Thus, by the Riemann-Lebesgue theorem,

$$\lim_{x \to \infty} |x^3 \cdot \hat{h}_p(x)| = \lim_{t \to \infty} |h_p^{(3)}(t)| = 0,$$

so that

$$\begin{aligned} 2\pi \cdot \varphi_p(x) &= \hat{h}_p(x) - \sum_{j=1}^{\lceil 3/p \rceil} \frac{(-1)^j}{j!} \cdot \left(\int_{\mathbb{R}} |t|^{jp} e^{-|t|} e^{itx} + \int_{\mathbb{R}} |t|^{jp+1} e^{-|t|} e^{itx} \right. \\ &\qquad \left. + \frac{1}{2} \cdot \int_{\mathbb{R}} |t|^{jp+2} e^{-|t|} e^{itx} \right) \\ &= o(1/x^3) - \sum_{j=1}^{\lceil 3/p \rceil} \frac{(-1)^j}{j!} \cdot \left(\int_{\mathbb{R}} |t|^{jp} e^{-|t|} e^{itx} + \int_{\mathbb{R}} |t|^{jp+1} e^{-|t|} e^{itx} \right. \qquad (3.18) \\ &\qquad \left. + \frac{1}{2} \cdot \int_{\mathbb{R}} |t|^{jp+2} e^{-|t|} e^{itx} \right). \end{aligned}$$

As we will see, the dominant term in the above summation comes from the $|t|^{jp} e^{-|t|}$ term for $j = 1$, and all other terms are asymptotically smaller. From this point we simply need to make the constant in the $o(1/x^3)$ term in Eq. (3.18) explicit, as well as to plug in Lemma 41 to bound the terms in the summation.

We first deal with the $o(1/x^3)$ term. We have

$$
\begin{aligned}
|x^3 \cdot \hat{h}_p(x)| &= \left| \int_{\mathbb{R}} \hat{h}_p^{(3)}(t) e^{-itx} dt \right| \\
&\leq \int_{\mathbb{R}} \left| \hat{h}_p^{(3)}(t) \right| dt.
\end{aligned}
$$

Then, for $t \neq 0$, and defining $q = jp$ in the inner sum for notational convenience,

$$
\begin{aligned}
\hat{h}_p^{(3)}(t) = (-p(p-1)(p-2)|t|^{p-3} + 3p^2(p-1)|t|^{2p-3} - p^3|t|^{3p-3}) \cdot \operatorname{sign}(t) e^{-|t|^p} \\
- \sum_{j=1}^{\lceil 3/p \rceil} \frac{(-1)^j}{j!} \cdot (q(q-1)(q-2)|t|^{q-3} + (q^3 - 3q^2 + 2q)|t|^{q-2} \\
- \frac{1}{2}(5q^2 - 3q - 2)|t|^{q-1} - \frac{1}{2}(3q^2 + 3q + 1)|t|^q \\
+ \frac{1}{2}(3q + 4)|t|^{q+1} - \frac{1}{2}|t|^{q+2}) \cdot \operatorname{sign}(t) e^{-|t|}
\end{aligned}
$$

Thus for $|t| > 1$, using that $p < 2$ we have

$$
|\hat{h}_p^{(3)}(t)| \leq 40|t|^6 e^{-|t|^p} + 463 \cdot \left\lceil \frac{3}{p} \right\rceil \cdot |t|^6 e^{-|t|}.
$$

By substitution,

$$
\begin{aligned}
\int_{|t|>1} |\hat{h}_p^{(3)}(t)| dt &\leq 40 \int_{\mathbb{R}} |t|^6 e^{-|t|^p} dt + 463 \cdot \left\lceil \frac{3}{p} \right\rceil \int_{\mathbb{R}} |t|^6 e^{-|t|} dt \\
&= \frac{40}{p} \int_{\mathbb{R}} |u|^{\frac{1}{p}+5} e^{-|u|} du + 463 \cdot \left\lceil \frac{3}{p} \right\rceil \int_{\mathbb{R}} |t|^6 e^{-|t|} dt \\
&= \frac{80}{p} \cdot \Gamma(6 + 1/p) + 926 \cdot \left\lceil \frac{3}{p} \right\rceil \cdot 6! \\
&\stackrel{\text{def}}{=} a_{1,p}
\end{aligned}
$$

Next, to deal with the contribution to $\|\hat{h}_p^{(3)}\|_1$ from $|t| < 1$, consider the series expansion

$$
\begin{aligned}
\hat{h}_p^{(3)}(t) = \sum_{j=1}^{\infty} \frac{(-1)^j}{j!} \cdot jp(jp-1)(jp-2) \cdot \operatorname{sign}(t) \cdot |t|^{jp-3} \\
+ \sum_{j=1}^{\lceil 3/p \rceil} \left(\sum_{k=0}^{\infty} \frac{(-1)^k}{k!} \cdot (k+jp)(k+jp-1)(k+jp-2) \cdot \operatorname{sign}(t) \cdot |t|^{k+jp-3} \right. \\
+ \sum_{k=0}^{\infty} \frac{(-1)^k}{k!} \cdot (k+jp+1)(k+jp)(k+jp-1) \cdot \operatorname{sign}(t) \cdot |t|^{k+jp-2}
\end{aligned}
$$

$$+\frac{1}{2}\sum_{k=0}^{\infty}\frac{(-1)^k}{k!}\cdot(k+jp+2)(k+jp+1)(k+jp)\cdot\operatorname{sign}(t)\cdot|t|^{k+jp-1}\Bigg)$$

Note h_p was defined so that all terms in its series expansion have $|t|$ with an exponent larger than 3 via cancellations, so that all terms in the series expansion of its third derivative have $|t|$ raised to a nonnegative exponent. Thus, for $|t|<1$, $\hat{h}_p^{(3)}(t)$ is at most the sum of magnitudes of its series expansion, which is at most

$$\begin{aligned}
\sum_{j=1}^{\infty}\frac{1}{j!}\cdot|jp(jp-1)(jp-2)| + \sum_{j=1}^{\lceil 3/p\rceil}\Bigg(&\sum_{k=0}^{\infty}\frac{1}{k!}\cdot|(k+jp)(k+jp-1)(k+jp-2)| \\
&+\sum_{k=0}^{\infty}\frac{1}{k!}\cdot|(k+jp+1)(k+jp)(k+jp-1)| \\
&+\frac{1}{2}\sum_{k=0}^{\infty}\frac{1}{k!}\cdot(k+jp+2)(k+jp+1)(k+jp)\Bigg) \\
&\le 60e+\left\lceil\frac{3}{p}\right\rceil\cdot\left[7^3\cdot\left(\frac{1}{0!}+\frac{1}{1!}+\frac{1}{2!}\right)+8^3\cdot\frac{3e}{2}\right] \\
&\stackrel{\text{def}}{=} a_{2,p}
\end{aligned}$$

Now, plugging back into Eq. (3.18), applying Lemma 41, and using that $\cos(\theta)=\sin(\pi/2-\theta)$,

$$\begin{aligned}
2\pi\varphi_p(x) &= \sum_{j=1}^{\lceil 3/p\rceil}\frac{(-1)^{j+1}}{j!}\cdot\frac{2\Gamma(jp+1)}{(1+x^2)^{\frac{jp+1}{2}}}\cdot\sin\left(\frac{\pi}{2}jp-(jp+1)\arcsin\frac{1}{\sqrt{1+x^2}}\right) \\
&\quad\pm\sum_{j=1}^{\lceil 3/p\rceil}\frac{1}{j!}\cdot\frac{2\Gamma(jp+2)}{(1+x^2)^{\frac{jp+2}{2}}}\pm\frac{1}{2}\sum_{j=1}^{\lceil 3/p\rceil}\frac{1}{j!}\cdot\frac{2\Gamma(jp+3)}{(1+x^2)^{\frac{jp+3}{2}}}\pm\left|\frac{a_{1,p}a_{2,p}}{x^3}\right| \\
&= \frac{2\Gamma(p+1)}{(1+x^2)^{\frac{p+1}{2}}}\cdot\sin\left(\frac{\pi}{2}p-(p+1)\arcsin\frac{1}{\sqrt{1+x^2}}\right)\pm\sum_{j=2}^{\lceil 3/p\rceil}\frac{1}{j!}\cdot\frac{2\Gamma(jp+1)}{(1+x^2)^{\frac{jp+1}{2}}} \\
&\quad\pm\sum_{j=1}^{\lceil 3/p\rceil}\frac{1}{j!}\cdot\frac{2\Gamma(jp+2)}{(1+x^2)^{\frac{jp+2}{2}}}\pm\frac{1}{2}\sum_{j=1}^{\lceil 3/p\rceil}\frac{1}{j!}\cdot\frac{2\Gamma(jp+3)}{(1+x^2)^{\frac{jp+3}{2}}}\pm\left|\frac{a_{1,p}a_{2,p}}{x^3}\right|
\end{aligned}$$

We thus have that

$$\begin{aligned}
&\frac{2\Gamma(p+1)}{(1+x^2)^{\frac{p+1}{2}}}\cdot\sin\left(\frac{\pi}{2}p\right)-\sum_{j=2}^{\lceil 3/p\rceil}\frac{1}{j!}\cdot\frac{2\Gamma(jp+1)}{(1+x^2)^{\frac{jp+1}{2}}}-\sum_{j=1}^{\lceil 3/p\rceil}\frac{1}{j!}\cdot\frac{2\Gamma(jp+2)}{(1+x^2)^{\frac{jp+2}{2}}} \\
&-\frac{1}{2}\sum_{j=1}^{\lceil 3/p\rceil}\frac{1}{j!}\cdot\frac{2\Gamma(jp+3)}{(1+x^2)^{\frac{jp+3}{2}}}\pm\left|\frac{a_{1,p}+a_{2,p}}{x^3}\right|
\end{aligned}$$

$$\leq 2\pi\varphi_p(x) \leq \frac{2\Gamma(p+1)}{(1+x^2)^{\frac{p+1}{2}}} \cdot + \sum_{j=2}^{\lceil 3/p \rceil} \frac{1}{j!} \cdot \frac{2\Gamma(jp+1)}{(1+x^2)^{\frac{jp+1}{2}}}$$
$$+ \sum_{j=1}^{\lceil 3/p \rceil} \frac{1}{j!} \cdot \frac{2\Gamma(jp+2)}{(1+x^2)^{\frac{jp+2}{2}}} + \frac{1}{2} \sum_{j=1}^{\lceil 3/p \rceil} \frac{1}{j!} \cdot \frac{2\Gamma(jp+3)}{(1+x^2)^{\frac{jp+3}{2}}} + \left| \frac{a_{1,p} + a_{2,p}}{x^3} \right|$$

Then, since the function $f(y) = (1+y^2)^{(q+1)/2}/y^{q+1}$ is strictly decreasing for any $q > 0$, we have for any $x > 1$ that

$$\frac{2\Gamma(p+1)}{x^{p+1}} \cdot \sin\left(\frac{\pi}{2}p\right) - \sum_{j=2}^{\lceil 3/p \rceil} \frac{2^{(jp+1)/2}}{j!} \cdot \frac{2\Gamma(jp+1)}{x^{jp+1}} - \sum_{j=1}^{\lceil 3/p \rceil} \frac{2^{(jp+2)/2}}{j!} \cdot \frac{2\Gamma(jp+2)}{x^{jp+2}}$$
$$- \frac{1}{2} \sum_{j=1}^{\lceil 3/p \rceil} \frac{2^{(jp+3)/2}}{j!} \cdot \frac{2\Gamma(jp+3)}{x^{jp+3}} \pm \left| \frac{a_{1,p} + a_{2,p}}{x^3} \right|$$
$$\leq 2\pi\varphi_p(x) \leq 2^{(p+1)/2} \cdot \frac{2\Gamma(p+1)}{x^{p+1}} \cdot + \sum_{j=2}^{\lceil 3/p \rceil} \frac{2^{(jp+1)/2}}{j!} \cdot \frac{2\Gamma(jp+1)}{x^{jp+1}}$$
$$+ \sum_{j=1}^{\lceil 3/p \rceil} \frac{2^{(jp+2)/2}}{j!} \cdot \frac{2\Gamma(jp+2)}{x^{jp+2}} + \frac{1}{2} \sum_{j=1}^{\lceil 3/p \rceil} \frac{2^{(jp+3)/2}}{j!} \cdot \frac{2\Gamma(jp+3)}{x^{jp+3}} + \left| \frac{a_{1,p} + a_{2,p}}{x^3} \right|$$

■

3.3 Fast Moment Estimation in Optimal Space

In this section we describe a more sophisticated algorithm for F_p estimation in data streams, which also uses optimal space but has a faster $O(\log^2(1/\varepsilon)\log\log(1/\varepsilon))$ update time. We first give an overview of how this algorithm works.

At the highest level, our faster algorithm in this section splits the coordinates $i \in [d]$ into a list L of *heavy hitters*, and the remaining *light* coordinates. A *ϕ-heavy hitter* with respect to F_p is a coordinate i such that $|x_i|^p \geq \phi\|x\|_p^p$.

To estimate the contribution of the light elements to F_p, we use $R = \Theta(1/\varepsilon^2)$ sketches $B_1, \ldots, B_R$, each with the property that it can be updated quickly and can provide a nearly unbiased estimate of the F_p contribution of items mapped to it, and with low variance. Upon receiving an update to x_i in the stream, we feed the update to $B_{h(i)}$ for some hash function $h : [d] \to [R]$. At the end of the stream, the estimate of the contribution to F_p from light elements is $(R/(R - |h(L)|)) \cdot \sum_{j \notin h(L)} \mathsf{Est}_p(B_j)$, where Est_p is a derandomized variant of Li's geometric mean estimator [87]. We could use Li's geometric mean estimator itself, and our analysis would only need that this estimator is unbiased and has a good variance bound. Unfortunately, the analysis of the geometric mean estimator assumes that the p-stable random variables being used are all independent. We show that a slight variant of Li's geometric mean estimator has bounded variance and is approximately unbiased (to within relative error ε)

even when the associated p-stable random variables are only r-wise independent for $r = \Omega(1/\varepsilon^p)$. This variant allows us to avoid Nisan's pseudorandom generator [99] and thus achieve optimal space. We evaluate the necessary r-wise independent hash function quickly by a combination of buffering and fast multipoint evaluation of a collection of pairwise independent polynomials. Our proof that bounded independence suffices uses FT-mollification.

In particular, analyzing Li's geometric mean estimator under bounded independence requires showing that $\mathbf{E}[f(\sum_i Q_i x_i)]$ is approximately preserved for r-wise independent p-stable random variables Q_i for $f(x) = |x|^{1/t}$. We express $\mathbf{E}[f(x)] = \int_0^\infty f(x)\varphi_p(x)dx$ as $\int_0^\infty f'(x)(1-\Phi_p(x))dx$ via integration by parts, where φ_p is the density function of the absolute value of the p-stable distribution, and Φ_p is the corresponding cumulative distribution function. We then note $1 - \Phi_p(x) = \mathbf{Pr}[|X| \ge x] = \mathbf{E}[I_{[x,\infty)\cup(-\infty,-x]}(X)]$ for X p-stable. We then FT-mollify $I_{[x,\infty)\cup(-\infty,-x]}$, which *is* the indicator function of some set, to write $\mathbf{E}[f(x)]$ as a weighted integral of indicator functions, from which point we can apply a similar proof technique to that in the proof of Theorem 32.

In order to estimate the contribution to F_p from coordinates in L, we develop a novel data structure we refer to as HighEnd. Suppose L contains all the α-heavy hitters, and every index in L is an $(\alpha/2)$-heavy hitter. We would like to compute $\|x_L\|_p^p \pm O(\varepsilon)\cdot\|x\|_p^p$, where $\alpha = \Omega(\varepsilon^2)$. We maintain a matrix of counters $D_{j,k}$ for $(j,k) \in [t]\times[s]$ for $t = O(\log(1/\varepsilon))$ and $s = O(1/\alpha)$. For each $j \in [t]$ we have a hash function $h^j : [d] \to [s]$ and $g^j : [d] \to [r]$ for $r = O(\log(1/\varepsilon))$. The counter $D_{j,k}$ then stores $\sum_{h^j(v)=k} e^{2\pi i g^j(v)/r} x_v$ for $i = \sqrt{-1}$. That is, our data structure is similar to the CountSketch data structure of Charikar, Chen, and Farach-Colton [29], but rather than taking the dot product with a random sign vector in each counter, we take the dot product with a vector whose entries are random complex roots of unity. At the end of the stream, our estimate of the F_p-contribution from heavy hitters is

$$\mathbf{Re}\left[\sum_{w\in L}\left(\frac{3}{t}\sum_{k=1}^{t/3} e^{-2\pi i g^{j(w,k)}(w)/r}\cdot \mathrm{sign}(x_w)\cdot D_{j(w,k),h^{j(w,k)}(w)}\right)^p\right].$$

The choice to use complex roots of unity is to ensure that our estimator is approximately unbiased, stemming from the fact that the real part of large powers of roots of unity is still 0 in expectation. Here $\mathbf{Re}[z]$ denotes the real part of z, and $j(w,k)$ denotes the kth smallest value $b \in [t]$ such that h^b isolates w from the other $w' \in L$ (if fewer than $t/3$ such b exist, we fail).

For related problems, e.g., estimating F_p for $p > 2$, using complex roots of unity leads to sub-optimal bounds [53]. Moreover, it seems that "similar" algorithms using sign variables in place of roots of unity do not work, as they have a constant factor bias in their expectation for which it is unclear how to remove. Our initial intuition was that an algorithm using p-stable random variables would be necessary to estimate the contribution to F_p from the heavy hitters. However, such approaches we explored suffered from too large a variance.

In parallel we must run an algorithm we develop to *find* the heavy hitters. Unfortunately, this algorithm, as well as HighEnd, use suboptimal space. To overcome this, we actually use a list of ϵ^2-heavy hitters for $\epsilon = \varepsilon \cdot \log(1/\varepsilon)$. This then improves the space, at the expense of increasing the variance of LightEstimator. We then run $O((\epsilon/\varepsilon)^2)$ pairwise independent instantiations of LightEstimator in parallel and take the average estimate, to bring the variance down. This increases some part of the update time of LightEstimator by a $\log^2(1/\varepsilon)$ factor, but this term turns out to anyway be dominated by the time to evaluate various hash functions. Though, even in the extreme case of balancing with $\epsilon = 1$, our algorithm for finding the heavy hitters algorithm requires $\Omega(\log d \log(mM))$ space, which is suboptimal. We remedy this by performing a dimensionality reduction down to dimension $\mathrm{poly}(1/\varepsilon)$ via hashing and dot products with random sign vectors. We then apply HighEnd to estimate the contribution from heavy hitters in this new vector, and we show that with high probability the correctness of our overall algorithm is still maintained.

3.3.1 Estimating the contribution from heavy hitters

Before giving our algorithm HighEnd for estimating $\|x_L\|_p^p$, we first give a few necessary lemmas and theorems.

The following theorem gives an algorithm for finding the ϕ-heavy hitters with respect to F_p. This algorithm uses the dyadic interval idea of [34] together with a black-box reduction of the problem of finding F_p heavy hitters to the problem of estimating F_p. Our proof is in Section 3.3.5. We note that our data structure both improves and generalizes that of [56], which gave an algorithm with slightly worse bounds that only worked in the case $p = 1$.

Theorem 43. *There is an algorithm* $\mathsf{F_pHH}$ *satisfying the following properties. Given* $0 < \phi < 1$ *and* $0 < \delta < 1$*, with probability at least* $1 - \delta$*,* $\mathsf{F_pHH}$ *produces a list* L *such that* L *contains all* ϕ*-heavy hitters and does not contain indices which are not* $\phi/2$*-heavy hitters. For each* $i \in L$*, the algorithm also outputs* $\mathrm{sign}(x_i)$*, as well as an estimate* $\tilde{x}_i$ *of* x_i *satisfying* $\tilde{x}_i^p \in [(6/7)|x_i|^p, (9/7)|x_i|^p]$*. Its space usage is* $O(\phi^{-1}\log(\phi d)\log(dmM)\log(\log(\phi d)/(\delta\phi)))$*. Its update time is* $O(\log(\phi d) \cdot \log(\log(\phi d)/(\delta\phi))$*. Its reporting time is* $O(\phi^{-1}(\log(\phi d) \cdot \log(\log(\phi d)/(\delta\phi))))$*.*

The following moment bound can be derived by integrating Bernstein's inequality (see [39, Theorem 1.2]).[2]

Lemma 44. *Let* $X_1, \ldots, X_n$ *be such that* X_i *has expectation* μ_i *and variance* σ_i^2*, and* $X_i \leq K$ *almost surely. Then if the* X_i *are* ℓ*-wise independent for some even integer* $\ell \geq 2$*,*

$$\mathbf{E}\left[\left(\sum_{i=1}^{n} X_i - \mu\right)^{\ell}\right] \leq 2^{O(\ell)} \cdot \left(\left(\sigma\sqrt{\ell}\right)^{\ell} + (K\ell)^{\ell}\right),$$

[2] A tail bound can be converted into a moment bound since $\mathbf{E}[|Z|^{\ell}] = \ell \cdot \int_0^{\infty} z^{\ell-1}\mathbf{Pr}[|Z| \geq z]dz$, by integration by parts.

where $\mu = \sum_i \mu_i$ *and* $\sigma^2 = \sum_i \sigma_i^2$. *In particular,*

$$\mathbf{Pr}\left[\left|\sum_{i=1}^n X_i - \mu\right| \geq \lambda\right] \leq 2^{O(\ell)} \cdot \left(\left(\sigma\sqrt{\ell}/\lambda\right)^\ell + (K\ell/\lambda)^\ell\right),$$

by Markov's inequality on the random variable $(\sum_i X_i - \mu)^\ell$.

Lemma 45 (Khintchine inequality [61])**.** *For* $x \in \mathbb{R}^n$, $t \geq 2$, *and uniformly random* $z \in \{-1,1\}^n$, $\mathbf{E}_z[|\langle x, z\rangle|^t] \leq \|x\|_2^t \cdot \sqrt{t}^t$.

In the following lemma, and henceforth in this section, i denotes $\sqrt{-1}$.

Lemma 46. *Let* $x \in \mathbb{R}^n$ *be arbitrary. Let* $z \in \{e^{2\pi i/r}, e^{2\pi i\cdot 2/r}, e^{2\pi i\cdot 3/r}, \ldots, e^{2\pi i\cdot r/r}\}^n$ *be a random such vector for* $r \geq 2$ *an even integer. Then for* $t \geq 2$ *an even integer,* $\mathbf{E}_z[|\langle x, z\rangle|^t] \leq \|x\|_2^t \cdot 2^{t/2}\sqrt{t}^t$.

Proof. Since x is real, $|\langle x, z\rangle|^2 = \left(\sum_{j=1}^n \mathbf{Re}[z_j]\cdot x_j\right)^2 + \left(\sum_{j=1}^n \mathbf{Im}[z_j]\cdot x_j\right)^2$. Then by Minkowski's inequality,

$$\begin{aligned}
\mathbf{E}[|\langle x, z\rangle|^t] &= \mathbf{E}\left[\left|\left(\sum_{j=1}^n \mathbf{Re}[z_j]\cdot x_j\right)^2 + \left(\sum_{j=1}^n \mathbf{Im}[z_j]\cdot x_j\right)^2\right|^{t/2}\right] \\
&\leq \left(2\cdot\max\left\{\mathbf{E}\left[\left(\sum_{j=1}^n \mathbf{Re}[z_j]\cdot x_j\right)^t\right]^{2/t}, \mathbf{E}\left[\left(\sum_{j=1}^n \mathbf{Im}[z_j]\cdot x_j\right)^t\right]^{2/t}\right\}\right)^{t/2} \\
&\leq 2^{t/2}\cdot\left(\mathbf{E}\left[\left(\sum_{j=1}^n \mathbf{Re}[z_j]\cdot x_j\right)^t\right] + \mathbf{E}\left[\left(\sum_{j=1}^n \mathbf{Im}[z_j]\cdot x_j\right)^t\right]\right). \qquad (3.19)
\end{aligned}$$

Since r is even, we may write $\mathbf{Re}[z_j]$ as $(-1)^{y_j}|\mathbf{Re}[z_j]|$ and $\mathbf{Im}[z_j]$ as $(-1)^{y'_j}|\mathbf{Im}[z_j]|$, where $y, y' \in \{-1,1\}^n$ are random sign vectors chosen independently of each other. Let us fix the values of $|\mathbf{Re}[z_j]|$ and $|\mathbf{Im}[z_j]|$ for each $j \in [n]$, considering just the randomness of y and y'. Applying Lemma 45 to bound each of the expectations in Eq. (3.19), we obtain the bound $2^{t/2}\cdot\sqrt{t}^t\cdot(\|b\|_2^t + \|b'\|_2^t) \leq 2^{t/2}\cdot\sqrt{t}^t\cdot(\|b\|_2^2 + \|b'\|_2^2)^{t/2}$ where $b_j = \mathbf{Re}[z_j]\cdot x_j$ and $b'_j = \mathbf{Im}[z_j]\cdot x_j$. But this is just $2^{t/2}\cdot\sqrt{t}^t\cdot\|x\|_2^t$ since $|z_j|^2 = 1$. ∎

The HighEnd data structure

In this section, we assume we know a subset $L \subseteq [d]$ of indices j so that

1. for all j for which $|x_j|^p \geq \alpha\|x\|_p^p$, $j \in L$,
2. if $j \in L$, then $|x_j|^p \geq (\alpha/2)\|x\|_p^p$,
3. for each $j \in L$, we know $\mathrm{sign}(x_j)$.

for some $0 < \alpha < 1/2$ which we know. We also are given some $0 < \varepsilon < 1/2$. We would like to output a value $\|x_L\|_p^p \pm O(\varepsilon)\|x\|_p^p$ with large constant probability. We assume $1/\alpha = O(1/\varepsilon^2)$.

We first define the BasicHighEnd data structure. Put $s = \lceil 4/\alpha \rceil$. We choose a hash function $h : [d] \to [s]$ at random from an r_h-wise independent family for $r_h = \Theta(\log(1/\alpha))$. Also, let $r = \Theta(\log 1/\varepsilon)$ be a sufficiently large even integer. For each $j \in [d]$, we associate a random complex root of unity $e^{2\pi i g(j)/r}$, where $g : [d] \to [r]$ is drawn at random from an r_g-wise independent family for $r_g = r$. We initialize s counters $b_1, \ldots, b_s$ to 0. Given an update of the form (j, v), add $e^{2\pi i g(j)/r} \cdot v$ to $b_{h(j)}$.

We now define the HighEnd data structure. Define $T = \tau \cdot \max\{\log(1/\varepsilon), \log(2/\alpha)\}$ for a sufficiently large constant τ to be determined later. Define $t = 3T$ and instantiate t independent copies of the BasicHighEnd data structure. Given an update (j, v), perform the update described above to each of the copies of BasicHighEnd. We think of this data structure as a $t \times s$ matrix of counters $D_{j,k}$, $j \in [t]$ and $k \in [s]$. We let g^j be the hash function g in the jth independent instantiation of BasicHighEnd, and similarly define h^j. We sometimes use g to denote the tuple $(g^1, \ldots, g^t)$, and similarly for h.

We now define our estimator, but first we give some notation. For $w \in L$, let $j(w, 1) < j(w, 2) < \ldots < j(w, n_w)$ be the set of n_w indices $j \in [t]$ such that w is *isolated* by h^j from other indices in L; that is, indices $j \in [t]$ where no other $w' \in L$ collides with w under h^j.

Event $\mathcal{E}$. Define $\mathcal{E}$ to be the event that $n_w \geq T$ for all $w \in L$.

If $\mathcal{E}$ does not hold, our estimator simply fails. Otherwise, define

$$x_w^* = \frac{1}{T} \cdot \sum_{k=1}^{T} e^{-2\pi i g^{j(w,k)}(w)/r} \cdot \mathrm{sign}(x_w) \cdot D_{j(w,k), h^{j(w,k)}(w)}.$$

If $\mathbf{Re}[x_w^*] < 0$ for any $w \in L$, then we output fail. Otherwise, define

$$\Psi' = \sum_{w \in L} (x_w^*)^p .$$

Our estimator is then $\Psi = \mathbf{Re}[\Psi']$. Note x^* is a complex number. By z^p for complex z, we mean $|z|^p \cdot e^{ip \cdot \arg(z)}$, where $\arg(z) \in (-\pi, \pi]$ is the angle formed by the vector from the origin to z in the complex plane.

A useful random variable

For $w \in L$, we make the definitions

$$y_w \stackrel{\mathrm{def}}{=} \frac{x_w^* - |x_w|}{|x_w|}, \qquad \Phi_w \stackrel{\mathrm{def}}{=} |x_w|^p \cdot \left(\sum_{k=0}^{r/3} \binom{p}{k} \cdot y_w^k \right)$$

as well as $\Phi \stackrel{\text{def}}{=} \sum_{w\in L} \Phi_w$. We assume $\mathcal{E}$ occurs so that the y_w and Φ_w (and hence Φ) are defined. Also, we use the definition $\binom{p}{k} = (\prod_{j=0}^{k-1}(p-j))/k!$ (note p may not be an integer).

Our overall goal is to show that $\Psi = \|x_L\|_p^p \pm O(\varepsilon)\cdot\|x\|_p^p$ with large constant probability. Our proof plan is to first show that $|\Phi - \|x_L\|_p^p| = O(\varepsilon)\cdot\|x\|_p^p$ with large constant probability, then to show that $|\Psi' - \Phi| = O(\varepsilon)\cdot\|x\|_p^p$ with large constant probability, at which point our claim follows by a union bound and the triangle inequality since $|\Psi - \|x_L\|_p^p| \le |\Psi' - \|x_L\|_p^p|$ since $\|x_L\|_p^p$ is real.

Before analyzing Φ, we define the following event.

Event $\mathcal{D}$. Let $\mathcal{D}$ be the event that for all $w \in L$ we have

$$\frac{1}{T^2}\sum_{k=1}^{T}\sum_{\substack{v\notin L\\ h^{j(w,k)}(v)=h^{j(w,k)}(w)}} x_v^2 < \frac{(\alpha\cdot\|x\|_p^p)^{2/p}}{r}.$$

We also define

$$V = \frac{1}{T^2}\sum_{w\in L}\sum_{j=1}^{t}\sum_{\substack{v\notin L\\ h^j(w)=h^j(v)}} |x_w|^{2p-2}\cdot|x_v|^2.$$

Theorem 47. *Conditioned on h, $\mathbf{E}_g[\Phi] = \|x_L\|_p^p$ and $\mathbf{Var}_g[\Phi \mid \mathcal{D}] = O(V)$.*

Proof. By linearity of expectation,

$$\mathbf{E}_g[\Phi] = \sum_{w\in L}|x_w|^p\cdot\left[\sum_{k=0}^{r/3}\binom{p}{k}\mathbf{E}_g[y_w^k]\right] = \sum_{w\in L}|x_w|^p + \sum_{w\in L}|x_w|^p\cdot\sum_{k=1}^{r/3}\binom{p}{r}\mathbf{E}_g\left[y_w^k\right],$$

where we use that $\binom{p}{0} = 1$. Then $\mathbf{E}_g[y_w^k] = 0$ for $k > 0$ by using linearity of expectation and r_g-wise independence, since each summand involves at most $k < r$ rth roots of unity. Hence,

$$\mathbf{E}_g[\Phi] = \sum_{w\in L}|x_w|^p.$$

We now compute the variance. Note that if the g^j were each fully independent, then we would have $\mathbf{Var}_g[\Phi \mid \mathcal{D}] = \sum_{w\in L}\mathbf{Var}_g[\Phi_w \mid \mathcal{D}]$ since different Φ_w depend on evaluations of the g^j on disjoint $v \in [d]$. However, since $r_g > 2r/3$, $\mathbf{E}_g[|\Phi|^2]$ is identical as in the case of full independence of the g^j. We thus have $\mathbf{Var}_g[\Phi \mid \mathcal{D}] =$

$\sum_{w\in L} \mathbf{Var}_g[\Phi_w \mid \mathcal{D}]$ and have reduced to computing $\mathbf{Var}_g[\Phi_w \mid \mathcal{D}]$.

$$\begin{aligned}
\mathbf{Var}_g[\Phi_w \mid \mathcal{D}] &= \mathbf{E}_g[|\Phi_w - \mathbf{E}_g[\Phi_w]|^2 \mid \mathcal{D}] \\
&= |x_w|^{2p} \cdot \mathbf{E}_g\left[\left|\sum_{k=1}^{r/3} \binom{p}{k} y_w^k\right|^2 \mid \mathcal{D}\right] \\
&= |x_w|^{2p} \cdot \left(p^2 \cdot \mathbf{E}_g[|y_w|^2 \mid \mathcal{D}] + \sum_{k=2}^{r/3} O(\mathbf{E}_g[|y_w|^{2k} \mid \mathcal{D}])\right)
\end{aligned}$$

We have

$$\mathbf{E}_g[|y_w|^2 \mid \mathcal{D}] \overset{\text{def}}{=} u_w^2 = \frac{1}{T^2}\sum_{k=1}^{T} \sum_{\substack{v\notin L \\ h^{j(w,k)}(v)=h^{j(w,k)}(w)}} \frac{x_v^2}{x_w^2}, \tag{3.20}$$

so that

$$\sum_{w\in L} p^2 \cdot \mathbf{E}_g[|y_w|^2 \mid \mathcal{D}] \le p^2 V.$$

Eq. (3.20) follows since, conditioned on $\mathcal{E}$ so that y_w is well-defined,

$$\begin{aligned}
\mathbf{E}_g[|y_w|^2] = \frac{1}{T^2 x_w^2} \sum_{k,k'=1}^{T} \sum_{v\notin L} \sum_{v'\notin L} &\mathbf{1}_{h^{j(w,k)}(v)=h^{j(w,k)}(w)} \cdot \mathbf{1}_{h^{j(w,k')}(v')=h^{j(w,k')}(w)} \\
&\times \mathbf{E}[e^{-2\pi i (g^{j(w,k)}(v) - g^{j(w,k')}(v'))/r}] x_v x_{v'}.
\end{aligned}$$

When $j(w,k) \neq j(w,k')$ the above expectation is 0 since the g^j are independent across different j. When $j(w,k) = j(w,k')$ the above expectation is only non-zero for $v = v'$ since $r_g \ge 2$.

We also have for $k \ge 2$ that

$$\mathbf{E}_g[|y_w|^{2k} \mid \mathcal{D}] \le 2^{O(k)} \cdot u_w^{2k} \cdot (2k)^k$$

by Lemma 46, so that

$$\sum_{k=2}^{r/3} \mathbf{E}_g[|y_w|^{2k} \mid \mathcal{D}] = O(u_w^2)$$

since $\mathcal{D}$ holds and so the sum is dominated by its first term. Thus, $\mathbf{Var}_g[\Phi \mid \mathcal{D}] = O(V)$. ■

Lemma 48. $\mathbf{E}_h[V] \le 3\alpha \cdot \|x\|_p^{2p}/(4T)$.

Proof. For any $w \in L$, $v \notin L$, and $j \in [t]$, we have $\mathbf{Pr}_h[h^j(w) = h^j(v)] = 1/s \le \alpha/4$

since $r_h \geq 2$. Thus,

$$\begin{aligned}
\mathbf{E}_h[V] &\leq \frac{\alpha}{4T^2} \sum_{\substack{w \in L \\ v \notin L \\ j \in [t]}} |x_w|^{2p-2} |x_v|^2 \\
&= \frac{3\alpha}{4T} \left(\sum_{w \in L} |x_w|^p |x_w|^{p-2} \right) \left(\sum_{v \notin L} |x_v|^2 \right) \\
&\leq \frac{3\alpha}{4T} \left(\sum_{w \in L} \|x\|_p^p (\alpha \cdot \|x\|_p^p)^{(p-2)/p} \right) \left(\frac{1}{\alpha} (\alpha \cdot \|x\|_p^p)^{2/p} \right) \qquad (3.21) \\
&= \frac{3}{4} \cdot \alpha \cdot \|x\|_p^{2p} / T.
\end{aligned}$$

where Eq. (3.21) used that $\|x_{[d]\setminus L}\|_2^2$ is maximized when $[d]\setminus L$ contains exactly $1/\alpha$ coordinates v each with $|x_v|^p = \alpha \|x\|_p^p$, and that $|x_w|^{p-2} \leq (\alpha \cdot \|x\|_p^p)^{(p-2)/p}$ since $p \leq 2$. ■

Lemma 49. $\mathbf{Pr}_h[\mathcal{E}] \geq 1 - \varepsilon$.

Proof. For any $j \in [t]$, the probability that w is isolated by h^j is at least $1/2$, since the expected number of collisions with w is at most $1/2$ by pairwise independence of the h^j and the fact that $|L| \leq 2/\alpha$ so that $s \geq 2|L|$. If X is the expected number of buckets where w is isolated, the Chernoff bound gives $\mathbf{Pr}_h[X < (1-\epsilon)\mathbf{E}_h[X]] < \exp(-\epsilon^2 \mathbf{E}_h[X]/2)$ for $0 < \epsilon < 1$. The claim follows for $\tau \geq 24$ by setting $\epsilon = 1/3$ then applying a union bound over $w \in L$. ■

Lemma 50. $\mathbf{Pr}_h[\mathcal{D}] \geq 63/64$.

Proof. We apply the bound of Lemma 44 for a single $w \in L$. Define $X_{j,v} = (x_v^2/T^2) \cdot \mathbf{1}_{h^j(v)=h^j(w)}$ and $X = \sum_{j=1}^t \sum_{v \notin L} X_{j,v}$. Note that X is an upper bound for the left hand side of the inequality defining $\mathcal{D}$, and thus it suffices to show a tail bound for X. In the notation of Lemma 44, we have $\sigma^2 \leq (3/(sT^3)) \cdot \|x_{[d]\setminus L}\|_4^4$, $K = (\alpha \cdot \|x\|_p^p)^{2/p}/T^2$, and $\mu = (3/(sT)) \cdot \|x_{[d]\setminus L}\|_2^2$. Since $\|x_{[d]\setminus L}\|_2^2$ and $\|x_{[d]\setminus L}\|_4^4$ are each maximized when there are exactly $1/\alpha$ coordinates $v \notin L$ with $|x_v|^p = \alpha \cdot \|x\|_p^p$,

$$\sigma^2 \leq \frac{3}{4T^3} \cdot (\alpha \cdot \|x\|_p^p)^{4/p}, \qquad \mu \leq \frac{3}{4T} \cdot (\alpha \cdot \|x\|_p^p)^{2/p}.$$

Setting $\lambda = (\alpha \cdot \|x\|_p^p)^{2/p}/(2r)$, noting that $\mu < \lambda$ for τ sufficiently large, and assuming $\ell \leq r_h$ is even, we apply Lemma 44 to obtain

$$\mathbf{Pr}[X \geq 2\lambda] \leq 2^{O(\ell)} \cdot \left(\left(\frac{\sqrt{3}r \cdot \sqrt{\ell}}{T^{3/2}} \right)^\ell + \left(\frac{2r \cdot \ell}{T^2} \right)^\ell \right).$$

By setting τ sufficiently large and $\ell = \log(2/\alpha) + 6$, the above probability is at most $(1/64) \cdot (\alpha/2)$. The lemma follows by a union bound over all $w \in L$, since $|L| \leq 2/\alpha$. ■

We now define another event.

Event $\mathcal{F}$. Let $\mathcal{F}$ be the event that for all $w \in L$ we have $|y_w| < 1/2$.

Lemma 51. $\mathbf{Pr}_g[\mathcal{F} \mid \mathcal{D}] \geq 63/64$.

Proof. $\mathcal{D}$ occurring implies that $u_w \leq \sqrt{1/r} \leq \sqrt{1/(64(\log(2/\alpha)+6)}$ (recall we assume $1/\alpha = O(1/\varepsilon^2)$ and pick $r = \Theta(\log(1/\varepsilon))$ sufficiently large, and u_w is as is defined in Eq. (3.20)), and we also have $\mathbf{E}_g[|y_w|^\ell \mid \mathcal{D}] < u_w^\ell \sqrt{\ell}^\ell 2^\ell$ by Lemma 46. Applying Markov's bound on the random variable $|y_w|^\ell$ for even $\ell \leq r_g$, we have $|y_w|^\ell$ is determined by r_g-wise independence of the g^j, and thus

$$\mathbf{Pr}_g[|y_w| \geq 1/2 \mid \mathcal{D}] < \left(\sqrt{\frac{16\ell}{64(\log(2/\alpha)+6)}}\right)^\ell,$$

which equals $(1/64) \cdot (\alpha/2)$ for $\ell = \log(2/\alpha) + 6$. We then apply a union bound over all $w \in L$. ■

Lemma 52. *If* $\mathcal{F}$ *occurs, then* $|\Psi' - \Phi| < \varepsilon \|x_L\|_p^p$.

Proof. Observe

$$\Psi' = \sum_{w \in L} |x_w|^p \cdot (1 + y_w)^p.$$

We have that $\ln(1+z)$, as a function of z, is holomorphic on the open disk of radius 1 about 0 in the complex plane, and thus $f(z) = (1+z)^p$ is holomorphic in this region since it is the composition $\exp(p \cdot \ln(1+z))$ of holomorphic functions. Therefore, $f(z)$ equals its Taylor expansion about 0 for all $z \in \mathbb{C}$ with $|z| < 1$ (see for example [123, Theorem 11.2]). Then since $\mathcal{F}$ occurs, we can Taylor-expand f about 0 for $z = y_w$ and apply Taylor's theorem to obtain

$$\begin{aligned}
\Psi' &= \sum_{w \in L} |x_w|^p \left(\sum_{k=0}^{r/3} \binom{p}{k} y_w^k \pm O\left(\binom{p}{r/3+1} \cdot |y_w|^{-r/3-1} \right) \right) \\
&= \Phi + O\left(\|x_L\|_p^p \cdot \left(\binom{p}{r/3+1} \cdot |y_w|^{-r/3-1} \right) \right)
\end{aligned}$$

The lemma follows since $\binom{p}{r/3+1} < 1$ and $|y_w|^{-r/3-1} < \varepsilon$ for $|y_w| < 1/2$. ■

Theorem 53. HighEnd *uses* $O(\alpha^{-1} \log(1/\varepsilon) \log(mM/\varepsilon) + O(\log^2(1/\varepsilon) \log d))$ *space. The update time is* $O(\log^2(1/\varepsilon))$. *The reporting time is* $O(\alpha^{-1} \log(1/\varepsilon) \log(1/\alpha))$. *Also,* $\mathbf{Pr}_{h,g}[|\Psi - \|x_L\|_p^p| < O(\varepsilon) \cdot \|x\|_p^p] > 7/8$.

Proof. We first argue correctness. By a union bound, $\mathcal{E}$ and $\mathcal{D}$ hold simultaneously with probability 31/32. By Markov's inequality and Lemma 48, $V = O(\alpha \cdot \|x\|_p^{2p}/T)$ with probability 63/64. We then have by Chebyshev's inequality and Theorem 47 that $|\Phi - \|x_L\|_p^p| = O(\varepsilon) \cdot \|x\|_p^p$ with probability 15/16. Lemma 52 then implies

$|\Psi' - \|x_L\|_p^p| = O(\varepsilon) \cdot \|x\|_p^p$ with probability $15/16 - \mathbf{Pr}[\neg\mathcal{F}] > 7/8$ by Lemma 51. In this case, the same must hold true for Ψ since $\Psi = \mathbf{Re}[\Psi']$ and $\|x_L\|_p^p$ is real.

Next we discuss space complexity. We start with analyzing the precision required to store the counters $D_{j,k}$. Since our correctness analysis conditions on $\mathcal{F}$, we can assume $\mathcal{F}$ holds. We store the real and imaginary parts of each counter $D_{j,k}$ separately. If we store each such part to within precision $\gamma/(2mT)$ for some $0 < \gamma < 1$ to be determined later, then each of the real and imaginary parts, which are the sums of at most m summands from the m updates in the stream, is stored to within additive error $\gamma/(2T)$ at the end of the stream. Let $\tilde{x}_w^*$ be our calculation of x_w^* with such limited precision. Then, each of the real and imaginary parts of $\tilde{x}_w^*$ is within additive error $\gamma/2$ of those for x_w^*. Since $\mathcal{F}$ occurs, $|x_w^*| > 1/2$, and thus $\gamma/2 < \gamma|x_w^*|$, implying $|\tilde{x}_w^*| = (1 \pm O(\gamma))|x_w^*|$. Now we argue $\arg(\tilde{x}_w^*) = \arg(x_w^*) \pm O(\sqrt{\gamma})$. Write $x_w^* = a + ib$ and $\tilde{x}_w^* = \tilde{a} + i\tilde{b}$ with $\tilde{a} = a \pm \gamma/2$ and $\tilde{b} = b \pm \gamma/2$. We have $\cos(\arg(x_w^*)) = a/\sqrt{a^2+b^2}$. Also, $\cos(\arg(\tilde{x}_w^*)) = (a \pm \gamma/2)/((1 \pm O(\gamma))\sqrt{a^2+b^2}) = (1 \pm O(\gamma))\cos(\arg(x_w^*)) \pm O(\gamma) = \cos(\arg(x_w^*)) \pm O(\gamma)$, implying $\arg(\tilde{x}_w^*) = \arg(x_w^*) \pm O(\sqrt{\gamma})$. Our final output is $\sum_{w\in L} |\tilde{x}_w^*|^p \cdot \cos(p \cdot \arg(\tilde{x}_w^*))$. Since cos never has derivative larger than 1 in magnitude, this is $\sum_{w\in L}[(1 \pm O(\gamma))|x_w^*|^p \cos(p \cdot \arg(x_w^*)) \pm O(\sqrt{\gamma}) \cdot (1 \pm O(\gamma))|x_w^*|^p]$. Since $\mathcal{F}$ occurs, $|x_w^*|^p < (3/2)^p \cdot |x_w|^p$, and thus our overall error introduced from limited precision is $O(\sqrt{\gamma} \cdot \|x_L\|_p^p)$, and it thus suffices to set $\gamma = O(\varepsilon^2)$, implying each $D_{j,k}$ requires $O(\log(mM/\varepsilon))$ bits of precision. For the remaining part of the space analysis, we discuss storing the hash functions. The hash functions h^j, g^j each require $O(\log(1/\varepsilon)\log d)$ bits of seed, and thus in total consume $O(\log^2(1/\varepsilon)\log d)$ bits.

Finally we discuss time complexity. To perform an update, for each $j \in [t]$ we must evaluate g^j and h^j then update a counter. Each of g^j, h^j require $O(\log(1/\varepsilon))$ time to evaluate. For the reporting time, we can mark all counters with the unique $w \in L$ which hashes to it under the corresponding h^j (if a unique such w exists) in $|L| \cdot t \cdot r_h = O(\alpha^{-1}\log(1/\varepsilon)\log(1/\alpha))$ time. Then, we sum up the appropriate counters for each $w \in L$, using the Taylor expansion of $\cos(p \cdot \arg(z))$ up to the $\Theta(\log(1/\varepsilon))$th degree to achieve additive error ε. Note that conditioned on $\mathcal{F}$, $\arg(x_w^*) \in (-\pi/4, \pi/4)$, so that $|p \cdot \arg(x_w^*)|$ is bounded away from $\pi/2$ for p bounded away from 2; in fact, one can even show via some calculus that $\arg(x_w^*) \in (-\pi/6, \pi/6)$ when $\mathcal{F}$ occurs by showing that $\cos(\arg(x_w^*)) = \cos(\arg(1 - y_w))$ is minimized for $|y_w| \le 1/2$ when $y_w = 1/4 + i\sqrt{3}/4$. Regardless, additive error ε is relative error $O(\varepsilon)$, since if $|p \cdot \arg(z)|$ is bounded away from $\pi/2$, then $|\cos(p \cdot \arg(z))| = \Omega(1)$. ■

3.3.2 Estimating the contribution from light elements

In this section, we show how to estimate the contribution to F_p from coordinates of x which are not heavy hitters. More precisely, given a list $L \subseteq [d]$ such that $|L| \le 2/\varepsilon^2$ and $|x_i|^p \le \varepsilon^2\|x\|_p^p$ for all $i \notin L$, we describe a subroutine $\mathsf{LightEstimator}$ that outputs a value that is $\|x_{[d]\setminus L}\|_p^p \pm O(\varepsilon) \cdot \|x\|_p^p$ with probability at least $7/8$.

We first need the following theorem, which comes from a derandomized variant of the geometric mean estimator of Li [87]. Our proof is in Section 3.3.4.

Given: $L \subseteq [d]$ at stream end such that $|L| \leq 2/\varepsilon^2$ and $|x_i|^p < \varepsilon^2 \|x\|_p^p$ for all $i \notin L$

1. Initialize $R = 4/\varepsilon^2$ instantiations $B_1, \ldots, B_R$ of structure D_p from Theorem 54. The randomness used across instantiations is pairwise independent.
2. Pick a random hash function $h : [d] \to [R]$ as in Theorem 13 with $z = R$ and $c = 2$.
3. **Update(i, v):** Feed update (i, v) to $B_{h(i)}$.
4. **Estimator:** Let $I = [R] \backslash h(L)$. Output $\frac{R}{|I|} \cdot \sum_{j \in I} \mathsf{Est}_p(B_j)$.

Figure 3-2: LightEstimator, for estimating contribution from non-heavy hitters

Theorem 54. *For any $0 < p < 2$, there is a randomized data structure D_p, and a deterministic algorithm Est_p mapping the state space of the data structure to reals, such that*

1. $\mathbf{E}[\mathsf{Est}_p(D_p(x))] = (1 \pm \varepsilon)\|x\|_p^p$
2. $\mathbf{E}[\mathsf{Est}_p(D_p(x))^2] \leq C_p \cdot \|x\|_p^{2p}$

for some constant $C_p > 0$ depending only on p, and where the expectation is taken over the randomness used by D_p. Aside from storing a length-$O(\varepsilon^{-p} \log(dmM))$ random string, the space complexity is $O(\log(dmM))$. The update time is the time to evaluate a $\Theta(1/\varepsilon^p)$-wise independent hash function over a field of size $\mathrm{poly}(dmM)$, and the reporting time is $O(1)$.

We also need the following algorithm for fast multipoint evaluation of polynomials.

Theorem 55 ([120, Ch. 10])**.** *Let $\mathbf{R}$ be a ring, and let $q \in \mathbf{R}[x]$ be a degree-r polynomial. Then, given distinct $x_1, \ldots, x_r \in \mathbf{R}$, all the values $q(x_1), \ldots, q(x_r)$ can be computed using $O(r \log^2 r \log\log r)$ operations over $\mathbf{R}$.*

The guarantees of the final LightEstimator are then given in Theorem 56.

Theorem 56. *Suppose we are given $0 < \varepsilon < 1/2$, and given a list $L \subseteq [d]$ at the end of the data stream such that $|L| \leq 2/\varepsilon^2$ and $|x_i|^p < \varepsilon^2\|x\|_p^p$ for all $i \notin L$. Then, given access to a randomized data structure satisfying properties (1) and (2) of Theorem 54, the algorithm* LightEstimator *(described in Figure 3-2) satisfies the following: The randomness used by* LightEstimator *can be broken up into a certain random hash function h, and another random string s.* LightEstimator *outputs a value Φ' satisfying $\mathbf{E}_{h,s}[\Phi'] = (1 \pm O(\varepsilon))\|x_{[d]\backslash L}\|_p^p$, and $\mathbf{E}_h[\mathbf{Var}_s[\Phi']] = O(\varepsilon^2\|x\|_p^{2p})$. The space usage is $O(\varepsilon^{-2} \log(dmM))$, the update time is $O(\log^2(1/\varepsilon) \log\log(1/\varepsilon))$, and the reporting time is $O(1/\varepsilon^2)$.*

Proof. The random string s in the theorem statement is the total randomness required by $B_1, \ldots, B_R$. First we show the computation of $\mathbf{E}_{h,s}[\Phi']$. In this proof, we let x_{tail} denote $x_{[d]\setminus L}$ and x_{head} denote x_L. Also, for $S \subseteq [d]$ we let $\mathcal{F}_S$ be the event that the hash family $\mathcal{H}$ we randomly select in Step 3 via Theorem 13 is $|S|$-wise independent when restricted to S.

$\mathbf{E_{h,s}[\Phi'] = (1 \pm O(\varepsilon))\|x_{\text{tail}}\|_p^p}$: For $\rho = 1 - \mathbf{Pr}[\mathcal{F}_L]$,

$$\begin{aligned}
\mathbf{E}_{s,h}\left[\frac{R}{|I|}\cdot\sum_{j\in I}\mathsf{Est}_p(B_j)\right] &= \mathbf{E}_{s,h}\left[\frac{R}{|I|}\cdot\sum_{j\in I}\mathsf{Est}_p(B_j)\ \Big|\ \mathcal{F}_L\right]\cdot\mathbf{Pr}[\mathcal{F}_L] \\
&\quad + \mathbf{E}_{s,h}\left[\frac{R}{|I|}\cdot\sum_{j\in I}\mathsf{Est}_p(B_j)\ \Big|\ \neg\mathcal{F}_L\right]\cdot\mathbf{Pr}[\neg\mathcal{F}_L] \\
&= (1-\rho)\mathbf{E}_{s,h}\left[\frac{R}{|I|}\cdot\sum_{j\in I}\mathsf{Est}_p(B_j)\ \Big|\ \mathcal{F}_L\right] \\
&\quad + \rho\cdot\mathbf{E}_s\left[\mathbf{E}_h\left[\frac{R}{|I|}\cdot\sum_{j\in I}\mathsf{Est}_p(B_j)\ \Big|\ \neg\mathcal{F}_L\right]\right] \\
&= (1-\rho)\mathbf{E}_{s,h}\left[\frac{R}{|I|}\cdot\sum_{j\in I}\mathsf{Est}_p(B_j)\ \Big|\ \mathcal{F}_L\right] \pm (\rho\cdot R)\|x_{\text{tail}}\|_p^p
\end{aligned} \tag{3.22}$$

by Theorem 13. We now compute the above expectation conditioned on I. Let $\mathcal{E}_{I'}$ be the event $I = I'$ for an arbitrary I'. Then,

$$\begin{aligned}
\mathbf{E}_{s,h}\left[\frac{R}{|I'|}\cdot\sum_{j\in I'}\mathsf{Est}_p(B_j)\ \Big|\ \mathcal{F}_L,\mathcal{E}_{I'}\right] &= \frac{R}{|I'|}\cdot\sum_{j\in I'}\mathbf{E}_{s,h}\left[\mathsf{Est}_p(B_j)\ \Big|\ \mathcal{F}_L,\mathcal{E}_{I'}\right] \\
&= \frac{R}{|I'|}\cdot\sum_{j\in I'}\mathbf{E}_h\left[\sum_{i\notin L}\mathbf{1}_{h(i)=j}\cdot|x|_i^p\ \Big|\ \mathcal{F}_L,\mathcal{E}_{I'}\right] \\
&= \frac{R}{|I'|}\cdot\sum_{j\in I'}\sum_{i\notin L}|x|_i^p\cdot\mathbf{Pr}_h\left[h(i)=j\mid\mathcal{F}_L,\mathcal{E}_{I'}\right] \\
&= \frac{R}{|I'|}\cdot\sum_{j\in I'}\sum_{i\notin L}|x|_i^p\cdot\frac{\mathbf{Pr}_h\left[(h(i)=j)\wedge\mathcal{E}_{I'}\mid\mathcal{F}_L\right]}{\mathbf{Pr}[\mathcal{E}_{I'}\mid\mathcal{F}_L]}
\end{aligned} \tag{3.23}$$

Now we note

$$\begin{aligned}
\mathbf{Pr}_h\left[(h(i)=j)\wedge\mathcal{E}_{I'}\mid\mathcal{F}_L\right] &= \mathbf{Pr}_h\left[(h(i)=j)\wedge\mathcal{E}_{I'}\mid\mathcal{F}_{L\cup\{i\}},\mathcal{F}_L\right]\cdot\mathbf{Pr}[\mathcal{F}_{L\cup\{i\}}\mid\mathcal{F}_L] \\
&\quad + \mathbf{Pr}_h\left[(h(i)=j)\wedge\mathcal{E}_{I'}\mid\neg\mathcal{F}_{L\cup\{i\}},\mathcal{F}_L\right]\cdot\mathbf{Pr}[\neg\mathcal{F}_{L\cup\{i\}}\mid\mathcal{F}_L] \\
&= \mathbf{Pr}_h\left[h(i)=j\mid\mathcal{F}_{L\cup\{i\}},\mathcal{F}_L\right]\cdot\mathbf{Pr}_h\left[\mathcal{E}_{I'}\mid\mathcal{F}_L,\mathcal{F}_{L\cup\{i\}},h(i)=j\right]
\end{aligned}$$

$$\times \mathbf{Pr}[\mathcal{F}_{L\cup\{i\}} \mid \mathcal{F}_L]$$
$$+ \mathbf{Pr}[\neg\mathcal{F}_{L\cup\{i\}} \mid \mathcal{F}_L] \cdot \mathbf{Pr}[\mathcal{E}_{I'} \mid \mathcal{F}_L, \neg\mathcal{F}_{L\cup\{i\}}]$$
$$\times \mathbf{Pr}[h(i) = j \mid \neg\mathcal{F}_{L\cup\{i\}}, \mathcal{F}_L, \mathcal{E}_{I'}]$$

Now note a couple things. First, if $\mathcal{F}_{L\cup\{i\}}$ occurs, then $\mathcal{E}_{I'}$ is independent of the event $h(i) = j$. Also, if $\mathcal{F}_L$ occurs, then $\mathcal{E}_{I'}$ is independent of $\mathcal{F}_{L\cup\{i\}}$. Thus, the above equals

$$\begin{aligned}&\mathbf{Pr}_h\left[h(i) = j \mid \mathcal{F}_{L\cup\{i\}}\right] \cdot \mathbf{Pr}_h\left[\mathcal{E}_{I'} \mid \mathcal{F}_L\right] \cdot \mathbf{Pr}[\mathcal{F}_{L\cup\{i\}} \mid \mathcal{F}_L]\\ &\quad + \mathbf{Pr}[\neg\mathcal{F}_{L\cup\{i\}} \mid \mathcal{F}_L] \cdot \mathbf{Pr}[\mathcal{E}_{I'} \mid \mathcal{F}_L] \cdot \mathbf{Pr}[h(i) = j \mid \neg\mathcal{F}_{L\cup\{i\}}, \mathcal{F}_L, \mathcal{E}_{I'}]\end{aligned}$$

Note $\mathbf{Pr}[\neg\mathcal{F}_{L\cup\{i\}} \mid \mathcal{F}_L] \le \mathbf{Pr}[\neg\mathcal{F}_{L\cup\{i\}}]/\mathbf{Pr}[\mathcal{F}_L] = \rho'_i/(1-\rho)$ for $\rho'_i = 1 - \mathbf{Pr}[\mathcal{F}_{L\cup\{i\}}]$. Also, $\mathbf{Pr}[\mathcal{F}_{L\cup\{i\}} \mid \mathcal{F}_L] \ge \mathbf{Pr}[\mathcal{F}_{L\cup\{i\}}]$ since $\mathbf{Pr}[\mathcal{F}_{L\cup\{i\}}]$ is a weighted average of $\mathbf{Pr}[\mathcal{F}_{L\cup\{i\}} \mid \mathcal{F}_L]$ and $\mathbf{Pr}[\mathcal{F}_{L\cup\{i\}} \mid \neg\mathcal{F}_L]$, and the latter is 0. Thus, for some $\rho''_i \in [0, \rho'_i]$ Eq. (3.23) is

$$\frac{R}{|I'|} \cdot \sum_{j\in I'} \sum_{i\notin L} |x|_i^p \cdot \left(\frac{1-\rho''_i}{R} \pm \frac{\rho'_i}{1-\rho}\right) = \|x_{\text{tail}}\|_p^p - \sum_{i\notin L} \rho''_i |x|_i^p \pm \left(\frac{\max_i \rho'_i}{1-\rho}\right) \cdot R \cdot \|x_{\text{tail}}\|_p^p$$

By our setting of $c = 2$ when picking the hash family of Theorem 13 in Step 3, we have $\rho, \rho'_i, \rho''_i = O(\varepsilon^3)$ for all i, and thus $\rho'_i/(1-\rho) \cdot R = O(\varepsilon)$, implying the above is $(1 \pm O(\varepsilon))\|x_{\text{tail}}\|_p^p$. Plugging this in Eq. (3.22) then shows that the desired expectation is $(1 \pm O(\varepsilon))\|x_{\text{tail}}\|_p^p$.

We now bound the expected variance of $(R/|I|) \cdot \sum_{j\in I} \mathsf{Est}_p(B_j)$.

$\mathbf{E_h}\left[\mathbf{Var_s}\left[\frac{\mathbf{R}}{|\mathbf{I}|} \cdot \sum_{\mathbf{j}\in\mathbf{I}} \mathsf{Est}_\mathbf{p}(\mathbf{B_j})\right]\right] = \mathbf{O}(\varepsilon^\mathbf{2} \cdot \|\mathbf{x}\|_\mathbf{p}^\mathbf{2p})$:

For any fixed h, $R/|I|$ is determined, and the $\mathsf{Est}_p(B_j)$ are pairwise independent. Thus, for fixed h,

$$\mathbf{Var}_s\left[\frac{R}{|I|} \cdot \sum_{j\in I} \mathsf{Est}_p(B_j)\right] = \left(\frac{R}{|I|}\right)^2 \cdot \sum_{j\in I} \mathbf{Var}_s[\mathsf{Est}_p(B_j)]$$

First observe that since $|I| \ge R - |L| \ge 2/\varepsilon^2$, for any choice of h we have that $R/|I| \le 2$. Thus, up to a constant factor, the expectation we are attempting to compute is

$$\mathbf{E}_h\left[\mathbf{Var}_s\left[\sum_{j\in I} \mathsf{Est}_p(B_j)\right]\right].$$

For notational convenience, say $\mathsf{Est}_p(B_j) = 0$ if $j \notin I$. Now,

$$\mathbf{E}_h\left[\mathbf{Var}_s\left[\sum_{j\in I} \mathsf{Est}_p(B_j)\right]\right] = \mathbf{E}_h\left[\sum_{j=1}^R \mathbf{1}_{j\in I} \cdot \mathbf{Var}_s\left[\mathsf{Est}_p(B_j)\right]\right]$$

$$
\begin{aligned}
&= \sum_{j=1}^{R} \mathbf{E}_h \left[\mathbf{1}_{j \in I} \cdot \mathbf{Var}_s \left[\mathsf{Est}_p(B_j) \right] \right] \\
&\le \sum_{j=1}^{R} \mathbf{E}_h \left[\mathbf{Var}_s \left[\mathsf{Est}_p(B_j) \right] \;\middle|\; j \in I \right] \\
&= \sum_{j=1}^{R} \mathbf{E}_h \left[C_p \cdot \left(\sum_{i \notin L} \mathbf{1}_{h(i)=j} \cdot |x_i|^p \right)^2 \;\middle|\; j \in I \right],
\end{aligned}
$$

which equals

$$
\begin{aligned}
C_p \cdot \Bigg(&\sum_{j=1}^{R} \sum_{i \notin L} |x_i|^{2p} \cdot \mathbf{Pr}_h[h(i) = j \mid j \in I] \\
&+ \sum_{j=1}^{R} \sum_{\substack{i \neq i' \\ i, i' \notin L}} |x_i|^p |x_{i'}|^p \cdot \mathbf{Pr}_h[(h(i) = j) \wedge (h(i') = j) \mid j \in I] \Bigg)
\end{aligned}
\tag{3.24}
$$

Now consider the quantity $\mathbf{Pr}_h[h(i) = j \mid j \in I]$. Then

$$
\begin{aligned}
\mathbf{Pr}_h[h(i) = j \mid j \in I] &= \mathbf{Pr}_h[h(i) = j \mid \mathcal{F}_{L \cup \{i\}}, j \in I] \cdot \mathbf{Pr}_h[\mathcal{F}_{L \cup \{i\}} \mid j \in I] \\
&\quad + \mathbf{Pr}_h[h(i) = j \mid \neg \mathcal{F}_{L \cup \{i\}}, j \in I] \cdot \mathbf{Pr}_h[\neg \mathcal{F}_{L \cup \{i\}} \mid j \in I] \\
&\le \mathbf{Pr}_h[h(i) = j \mid \mathcal{F}_{L \cup \{i\}}, j \in I] + \mathbf{Pr}_h[\neg \mathcal{F}_{L \cup \{i\}} \mid j \in I] \\
&= \frac{1}{R} + \mathbf{Pr}_h[\neg \mathcal{F}_{L \cup \{i\}} \mid j \in I]
\end{aligned}
$$

Then by Bayes' theorem, the above is at most

$$
\begin{aligned}
\frac{1}{R} + \frac{\mathbf{Pr}_h[\neg \mathcal{F}_{L \cup \{i\}}]}{\mathbf{Pr}_h[j \in I]} &\le \frac{1}{R} + \frac{\mathbf{Pr}_h[\neg \mathcal{F}_{L \cup \{i\}}]}{\mathbf{Pr}_h[j \in I \mid \mathcal{F}_L] \cdot \mathbf{Pr}[\mathcal{F}_L]} \\
&= \frac{1}{R} + \frac{\mathbf{Pr}_h[\neg \mathcal{F}_{L \cup \{i\}}]}{\left(1 - \frac{1}{R}\right)^{|L|} \cdot \mathbf{Pr}[\mathcal{F}_L]} \\
&\le \frac{1}{R} + \frac{\mathbf{Pr}_h[\neg \mathcal{F}_{L \cup \{i\}}]}{\left(1 - \frac{|L|}{R}\right) \cdot \mathbf{Pr}[\mathcal{F}_L]}
\end{aligned}
$$

Note $|L|/R \le 1/2$. Also, by choice of c, z in the application of Theorem 13 in Step 3, $\mathbf{Pr}[\mathcal{F}_L] = 1 - O(\varepsilon)$ and $\mathbf{Pr}[\neg \mathcal{F}_{L \cup \{i\}}] = O(1/R^2)$. Thus overall,

$$
\mathbf{Pr}_h[h(i) = j \mid j \in I] = O(1/R).
$$

An essentially identical calculation, but conditioning on $\mathcal{F}_{L \cup \{i, i'\}}$ instead of $\mathcal{F}_{L \cup \{i\}}$, gives that

$$\mathbf{Pr}_h[(h(i) = j) \wedge (h(i') = j) \mid j \in I] = O(1/R^2).$$

Combining these bounds with Eq. (3.24), we have that the expected variance we are aiming to compute is

$$O(\|x_{\text{tail}}\|_{2p}^{2p} + \|x_{\text{tail}}\|_p^{2p}/R).$$

The second summand is $O(\varepsilon^2\|x\|_p^{2p})$. For the first summand, given our properties of L, every $|x_i|$ for $i \notin L$ has $|x_i| \le \varepsilon^2\|x\|_p^p$. Under this constraint, $\|x_{\text{tail}}\|_{2p}^{2p}$ is maximized when there are exactly $1/\varepsilon^2$ coordinates $i \notin L$ each with $|x_i| = \varepsilon^2\|x\|_p^p$, in which case $\|x_{\text{tail}}\|_{2p}^{2p} = \varepsilon^2\|x\|_p^{2p}$.

To achieve the desired update time, we buffer every $r = 1/\varepsilon^p$ updates then perform the fast multipoint evaluation of Theorem 55 in batch (note this does not affect our space bound since $p < 2$). That is, although the hash function h can be evaluated in constant time, updating any B_j requires evaluating a degree-$\Omega(1/\varepsilon^p)$ polynomial, which naïvely requires $\Omega(1/\varepsilon^p)$ time. Note that one issue is that the different data structures B_j use different polynomials, and thus we may need to evaluate $1/\varepsilon^p$ different polynomials on the $1/\varepsilon^p$ points, defeating the purpose of batching. To remedy this, note that these polynomials are themselves pairwise independent, so we can use the same approach described in Remark 39 to evaluate all the polynomials in combined time $O(\varepsilon^{-p}\log(1/\varepsilon)\log\log(1/\varepsilon) + r)$. ■

3.3.3 The final algorithm: putting it all together

To obtain our final algorithm, one option is to run HighEnd and LightEstimator in parallel after finding L, then output the sum of their estimates. Note that by the variance bound in Theorem 56, the output of a single instantiation of LightEstimator is $\|x_{[d]\setminus L}\|_p^p \pm O(\varepsilon)\|x\|_p^p$ with large constant probability. The downside to this option is that Theorem 43 uses space that would make our overall F_p estimation algorithm suboptimal by polylog(d/ε) factors, and HighEnd by an $O(\log(1/\varepsilon))$ factor for $\alpha = \varepsilon^2$ (Theorem 53). We can overcome this by a combination of balancing and universe reduction. Specifically, for balancing, notice that if instead of having L be a list of ε^2-heavy hitters, we instead defined it as a list of ϵ^2-heavy hitters for some $\epsilon > \varepsilon$, we could improve the space of both Theorem 43 and Theorem 53. To then make the variance in LightEstimator sufficiently small, i.e. $O(\varepsilon^2\|x\|_p^2)$, we could run $O((\epsilon/\varepsilon)^2)$ instantiations of LightEstimator in parallel and output the average estimate, keeping the space optimal but increasing the update time to $\Omega((\epsilon/\varepsilon)^2)$. This balancing gives a smooth tradeoff between space and update time; in fact note that for $\epsilon = 1$, our overall algorithm simply becomes a derandomized variant of Li's geometric mean estimator. We would like though to have $\epsilon \ll 1$ to have small update time.

Doing this balancing does not resolve all our issues though, since Theorem 43 is suboptimal by a $\log d$ factor. That is, even if we picked $\epsilon = 1$, Theorem 43 would cause our overall space to be $\Omega(\log d \log(mM))$, which is suboptimal. To overcome this issue we use universe reduction. Specifically, we set $N = 1/\varepsilon^{18}$ and pick hash functions $h_1 : [d] \to [N]$ and $\sigma : [d] \to \{-1, 1\}$. We define a new N-dimensional vector y by $y_i = \sum_{h_1(j)=i} \sigma(j)x_j$. Henceforth in this section, y, h_1, and σ are as discussed here. Rather than computing a list L of heavy hitters of x, we instead compute a list L' of heavy hitters of y. Then, since y has length only $\mathrm{poly}(1/\varepsilon)$, Theorem 43 is only suboptimal by $\mathrm{polylog}(1/\varepsilon)$ factors and our balancing trick applies. The list L' is also used in place of L for both HighEnd and LightEstimator. Though, since we never learn L, we must modify LightEstimator. Namely, the hash function $h : [d] \to [R]$ should be implemented as the composition of h_1, and a hash function $h_2 : [N] \to [R]$ chosen as Theorem 13 (again with $z = R$ and $c = 2$). Then, we let $I = [R]\backslash h_2(L')$. The remaining parts of the algorithm remain the same.

There are several issues we must address to show that our universe reduction step still maintains correctness. Informally, we need that (a) any i which is a heavy hitter for y should have exactly one $j \in [d]$ with $h_1(j) = i$ such that j was a heavy hitter for x, (b) if i is a heavy hitter for x, then $h_1(i)$ is a heavy hitter for y, and $|y_{h_1(i)}|^p = (1 \pm O(\varepsilon))|x_i|^p$ so that x_i's contribution to $\|x\|_p^p$ is properly approximated by HighEnd, (c) $\|y\|_p^p = O(\|x\|_p^p)$ with large probability, since the error term in HighEnd is $O(\varepsilon \cdot \|y\|_p^p)$, and (d) the amount of F_p mass not output by LightEstimator because it collided with a heavy hitter for x under h_1 is negligible. Also, the composition $h = h_1 \circ h_2$ for LightEstimator does not satisfy the conditions of Theorem 13 even though h_1 and h_2 might do so individually. To see why, as a simple analogy, consider that the composition of two purely random functions is no longer random. For example, as the number of compositions increases, the probability of two items colliding increases as well. Nevertheless, the analysis of LightEstimator carries over essentially unchanged in this setting, since whenever considering the distribution of where two items land under h, we can first condition on them not colliding under h_1. Not colliding under h_1 happens with $1 - O(\varepsilon^{18})$ probability, and thus the probability that two items land in two particular buckets $j, j' \in [R]$ under h is still $(1 \pm o(\varepsilon))/R^2$.

We now give our full description and analysis. We pick h_1 as in Theorem 13 with $z = R$ and $c = c_h$ a sufficiently large constant. We also pick σ from an $\Omega(\log N)$-wise independent family. We run an instantiation of $\mathsf{F_pHH}$ for the vector y with $\phi = \varepsilon^2/(34C)$ for a sufficiently large constant $C > 0$. We also obtain a value $\tilde{F}_p \in [F_p/2, 3F_p/2]$ using the algorithm of Section 3.2. We define L' to be the sublist of those w output by our $\mathsf{F_pHH}$ instantiation such that $|\tilde{y}_w|^p \geq (2\varepsilon^2/7)\tilde{F}_p$.

For ease of presentation in what follows, define L_ϕ to be the list of ϕ-heavy hitters of x with respect to F_p ("L", without a subscript, always denotes the ε^2-heavy hitters with respect to x), and define $z_i = \sum_{w \in h_1^{-1}(i)\backslash L_{\varepsilon^8}} \sigma(w)x_w$, i.e. z_i is the contribution to y_i from the significantly light elements of x.

Lemma 57. *For $x \in \mathbb{R}^n$, $\lambda > 0$ with λ^2 a multiple of 8, and random $z \in \{-1, 1\}^n$ drawn from a $(\lambda^2/4)$-wise independent family, $\mathbf{Pr}[|\langle x, z\rangle| > \lambda\|x\|_2] < 2^{-\lambda^2/4}$.*

Proof. By Markov's inequality on the random variable $\langle x, z\rangle^{\lambda^2/4}$, $\mathbf{Pr}[|\langle x, z\rangle| > \lambda] < \lambda^{-\lambda^2/4} \cdot \mathbf{E}[\langle x, z\rangle^{\lambda^2/4}]$. The claim follows by applying Lemma 45. ■

Lemma 58. *For any $C > 0$, there exists ε_0 such that for $0 < \varepsilon < \varepsilon_0$, $\mathbf{Pr}[\|y\|_p^p > 17C\|x\|_p^p] < 2/C$.*

Proof. Condition on h_1. Define $Y(i)$ to be the vector $x_{h_1^{-1}(i)}$. For any vector v we have $\|v\|_2 \le \|v\|_p$ since $p < 2$. Letting $\mathcal{E}$ be the event that no $i \in [N]$ has $|y_i| > 4\sqrt{\log N}\|Y(i)\|_p$, we have $\mathbf{Pr}[\mathcal{E}] \ge 1 - 1/N^4$ by Lemma 57. For $i \in [N]$, again by Lemma 57 any $i \in [N]$ has $|y_i| \le 2t \cdot \|Y(i)\|_2 \le 2t \cdot \|Y(i)\|_p$ with probability at least $1 - \max\{1/(2N), 2^{-t^2}\}$. Then for fixed $i \in [N]$,

$$\begin{aligned}
\mathbf{E}[|y_i|^p \mid \mathcal{E}] &\le 2^p\|Y(i)\|_p^p + \sum_{t=0}^{\infty} \mathbf{Pr}\left[(2\cdot 2^t)^p\|Y(i)\|_p^p < |y_i|^p \le (2\cdot 2^{t+1})^p\|Y(i)\|_p^p \mid \mathcal{E}\right] \\
&\qquad\qquad\qquad\qquad\qquad\qquad \times (2\cdot 2^{t+1})^p\|Y(i)\|_p^p \\
&\le 2^p\|Y(i)\|_p^p + (1/\mathbf{Pr}[\mathcal{E}]) \cdot \sum_{t=0}^{\log(2\sqrt{\log N})} 2^{-2^{2t}} \cdot (2\cdot 2^{t+1})^p\|Y(i)\|_p^p \\
&< 4\|Y(i)\|_p^p + (1/\mathbf{Pr}[\mathcal{E}]) \cdot \sum_{t=0}^{\log(2\sqrt{\log N})} 2^{-2^{2t}} \cdot (2\cdot 2^{t+1})^2\|Y(i)\|_p^p \\
&< 17\|Y(i)\|_p^p
\end{aligned}$$

since $\mathbf{Pr}[\mathcal{E}] \ge 1 - 1/N^4$ and ε_0 is sufficiently small. Thus by linearity of expectation, $\mathbf{E}[\|y\|_p^p \mid \mathcal{E}] \le 17\|x\|_p^p$, which implies $\|y\|_p^p \le 17C\|x\|_p^p$ with probability $1 - 1/C$, conditioned on $\mathcal{E}$ holding. We conclude by again using $\mathbf{Pr}[\mathcal{E}] \ge 1 - 1/N^4$. ■

Lemma 59. *With probability at least* $1 - \mathrm{poly}(\varepsilon)$ *over* σ*, simultaneously for all* $i \in [N]$ *we have that* $|z_i| = O(\sqrt{\log(1/\varepsilon)} \cdot \varepsilon^{6/p}\|x\|_p)$.

Proof. By Lemma 57, any $i \in [N]$ has $|z_i| \le 4\sqrt{\log(1/\varepsilon)} \cdot (\sum_{w \in h_1^{-1}(i)\setminus L_{\varepsilon^8}} |x_w|^2)^{1/2}$ with probability at least $1 - 1/N^4$. We then apply a union bound and use the fact that $\ell_p \le \ell_2$ for $p < 2$, so that $|z_i| \le 4\sqrt{\log(1/\varepsilon)} \cdot (\sum_{w \in h_1^{-1}(i)\setminus L_{\varepsilon^8}} |x_w|^p)^{1/p}$ (call this event $\mathcal{E}$) with probability $1 - \mathrm{poly}(\varepsilon)$.

We now prove our lemma, i.e. we show that with probability $1 - \mathrm{poly}(\varepsilon)$, $|z_i|^p = O(\log^{p/2}\varepsilon^6\|x\|_p^p)$ simultaneously for all $i \in [N]$. We apply Lemma 44. Specifically, fix an $i \in [N]$. For all j with $|x_j|^p \le \varepsilon^8\|x\|_p^p$, let $X_j = |x_j|^p \cdot \mathbf{1}_{h_1(j)=i}$. Then, in the notation of Lemma 44, $\mu_j = |x_j|^p/N$, and $\sigma_j^2 \le |x_j|^{2p}/N$, and thus $\mu = \|x\|_p^p/N$ and $\sigma^2 \le \|x\|_{2p}^{2p}/N \le \varepsilon^8\|x\|_p^p/N$. Also, $K = \varepsilon^8\|x\|_p^p$. Then if h_1 were ℓ-wise independent for $\ell = 10$, Lemma 44 would give

$$\mathbf{Pr}\left[\left|\sum_i X_i - \|x\|_p^p/N\right| > \varepsilon^6\|x\|_p^p\right] < 2^{O(\ell)} \cdot (\varepsilon^{7\ell} + \varepsilon^{2\ell}) = O(\varepsilon/N).$$

A union bound would then give that with probability $1-\varepsilon$, the F_p mass in any bucket from items i with $|x_i|^p \le \varepsilon^8\|x\|_p^p$ is at most $\varepsilon^6\|x\|_p^p$. Thus by a union bound with event $\mathcal{E}$, $|z_i|^p = O(\log^{p/2}\varepsilon^6\|x\|_p^p)$ for all $i \in [N]$ with probability $1 - \mathrm{poly}(\varepsilon)$.

Though, h_1 is not 10-wise independent. Instead, it is selected as in Theorem 13. However, for any constant ℓ, by increasing the constant c_h in our definition of h_1 we can ensure that our ℓth moment bound for $(\sum_i X_i - \mu)$ is preserved to within a constant factor, which is sufficient to apply Lemma 44. ■

Lemma 60. *With probability* $1 - \mathrm{poly}(\varepsilon)$*, for all* $w \in L$ *we have* $|y_{h_1(w)}|^p = (1 \pm O(\varepsilon))|x_w|^p$*, and thus with probability* $1-\mathrm{poly}(\varepsilon)$ *when conditioned on* $\|y\|_p^p \le 17C\|x\|_p^p$*, we have that if* w *is an* α*-heavy hitter for* x*, then* $h_1(w)$ *is an* $\alpha/(34C)$*-heavy hitter for* y*.*

Proof. Let w be in L. We know from Lemma 59 that $|z_{h_1(w)}| \le 2\sqrt{\log(1/\varepsilon)}\varepsilon^{6/p}\|x\|_p$ with probability $1 - \mathrm{poly}(\varepsilon)$, and that the elements of L are perfectly hashed under h_1 with probability $1 - \mathrm{poly}(\varepsilon)$. Conditioned on this perfect hashing, we have that $|y_{h_1(w)}| \ge |x_w| - 2\varepsilon^{6/p}\sqrt{\log(1/\varepsilon)}\|x\|_p$. Since for $w \in L$ we have $|x_w| \ge \varepsilon^{2/p}\|x\|_p$, and since $p < 2$, we have $|y_{h_1(w)}| \ge (1 - O(\varepsilon))|x_w|$.

For the second part of the lemma, $(1 - O(\varepsilon))|x_w| > |x_w|/2^{1/p}$ for ε_0 sufficiently small. Thus if w is an α-heavy hitter for x, then $h_1(w)$ is an $\alpha/(34C)$-heavy hitter for y. ■

Finally, the following lemma follows from a Markov bound followed by a union bound.

Lemma 61. *For* $w \in [d]$ *consider the quantity* $s_w = \sum_{\substack{v\neq w\\ h(v)=h(w)}} |x_v|^p$*. Then, with probability at least* $1 - O(\varepsilon)$*,* $s_w \le \varepsilon^{15}\|x\|_p^p$ *simultaneously for all* $w \in L$*.*

We now put everything together. We set $\epsilon = \varepsilon\log(1/\varepsilon)$. As stated earlier, we define L' to be the sublist of those w output by our $\mathsf{F_pHH}$ instantiation with $\phi = \epsilon^2$ such that $|\tilde{y}_w|^p \ge (2\varepsilon^2/7)\tilde{F}_p$. We interpret updates to x as updates to y to then be fed into $\mathsf{HighEnd}$, with $\alpha = \epsilon^2/(34C)$. Thus both $\mathsf{HighEnd}$ and $\mathsf{F_pHH}$ require $O(\varepsilon^{-2}\log(nmM/\varepsilon))$ space. We now define some events.

Event $\mathcal{A}$. L_{ε^8} is perfectly hashed under h_1, and $\forall i \in [N], |z_i|^p = O(\log(1/\varepsilon)^{p/2} \cdot \varepsilon^6\|x\|_p^p)$.

Event $\mathcal{B}$. $\forall w \in L_{\epsilon^2}$, $h_1(w)$ is output as an $\epsilon^2/(34C)$-heavy hitter by $\mathsf{F_pHH}$.

Event $\mathcal{C}$. $\forall w \in L_{\epsilon^2/18}$, $|y_{h_1(w)}| = (1 \pm O(\varepsilon))|x_w|$.

Event $\mathcal{D}$. $\tilde{F}_p \in [(1/2)\cdot\|x\|_p^p, (3/2)\cdot\|x\|_p^p]$, and $\mathsf{HighEnd}$, $\mathsf{LightEstimator}$, and $\mathsf{F_pHH}$ succeed.

Now, suppose $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$ all occur. Then for $w \in L_{\epsilon^2}$, w is output by $\mathsf{F_pHH}$, and furthermore $|y_{h_1(w)}|^p \ge (1 - O(\varepsilon))|x_w|^p \ge |x_w|^p/2 \ge \epsilon^2\|x\|_p^p/2$. Also, $\tilde{y}^p_{h_1(w)} \ge (6/7)\cdot|y_{h_1(w)}|^p$. Since $\tilde{F}_p \le 3\|x\|_p^p/2$, we have that $h_1(w) \in L'$. Furthermore, we also know that for i output by $\mathsf{F_pHH}$, $\tilde{y}_i^p \le (9/7)\cdot|y_i|^p$, and thus $i \in L'$ implies $|y_i|^p \ge (\epsilon^2/9)\cdot\|x\|_p^p$. Notice that by event $\mathcal{A}$, each y_i is z_i, plus potentially $x_{w(i)}$ for some $x_{w(i)} \in L_{\varepsilon^8}$. If $|y_i|^p \ge (\epsilon^2/9)\cdot\|x\|_p^p$, then there must exist such a $w(i)$, and furthermore it must be that $|x_{w(i)}|^p \ge (\epsilon^2/18)\cdot\|x\|_p^p$. Thus, overall, L' contains $h_1(w)$ for all $w \in L_{\epsilon^2}$, and furthermore if $i \in L'$ then $w(i) \in L_{\epsilon^2/18}$.

Since L' contains $h_1(L_{\epsilon^2})$, $\mathsf{LightEstimator}$ outputs $\|x_{[d]\setminus h^{-1}(L')}\|_p^p \pm O(\varepsilon\|x\|_p^p)$. Also, $\mathsf{HighEnd}$ outputs $\|y_{L'}\| \pm O(\varepsilon)\cdot\|y\|_p^p$. Now we analyze correctness. We have $\mathbf{Pr}[\mathcal{A}] =$

$1-\mathrm{poly}(\varepsilon)$, $\mathbf{Pr}[\mathcal{B} \mid \|y\|_p^p \le 17C\|x\|_p^p] = 1-\mathrm{poly}(\varepsilon)$, $\mathbf{Pr}[\mathcal{C}] = 1-\mathrm{poly}(\varepsilon)$, and $\mathbf{Pr}[\mathcal{D}] \ge 5/8$. We also have $\mathbf{Pr}[\|y\|_p^p \le 17C\|x\|_p^p] \ge 1-2/C$. Thus by a union bound and setting C sufficiently large, we have $\mathbf{Pr}[\mathcal{A} \wedge \mathcal{B} \wedge \mathcal{C} \wedge \mathcal{D} \wedge (\|y\|_p^p \le 17C\|x\|_p^p)] \ge 9/16$. Define L_{inv} to be the set $\{w(i)\}_{i \in L'}$, i.e. the heavy hitters of x corresponding to the heavy hitters in L' for y. Now, if all these events occur, then $\|x_{[d]\setminus h^{-1}(L')}\|_p^p = \|x_{[d]\setminus L_{\mathrm{inv}}}\|_p^p \pm O(\varepsilon^{15})\|x\|_p^p$ with probability $1-O(\varepsilon)$ by Lemma 61. We also have, since $\mathcal{C}$ occurs and conditioned on $\|y\|_p^p = O(\|x\|_p^p)$, that $\|y_{L'}\| \pm O(\varepsilon)\cdot\|y\|_p^p = \|x_{L_{\mathrm{inv}}}\|_p^p \pm O(\varepsilon)\cdot\|x\|_p^p$. Thus, overall, our algorithm outputs $\|x\|_p^p \pm O(\varepsilon)\cdot\|x\|_p^p$ with probability $17/32 > 1/2$ as desired. Notice this probability can be amplified to $1-\delta$ by outputting the median of $O(\log(1/\delta))$ independent instantiations.

We further note that for a single instantiation of LightEstimator, $\mathbf{E}_h[\mathbf{Var}_s[\Phi']] = O(\epsilon^2\|x\|_p^{2p})$. Once h is fixed, the variance of Φ' is simply the sum of variances across the B_j for $j \notin h_1(L')$. Thus, it suffices for the B_j to use pairwise independent randomness. Furthermore, in repeating $O((\epsilon/\varepsilon)^2)$ parallel repetitions of LightEstimator, it suffices that all the B_j across all parallel repetitions use pairwise independent randomness, and the hash function h can remain the same. Thus, as discussed in Remark 39, the coefficients of the degree-$O(1/\varepsilon^p)$ polynomials used in all B_j combined can be generated by just two coefficient vectors, and thus the update time of LightEstimator with $O((\epsilon/\varepsilon)^2)$ parallel repetitions is just $O((\epsilon/\varepsilon)^2 + O(\log^2(1/\varepsilon)\log\log(1/\varepsilon))) = O(\log^2(1/\varepsilon)\log\log(1/\varepsilon))$. Thus overall, we have the following theorem.

Theorem 62. *There exists an algorithm such that given $0 < p < 2$ and $0 < \varepsilon < 1/2$, the algorithm outputs $(1\pm\varepsilon)\|x\|_p^p$ with probability $2/3$ using $O(\varepsilon^{-2}\log(dmM/\varepsilon))$ space. The update time of the algorithm is $O(\log^2(1/\varepsilon)\log\log(1/\varepsilon))$, and its reporting time is $O(\varepsilon^{-2}\log^2(1/\varepsilon)\log\log(1/\varepsilon))$.*

The space bound above can be assumed $O(\varepsilon^{-2}\log(mM)+\log\log d)$ by Section 3.5.1.

3.3.4 A derandomized geometric mean estimator variant

In this section we prove Theorem 54. The data structure and estimator we give is a slightly modified version of the geometric mean estimator of Li [87]. Our modification allows us to show that only bounded independence is required amongst the p-stable random variables in our data structure.

Li's *geometric mean estimator* is as follows. For some positive integer $t > 2$, select a matrix $S \in \mathbb{R}^{t\times d}$ with independent p-stable entries, and maintain $y = Sx$ in the stream. Given y, the estimate of $\|x\|_p^p$ is then $C_{t,p}\cdot(\prod_{j=1}^t |y_j|^{p/t})$ for some constant $C_{t,p}$. For Theorem 54, we make the following adjustments. First, we require $t > 4$. Next, for any fixed row of S we only require that the entries be $\Omega(1/\varepsilon^p)$-wise independent, though the rows themselves we keep independent. Furthermore, in parallel we run the algorithm of Section 3.2 with constant error parameter to obtain a value $\tilde{F}_p$ in $[\|x\|_p^p/2, 3\|x\|_p^p/2]$. The D_p data structure of Theorem 54 is then simply y, together with the state maintained by the algorithm of Section 3.2. The estimator Est_p is $\min\{C_{t,p}\cdot(\prod_{j=1}^t |y_j|^{p/t}), \tilde{F}_p/\varepsilon\}$. To state the value $C_{t,p}$, we use the following theorem, which follows from [128, Theorem 2.6.3] after applying Euler's reflection formula.

Theorem 63. *For $Q \sim \mathcal{D}_p$ and $-1 < \lambda < p$,*

$$\mathbf{E}[|Q|^\lambda] = \frac{2}{\pi}\Gamma\left(1 - \frac{\lambda}{p}\right)\Gamma(\lambda)\sin\left(\frac{\pi}{2}\lambda\right).$$

Theorem 63 implies that we should set

$$C_{t,p} = \left[\frac{2}{\pi}\cdot\Gamma\left(1-\frac{1}{t}\right)\cdot\Gamma\left(\frac{p}{t}\right)\cdot\sin\left(\frac{\pi p}{2t}\right)\right]^{-t}.$$

We now prove a tail bound for linear forms over r-wise independent p-stable random variables. Note that for a random variable X whose moments are bounded, one has $\mathbf{Pr}[X - \mathbf{E}[X] > t] \le \mathbf{E}[(X - \mathbf{E}[X])^r]/t^r$ by applying Markov's inequality to the random variable $(X - \mathbf{E}[X])^r$ for some even integer $r \ge 2$. Unfortunately, for $0 < p < 2$, it is known that even the second moment of $\mathcal{D}_p$ is already infinite, so this method cannot be applied. We instead prove our tail bound via FT-mollification of $I_{[t,\infty)}$, since $\mathbf{Pr}[X \ge t] = \mathbf{E}[I_{[t,\infty)}(X)]$.

We now prove our tail bound.

Lemma 64. *Suppose $x \in \mathbb{R}^d$, $\|x\|_p = 1$, $0 < \varepsilon < 1/2$ is given, and $R_1, \ldots, R_d$ are r-wise independent p-stable random variables for $r \ge 2$. Let $Q \sim \mathcal{D}_p$. Then for all $t \ge 0$, $R = \sum_{i=1}^d R_i x_i$ satisfies*

$$|\mathbf{Pr}[|Q| \ge t] - \mathbf{Pr}[|R| \ge t]| = O(r^{-1/p}/(1+t^{p+1}) + r^{-2/p}/(1+t^2) + 2^{-\Omega(r)}).$$

Proof. We have $\mathbf{Pr}[|Z| \ge t] = \mathbf{E}[I_{[t,\infty)}(Z)] + \mathbf{E}[I_{(-\infty,t]}(Z)]$ for any random variable Z, and thus we will argue $|\mathbf{E}[I_{[t,\infty)}(Q)] - \mathbf{E}[I_{[t,\infty)}(R)]| = O(r^{-1/p}/(1+t^{p+1}) + r^{-2/p}/(1+t^2) + 2^{-\Omega(r)})$; a similar argument shows the same bound for $|\mathbf{E}[I_{(-\infty,t]}(Q)] - \mathbf{E}[I_{(-\infty,t]}(R)]|$.

We argue the following chain of inequalities for $c = r^{1/p}/(3C)$, for C the constant in Lemma 35, and we define $\gamma = r^{-1/p}/(1+t^{p+1}) + r^{-2/p}/(1+t^2)$:

$$\mathbf{E}[I_{[t,\infty)}(Q)] \approx_\gamma \mathbf{E}[\tilde{I}^c_{[t,\infty)}(Q)] \approx_{2^{-c^p}} \mathbf{E}[\tilde{I}^c_{[t,\infty)}(R)] \approx_{\gamma+2^{-c^p}} \mathbf{E}[I_{[t,\infty)}(R)].$$

$\mathbf{E}[\mathbf{I}_{[\mathbf{t},\infty)}(\mathbf{Q})] \approx_\gamma \mathbf{E}[\tilde{\mathbf{I}}^{\mathbf{c}}_{[\mathbf{t},\infty)}(\mathbf{Q})]$: Assume $t \ge 1$. We have

$$\begin{aligned}
|\mathbf{E}[I_{[t,\infty)}(Q)] - \mathbf{E}[\tilde{I}^c_{[t,\infty)}(Q)]| &\le \mathbf{E}[|I_{[t,\infty)}(Q) - \tilde{I}^c_{[t,\infty)}(Q)|] \\
&\le \mathbf{Pr}[|Q-t| \le 1/c] + \left(\sum_{s=1}^{\log(ct)-1} \mathbf{Pr}[|Q-t| \le 2^s/c]\cdot O(2^{-2s})\right) \qquad (3.25) \\
&\quad + \mathbf{Pr}[|Q-t| > t/2]\cdot O(c^{-2}t^{-2}) \\
&= O(1/(c\cdot t^{p+1})) + O(c^{-2}t^{-2})
\end{aligned}$$

since $\mathbf{Pr}[|Q-t| \le 2^s/c$ is $O(2^s/(c\cdot t^{p+1})$ as long as $2^s/c \le t/2$.

In the case $0 < t < 1$, we repeat the same argument as above but replace Eq. (3.25) with a summation from $s = 1$ to ∞, and also remove the additive $\mathbf{Pr}[|Q - t| >$

$t/2] \cdot O(c^{-2}t^{-2})$ term. Doing so gives an overall upper bound of $O(1/c)$ in this case.

$\mathbf{E}[\tilde{\mathbf{I}}^{\mathbf{c}}_{[\mathbf{t},\infty)}(\mathbf{Q})] \approx_{\mathbf{2^{-c^p}}} \mathbf{E}[\tilde{\mathbf{I}}^{\mathbf{c}}_{[\mathbf{t},\infty)}(\mathbf{R})]$: This follows from Lemma 35 with $\varepsilon = 2^{-c^p}$ and $\alpha = c$.

$\mathbf{E}[\tilde{\mathbf{I}}^{\mathbf{c}}_{[\mathbf{t},\infty)}(\mathbf{R})] \approx_{\gamma+\mathbf{2^{-c^p}}} \mathbf{E}[\mathbf{I}_{[\mathbf{t},\infty)}(\mathbf{R})]$: We would like to apply the same argument as when showing $\mathbf{E}[\tilde{I}^c_{[t,\infty)}(Q)] \approx_\gamma \mathbf{E}[I_{[t,\infty)}(Q)]$ above. The trouble is, we must bound $\mathbf{Pr}[|R - t| > t/2]$ and $\mathbf{Pr}[|R-t| \le 2^s/c]$ given that the R_i are only r-wise independent. For the first probability, we above only used that $\mathbf{Pr}[|Q-t| > t/2] \le 1$, which still holds with Q replaced by R.

For the second probability, observe $\mathbf{Pr}[|R-t| \le 2^s/c] = \mathbf{E}[I_{[t-2^s/c,t+2^s/c]}(R)]$. Define $\delta = 2^s/c + b/c$ for a sufficiently large constant $b > 0$ to be determined later. Then, arguing as above, we have

$$\mathbf{E}[\tilde{I}^c_{[t-\delta,t+\delta]}(R)] \approx_{2^{-c^p}} \mathbf{E}[\tilde{I}^c_{[t-\delta,t+\delta]}(Q)] \approx_\gamma \mathbf{E}[I_{[t-\delta,t+\delta]}(Q)],$$

and we also know

$$\mathbf{E}[I_{[t-\delta,t+\delta]}(Q)] = O(\mathbf{E}[I_{[t-2^s/c,t+2^s/c]}(Q)]) = O(\mathbf{Pr}[|Q-t| \le 2^s/c]) = O(2^s/(c \cdot t^{p+1})).$$

Now, for $x \notin [t-2^s/c, t+2^s/c]$, $I_{[t-2^s/c,t+2^s/c]}(x) = 0$ while $I_{[t-\delta,t+\delta]}(x) = 1$. For $x \in [t-2^s/c, t+2^s/c]$, the distance from x to $\{t-\delta, t+\delta\}$ is at least b/c, implying $\tilde{I}^c_{[t-\delta,t+\delta]}(x) \ge 1/2$ for b sufficiently large by Corollary 31. Thus, $2 \cdot \tilde{I}^c_{[t-\delta,t+\delta]} \ge I_{[t-2^s/c,t+2^s/c]}$ on $\mathbb{R}$, and thus in particular, $\mathbf{E}[I_{[t-2^s/c,t+2^s/c]}(R)] \le 2 \cdot \mathbf{E}[\tilde{I}^c_{[t-\delta,t+\delta]}(R)]$. Thus, in summary, $\mathbf{E}[I_{[t-2^s/c,t+2^s/c]}(R)] = O(2^s/(c \cdot t^{p+1}) + \gamma + 2^{-c^p})$. ■

We now prove the main lemma of this section, which implies Theorem 54.

Lemma 65. *Let $x \in \mathbb{R}^d$ be such that $\|x\|_p = 1$, and suppose $0 < \varepsilon < 1/2$. Let $0 < p < 2$, and let t be any constant greater than $4/p$. Let $R_1, \ldots, R_d$ be r-wise independent p-stable random variables for $r = \Omega(1/\varepsilon^p)$, and let Q be a p-stable random variable. Define $f(x) = \min\{|x|^{1/t}, T\}$, for $T = 1/\varepsilon$. Then, $|\mathbf{E}[f(R)] - \mathbf{E}[|Q|^{1/t}] = O(\varepsilon)$ and $\mathbf{E}[f^2(R)] = O(\mathbf{E}[|Q|^{2/t}])$.*

Proof. We first argue $|\mathbf{E}[f(R)] - \mathbf{E}[|Q|^{1/t}] = O(\varepsilon)$. We argue through the chain of inequalities

$$\mathbf{E}[|Q|^{1/t}] \approx_\varepsilon \mathbf{E}[f(Q)] \approx_\varepsilon \mathbf{E}[f(R)].$$

$\mathbf{E}[|\mathbf{Q}|^{\mathbf{1/t}}] \approx_\varepsilon \mathbf{E}[\mathbf{f}(\mathbf{Q})]$: We have

$$\begin{aligned}|\mathbf{E}[|Q|^{1/t}] - \mathbf{E}[f(Q)]| &= 2\int_{T^t}^{\infty}(x^{1/t}-T)\cdot\varphi_p(x)dx\\ &= \int_{T^t}^{\infty}(x^{1/t}-T)\cdot O(1/x^{p+1})dx\\ &= O\left(T^{1-tp}\cdot\left(\frac{t}{pt-1}+\frac{1}{p}\right)\right)\\ &= O(1/(Tp))\\ &= O(\varepsilon)\end{aligned}$$

$\mathbf{E}[\mathbf{f}(\mathbf{Q})] \approx_\varepsilon \mathbf{E}[\mathbf{f}(\mathbf{R})]$: Let φ_p^+ be the probability density function corresponding to the distribution of $|Q|$, and let Φ_p^+ be its cumulative distribution function. Then, by integration by parts and Lemma 64,

$$\begin{aligned}\mathbf{E}[f(Q)] &= \int_0^{T^t} x^{1/t}\varphi_p^+(x)dx + T\cdot\int_{T^t}^{\infty}\varphi_p^+(x)dx\\ &= -[x^{1/t}\cdot(1-\Phi_p^+(x))]_0^{T^t} - T\cdot[(1-\Phi_p^+(x))]_{T^t}^{\infty} + \frac{1}{t}\int_0^{T^t}\frac{1}{x^{1-1/t}}(1-\Phi_p^+(x))dx\\ &= \frac{1}{t}\int_0^{T^t}\frac{1}{x^{1-1/t}}\cdot\mathbf{Pr}[|Q|\geq x]dx\\ &= \frac{1}{t}\int_0^{T^t}\frac{1}{x^{1-1/t}}\cdot(\mathbf{Pr}[|R|\geq x] + O(r^{-1/p}1/(1+x^{p+1})\\ &\qquad\qquad + r^{-2/p}/(1+x^2)+2^{-\Omega(r)}))dx\\ &= \mathbf{E}[f(R)] + \int_0^1 x^{1/t-1}\cdot O(r^{-1/p}+r^{-2/p}+2^{-\Omega(r)}))dx\\ &\quad + \int_1^{T^t} x^{1/t-1}\cdot O(r^{-1/p}/x^{p+1}+r^{-2/p}/x^2+2^{-\Omega(r)}))dx \qquad (3.26)\\ &= \mathbf{E}[f(R)] + O(\varepsilon) + O\left(\frac{1}{r^{1/p}}\cdot\left(\frac{1}{T^{tp+t-1}}-1\right)\cdot\frac{1}{\frac{1}{t}-p-1}\right)\\ &\quad + O\left(\frac{1}{r^{2/p}}\cdot\left(\frac{1}{T^{2t-1}}-1\right)\cdot\frac{1}{\frac{1}{t}-2}\right) + O(2^{-\Omega(r)}\cdot(T-1))\\ &= \mathbf{E}[f(R)] + O(\varepsilon)\end{aligned}$$

We show $\mathbf{E}[f^2(R)] = O(|Q|^{2/t})$ similarly. Namely, we argue through the chain of inequalities

$$\mathbf{E}[|Q|^{2/t}] \approx_\varepsilon \mathbf{E}[f^2(Q)] \approx_\varepsilon \mathbf{E}[f^2(R)],$$

which proves our claim since $\mathbf{E}[|Q|^{2/t}] = \Omega(1)$ by Theorem 63.

$\mathbf{E}[|\mathbf{Q}|^{\mathbf{1/t}}] \approx_\varepsilon \mathbf{E}[\mathbf{f^2}(\mathbf{Q})]$: We have

$$\begin{aligned}
|\mathbf{E}[|Q|^{2/t}] - \mathbf{E}[f^2(Q)]| &= 2\int_{T^t}^{\infty} (x^{2/t} - T^2)\cdot \varphi_p(x)dx \\
&= \int_{T^t}^{\infty} (x^{2/t} - T^2)\cdot O(1/x^{p+1})dx \\
&= O\left(T^{2-tp}\cdot\left(\frac{t}{pt-2} - \frac{1}{p}\right)\right) \\
&= O(1/(Tp)) \\
&= O(\varepsilon)
\end{aligned}$$

$\mathbf{E}[\mathbf{f^2}(\mathbf{Q})] \approx_\varepsilon \mathbf{E}[\mathbf{f^2}(\mathbf{R})]$: This is argued similarly as our proof that $\mathbf{E}[f(Q)] \approx_\varepsilon \mathbf{E}[f(R)]$ above. The difference is that our error term corresponding to Eq. (3.26) is now

$$\begin{aligned}
&\int_0^1 x^{2/t-1}\cdot O(r^{-1/p} + r^{-2/p} + 2^{-\Omega(r)}))dx \\
&\qquad + \int_1^{T^t} x^{2/t-1}\cdot O(r^{-1/p}/x^{p+1} + r^{-2/p}/x^2 + 2^{-\Omega(r)}))dx \\
&\qquad = O(\varepsilon) + O\left(\frac{1}{r^{1/p}}\cdot\left(\frac{1}{T^{tp+t-2}} - 1\right)\cdot\frac{1}{\frac{2}{t} - p - 1}\right) \\
&\qquad\quad + O\left(\frac{1}{r^{2/p}}\cdot\left(\frac{1}{T^{2t-2}} - 1\right)\cdot\frac{1}{\frac{2}{t} - 2}\right) + O(2^{-\Omega(r)}\cdot(T^2-1)) \\
&\qquad = O(\varepsilon)
\end{aligned}$$

■

3.3.5 A heavy hitter algorithm for F_p

Note that $\mathsf{F_pReport}$, $\mathsf{F_pUpdate}$, and $\mathsf{F_pSpace}$ below can be as in the statement in Section 3.3.1 by using the algorithm of Section 3.2.

Theorem 43 (restatement). *There is an algorithm* $\mathsf{F_pHH}$ *satisfying the following properties. Given* $0 < \phi, \delta < 1/2$ *and black-box access to an* F_p *estimation algorithm* $\mathsf{F_pEst}(\varepsilon', \delta')$ *with* $\varepsilon' = 1/7$ *and* $\delta' = \phi\delta/(12(\log(\phi d) + 1))$, $\mathsf{F_pHH}$ *produces a list* L *such that* L *contains all* ϕ*-heavy hitters and does not contain indices which are not* $\phi/2$*-heavy hitters with probability at least* $1-\delta$*. For each* $i \in L$*, the algorithm also outputs* $\mathrm{sign}(x_i)$*, as well as an estimate* $\tilde{x}_i$ *of* x_i *satisfying* $\tilde{x}_i^p \in [(6/7)|x_i|^p, (9/7)|x_i|^p]$*. Its space usage is* $O(\phi^{-1}\log(\phi d)\cdot \mathsf{F_pSpace}(\varepsilon', \delta') + \phi^{-1}\log(1/(\delta\phi))\log(dmM))$*. Its update time is* $O(\log(\phi d)\cdot\mathsf{F_pUpdate}(\varepsilon', \delta') + \log(1/(\delta\phi)))$*. Its reporting time is* $O(\phi^{-1}(\log(\phi d)\cdot \mathsf{F_pReport}(\varepsilon', \delta') + \log(1/(\delta\phi))))$*.* $\mathsf{F_pReport}(\varepsilon', \delta')$*,* $\mathsf{F_pUpdate}(\varepsilon', \delta')$*, and* $\mathsf{F_pSpace}(\varepsilon', \delta')$ *are the reporting time, update time, and space consumption of* $\mathsf{F_pEst}$ *when a* $(1 \pm \varepsilon')$*-approximation to* F_p *is desired with probability at least* $1 - \delta'$*.*

Proof. First we argue with $\delta' = \phi\delta/(12(\log d + 1))$. We assume without loss of generality that d is a power of 2. Consider the following data structure $\mathsf{BasicF_pHH}(\phi', \delta, \varepsilon', k)$, where $k \in \{0, \ldots, \log d\}$. We set $R = \lceil 1/\phi' \rceil$ and pick a function $h : \{0, \ldots, 2^k - 1\} \to [R]$ at random from a pairwise independent hash family. We also create instantiations $D_1, \ldots, D_R$ of $\mathsf{F_pEst}(\varepsilon', 1/5)$. This entire structure is then repeated independently in parallel $T = \Theta(\log(1/\delta))$ times, so that we have hash functions $h_1, \ldots, h_T$, and instantiations D_i^j of $\mathsf{F_pEst}$ for $i, j \in [R] \times [T]$. For an integer i in $[d]$, let $\mathsf{prefix}(i, k)$ denote the length-k prefix of $i - 1$ when written in binary, treated as an integer in $\{0, \ldots, 2^k - 1\}$. Upon receiving an update (i, v) in the stream, we feed this update to $D^j_{h_j(\mathsf{prefix}(i,k))}$ for each $j \in [T]$.

For $t \in \{0, \ldots, 2^k - 1\}$, let $F_p(t)$ denote the F_p value of the vector x restricted to indices $i \in [d]$ with $\mathsf{prefix}(i) = t$. Consider the procedure $\mathsf{Query}(t)$ which outputs the median of F_p estimates given by $D^j_{h_j(t)}$ over all $j \in [T]$. We now argue that the output of $\mathsf{Query}(t)$ is in the interval $[(1 - \varepsilon') \cdot F_p(t), (1 + \varepsilon') \cdot (F_p(t) + 5\phi' \|x\|_p^p)]$, i.e. $\mathsf{Query}(t)$ "succeeds", with probability at least $1 - \delta$.

For any $j \in [T]$, consider the actual F_p value $F_p(t)^j$ of the vector x restricted to coordinates i such that $h_j(\mathsf{prefix}(i, k)) = h_j(t)$. Then $F_p(t)^j = F_p(t) + R(t)^j$, where $R(t)^j$ is the F_p contribution of the i with $\mathsf{prefix}(i, k) \neq t$, yet $h_j(\mathsf{prefix}(i, k)) = h(t)$. We have $R(t)^j \geq 0$ always, and furthermore $\mathbf{E}[R(t)^j] \leq \|x\|_p^p/R$ by pairwise independence of h_j. Thus by Markov's inequality, $\mathbf{Pr}[R(t)^j > 5\phi'\|x\|_p^p] < 1/5$. Note for any fixed $j \in [T]$, the F_p estimate output by $D^j_{h(t)}$ is in $[(1-\varepsilon') \cdot F_p(t), (1+\varepsilon') \cdot (F_p(t)+5\phi'\|x\|_p^p)]$ as long as both the events "$D^j_{h(t)}$ successfully gives a $(1 \pm \varepsilon')$-approximation" and "$R(t)^j \leq 5\phi'\|x\|_p^p$" occur. This happens with probability at least 3/5. Thus, by a Chernoff bound, the output of $\mathsf{Query}(t)$ is in the desired interval with probability at least $1 - \delta$.

We now define the final $\mathsf{F_pHH}$ data structure. We maintain one global instantiation D of $\mathsf{F_pEst}(1/7, \delta/2)$. We also use the dyadic interval idea for ℓ_1-heavy hitters given in [34]. Specifically, we imagine building a binary tree $\mathcal{T}$ over the universe $[d]$ (without loss of generality assume n is a power of 2). The number of levels in the tree is $\ell = 1 + \log d$, where the root is at level 0 and the leaves are at level $\log d$. For each level $j \in \{0, \ldots, \ell\}$, we maintain an instantiation B_j of $\mathsf{BasicF_pHH}(\phi/80, \delta', 1/7, j)$ for δ' as in the theorem statement. When we receive an update (i, v) in the stream, we feed the update to D and also to each B_j.

We now describe how to answer a query to output the desired list L. We first query D to obtain $\tilde{F}_p$, an approximation to F_p. We next initiate an iterative procedure on our binary tree, beginning at the root, which proceeds level by level. The procedure is as follows. Initially, we set $L = \{0\}$, $L' = \emptyset$, and $j = 0$. For each $i \in L$, we perform $\mathsf{Query}(i)$ on B_j then add $2i$ and $2i + 1$ to L' if the output of $\mathsf{Query}(i)$ is at least $3\phi\tilde{F}_p/4$. After processing every $i \in L$, we then set $L \leftarrow L'$ then $L' \leftarrow \emptyset$, and we increment j. This continues until $j = 1 + \log d$, at which point we halt and return L. We now show why the list L output by this procedure satisfies the claim in the theorem statement. We condition on the event $\mathcal{E}$ that $\tilde{F}_p = (1 \pm 1/7)F_p$, and also on the event $\mathcal{E}'$ that every query made throughout the recursive procedure is successful. Let i be such that $|x_i|^p \geq \phi F_p$. Then, since $F_p(\mathsf{prefix}(i, j)) \geq |x_i|^p$ for any j, we always have that $\mathsf{prefix}(i, j) \in L$ at the end of the jth round of our iterative procedure, since

$(6/7)|x_i|^p \geq (3/4)\phi\tilde{F}_p$ given $\mathcal{E}$. Now, consider an i such that $|x_i|^p < (\phi/2)F_p$. Then, $(8/7) \cdot (|x_i|^p - 5 \cdot (\phi/80)) < 3\phi\tilde{F}_p/4$, implying i is not included in the final output list. Also, note that since the query at the leaf corresponding to $i \in L$ is successful, then by definition of a successful query, we are given an estimate $\tilde{x}_i^p$ of $|x_i|^p$ by the corresponding $\mathsf{BasicF_pHH}$ structure satisfying $\tilde{x}_i^p \in [(6/7)|x_i|^p, (8/7)|x_i|^p + (\phi/16)F_p]$, which is $[(6/7)|x_i|^p, (9/7)|x_i|^p]$ since $|x_i|^p \geq (\phi/2)F_p$.

We now only need to argue that $\mathcal{E}$ and $\mathcal{E}'$ occur simultaneously with large probability. We have $\mathbf{Pr}[\mathcal{E}] \geq 1 - \delta/2$. For $\mathcal{E}'$, note there are at most 2ϕ $\phi/2$-heavy hitters at any level of the tree, where at level j we are referring to heavy hitters of the 2^j-dimensional vector y_j satisfying $(y_j)_i^p = \sum \mathsf{prefix}(t,j) = i|x_t|^p$. As long as the $\mathsf{Query}(\cdot)$ calls made for all $\phi/2$-heavy hitters and their two children throughout the tree succeed (including at the root), $\mathcal{E}'$ holds. Thus, $\mathbf{Pr}[\mathcal{E}'] \geq 1 - \delta' \cdot 6(\log d + 1)\phi^{-1} = 1 - \delta/2$. Therefore, by a union bound $\mathbf{Pr}[\mathcal{E} \wedge \mathcal{E}'] \geq 1 - \delta$.

Finally, notice that the number of levels in $\mathsf{F_pHH}$ can be reduced from $\log d$ to $\log d - \log\lceil 1/\phi \rceil = O(\log(\phi d))$ by simply ignoring the top $\log\lceil 1/\phi \rceil$ levels of the tree. Then, in the topmost level of the tree which we maintain, the universe size is $O(1/\phi)$, so we can begin our reporting procedure by querying all these universe items to determine which subtrees to recurse upon.

To recover $\text{sign}(x_w)$ for each $w \in L$, we use the $\mathsf{CountSketch}$ data structure of [29] with $T = (21 \cdot 2^p)/\phi$ columns and $C = \Theta(\log(1/(\delta\phi)))$ rows; the space is $O(\phi^{-1}\log(1/(\delta\phi))\log(dmM))$, and the update time is $O(\log(1/(\delta\phi)))$. $\mathsf{CountSketch}$ operates by, for each row i, having a pairwise independent hash function $h_i : [d] \to [T]$ and a 4-wise independent hash function $\sigma_i : [d] \to \{-1, 1\}$. There are $C \cdot T$ counters $A_{i,j}$ for $(i,j) \in [C] \times [T]$. Counter $A_{i,j}$ maintains $\sum_{h_i(v)=j} \sigma_i(v) \cdot x_v$. For $(i,j) \in [C] \times [T]$, let x^i be the vector x restricted to coordinates v with $h_i(v) = h_i(w)$, other than w itself. Then for fixed i, the expected contribution to $\|x^i\|_p^p$ is at most $\|x\|_p^p/T$, and thus is at most $10\|x\|_p^p/T$ with probability $9/10$ by Markov's inequality. Conditioned on this event, $|x_w| > \|x^i\|_p/2 \geq \|x^i\|_2/2$. The analysis of $\mathsf{CountSketch}$ also guarantees $|A_{i,h_i(w)} - \sigma_i(w)x_w| \leq 2\|x^i\|_2$ with probability at least $2/3$, and thus by a union bound, $|x_w| > |A_{i,h_i(w)} - \sigma_i(w)x_w|$ with probability at least $11/20$, in which case $\sigma_i(w) \cdot \text{sign}(A_{i,h_i(w)}) = \text{sign}(x_w)$. Thus, by a Chernoff bound over all rows, together with a union bound over all $w \in L$, we can recover $\text{sign}(x_w)$ for all $w \in L$ with probability $1 - \delta$. ■

3.4 Lower Bounds

In this section we prove our lower bound for $(1 \pm \varepsilon)$-multiplicative approximation of F_p for any positive real constant p bounded away from 0 in the turnstile model. We show a lower bound of $\Omega(\min\{d, m, \varepsilon^{-2}(\log(\varepsilon^2 mM))\})$. Note that if $\varepsilon \geq 1/\min\{d, m\}^{1/2-\delta}$ for any constant $\delta > 0$, the lower bound becomes $\Omega(\varepsilon^{-2}\log mM)$, matching our upper bound. We also describe a folklore lower bound of $\Omega(\log\log d)$ in Section 3.5.2. Our lower bound holds for all ranges of the parameters ε, d, m, M varying independently.

Our proof in part uses that Augmented-Indexing requires linear communication in the one-way, one-round model [91] (an alternative proof was also given later in [10, Lemma 2]). We also use a known reduction [70, 125] from Indexing to Gap-Hamdist. Henceforth all communication games discussed will be one-round and two-player, with the first player to speak named "Alice", and the second "Bob". We assume that Alice and Bob have access to public randomness.

Definition 66. *In the* Augmented-Indexing *problem, Alice receives a vector $x \in \{0,1\}^n$, Bob receives some $i \in [n]$ as well as all x_j for $j > i$, and Bob must output x_i. The problem* Indexing *is defined similarly, except Bob receives only $i \in [n]$, without receiving x_j for $j > i$.*

Definition 67. *In the* Gap-Hamdist *problem, Alice receives $x \in \{0,1\}^n$ and Bob receives $y \in \{0,1\}^n$. Bob is promised that either $\Delta(x,y) \le n/2 - \sqrt{n}$ (*NO *instance), or $\Delta(x,y) \ge n/2 + \sqrt{n}$ (*YES *instance) and must decide which holds. Here $\Delta(\cdot,\cdot)$ denotes Hamming distance.*

We also make use of the following two theorems.

Theorem 68 (Miltersen et al. [91, Lemma 13])**.** *The randomized one-round, one-way communication complexity of* Augmented-Indexing *with error probability at most $1/3$ is $\Omega(n)$.* ■

Theorem 69 ([70], [125, Section 4.3])**.** *There is a reduction from* Indexing *to* Gap-Hamdist *such that deciding* Gap-Hamdist *with probability at least $11/12$ implies a solution to* Indexing *with probability at least $2/3$. Furthermore, in this reduction the parameter n in* Indexing *is within a constant factor of that for the reduced* Gap-Hamdist *instance.* ■

Theorem 70 (Main lower bound)**.** *For any real constant $p > 0$, any one-pass streaming algorithm for $(1 \pm \varepsilon)$-multiplicative approximation of F_p with probability at least $11/12$ in the turnstile model requires $\Omega(\min\{d, m, \varepsilon^{-2}(1 + p \cdot \log(\lceil \varepsilon^2 mM \rceil))\})$ bits of space.*

Proof. Let $q = 2^{1/p}$ and define $k = \lfloor 1/\varepsilon^2 \rfloor$ and $t = \min\{\lfloor \min\{d,m\}/k \rfloor, \lfloor \log_q(M) \rfloor + 1\}$. Given an algorithm A providing a $(1 \pm r\varepsilon/2^p)$-multiplicative approximation of F_p with probability at least $11/12$, where $r > 0$ is some small constant to be fixed later, we devise a protocol to decide Augmented-Indexing on strings of length kt, where the number of bits communicated in the protocol is dominated by the space complexity of A. Since $kt = \Omega(\min\{d, m, \varepsilon^{-2}(1 + p \cdot \log(\lceil \varepsilon^2 mM \rceil))\})$ and $2^p = O(1)$ for constant p, the lower bound (ignoring the $\varepsilon^2 m$ term) follows. At the end of the proof, we describe how to obtain our stated dependence on m in the lower bound as well.

Let Alice receive $x \in \{0,1\}^{kt}$, and Bob receive $z \in [kt]$. Alice conceptually divides x into t contiguous blocks where the ith block b_i is of size k. Bob's index z lies in some $b_{i(z)}$, and Bob receives bits x_j that lie in a block b_i with $i > i(z)$. Alice applies the Gap-Hamdist reduction of Theorem 69 to each b_i separately to obtain new vectors y_i each of length at most $c \cdot k$ for some constant c for all $0 \le i < t$. Alice then creates a stream from the set of y_i by, for each i and each bit $(y_i)_j$ of y_i, inserting an update

$((i,j), \lfloor q^i \rfloor)$ into the stream if $(y_i)_j = 1$. Alice processes this stream with A then sends the state of A to Bob along with the Hamming weight $w(y_i)$ of y_i for all i. Note the size of the universe in the stream is at most $ckt = O(\min\{d,m\}) = O(d)$.

Now, since Bob knows the bits in b_i for $i > i(z)$ and shares randomness with Alice, he can run the same Gap-Hamdist reduction as Alice to obtain the y_i for $i > i(z)$ then delete all the insertions Alice made for these y_i. Bob then performs his part of the reduction from Indexing on strings of length k to Gap-Hamdist within the block $b_{i(z)}$ to obtain a vector $y(B)$ of length ck such that deciding whether $\Delta(y(B), y_{i(z)}) > ck/2 + \sqrt{ck}$ or $\Delta(y(B), y_{i(z)}) < ck/2 - \sqrt{ck}$ with probability at least $11/12$ allows one to decide the Indexing instance on block $b_{i(z)}$ with probability at least $2/3$. Here $\Delta(\cdot,\cdot)$ denotes Hamming distance. For each j such that $y(B)_j = 1$, Bob inserts $((i(z),j), -\lfloor q^{i(z)} \rfloor)$ into the stream being processed by A. We have so far described all stream updates, and thus the number of updates is at most $2ckt = O(\min\{d,m\}) = O(m)$. Note the pth moment L'' of the underlying vector of the stream now exactly satisfies

$$L'' = \left\lfloor q^{i(z)} \right\rfloor^p \cdot \Delta(y(B), y_{i(z)}) + \sum_{i<i(z)} w(y_i) \left\lfloor q^i \right\rfloor^p . \tag{3.27}$$

Setting $\eta = \sum_{i<i(z)} w(y_i) \lfloor q^i \rfloor^p$ and rearranging terms, $\Delta(y(B), y_{i(z)}) = (L''-\eta)/\left\lfloor q^{i(z)} \right\rfloor^p$. Recall that in this Gap-Hamdist instance, Bob must decide whether $\Delta(y(B), y_{i(z)}) < ck/2 - \sqrt{ck}$ or $\Delta(y(B), y_{i(z)}) > ck/2 + \sqrt{ck}$. Bob can compute η exactly given Alice's message. To decide Gap-Hamdist it thus suffices to obtain a $(\sqrt{ck}/4)$-additive approximation to $\left\lfloor q^{i(z)} \right\rfloor^{-p} L''$. Note $\lfloor q^i \rfloor^p = 2^i \cdot (\lfloor q^i \rfloor / q^i)^p$, and thus $2^i/2^p \le \lfloor q^i \rfloor^p \le 2^i$. Thus, $\left\lfloor q^{i(z)} \right\rfloor^{-p} L''$ is upper-bounded by

$$\left\lfloor q^{i(z)} \right\rfloor^{-p} \sum_{i=0}^{i(z)} \left\lfloor q^i \right\rfloor^p \cdot ck \le 2^{p-i(z)}(2^{i(z)+1} - 1)ck \le 2^{p+1}ck,$$

implying our desired additive approximation is guaranteed by obtaining a $(1 \pm \varepsilon')$-multiplicative approximation to L'' for $\varepsilon' = (\sqrt{ck}/4)/(4 \cdot 2^{p+1}ck) = 1/(2^{p+3}\sqrt{ck})$. By choice of k, this is a $(1 \pm O(\varepsilon/2^p))$-multiplicative approximation, which we can obtain from A by setting r to be a sufficiently small constant. Recalling that A provides this $(1 \pm O(\varepsilon/2^p))$-approximation with probability at least $11/12$, we solve Gap-Hamdist in the block $i(z)$ with probability at least $11/12$, and thus Indexing in block $i(z)$ with probability at least $2/3$ by Theorem 69, which is equivalent to solving the original Augmented-Indexing instance. This implies that the total number of bits communicated in this protocol must be $\Omega(kt)$. Now note that the only bits communicated other than the state of A are the transmissions of $w(y_i)$ for $0 \le i < t$. Since $w(y_i) \le k$, all Hamming weights can be communicated in $O(t \log(k)) = o(kt)$ bits. Thus, indeed, the communication of the protocol is dominated by the space complexity of A, implying A uses space $\Omega(kt)$.

The above argument yields the lower bound $\Omega(\min\{d, m, \varepsilon^2 \log(M)\})$. We can similarly obtain the lower bound $\Omega(\min\{d, m, \varepsilon^2 \log(\lceil \varepsilon^2 m \rceil)\})$ by, rather than updating an item in the stream by $f_i = \lfloor q^i \rfloor$ in one update, we update the same item f_i times by 1. The number of total updates in the ith block is then at most $\lfloor q^i \rfloor \cdot k$, and thus the maximum number of blocks of length k we can give Alice to ensure that both the stream length and number of used universe elements is at most $\min\{d, m\}$ is $t = \min\{\lfloor \min\{d,m\}/k \rfloor, O(\log(\lceil m/k \rceil))\}$. The proof is otherwise identical. ■

Our lower bound technique also improves lower bounds for $p = 0$, ℓ_p estimation in the strict turnstile model (where we are promised $x_i \geq 0$ always). We first make the following simple observation.

Observation 71. *For vectors u, v of equal length with entries in $\{0, r\}$, let $\Delta(u, v) = |\{i : u_i \neq v_i\}|$ denote their Hamming distance. Let $w(z) = |\{i : z_i \neq 0\}|$ denote the weight of z. Then for any $p \geq 0$,*

$$r^p(2^p - 2)\Delta(u, v) = (2r)^p w(u) + (2r)^p w(v) - 2\|u + v\|_p^p.$$

Theorem 72. *For any real constant $p > 0$, any one-pass streaming algorithm for $(1 \pm \varepsilon)$-multiplicative approximation of F_p with probability at least $11/12$ in the strict turnstile model requires $\Omega(\min\{d, m, |p-1|^2\varepsilon^{-2}(\log(\varepsilon^2 mM/|p-1|^2))\})$ bits of space.*

Proof (Sketch). The proof is very similar to that of Theorem 70, so we only explain the differences. The main difference is the following. In the proof of Theorem 70, Bob inserted item $(i(z), j)$ into the stream with frequency $-\lfloor q^{i(z)} \rfloor$ for each j satisfying $y(B)_j = 1$. Doing this may not yield a strict turnstile stream, since $(i(z), j)$ may never have received a positive update from Alice. We instead have Bob insert $(i(z), j)$ with *positive* frequency $\lfloor q^{i(z)} \rfloor$.

Now, after all updates have been inserted into the stream, Observation 71 implies that the pth frequency moment of the stream is exactly

$$\begin{aligned} L'' = & \frac{(2\lfloor q^{i(z)} \rfloor)^p}{2} w(y_{i(z)}) + \frac{(2\lfloor q^{i(z)} \rfloor)^p}{2} w(y(B)) \\ & - \frac{\lfloor q^{i(z)} \rfloor^p (2^p - 2)}{2} \Delta(y_{i(z)}, y(B)) \\ & + \sum_{i < i(z)} w(y_i) \lfloor q^i \rfloor^p . \end{aligned}$$

Setting $\eta = \sum_{i<i(z)} w(y_i) \lfloor q^i \rfloor^p$ and rearranging terms,

$$\begin{aligned} \Delta(y_{i(z)}, y(B)) = & \frac{2^{p-1}}{2^{p-1} - 1} w(y_{i(z)}) + \frac{2^{p-1}}{2^{p-1} - 1} w(y(B)) \\ & + \frac{\lfloor q^{i(z)} \rfloor^{-p} (\eta - L'')}{2^{p-1} - 1}. \end{aligned}$$

Bob knows η, $w(y_{i(z)})$, and $w(y(B))$ exactly. To decide Gap-Hamdist for vectors $y_{i(z)}, y(B)$, it thus suffices to obtain a $((2^{p-1} - 1)/(4\sqrt{ck}))$-additive approximation to $\lfloor q^{i(z)} \rfloor^{-p} L''$. Since $2^{-i(z)} L''$ is upper-bounded in absolute value by $(1 + 2^p)ck$, our desired additive approximation is guaranteed by obtaining a $(1 \pm ((2^{p-1}-1)\sqrt{ck}/(4 \cdot (1+2^p))))$-multiplicative approximation to L''. Since $p \neq 1$ is a constant, $|2^x - 1| = \Theta(|x|)$ as $|x| \to 0$, and $k = \Theta(1/\varepsilon^2)$, this is a $(1 \pm O(|p-1|\varepsilon))$-multiplicative approximation. To conclude, a $(1 \pm O(|p-1|\varepsilon))$-multiplicative approximation to F_p with probability $11/12$ gives a protocol for Augmented-Indexing with success probability $2/3$, with Alice having a string of length kt for k, t as in the proof of Theorem 70. The theorem follows. ∎

The lower bounds of Theorem 70 and Theorem 72 fail to give any improved lower bound over the previously known $\Omega(\min\{d, m, 1/\varepsilon^2\})$ lower bound for p near (and including) 0. The reason is that we gave items in block j a frequency of roughly $2^{j/p}$, so that contributions to F_p increase geometrically as block ID increases. This fails for, say, $p = 0$, since in this case increasing frequency does not increase contribution to F_0 at all. We fix this issue by, rather than giving items in large blocks a large frequency, instead blow up the universe size. Specifically, we use a proof identical to that of Theorem 72, but rather than give an index i in block j frequency roughly $2^{j/p}$, we instead create 2^j indices $(i, 1), \ldots, (i, 2^j)$ and give them each a frequency of 1. The setting of t, the number of blocks, can then be at most $O(\log(\varepsilon^2 \min\{d, m\}))$ since $n, m \leq 2\varepsilon^{-2} \sum_{j=0}^{t-1} 2^j$, which we require to be at most $\min\{d, m\}$. We thus have:

Theorem 73. *For any real constant $p \geq 0$, one-pass $(1 \pm \varepsilon)$-multiplicative approximation of F_p with probability at least $11/12$ in the strict turnstile model requires $\Omega(|p-1|^2 \varepsilon^{-2} \log(\varepsilon^2 \min\{d, m\}/|p-1|^2))$ bits of space.*

The decay of our lower bounds in the strict turnstile model as $p \to 1$ is necessary since Li gave an algorithm in this model whose dependence on ε becomes subquadratic as $p \to 1$ [88]. Furthermore, when $p = 1$ there is a simple, deterministic $O(\log(mM))$-space algorithm for computing F_1: maintain a counter.

3.5 The additive $\log \log d$ in ℓ_p bounds

Throughout this chapter and in the introduction and in the coming chapter on entropy estimation (Chapter 4), we omit an additive $\log \log d$ in the statements of various upper and lower bounds. We here discuss why this term arises.

3.5.1 The upper bounds

Recall we assume that stream updates (i, v) come from $[d] \times \{-M, \ldots, M\}$, causing the change $x_i \leftarrow x_i + v$. We note that in fact, we can assume that i comes not from the universe $[d]$, but rather from $[U]$ for $U = O(m^2)$, at the cost of an additive $O(\log \log d)$ in the space complexity of any streaming algorithm with constant error in the turnstile model. We discuss why this is the case below, using an idea of [51].

Let $\{i_1, \ldots, i_r\}$ be the indices appearing in the stream. Picking a prime q and treating all updates (i, v) as $(i \bmod q, v)$, our ℓ_p estimate is unaffected as long as $i_{j_1} \neq i_{j_2} \bmod q$ for all $j_1 \neq j_2$. There are at most $r^2/2$ differences $|i_{j_1} - i_{j_2}|$, and each difference is an integer bounded by d, thus having at most $\log d$ prime factors.

There are thus at most $r^2 \log d$ prime factors dividing *some* $|i_{j_1} - i_{j_2}|$. If we pick a random prime $q = \widetilde{O}(r^2 \log d)$, we can ensure with constant probability arbitrarily close to 1 (by increasing the constant in the "big-Oh") that no indices collide modulo q. Since $r \le m$, we thus have $q = \widetilde{O}(m^2 \log d)$. We then pick a hash function $h : \{0, \ldots, q-1\} \to [O(m^2)]$ at random from pairwise independent family. With constant probability which can be made arbitrarily high (again by increasing the constant in the "big-Oh"), the mapping $i \mapsto h(i \bmod q)$ perfectly hashes the indices appearing in the stream. Storing both h and q requires $O(\log q + \log m) = O(\log m + \log\log d)$ bits.

3.5.2 The lower bounds

An $\Omega(\log\log d)$ lower bound for moment estimation in the turnstile model follows by a simple reduction from the communication problem of Equality in the private coin model. The lower bound holds for streams of length at least 2 and arbitrary M. In Equality, Alice receives $x \in [d]$, and Bob receives $y \in [d]$, and they must decide whether $x = y$ with probability $2/3$. This problem requires $\Omega(\log\log d)$ communication [84]. Given a streaming algorithm A for turnstile F_p estimation for $p \ge 0$ with success probability $2/3$, Alice performs the update $(x, +1)$, sends A's state to Bob, and Bob performs the update $(y, -1)$. Either $x = y$, implying $F_p = 0$, else $F_p \neq 0$. Thus, any multiplicative approximation of F_p gives a solution to Equality, implying A uses $\Omega(\log\log d)$ space. This same argument also applies for entropy estimation, or for any other problem in which $x = 0$ must be distinguished from $x \neq 0$.

Chapter 4

Entropy Estimation

In this chapter we describe our algorithm for additive and multiplicative empirical entropy estimation in turnstile streams. Recall that for a discrete probability distribution with event probabilities $q_1, \ldots, q_n$, its *entropy* is defined as

$$H = -\sum_{i=1}^{n} q_i \ln q_i.$$

Throughout this chapter we consider the case $q_i = |x_i|/\|x\|_1$ for a vector x updated in a turnstile stream, and we would like to additively or multiplicatively estimate H given some error parameter $0 < \varepsilon < 1/2$. So that H is well-defined, we assume $x \neq 0$; the case $x = 0$ can be detected with probability $1 - \delta$ by taking the dot product of x with a random sign vector drawn from a 4-wise independent family $O(\log(1/\delta))$ times, then checking whether any dot product is 0 [14].

Let $F_p(q)$ denote the pth frequency moment of the q vector, and let F_p as usual denote the pth frequency moment of x. The main idea behind all our algorithms in this chapter is that if one defines the *pth Tsallis entropy* T_p by

$$T_p \stackrel{\text{def}}{=} \frac{1 - F_p(q)}{p - 1},$$

then L'Hôpital's rule easily shows that $\lim_{p \to 1} T_p = H$. Thus, at least for additive entropy estimation, it suffices to take some $p = p(\varepsilon)$ sufficiently close to 1 and produce an estimate $\tilde{F}_p(q) = (1 \pm \tilde{\varepsilon})F_p(q)$, since

$$\tilde{T}_p \stackrel{\text{def}}{=} \frac{1 - \tilde{F}_p(q)}{p - 1} = T_p \pm \frac{\tilde{\varepsilon}}{p - 1}.$$

One then should choose p so that $|T_p - H| < \varepsilon/2$, then choose $\tilde{\varepsilon} = \varepsilon(p-1)/2$. In other words, entropy estimation with additive error directly reduces to moment estimation with multiplicative error, albeit with a different error parameter $\tilde{\varepsilon}$. Unfortunately, consider the vector $x = (1, \ldots, 1)$ formed by a stream of length $m = d$. This vector yields $H = \ln m$, however

$$T_{1+\varepsilon/\ln m} = \frac{\ln m}{\varepsilon} \cdot \left(1 - m \cdot \left(\frac{1}{m}\right)^{1+\varepsilon/\ln m}\right) = \frac{\ln m}{\varepsilon} \cdot \left(1 - e^{-\varepsilon}\right) = (1 + \Theta(\varepsilon)) \ln m.$$

A similar calculation can be performed for $T_{1-\varepsilon/\ln m}$. Thus, in general ε-additive entropy estimation requires setting $|p-1| = 1+O(\varepsilon/\ln^2 m)$, and thus requires setting $\tilde{\varepsilon} = O(\varepsilon^2/\ln^2 m)$. Due to Theorem 70, this reduction cannot yield an algorithm using $o(1/\varepsilon^4)$ space. Our approach circumvents this problem using interpolation. Specifically, we estimate T_p for many p, then interpolate a polynomial which we evaluate at $p = 1$.

Remark 74. In the case of the strict turnstile model, the above argument does not actually imply that a direction reduction to moment estimation would require $\Omega(1/\varepsilon^4)$ space. This is because the lower bound for moment estimation in the strict turnstile model is only $\Omega(\varepsilon^{-2}|p-1|^2)$ (Theorem 72). Recall that when $p = 1$, a very simple algorithm estimates F_1 with no error: maintain a counter.

We assume in our algorithms that m, the length of the stream, is known in advance, though in fact our algorithms work with only a constant factor increase in space as long as a value m' satisfying $m \le m' \le m^{O(1)}$ is known. In fact, we actually only need to know an upper bound for $\|x\|_\infty$, which mM provides.

Bibliographic remarks: The algorithms in this chapter were developed in joint work with Harvey and Onak [63]. The lower bound in Section 4.4 was developed in joint work with Kane and Woodruff [78]. See also a later improvement to the lower bound by Jayram and Woodruff [72] which takes error probability into account.

4.1 Noisy Extrapolation

In this section we describe an extrapolation technique that lies at the heart of our streaming algorithms for entropy estimation. 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 $f(0)$. We do this by approximating 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, where Δ_i is the error term. We then interpolate to obtain the only polynomial h of degree at most k such that $h(y_i) = z_i$ and hope that $h(0)$ is a good approximation to $f(0)$.

The polynomial h can be decomposed into two polynomials h_f and h_Δ of degree at most k such that $h = h_f + h_\Delta$, and for each i, $h_f(y_i) = f(y_i)$ and $h_\Delta(y_i) = \Delta_i$. We have

$$|h(0) - f(0)| \le |h_f(0) - f(0)| + |h_\Delta(0)|. \tag{4.1}$$

We analyze and bound each of the terms on the right hand side of Eq. (4.1) 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 requires a careful choice of y_i and employs extremal properties of Chebyshev polynomials.

4.1.1 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.

Fact 75 (Phillips and Taylor [107], 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*

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

where $h_f(y)$ 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.

4.1.2 Bounding the Second Error Term

We now show how to bound $|h_\Delta(0)|$, the error from learning 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 $|h_\Delta(0)|$. 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 h_Δ can be bounded at 0.

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 [106, 107, 108]. A thorough treatment of these objects can be found in [110]. We now present the background relevant for our purposes.

Definition 76. *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*

$$P_k(x) \;=\; \begin{cases} 1, & (k = 0) \\ x, & (k = 1) \\ 2xP_{k-1}(x) - P_{k-2}(x), & (k \ge 2) \end{cases} \tag{4.2}$$

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* $h \in \mathcal{P}_k$ *satisfying* $\max_{0 \le j \le k} |h(\eta_{j,k})| \le 1$.

The following can be found in [110, Ex. 1.5.11] or [111].

Fact 77 (Extremal Growth Property)**.** *If* $h \in \mathcal{C}_k$ *and* $|t| \ge 1$*, then* $|h(t)| \le |P_k(t)|$.

Fact 77 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.

The Choice of y_i

We will use Fact 77 to bound $h_\Delta(0)$. Since this fact provides no guarantees at $t = 0$, we produce a new polynomial from h_Δ by applying an affine map to its domain, then bound this new polynomial at a point t slightly larger than 1. Fact 77 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 4.2 to control how close the y_i's are to 0, and consequently the efficiency of estimating $f(y_i)$.

Formally, define

$$\begin{aligned} g_\ell(t) &\stackrel{\text{def}}{=} \left(\frac{\ell}{2k^2+1}\right)\left(k^2 \cdot t \;-\; (k^2+1)\right), \quad \text{and} \\ y_i &\stackrel{\text{def}}{=} g_\ell(\eta_{i,k}). \end{aligned} \tag{4.3}$$

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

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

Let Δ be an upper bound on the error with which we can approximate f, i.e., $|\Delta_i| = |h_\Delta(y_i)| \le \Delta$. Then the polynomial $\tilde{h}_\Delta(t) \stackrel{\text{def}}{=} h_\Delta(g_\ell(t))/\Delta$ has the property $|\tilde{h}_\Delta(\eta_{i,k})| \le 1$, that is, $\tilde{h}_\Delta(t)$ belongs to $\mathcal{C}_k$. Furthermore, $h_\Delta(0) = \Delta \cdot \tilde{h}_\Delta(g_\ell^{-1}(0)) = \Delta \cdot \tilde{h}_\Delta(1+1/k^2)$. By Fact 77, we then have $|h_\Delta(0)| = \Delta \cdot |\tilde{h}_\Delta(1+1/k^2)| \le \Delta \cdot |P_k(1+1/k^2)|$. To finish bounding $|h_\Delta(0)|$, we use the following standard lemma (in fact even sharper bounds can be obtained, see [109]).

Lemma 78. *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. The proof is by induction and Eq. (4.2). Fix $x = 1 + k^{-c}$. We will prove for all $j \ge 1$ that

$$P_{j-1}(x) \;\le\; P_j(x) \;\le\; P_{j-1}(x)\left(1 + \frac{2j}{k^c}\right).$$

Additive entropy estimation

1. $k \leftarrow 2 \cdot \max\{\ln(8e/\varepsilon)/\ln\ln(8e/\varepsilon), (\ln\ln m)/(\ln\ln\ln m)\}$
2. $\tilde{\varepsilon} \leftarrow \varepsilon/(4e^3(k+1)(2k^2+1)\ln mM)$
3. Define $y_0, \ldots, y_k$ as in Eq. (4.3), with $\ell = 1/(e(k+1)\ln mM)$
4. For each $i \in \{0, \ldots, k\}$, compute $\tilde{F}_{1+y_i}$, a $(1+\tilde{\varepsilon})$-approximation of F_{1+y_i} with success probability $1 - 1/(6(k+1))$
5. Compute $\tilde{F}_1$, a $(1+\tilde{\varepsilon})$-approximation of F_1 with success probability $5/6$
6. For each i, compute $\tilde{T}(y_i) = \left(1 - \tilde{F}_{1+y_i}/\tilde{F}_1^{1+y_i}\right)/y_i$
7. Return an estimate of $T(0)$ by polynomial interpolation using the points $\tilde{T}(y_i)$

Figure 4-1: Algorithm for additively approximating empirical entropy.

For the first inequality, we observe $P_{j-1} \in \mathcal{C}_j$, so we apply Fact 77 together with the fact that $P_j(y)$ is strictly positive for all $y > 1$ and all j.

For the second inequality, we induct on j. For the sake of the proof define $P_{-1}(x) = 1$ so that the inductive hypothesis holds at the base case $j = 0$. For the inductive step with $j \geq 1$, we use the definition of $P_j(x)$ in Eq. (4.2) and we have

$$\begin{aligned}
P_{j+1}(x) &= P_j(x)\left(1 + \frac{2}{k^c}\right) + (P_j(x) - P_{j-1}(x)) \\
&\leq P_j(x)\left(1 + \frac{2}{k^c}\right) + P_{j-1}(x)\frac{2j}{k^c} \\
&\leq P_j(x)\left(1 + \frac{2}{k^c}\right) + P_j(x)\frac{2j}{k^c} \\
&= P_j(x)\left(1 + \frac{2(j+1)}{k^c}\right).
\end{aligned}$$

■

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

4.2 Additive Estimation Algorithm

In this section, we apply the noisy extrapolation technique of Section 4.1 to ε-additive entropy estimation. Our algorithm is in Figure 4-1. There and henceforth, we use the notation $T(y)$ to denote T_{1+y}. The algorithm estimates Tsallis entropy with error parameter $\tilde{\varepsilon} = \varepsilon/(4e^3(k+1)(2k^2+1)\ln mM)$, where $k = 2 \cdot \max\{\ln(8e/\varepsilon)/\ln\ln(8e/\varepsilon), (\ln\ln m)/(\ln\ln\ln m)\}$. We define the points $y_0, y_1, \ldots, y_k$ by setting $\ell = 1/(e(k+1)\ln mM)$ and $y_i = g_\ell(\eta_{i,k})$, as in Eq. (4.3).

The correctness of the algorithm is proven in Section 4.2. For the space requirements, we use either of the algorithms in Chapter 3. To compute each estimate $\tilde{F}_{1+y_i}$, the space required is $O(\tilde{\varepsilon}^{-2}\log(mM)\log k)$. The space required for estimating F_1 is $O(\tilde{\varepsilon}^{-2}\log(mM))$. Thus, the total space is $O(\varepsilon^{-2}k^6\log k\log^3(mM)) = \tilde{O}(\varepsilon^{-2}\log^3 mM)$. Using the faster algorithm from Section 3.3, the update time is $O(k\log k\log^2(1/\tilde{\varepsilon})\log\log(1/\tilde{\varepsilon})) = \mathrm{polylog}((\log mM)/\varepsilon)$.

A Technical Claim

Before showing correctness of our algorithm, we first need a bound on the high order derivatives of T. When analyzing correctness, we will plug this bound into Fact 75 to bound the first error term in Eq. (4.1) in our approximation by interpolation.

Lemma 79. *Let ε be in $(0, 1/2]$. Then,*

$$\left|T^{(k)}\left(-\frac{\varepsilon}{(k+1)\ln\|x\|_\infty}\right)\right| \;\leq\; 4\left(\ln^k\|x\|_\infty\right)H/(k+1).$$

Before proving Lemma 79, we first define a a family of functions $\{G_t\}_{t=0}^{\infty}$ and prove a few lemmas. For any integer $t \geq 0$ and real number $a \geq -1$, define

$$G_t(a) \;=\; \sum_{i=1}^{d} q_i^{1+a}\,\ln^k q_i,$$

so $G_0(a) = F_{1+a}/\|x\|_1^{1+a}$. Note that $G_t^{(1)}(a) = G_{t+1}(a)$ for $t \geq 0$, and $T(a) = (1-G_0(a))/a$.

Lemma 80. *The tth derivative of the Tsallis entropy has the following expression.*

$$T^{(t)}(a) \;=\; \frac{(-1)^t\,t!\,\big(1-G_0(a)\big)}{a^{t+1}} \;-\; \left(\sum_{j=1}^{t}\frac{(-1)^{t-j}\,k!\,G_j(a)}{a^{t-j+1}j!}\right)$$

Proof. The proof is by induction on t, the case $t = 0$ being trivial. Now, assume

$t \geq 1$. Taking the derivative of the expression for $T^{(t)}(a)$ above, we obtain:

$$\begin{aligned}
&T^{(t+1)}(a) \\
&= \left(\sum_{j=1}^{t} \frac{t!(t-j+1)(-1)^{(t+1)-j} G_j(a)}{a^{(t+1)-j+1} j!} + \frac{t!(-1)^{t-j} G_{j+1}(a)}{a^{t-j+1} j!} \right) \\
&\quad + \frac{(-1)^{t+1}(t+1)!(G_0(a)-1)}{a^{t+2}} + \frac{(-1)^t t! G_1(a)}{a^{t+1}} \\
&= \left(\sum_{j=1}^{t} \frac{t!(-1)^{(t+1)-j} G_j(a)}{a^{(t+1)-j+1}(j-1)!} \left(1 + \frac{t-j+1}{j}\right) \right) \\
&\quad + \frac{G_{t+1}(a)}{a} + \frac{(-1)^{t+1}(t+1)!(G_0(a)-1)}{a^{t+2}} \\
&= \left(\sum_{j=1}^{t+1} \frac{(t+1)!(-1)^{(t+1)-j} G_j(a)}{a^{(t+1)-j+1} j!} \right) + \frac{(-1)^{t+1}(t+1)!(G_0(a)-1)}{a^{t+2}}
\end{aligned}$$

as claimed. ∎

Lemma 81. *Define $S_t(a) = a^{t+1} T^{(t)}(a)$. Then, for $1 \leq j \leq t+1$,*

$$S_t^{(j)}(a) = \sum_{i=0}^{j-1} \binom{j-1}{i} \frac{t!}{(t-j+i+1)!} a^{t-j+i+1} G_{t+1+i}(a)$$

In particular, for $1 \leq j \leq t$, we have

$$\lim_{a \to 0} S_t^{(j)}(a) = 0 \quad \textit{and} \quad \lim_{a \to 0} S_t^{(t+1)}(a) = t!\, G_{t+1}(0) \quad \textit{so that} \quad \lim_{a \to 0} T^{(t)}(a) = \frac{G_{t+1}(0)}{t+1}.$$

Proof. We prove the claim by induction on j. First, note

$$S_t(a) = (-1)^t t!(1 - G_0(a)) - \left(\sum_{j=1}^{t} \frac{a^j (-1)^{t-j} t! G_j(a)}{j!} \right)$$

so that

$$\begin{aligned}
S_t^{(1)}(a) &= (-1)^{t-1} t! G_1(a) \\
&\quad - \left(\sum_{j=1}^{t} -\frac{a^{(j+1)-1}(-1)^{t-(j+1)} t! G_{j+1}(a)}{((j+1)-1)!} + \frac{a^{j-1}(-1)^{t-j} t! G_j(a)}{(j-1)!} \right) \\
&= a_{t+1}^{t}(a)
\end{aligned}$$

Thus, the base case holds. For the inductive step with $2 \leq j \leq t+1$, we have

$$\begin{aligned}
S_t^{(j)}(a) &= \frac{\partial}{\partial a}\left(\sum_{i=0}^{j-2}\binom{j-2}{i}\frac{t!}{(t-j+i+2)!}a^{t-j+i+2}G_{t+1+i}(a)\right) \\
&= \sum_{i=0}^{j-2}\left(\binom{j-2}{i}\frac{t!}{(t-j+i+1)!}a^{t-j+i+1}G_{t+1+i}(a)\right. \\
&\quad \left. + \binom{j-2}{i}\frac{t!}{(t-j+(i+1)+1)!}a^{t-j+(i+1)+1}G_{t+1+(i+1)}(a)\right) \\
&= \sum_{i=0}^{j-1}\binom{j-1}{i}\frac{t!}{(t-j+i+1)!}a^{t-j+i+1}G_{t+1+i}(a)
\end{aligned}$$

The final equality holds since $\binom{j-2}{0} = \binom{j-1}{0} = 1$, $\binom{j-2}{j-2} = \binom{j-1}{j-1} = 1$, and by Pascal's formula $\binom{j-2}{i} + \binom{j-2}{i+1} = \binom{j-1}{i+1}$ for $0 \leq i \leq j-3$.

For $1 \leq j \leq t$, every term in the above sum is well-defined for $a = 0$ and contains a power of a which is at least 1, so $\lim_{a\to 0} S_t^{(j)}(a) = 0$. When $j = t+1$, all terms but the first term contain a power of a which is at least 1, and the first term is $t! \cdot G_{t+1}(a)$, so $\lim_{a\to 0} S_t^{(t+1)}(a) = t! G_{t+1}(0)$. The claim on $\lim_{a\to 0} T(t)(a)$ thus follows by writing $T^{(t)}(a) = S_t(a)/a^{t+1}$ then applying L'Hôpital's rule $t+1$ times. ■

To finally finish off our proof of Lemma 79, we need the following form of L'Hôpital's rule.

Theorem 82 (see Rudin [112, p.109]). *Let $f : \mathbb{R} \to \mathbb{R}$ and $g : \mathbb{R} \to \mathbb{R}$ be differentiable functions such that the following limits exist*

$$\lim_{\alpha\to 1} f(\alpha) = 0, \qquad \lim_{\alpha\to 1} g(\alpha) = 0, \quad \text{and} \quad \lim_{\alpha\to 1} f'(\alpha)/g'(\alpha) = L.$$

Let ε and δ be such that $|\alpha - 1| < \delta$ implies that $|f'(\alpha)/g'(\alpha) - L| < \varepsilon$. Then $|\alpha - 1| < \delta$ also implies that $|f(\alpha)/g(\alpha) - L| < \varepsilon$.

Proof (of Lemma 79). We will first show that

$$\left| T^{(t)}\left(-\frac{\varepsilon}{(t+1)\ln\|x\|_\infty}\right) - \frac{G_{t+1}(0)}{t+1}\right| \leq \frac{6\varepsilon \ln^t(\|x\|_\infty) H}{t+1}$$

Let $S_t(a) = a^{t+1}T^{(t)}(a)$ and note $T^{(t)}(a) = S_t(a)/a^{t+1}$. By Lemma 80, $\lim_{a\to 0} S_t(a) = 0$. Furthermore, $\lim_{a\to 0} S_t^{(j)} = 0$ for all $1 \leq j \leq t$ by Lemma 81. Thus, when analyzing $\lim_{a\to 0} S_t^{(j)}(a)/(a^{t+1})^{(j)}$ for $0 \leq j \leq t$, both the numerator and denominator approach 0 and we can apply L'Hôpital's rule (here $(a^{t+1})^{(j)}$ denotes the jth derivative of the function a^{t+1}). By $t+1$ applications of L'Hôpital's rule, we can thus say that $T^{(t)}(a)$ converges to its limit at least as quickly as $S_t^{(t+1)}(a)/(a^{t+1})^{(t+1)} = S_t^{(t+1)}(a)/(t+1)!$ does (using Theorem 82). We note that $G_j(a)$ is nonnegative for j even and nonpositive otherwise. Thus, for negative a, each term in the summand of the expression for $S_t^{(t+1)}(a)$ in Lemma 81 is nonnegative for odd t and nonpositive for even t. As the analyses for even and odd t are nearly identical, we focus below on odd t, in which case every term in the summand is nonnegative. For odd t, $S_t^{(t+2)}(a)$ is nonpositive so that $S_t^{(t+1)}(a)$ is monotonically decreasing. Thus, it suffices to show that $S_t^{(t+1)}(-\varepsilon/((t+1)\ln\|x\|_\infty))/(t+1)!$ is not much larger than its limit.

$$
\begin{aligned}
\frac{S_t^{(t+1)}\left(-\frac{\varepsilon}{(t+1)\ln\|x\|_\infty}\right)}{(t+1)!} &= \frac{\sum_{i=0}^{t}\binom{t}{i}\frac{t!}{i!}\left(-\frac{\varepsilon}{(t+1)\ln\|x\|_\infty}\right)^i G_{t+1+i}\left(-\frac{\varepsilon}{(t+1)\ln\|x\|_\infty}\right)}{(t+1)!} \\
&\le \frac{1+2\varepsilon}{t+1}\sum_{i=0}^{t}\binom{t}{i}\left(\frac{\varepsilon}{(t+1)\ln\|x\|_\infty}\right)^i |G_{t+1+i}(0)| \\
&\le \frac{1+2\varepsilon}{t+1}\sum_{i=0}^{t} t^i\left(\frac{\varepsilon}{(t+1)\ln\|x\|_\infty}\right)^i |G_{t+1+i}(0)| \\
&\le \frac{1+2\varepsilon}{t+1}\sum_{i=0}^{t}\left(\frac{\varepsilon}{\ln\|x\|_\infty}\right)^i |G_{t+1+i}(0)| \\
&\le \frac{1+2\varepsilon}{t+1}\sum_{i=0}^{t}\varepsilon^i |G_{t+1}(0)| \\
&\le \frac{(1+2\varepsilon)|G_{t+1}(0)|}{t+1} + \frac{1+2\varepsilon}{t+1}\sum_{i=1}^{t}\varepsilon^i |G_{t+1}(0)| \\
&\le \frac{(1+2\varepsilon)|G_{t+1}(0)|}{t+1} + \frac{2}{t+1}\sum_{i=1}^{t}\varepsilon^i (\ln^t\|x\|_\infty)H \\
&\le \frac{|G_{t+1}(0)|}{t+1} + \frac{6\varepsilon(\ln^t\|x\|_\infty)H}{t+1}
\end{aligned}
$$

The first inequality holds since $q_i \ge 1/\|x\|_\infty$ for each i with $q_i \neq 0$, so that $q_i^{-\varepsilon/((t+1)\ln\|x\|_\infty)} \le \|x\|_\infty^{\varepsilon/((t+1)\ln\|x\|_\infty)} \le \|x\|_\infty^{\varepsilon/\ln m} \le e^\varepsilon \le 1+2\varepsilon$ for $\varepsilon \le 1/2$. The final inequality above holds since $\varepsilon \le 1/2$.

The lemma follows since $|G_{t+1}(0)| \le (\ln^t\|x\|_\infty)H$. ■

Correctness

We now prove correctness of our algorithm. It suffices to bound the two error terms in Eq. (4.1). We show that both terms are at most $\varepsilon/2$.

To bound the first error term, we use Fact 75. 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 $\ell = 1/(e(k+1)\ln mM)$, Lemma 79 implies that

$$|T^{(k+1)}(\xi)| \ \le\ \frac{4\,(\ln^{k+1}\|x\|_\infty)\,H}{k+2} \quad \forall \xi \in [-\ell, 0). \tag{4.5}$$

Thus, by Fact 75 and Eq. (4.5), and using $\|x\|_\infty \le mM$ and $H \le \ln m$, we have

$$\begin{aligned} |T(0) - h_T(0)| &\le |\ell|^{k+1} \cdot \frac{4\ \ln^{k+1}(mM)\, H}{(k+1)!\,(k+2)} \\ &\le \frac{1}{e^{k+1}\ln^{k+1}(mM)} \cdot \frac{4\ \ln^{k+1}(mM)\, H}{(k+2)!} \\ &\le \frac{4eH}{(k+2)^{k+2}} \le \frac{\varepsilon}{2}. \end{aligned} \tag{4.6}$$

We now have to bound the second error term $|h_\Delta|$, since Figure 4-1 does not compute the exact values $T(y_i)$ but rather only computes approximations. The accuracy of these approximations can be determined as follows.

$$\tilde{T}(y_i) = \frac{1 - \tilde{F}_{1+y_i}/\|x\|_1^{1+y_i}}{y_i} \le T(y_i) - \tilde{\varepsilon} \cdot \frac{\sum_{j=1}^{d} q_j^{1+y_i}}{y_i}. \tag{4.7}$$

To analyze the last term, recall that $q_j \ge 1/mM$ for each i and $y_i \ge -\ell$, so that $q_i^{y_i} \le (mM)^\ell = (mM)^{1/e(k+1)\ln mM} < 2$. Thus $\sum_{j=1}^{d} q_j^{1+y_i} < 2\sum_{j=1}^{d} q_j = 2$. By Eq. (4.4),

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

Thus, we have

$$T(y_i) \le \tilde{T}(y_i) \le T(y_i) + \frac{\varepsilon}{2e^2}. \tag{4.8}$$

Hence, the additive error $|h_\Delta(y_i)| = |\Delta_i|$ on each $T(y_i)$ is bounded by $\Delta = \varepsilon/(2e^2)$. In Section 4.1, we showed that $|h_\Delta(0)| \le e^2 \cdot \Delta \le \frac{\varepsilon}{2}$. This completes the analysis.

4.3 Multiplicative Estimation Algorithm

We now discuss how to obtain a multiplicative approximation of entropy. The main modification that we require is a multiplicative estimate of Tsallis entropy.

Our multiplicative algorithm for entropy estimation is similar to Figure 4-1, though we make certain modifications. We set the number of interpolation points to be $k = \ln(8e/\varepsilon)/\ln\ln(8e/\varepsilon)$ and keep ℓ the same. The y_i are as in Eq. (4.3). We then argue as in Eq. (4.6) to have $|T(0) - h_T(0)| \le \varepsilon H/2$, where h_T is the interpolated polynomial of degree k at the true Tsallis entropies, without error. We then let $\tilde{T}(y_i)$ be a $(1+\tilde{\varepsilon})$-*multiplicative* estimation of $T(y_i)$ instead of an $\tilde{\varepsilon}$-additive estimation by using Theorem 87 below, where we set $\tilde{\varepsilon} = \varepsilon/(8e^2)$. We then have $T(y_i) \le \tilde{T}(y_i) \le T(y_i) + \tilde{\varepsilon}T(y_i) \le T(y_i) + \varepsilon H/(2e^2)$. The final inequality follows from Lemma 79 with $k = 0$. We then have

$$|h(0) - H| \le |h_T(0) - H| + |h_\Delta(0)| \le \varepsilon H/2 + e^2 \cdot (\varepsilon H/(2e^2)) = \varepsilon H.$$

Our algorithm for $(1 \pm \varepsilon)$-approximation of T_p requires $O(\log(mM)/(|p-1|^2\varepsilon^2))$

space and $O(\log(1/(|p-1|\varepsilon))\log\log(1/(|p-1|\varepsilon)))$ update time (Theorem 87), and we use error parameter $\varepsilon/(8e^2)$ and error probability $1/(6(k+1))$. All our values for the y_i give $|p-1| = \Omega(\ell/k^2)$. Thus, our overall space is $O(k^6 \log k \log^2(mM)/\varepsilon^2)$, and our update time is $O(k \log(\log(mM)/\varepsilon) \log\log(\log(mM)/\varepsilon))$. This gives the following theorem, which is the main theorem of this section.

Theorem 83. *There is an algorithm for $(1 \pm \varepsilon)$-approximation of entropy with error probability $1/6$ using $\tilde{O}(\varepsilon^2 \log^2(mM))$ space, and with $\tilde{O}(\log(\log(mM)/\varepsilon))$ update time.*

We now describe our algorithm for multiplicative estimation of Tsallis entropy. First we must prove a few facts. The next lemma says that if there is no heavy element in the empirical distribution, then Tsallis entropy is at least a constant.

Lemma 84. *Let $q_1, q_2, \ldots, q_d$ be values in $[0,1]$ of total sum 1. There exists a positive constant C independent of d such that if $q_i \le 5/6$ for all i then, for $p \in (0,1) \cup (1,2]$,*

$$\left|1 - \sum_{i=1}^{d} q_i^p\right| \ge C \cdot |p-1|.$$

Proof. Consider first $p \in (0,1)$. For $y \in (0, 5/6]$,

$$y^{p-1} \ge \left(\frac{5}{6}\right)^{p-1} \ge 1 + C_1 \cdot (1-p),$$

for some positive constant C_1. The last equality follows from convexity of $(5/6)^z$ as a function of z. Hence,

$$\sum_{i=1}^{d} q_i^p \ge \sum_{i=1}^{d} (1 + C_1(1-p))q_i = 1 + C_1(1-p),$$

and furthermore,

$$\left|1 - \sum_{i=1}^{d} q_i^p\right| = \left(\sum_{i=1}^{d} q_i^p\right) - 1 \ge C_1 \cdot (1-p) = C_1 \cdot |p-1|$$

When $p \in (1,2]$, then for $y \in (0, 5/6]$,

$$y^{p-1} \le \left(\frac{5}{6}\right)^{p-1} \le 1 - C_2 \cdot (p-1),$$

for some positive constant C_2. This implies that

$$\sum_{i=1}^{d} q_i^p \le \sum_{i=1}^{d} q_i(1 - C_2 \cdot (p-1)) = 1 - C_2 \cdot (p-1),$$

and

$$\left|1 - \sum_{i=1}^{d} q_i^p\right| = 1 - \sum_{i=1}^{d} q_i^p \geq C_2 \cdot (p-1) = C_2 \cdot |p-1|.$$

To finish the proof of the lemma, we set $C = \min\{C_1, C_2\}$. ■

Corollary 85. *There exists a constant C such that if the probability of each element is at most $5/6$, then the pth Tsallis entropy is at least C for any $p \in (0,1) \cup (1,2]$.*

Proof. We have

$$T_p = \frac{1 - \sum_{i=1}^{d} q^p}{p-1} = \frac{|1 - \sum_{i=1}^{d} q_i^p|}{|p-1|} \geq C.$$

■

We now show how to deal with the case when there is an element of large probability. It turns out that in this case we can obtain a multiplicative approximation of Tsallis entropy by combining two residual moments.

Lemma 86. *There is a positive constant C such that if there is an element i of probability $q_i \geq 2/3$, then the sum of a multiplicative $(1 + C \cdot |1-p| \cdot \varepsilon)$-approximation to $1 - q_i$ and a multiplicative $(1 + C \cdot |1-p| \cdot \varepsilon)$-approximation to $\sum_{j \neq i} q_j^p$ gives a multiplicative $(1+\varepsilon)$-approximation to $|1 - \sum_i q_i^p|$, for any $p \in (0,1) \cup (1,2]$.*

Proof. We first argue that a multiplicative approximation to $|1 - q_i^p|$ can be obtained from a multiplicative approximation to $1 - q_i$. Let $g(y) = 1 - (1-y)^p$. Note that $g(1-q_i) = 1 - q_i^p$. Since $1 - q_i \in [0, 1/3]$, we restrict the domain of g to $[0, 1/3]$. The derivative of g is $g'(y) = p(1-y)^{p-1}$. Note that g is strictly increasing for $p \in (0,1) \cup (1,2]$. For $p \in (0,1)$, the derivative is in the range $[p, 3p/2]$. For $p \in (1,2]$, it always lies in the range $[2p/3, p]$. In both cases, a $(1 + 2\varepsilon/3)$-approximation to y suffices to compute a $(1+\varepsilon)$-approximation to $g(y)$.

We now consider two cases:

- Assume first that $p \in (0,1)$. For any $y \in (0, 1/3]$, we have

$$\frac{y^p}{x} \geq \left(\frac{1}{3}\right)^{p-1} = 3^{1-p} \geq 1 + C_1(1-p),$$

for some positive constant C_1. The last inequality follows from the convexity of the function 3^{1-p}. This means that if $q_i < 1$, then

$$\frac{\sum_{j \neq i} q_j^p}{1 - q_i} \geq \frac{\sum_{j \neq i} q_j (1 + C_1(1-p))}{1 - q_i} = \frac{(1-q_i)(1 + C_1(1-p))}{1 - q_i} = 1 + C_1(1-p).$$

Since $q_i \leq q_i^p < 1$, we also have

$$\frac{\sum_{j \neq i} q_j^p}{1 - q_i^p} \geq \frac{\sum_{j \neq i} q_j^p}{1 - q_i} \geq 1 + C_1(1-p).$$

This implies that if we compute a multiplicative $1+(1-p)\varepsilon/D_1$-approximations to both $1-q_i^p$ and $\sum_{j\neq i} q_j^p$, for sufficiently large constant D_1, we compute a multiplicative $(1+\varepsilon)$-approximation of $(\sum_{j=1}^d q_j^p)-1$.

- The case of $p\in(1,2]$ is similar. For any $y\in(0,1/3]$, we have

$$\frac{y^p}{y}\leq\left(\frac{1}{3}\right)^{p-1}\leq 1-C_2(p-1),$$

for some positive constant C_2. Hence,

$$\frac{\sum_{j\neq i} q_j^p}{1-q_i}\leq\frac{\sum_{j\neq i} q_j(1-C_2(p-1))}{1-q_i}=\frac{(1-q_i)(1-C_2(p-1))}{1-q_i}=1-C_2(p-1),$$

and because $q_i^p\leq q_i$,

$$\frac{\sum_{j\neq i} q_j^p}{1-q_i^p}\leq\frac{\sum_{j\neq i} q_j^p}{1-q_i}\leq 1-C_2(p-1).$$

This implies that if we compute a multiplicative $1+(p-1)\varepsilon/D_2$-approximations to both $1-q_i^p$ and $\sum_{j\neq i} q_j^p$, for sufficiently large constant D_2, we can compute a multiplicative $(1+\varepsilon)$-approximation to $1-\sum_{j=1}^d q_j^p$.

■

We these collect those facts in the following theorem.

Theorem 87. *There is a streaming algorithm for multiplicative $(1+\varepsilon)$-approximation of Tsallis entropy for any $p\in(0,1)\cup(1,2]$ using $O\left(\log(mM)/(|p-1|^2\varepsilon^2)\right)$ space and $O(\log(1/(|p-1|\varepsilon))\log\log(1/(|p-1|\varepsilon)))$ update time.*

Proof. Using the CountMin sketch of [34], we can decide whether there exists an i with $q_i\geq 2/3$, or whether no i has $q_i>5/6$. By first applying universe reduction to a universe of size $O(1)$ as discussed in Section 3.3.3 (here the gap between 5/6 and 7/8 is just a constant), the space complexity would be $O(\log mM)$, and the update time would be $O(1)$.

If there is no heavy element, then by Lemma 84 there is a constant $C\in(0,1)$ such that $|1-\sum_i q_i^p|\geq C|p-1|$. We want to compute a multiplicative approximation to $|1-\sum_i q_i^p|$. We know that the difference between $\sum_i q_i^p$ and 1 is large. Therefore, if we compute a multiplicative $(1+\frac{1}{2}|p-1|C\varepsilon)$-approximation to $\sum_i q_i^p$, we obtain an additive $(\frac{1}{2}|p-1|C\varepsilon\sum_i q_i^p)$-approximation to $\sum_i q_i^p$. If $\sum_i q_i^p\leq 2$, then

$$\frac{\frac{1}{2}|p-1|C\varepsilon\sum_i q_i^p}{|1-\sum_i q_i^p|}\leq\frac{|p-1|C\varepsilon}{C|p-1|}=\varepsilon.$$

If $\sum_i q_i^p\geq 2$, then

$$\frac{\frac{1}{2}|p-1|C\varepsilon\sum_i q_i^p}{|1-\sum_i q_i^p|}\leq\frac{1}{2}|p-1|C\varepsilon\cdot 2\leq\varepsilon.$$

In either case, we obtain a multiplicative $(1+\varepsilon)$-approximation to $|1-\sum_i q_i^p|$, which in turn yields a multiplicative approximation to the pth Tsallis entropy. By using the algorithm in Section 3.3, the space usage would be $O(\log(mM)/(|p-1|^2\varepsilon^2))$.

In the case that there is a heavy element, we need to estimate $F_p^{\mathrm{res}}(q)$ and $F_1^{\mathrm{res}}(q)$ with multiplicative error $1 \pm \varepsilon$, which reduces to multiplicatively estimating $F_p^{\mathrm{res}}, F_1^{\mathrm{res}}$ with error $1 \pm O(\varepsilon)$. Here we use F_p^{res} to denote the pth moment of x when restricted to all coordinates except the one with largest weight (break ties arbitrarily). $F_p^{\mathrm{res}}(q)$ is similar, but for the q vector. For this, we use a a slightly altered version of our algorithm in Section 3.3. We keep all parts of the algorithm the same, but we change our definition of Ψ' in HighEnd to be $\Psi' = \sum_{w \in L \setminus \{i^*\}} (x_w^*)^p$. That is, we do not include the contribution from the heaviest element, which we label as having index i^*. The same analysis shows that this gives an estimate of $x_{L\setminus\{i^*\}}$ with additive error $\varepsilon F_p^{\mathrm{res}}$. Thus again our space and time are the same as in the previous case. ■

4.4 Lower Bound

By combining ideas of [25] and our technique of embedding geometrically-growing hard instances in Section 3.4, we can provide an improved lower bound for additively estimating the entropy of a stream in the strict turnstile model.[1]

Theorem 88. *Any algorithm for ε-additive approximation of H, the entropy of a stream, in the strict turnstile model with probability at least $11/12$ requires space $\Omega(\varepsilon^{-2}\log(\min\{d,m\})/\log(1/\varepsilon))$.*

Proof. We reduce from Augmented-Indexing, as in Theorem 70. Alice receives a string of length $s = \log(\min\{d,m\})/(2\varepsilon^2\log(1/\varepsilon))$, and Bob receives an index $z \in [s]$. Alice conceptually divides her input into $b = \varepsilon^2 s$ blocks, each of size $1/\varepsilon^2$, and reduces each block using the Indexing→Gap-Hamdist reduction of Theorem 69 to obtain b Gap-Hamdist instances with strings $y_1, \ldots, y_b$, each of length $\ell = \Theta(1/\varepsilon^2)$. For each $1 \le i \le b$, and $1 \le j \le \ell$ Alice inserts universe elements $(i, j, 1, (y_i)_j), \ldots, (i, j, \varepsilon^{-2i}, (y_i)_j)$ into the stream and sends the state of a streaming algorithm to Bob.

Bob identifies the block $i(z)$ in which z lands and deletes all stream elements associated with blocks with index $i > i(z)$. He then does his part in the Indexing→Gap-Hamdist reduction to obtain a vector $y(\mathrm{Bob})$ of length ℓ. For all $1 \le j \le \ell$, he inserts the universe elements $(i(z), j, 1, y(\mathrm{Bob})_j), \ldots, (i(z), j, \varepsilon^{-2i(z)}, y(\mathrm{Bob})_j)$ into the stream.

The number of stream tokens from block indices $i < i(z)$ is $A = \varepsilon^{-2}\sum_{i=0}^{i(z)-1}\varepsilon^{-2i} = \Theta(\varepsilon^{-2i(z)})$. The number of tokens in block $i(z)$ from Alice and Bob combined is $2\varepsilon^{-(2i(z)+2)}$. Define $B = \varepsilon^{-2i(z)}$ and $C = \varepsilon^{-2}$. The L_1 weight of the stream is $R = A + 2BC$. Let Δ denote the Hamming distance between $y_{i(z)}$ and $y(\mathrm{Bob})$ and H denote the entropy of the stream.

[1] The previous proof of [25] though has the advantage of even holding in the weakest insertion-only model, i.e. no negative frequency updates, but with a weaker lower bound.

We have:

$$
\begin{aligned}
H &= \frac{A}{R}\log(R) + \frac{2B(C-\Delta)}{R}\log\left(\frac{R}{2}\right) + \frac{2B\Delta}{R}\log(R) \\
&= \frac{A}{R}\log(R) + \frac{2BC}{R}\log(R) - \frac{2BC}{R} + \frac{2B\Delta}{R}
\end{aligned}
$$

Rearranging terms gives

$$
\Delta = \frac{HR}{2B} + C - C\log(R) - \frac{A}{2B}\log(R) \tag{4.9}
$$

To decide the Gap-Hamdist instance, we must decide whether $\Delta < 1/2\varepsilon^2 - 1/\varepsilon$ or $\Delta > 1/2\varepsilon^2 + 1/\varepsilon$. By Eq. (4.9) and the fact that Bob knows A, B, C, and R, it suffices to obtain a $1/\varepsilon$-additive approximation to $HR/(2B)$ to accomplish this goal. In other words, we need a $2B/(\varepsilon R)$-additive approximation to H. Since $B/R = \Theta(\varepsilon^2)$, it suffices to obtain an additive $\Theta(\varepsilon)$-approximation to H. Let $\mathcal{A}$ be a streaming algorithm which can provide an additive $\Theta(\varepsilon)$-approximation with probability at least 11/12. Recalling that correctly deciding the Gap-Hamdist instance with probability 11/12 allows one to correctly decide the original Augmented-Indexing instance with probability 2/3 by Theorem 69, and given Theorem 68, $\mathcal{A}$ must use at least $\log(\min\{d,m\})/(\varepsilon^2\log(1/\varepsilon))$ bits of space. As required, the length of the vector being updated in the stream is at most $\sum_{i=1}^{s}\varepsilon^{-2i} = O(\min\{d,m\}) = O(d)$, and the length of the stream is exactly twice the vector length, and thus $O(\min\{d,m\}) = O(m)$. ■

Chapter 5

Johnson-Lindenstrauss Transforms

In this chapter we provide some results on the Johnson-Lindenstrauss transform. In Section 5.1 we exhibit two distributions over sparse matrices that both yield JL distributions. Each distribution is over linear mappings that embed into optimal target dimension $O(\varepsilon^{-2}\log(1/\delta))$ to achieve distortion $1 \pm \varepsilon$ with success probability $1-\delta$, and such that every matrix in the support of the distribution only has an $O(\varepsilon)$-fraction of its entries non-zero (in fact, every column has at most an $O(\varepsilon)$-fraction of its entries non-zero).

In Section 5.4 we provide a JL distribution which can be sampled from using few random bits. More precisely, a matrix can be sampled from the distribution using $O(\log d + \log(1/\varepsilon)\log(1/\delta) + \log(1/\delta)\log\log(1/\delta))$ uniform random bits.

Since all our JL families contain embeddings which are are linear, without loss of generality we henceforth assume that the vector to be embedded, x, has $\|x\|_2 = 1$.

Bibliographic remarks: Section 5.1 based on joint work with Kane [76], which used some ideas from a previous work of ours [75] analyzing a scheme of Dasgupta, Kumar, and Sarlós [35]. Section 5.4 is based on joint work with Kane and Meka [74].

5.1 Sparse JL transforms

We first provide an overview of our approach. For a JL distribution over linear mappings, we let s denote the *column sparsity* of that distribution, i.e. the maximum number of non-zero entries in any column in any matrix in the support of the distribution. Our constructions are depicted in Figure 5-1. Figure 5-1(a) represents the "DKS construction" of [35] in which each item is hashed to s random target coordinates with replacement. Our two schemes achieving $s = \Theta(\varepsilon^{-1}\log(1/\delta))$ are as follows. Construction (b) is much like (a) except that we hash coordinates s times *without* replacement. In (c), the target vector is divided up into s contiguous blocks each of equal size k/s, and a given coordinate in the original vector is hashed to a random location in each block (essentially this is the CountSketch of [29], though we use higher independence in our hash functions). In all cases (a), (b), and (c), we randomly flip the sign of a coordinate in the original vector and divide by $\sqrt{s}$ before

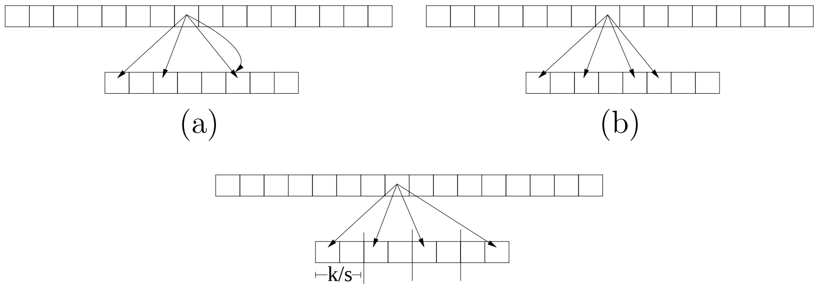

Figure 5-1: Three constructions to project a vector in $\mathbb{R}^d$ down to $\mathbb{R}^k$. Figure (a) is the DKS construction in [35], and the two constructions we give in this work are represented in (b) and (c). The out-degree in each case is s, the sparsity.

adding it in any location in the target vector.

We give two different analyses for both our constructions (b) and (c). Look at the random variable $Z = \|Sx\|_2^2 - \|x\|_2^2$, where S is a random matrix in the JL family. Our proofs all use Markov's bound on the ℓth moment Z^ℓ to give $\mathbf{Pr}[|Z| > \varepsilon\|x\|_2^2] < \varepsilon^{-\ell} \cdot \mathbf{E}[Z^\ell]$ for $\ell = \log(1/\delta)$ an even integer. The task is then to bound $\mathbf{E}[Z^\ell]$. In our first approach, we observe that Z is a quadratic form in the random signs, and thus its moments can be bounded via the Hanson-Wright inequality [62]. This analysis turns out to reveal that the hashing to coordinates in the target vector need not be done randomly, but can in fact be specified by any sufficiently good code. Specifically, in (b) it suffices for the columns of the embedding matrix (ignoring the random signs and division by $\sqrt{s}$) to be codewords in a constant-weight binary code of weight s and minimum distance $s - O(s^2/k)$. In (c), if for each $i \in [d]$ we let C_i be a length-s vector with entries in $[k/s]$ specifying where coordinate i is mapped to in each block, it suffices for $\{C_i\}_{i=1}^d$ to be a code of minimum distance $s - O(s^2/k)$. It is fairly easy to see that if one wants a deterministic hash function, it is necessary for the columns of the embedding matrix to be specified by a code: if two coordinates have small Hamming distance in their vectors of hash locations, it means they collide often. Since collision is the source of error, an adversary in this case could ask to embed a vector which has its mass equally spread on the two coordinates whose hash locations have small Hamming distance, causing large error with large probability over the choice of random signs. What our analysis shows is that not only is a good code necessary, but it is also sufficient.

In our second analysis approach, we expand Z^ℓ to obtain a polynomial with roughly $d^{2\ell}$ terms. We view its monomials as being in correspondence with graphs, group monomials whose graphs are isomorphic, then do some combinatorics to make the expectation calculation feasible. In this approach, we assume that the random signs as well as the hashing to coordinates in the target vector are done $2\log(1/\delta)$-wise independently. This graph-based approach played a large role in the analysis in [75], and was later also used in [18]. We point out here that Figure 5-1(c) is some-what simpler to implement than Figure 5-1(b), since there are simple constructions of $2\log(1/\delta)$-wise hash families [24]. Figure 5-1(b) on the other hand requires hashing without replacement, which amounts to using random permutations. We thus derandomize Figure 5-1(b) using almost $2\log(1/\delta)$-wise independent permutation families [80].

Our analyses in the case that the hash functions are specified by codes require the following moment bound due to Hanson and Wright [62]. We also include a proof in Section 5.3. Recall that the *Frobenius norm* $\|B\|_F$ of a matrix B is $\sqrt{\sum_{i,j} B_{i,j}^2}$, and its *operator norm* $\|B\|_2$ (when B is real and symmetric) is the magnitude of B's largest eigenvalue.

Theorem 89 (Hanson-Wright inequality)**.** *There is a universal constant $C > 0$ such that for $z = (z_1, \ldots, z_n)$ a vector of i.i.d. Bernoulli ± 1 random variables, $B \in \mathbb{R}^{n\times n}$ symmetric, and integer $\ell \geq 2$,*

$$\mathbf{E}\left[\left|z^T Bz - \mathrm{trace}(B)\right|^\ell\right] \leq C^\ell \cdot \max\left\{\sqrt{\ell}\cdot\|B\|_F, \ell\cdot\|B\|_2\right\}^\ell.$$

In Section 5.1.1, as a warmup we show how to use the Hanson-Wright inequality to show that the distribution over dense sign matrices (appropriately scaled) is a JL distribution. Then, in Section 5.1.2, we show how to use the Hanson-Wright inequality to analyze our schemes Figure 5-1(b) and Figure 5-1(c). In Section 5.1.3, we provide alternative analyses to our schemes in which the hash functions are pseudorandom, and not fixed by any specific code. In Section 5.1.4 we show that our analyses are tight and that column sparsity $\Omega(\varepsilon^{-1}\log(1/\delta))$ is required, even if the hash functions used are completely random.

5.1.1 A Warmup Proof of the JL Lemma

Note that for $y = Sx$,

$$\|y\|_2^2 = \|Sx\|_2^2 = \sum_{r=1}^k \sum_{i,j} x_i x_j S_{r,i} S_{r,j}.$$

That is, for any fixed x, $\|y\|_2^2$ is a quadratic form in the random variables $S_{r,j}$. In the case that the entries of S are (scaled) Bernoulli random variables, we show how to use Theorem 89 to show that any distribution over such matrices with $2\log(1/\delta)$-wise independent entries satisfies the JL lemma. This was initially shown by Achlioptas [2], and can also be derived from [9].

Theorem 90. *For $d > 0$ an integer and any $0 < \varepsilon, \delta < 1/2$, let S be a $k \times d$ random matrix with $\pm 1/\sqrt{k}$ entries that are r-wise independent for $k = \Omega(\varepsilon^{-2}\log(1/\delta))$ and $r = \Omega(\log(1/\delta))$. Then for any $x \in \mathbb{R}^d$ with $\|x\|_2 = 1$,*

$$\mathbf{Pr}_S[|\|Sx\|_2^2 - 1| > \varepsilon] < \delta.$$

Proof. We have

$$\|Sx\|_2^2 = \frac{1}{k} \cdot \sum_{i=1}^{k} \left(\sum_{(s,t)\in[d]\times[d]} x_s x_t \sigma_{i,s}\sigma_{i,t} \right), \tag{5.1}$$

where σ is a kd-dimensional vector formed by concatenating the rows of $\sqrt{k} \cdot S$. Define the matrix $T \in \mathbb{R}^{kd\times kd}$ to be the block-diagonal matrix where each block equals xx^T/k. Then, $\|Sx\|_2^2 = \sigma^T T\sigma$. Furthermore, $\mathrm{trace}(T) = \|x\|_2^2 = 1$. Thus, we would like to argue that $\sigma^T T\sigma$ is concentrated about $\mathrm{trace}(T)$, for which we can use Theorem 89. Specifically, if $\ell \geq 2$ is even,

$$\mathbf{Pr}[|\|Sx\|_2^2 - 1| > \varepsilon] = \mathbf{Pr}[|\sigma^T T\sigma - \mathrm{trace}(T)| > \varepsilon] < \varepsilon^{-\ell} \cdot \mathbf{E}[(\sigma^T T\sigma - \mathrm{trace}(T))^\ell]$$

by Markov's inequality. To apply Theorem 89, we also ensure $2\ell \leq r$ so that the ℓth moment of $\sigma^T T\sigma - \mathrm{trace}(T)$ is determined by r-wise independence of the σ entries. We also must bound $\|T\|_F$ and $\|T\|_2$. Direct computation gives $\|T\|_F^2 = (1/k) \cdot \|x\|_2^4 = 1/k$. Also, x is the only eigenvector of xx^T/k with non-zero eigenvalue, and furthermore its eigenvalue is $\|x\|_2^2/k = 1/k$, and thus $\|T\|_2 = 1/k$. Therefore,

$$\mathbf{Pr}[|\|Sx\|_2^2 - 1| > \varepsilon] < C^\ell \cdot \max\left\{\varepsilon^{-1}\sqrt{\frac{\ell}{k}}, \varepsilon^{-1}\frac{\ell}{k}\right\}^\ell, \tag{5.2}$$

which is at most δ for $\ell = \log(1/\delta)$ and $k \geq 4 \cdot C^2 \cdot \varepsilon^{-2}\log(1/\delta)$. ■

Remark 91. The conclusion of Theorem 89 holds even if the z_i are not necessarily Bernoulli but rather have mean 0, variance 1, and subGaussian tails. Thus, the above proof of Theorem 90 carries over unchanged to show that S could instead have $\Omega(\log(1/\delta))$-wise independent such z_i as entries. We also direct the reader to an older proof of this fact by Matousek [89], without discussion of independence requirements (though independence requirements can most likely be calculated from his proof, by converting the tail bounds he uses into moment bounds via integration).

5.1.2 Code-Based Sparse JL Constructions

In this section, we provide analyses of our constructions (b) and (c) in Figure 5-1 when the hash locations are determined by some fixed error-correcting code. We give the full argument for (c) below, then discuss in Remark 97 how essentially the same argument can be applied to analyze (b).

Define $k = C\cdot\varepsilon^{-2}\log(1/\delta)$ for a sufficiently large constant C. Let s be some integer dividing k satisfying $s \geq 2\varepsilon^{-1}\log(1/\delta)$. Let $\mathcal{C} = \{C_1, \ldots, C_d\}$ be any $(s, \log_{k/s} d, s - O(s^2/k))_{k/s}$ code. We specify our JL family by describing the embedded vector y. Define hash functions $\sigma : [d] \times [s] \to \{-1, 1\}$ and $h : [d] \times [s] \to [k/s]$. The former is drawn at random from a $2\log(1/\delta)$-wise independent family, and the latter has $h(i,j)$ being the jth entry of the ith codeword in $\mathcal{C}$. We conceptually view $y \in \mathbb{R}^k$ as being in $\mathbb{R}^{s\times(k/s)}$. Our embedded vector then has $y_{r,j} = \sum_{h(i,r)=j} \sigma(i,r)x_i/\sqrt{s}$. This describes our JL family, which is indexed by σ. Note the column sparsity is s.

Remark 92. It is important to know whether an $(s, \log_{k/s} d, s - O(s^2/k))_{k/s}$ code exists. By picking h at random from an $O(\log(d/\delta))$-wise independent family and setting $s \geq \Omega(\varepsilon^{-1}\sqrt{\log(d/\delta)\log(1/\delta)})$, it is not too hard to show via the Chernoff bound (or more accurately, Markov's bound applied with the $O(\log(d/\delta))$th moment bound implied by integrating the Chernoff bound) followed by a union bound over all pairs of $\binom{d}{2}$ vectors that h defines a good code with probability $1-\delta$. We do not perform this analysis here since Section 5.1.3 obtains better parameters. We also point out that we may assume without loss of generality that $d = O(\varepsilon^{-2}/\delta)$. This is because there exists an embedding into this dimension with sparsity 1 using only 4-wise independence with distortion $(1+\varepsilon)$ and success probability $1-\delta$ [29, 118]. It is worth noting that in the construction in this section, potentially h could be deterministic given an explicit code with our desired parameters.

Analysis of Figure 5-1(c) code-based construction: We first note

$$\|y\|_2^2 = \|x\|_2^2 + \frac{1}{s}\sum_{i\neq j}\sum_{r=1}^{s}\eta_{i,j,r}x_ix_j\sigma(i,r)\sigma(j,r),$$

where $\eta_{i,j,r}$ is 1 if $h(i,r) = h(j,r)$, and $\eta_{i,j,r} = 0$ otherwise. We thus would like that

$$Z = \frac{1}{s}\sum_{i\neq j}\sum_{r=1}^{s}\eta_{i,j,r}x_ix_j\sigma(i,r)\sigma(j,r) \tag{5.3}$$

is concentrated about 0. Note Z is a quadratic form in σ which can be written as $\sigma^T T\sigma$ for an $sd\times sd$ block-diagonal matrix T. There are s blocks, each $d\times d$, where in the rth block T_r we have $(T_r)_{i,j} = x_ix_j\eta_{i,j,r}/s$ for $i\neq j$ and $(T_r)_{i,i} = 0$ for all i. Now, $\mathbf{Pr}[|Z| > \varepsilon] = \mathbf{Pr}[|\sigma^T T\sigma| > \varepsilon]$. To bound this probability, we use the Hanson-Wright inequality combined with a Markov bound. Specifically, we prove our construction satisfies the JL lemma by applying Theorem 89 with $z = \sigma, B = T$.

Lemma 93. *$\|T\|_F^2 = O(1/k)$.*

Proof. We have that $\|T\|_F^2$ is equal to

$$\frac{1}{s^2}\cdot\sum_{i\neq j}x_i^2x_j^2\cdot\left(\sum_{r=1}^{s}\eta_{i,j,r}\right) = \frac{1}{s^2}\cdot\sum_{i\neq j}x_i^2x_j^2\cdot(s-\Delta(C_i,C_j)) \leq O(1/k)\cdot\|x\|_2^4 = O(1/k).$$

■

Lemma 94. *$\|T\|_2 \leq 1/s$.*

Proof. Since T is block-diagonal, its eigenvalues are the eigenvalues of each block. For a block T_r, write $T_r = (1/s)\cdot(S_r - D)$. D is diagonal with $D_{i,i} = x_i^2$, and $(S_r)_{i,j} = x_ix_j\eta_{i,j,r}$, including when $i = j$. Since S_r and D are both positive semidefinite, we have $\|T\|_2 \leq (1/s)\cdot\max\{\|S_r\|_2, \|D\|_2\}$. We have $\|D\|_2 = \|x\|_\infty^2 \leq 1$. For S_r, define u_t

for $t \in [k/s]$ by $(u_t)_i = x_i$ if $h(i,r) = t$, and $(u_t)_i = 0$ otherwise. Then $u_1, \ldots, u_{k/s}$ are eigenvectors of S_r each with eigenvalue $\|u_t\|_2^2$, and furthermore they span the image of S_r. Thus $\|S_r\|_2 = \max_t \|u_t\|_2^2 \le \|x\|_2^2 = 1$. ■

Theorem 95. $\mathbf{Pr}_\sigma[|\|y\|_2^2 - 1| > \varepsilon] < \delta$.

Proof. By a Markov bound applied to Z^ℓ for ℓ an even integer,

$$\mathbf{Pr}_\sigma[|Z| > \varepsilon] < \varepsilon^{-\ell} \cdot \mathbf{E}_\sigma[Z^\ell].$$

Since $Z = \sigma^T T \sigma$ and $\mathrm{trace}(T) = 0$, applying Theorem 89 with $B = T$, $z = \sigma$, and $\ell \le \log(1/\delta)$ gives

$$\mathbf{Pr}_\sigma[|Z| > \varepsilon] < C^\ell \cdot \max\left\{O(\varepsilon^{-1}) \cdot \sqrt{\frac{\ell}{k}}, \varepsilon^{-1}\frac{\ell}{s}\right\}^\ell. \tag{5.4}$$

since the ℓth moment is determined by $2\log(1/\delta)$-wise independence of σ. We conclude the proof by noting that the expression in Eq. (5.4) is at most δ for $\ell = \log(1/\delta)$ and our choices for s, k. ■

Remark 96. Only using that $\mathcal{C}$ has sufficiently high minimum distance, it is impossible to improve our analysis further. For example, for any $(s, \log_{k/s} d, s - O(s^2/k))_{k/s}$ code $\mathcal{C}$, create a new code $\mathcal{C}'$ which simply replaces the first letter of each codeword with "1"; $\mathcal{C}'$ then still has roughly the same minimum distance. However, in our construction this corresponds to all indices colliding in the first chunk of k/s coordinates, which creates an error term of $(1/s) \cdot \sum_{i \ne j} x_i x_j \sigma(i,r)\sigma(j,r)$. Now, suppose x consists of $t = (1/2) \cdot \log(1/\delta)$ entries each with value $1/\sqrt{t}$. Then, with probability $\sqrt{\delta} \gg \delta$, all these entries receive the same sign under σ and contribute a total error of $\Omega(t/s)$ in the first chunk alone. We thus need $t/s = O(\varepsilon)$, which implies $s = \Omega(\varepsilon^{-1}\log(1/\delta))$.

Remark 97. It is also possible to use a code to specify the hash locations in Figure 5-1(b). In particular, let the jth entry of the ith column of the embedding matrix be the jth symbol of the ith codeword (which we call $h(i,j)$) in a weight-s binary code of minimum distance $s - O(s^2/k)$ for $s \ge 2\varepsilon^{-1}\log(1/\delta)$. Define $\eta_{i,j,r}$ for $i, j \in [d], r \in [s]$ as an indicator variable for $h(i,r) = h(j,r) = 1$. Then, the error is again exactly as in Eq. (5.3). The Frobenius norm proof is identical, and the operator norm proof is nearly identical except that we have k blocks in our block-diagonal matrix instead of s. Also, as in Remark 96, such a code can be shown to exist via the probabilistic method (the Chernoff bound can be applied using negative dependence, followed by a union bound) as long as $s = \Omega(\varepsilon^{-1}\sqrt{\log(d/\delta)\log(1/\delta)})$. We omit the details since Section 5.1.3 obtains better parameters.

5.1.3 Random Hashing Sparse JL Constructions

In this section, we show that if the hash functions h described in Section 5.1.2 and Remark 97 are not specified by fixed codes, but rather are chosen at random from some family of sufficiently high independence, then one can achieve sparsity $O(\varepsilon^{-1}\log(1/\delta))$ (in the case of Figure 5-1(b), we actually need almost k-wise independent *permutations*). Recall our bottleneck in reducing the sparsity in Section 5.1.2 was actually obtaining the codes, discussed in Remark 92 and Remark 97.

Block Construction

Here we analyze the construction of Figure 5-1(c), except rather than let $\mathcal{C}$ be an arbitrary code, we let the underlying hash function $h : [d] \times [s] \to [k/s]$ be randomly selected from a $2\log(1/\delta)$-wise independent family. Note that one can sample a random matrix from this family using a $O(\log(1/\delta)\log d)$-length seed.

We perform our analysis by bounding the ℓth moment of Z from first principles for $\ell = \log(1/\delta)$ an even integer (for this particular scheme, it seems the Hanson-Wright inequality does not simplify any details of the proof). We then use Markov's inequality to give $\mathbf{Pr}_{h,\sigma}[|Z| > \varepsilon] < \varepsilon^{-\ell} \cdot \mathbf{E}_{h,\sigma}[Z^\ell]$.

Let $Z_r = \sum_{i\neq j} \eta_{i,j,r} x_i x_j \sigma(i,r)\sigma(j,r)$ so that $Z = (1/s)\cdot\sum_{r=1}^s Z_r$. We first bound the tth moment of each Z_r for $1 \le t \le \ell$. As in the Frobenius norm moment bound of [75], and also used later in [18], the main idea is to observe that monomials appearing in the expansion of Z_r^t can be thought of in correspondence with graphs. Notice

$$Z_r^t = \sum_{i_1\neq j_1,\ldots,i_t\neq j_t} \prod_{u=1}^t \eta_{i_u,j_u,r} x_{i_u} x_{j_u} \sigma(i_u,r)\sigma(j_u,r) \tag{5.5}$$

Each monomial corresponds to a directed multigraph with labeled edges whose vertices correspond to the distinct i_u and j_u. An $x_{i_u}x_{j_u}$ term corresponds to a directed edge with label u from the vertex corresponding to i_u to the vertex corresponding to j_u. The main idea to bound $\mathbf{E}_{h,\sigma}[Z_r^t]$ is then to group monomials whose corresponding graphs are isomorphic, then do some combinatorics.

Lemma 98. *For $t \le \log(1/\delta)$, $\mathbf{E}_{h,\sigma}[Z_r^t] \le 2^{O(t)} \cdot \begin{cases} s/k & t < \log(k/s) \\ (t/\log(k/s))^t & \text{otherwise} \end{cases}$.*

Proof. Let $\mathcal{G}_t$ be the set of isomorphism classes of directed multigraphs with t labeled edges with distinct labels in $[t]$, where each edge has positive and even degree (the sum of in- and out-degrees), and the number of vertices is between 2 and t. Let $\mathcal{G}'_t$ be similar, but with labeled vertices and connected components as well, where vertices have distinct labels between 1 and the number of vertices, and components have distinct labels between 1 and the number of components. Let f map the monomials appearing in Eq. (5.5) to the corresponding graph isomorphism class. By $2t$-wise independence of σ, any monomial in Eq. (5.5) whose corresponding graph does not have all even degrees has expectation 0. For a graph G, we let v denote the number of vertices, and m the number of connected components. Let d_u denote the degree of a vertex u. Then,

$$\mathbf{E}_{h,\sigma}[Z_r^t] = \sum_{i_1\neq j_1,\ldots,i_t\neq j_t} \left(\prod_{u=1}^t x_{i_u}x_{j_u}\right)\cdot \mathbf{E}\left[\prod_{u=1}^t \sigma(i_u,r)\sigma(j_u,r)\right]\cdot \mathbf{E}\left[\prod_{u=1}^t \eta_{i_u,j_u,r}\right]$$

$$= \sum_{G\in\mathcal{G}_t} \sum_{\substack{i_1\neq j_1,\ldots,i_t\neq j_t\\ f((i_u,j_u)_{u=1}^t)=G}} \left(\prod_{u=1}^t x_{i_u}x_{j_u}\right)\cdot \mathbf{E}\left[\prod_{u=1}^t \eta_{i_u,j_u,r}\right]$$

$$= \sum_{G\in\mathcal{G}_t} \sum_{\substack{i_1\neq j_1,\ldots,i_t\neq j_t\\ f((i_u,j_u)_{u=1}^t)=G}} \left(\frac{s}{k}\right)^{v-m}\cdot \left(\prod_{u=1}^t x_{i_u}x_{j_u}\right) \tag{5.6}$$

$$\leq \sum_{G\in\mathcal{G}_t} \left(\frac{s}{k}\right)^{v-m}\cdot v!\cdot \frac{1}{\binom{t}{d_1/2,\ldots,d_v/2}} \tag{5.7}$$

$$= \sum_{G\in\mathcal{G}'_t} \left(\frac{s}{k}\right)^{v-m}\cdot \frac{1}{m!}\cdot \frac{1}{\binom{t}{d_1/2,\ldots,d_v/2}}. \tag{5.8}$$

We now justify these inequalities. The justification of Eq. (5.6) is similar to that in the Frobenius norm bound in [75]. That is, $\prod_{u=1}^t \eta_{i_u,j_u,r}$ is determined by $h(i_u,r), h(j_u,r)$ for each $u\in[t]$, and hence its expectation is determined by $2t$-wise independence of h. This product is 1 if i_u and j_u hash to the same element for each u and is 0 otherwise. Every i_u, j_u pair hashes to the same element if and only if for each connected component of G, all elements of $\{i_1,\ldots,i_t,j_1,\ldots,j_t\}$ corresponding to vertices in that component hash to the same value. We can choose one element of $[k/s]$ for each component to be hashed to, thus giving $(k/s)^m$ possibilities. The probability of any particular hashing is $(k/s)^{-v}$, and this gives that the expectation of the product is $(s/k)^{v-m}$.

For Eq. (5.7), note that $(\|x\|_2^2)^t = 1$, and the coefficient of $\prod_{u=1}^v x_{a_u}^{d_u}$ in its expansion for $\sum_u d_u = t$ is $\binom{t}{d_1/2,\ldots,d_v/2}$. Meanwhile, the coefficient of this monomial when summing over all $i_1\neq j_1,\ldots,i_t\neq j_t$ for a particular $G\in\mathcal{G}_\ell$ is at most $v!$. For Eq. (5.8), we move from isomorphism classes in $\mathcal{G}_t$ to those in $\mathcal{G}'_t$. For any $G\in\mathcal{G}_t$, there are $v!\cdot m!$ ways to label vertices and connected components.

We now bound the sum of the $1/\binom{t}{d_1/2,\ldots,d_v/2}$ term. Fix $v_1,\ldots,v_m$, $t_1,\ldots,t_m$ (where there are v_i vertices and t_i edges in the ith component C_i), and the assignment of vertex and edge labels to connected components. We upper bound Eq. (5.8) by considering building G edge by edge, starting with 0 edges. Let the initial graph be G_0, and we form $G = G_t$ by adding edges in increasing label order. We then want to bound the sum of $1/\binom{t}{d_1/2,\ldots,d_v/2}$ over $G\in\mathcal{G}'_\ell$ which satisfy the quantities we have fixed. Note $1/\binom{t}{d_1/2,\ldots,d_v/2}$ equals $2^{O(t)}\cdot t^{-t}\cdot\prod_{u=1}^v\cdot\left(\sqrt{d_u}^{d_u}\right)$. Initially, when $t=0$, our sum is $S_0 = 1$. When considering all ways to add the next edge to G_u to form G_{u+1}, an edge $i\to j$ contributes $S_u\cdot\sqrt{d_id_j}/t$ to S_{u+1}. Since we fixed assignments of edge labels to connected components, this edge must come from some particular component C_w. Summing over vertices $i\neq j$ in C_w and applying Cauchy-Schwarz,

$$\sum_{i\neq j\in C_w}\sqrt{d_id_j}/t \leq \frac{1}{t}\cdot\left(\sum_{i\in \mathrm{C_w}}\sqrt{d_i}\right)^2 \leq (v_it_i)/t,$$

Since there are $\binom{v}{v_1,\ldots,v_m}\binom{t}{t_1,\ldots,t_m}$ ways to assign edge and vertex labels to components, Eq. (5.8) gives

$$\mathbf{E}_{h,\sigma}[Z_r^t] \leq 2^{O(t)} \cdot \sum_{v=2}^{t} \sum_{m=1}^{v/2} \sum_{v_1,\ldots,v_m} \sum_{t_1,\ldots,t_m} \left(\frac{s}{k}\right)^{v-m} \cdot \frac{1}{m^m} \cdot \binom{v}{v_1,\ldots,v_m} \times \binom{t}{t_1,\ldots,t_m} \cdot \frac{\left(\prod_{i=1}^{m}(v_i t_i)^{t_i}\right)}{t^t} \tag{5.9}$$

$$\leq 2^{O(t)} \cdot \sum_{v=2}^{t} \sum_{m=1}^{v/2} \left(\frac{s}{k}\right)^{v-m} \cdot \left(\frac{v^v}{m^m}\right) \cdot v^{t-v} \tag{5.10}$$

$$\leq 2^{O(t)} \cdot \sum_{v=2}^{t} \sum_{m=1}^{v/2} \left(\frac{s}{k}\right)^{v-m} \cdot (v-m)^t \tag{5.11}$$

$$\leq 2^{O(t)} \cdot \sum_{v=2}^{t} \sum_{q=1}^{v/2} \left(\frac{s}{k}\right)^{q} \cdot q^t.$$

Eq. (5.10) holds since there are at most 2^{v+t} ways to choose the v_i, t_i and $t_i \geq v_i$. Eq. (5.11) follows since $v \geq 2m$ and thus $v = O(v-m)$.Setting $q = v - m$ and under the constraint $q \geq 1$, $(s/k)^q \cdot q^t$ is maximized when $q = \max\{1, \Theta(t/\log(k/s))\}$. The lemma follows. ∎

Theorem 99. *Our construction in this section gives a JL family with sparsity* $s = O(\varepsilon^{-1} \cdot \log(1/\delta))$.

Proof. We have

$$\mathbf{E}_{h,\sigma}[Z^\ell] = \frac{1}{s^\ell} \cdot \sum_{\substack{r_1<\ldots<r_q \\ \ell_1,\ldots,\ell_q \\ \forall i\ \ell_i>1 \\ \sum_i \ell_i=\ell}} \binom{\ell}{\ell_1,\ldots,\ell_q} \cdot \sum_{i=1}^{q} \mathbf{E}_{h,\sigma}[Z_{r_i}^{\ell_i}]$$

$$\leq \frac{1}{s^\ell} \cdot 2^{O(\ell)} \cdot \sum_{q=1}^{\ell/2} \binom{s}{q} \cdot \frac{\ell^\ell}{\prod_{i=1}^{q} \ell_i^{\ell_i}} \cdot \left(\frac{s}{k}\right)^q \cdot \prod_{i=1}^{q} \left\lceil \frac{\ell_i}{\log(k/s)} \right\rceil^{\ell_i} \tag{5.12}$$

$$\leq \frac{1}{s^\ell} \cdot 2^{O(\ell)} \cdot \sum_{q=1}^{\ell/2} \binom{s}{q} \cdot \ell^\ell \cdot \left(\frac{s}{k}\right)^q$$

$$\leq \frac{1}{s^\ell} \cdot 2^{O(\ell)} \cdot \sum_{q=1}^{\ell/2} \ell^\ell \cdot \left(\frac{s^2}{qk}\right)^q$$

Eq. (5.12) follows since there are $\binom{s}{q}$ ways to choose the r_i, and there are at most $2^{\ell-1}$ ways to choose the ℓ_i. Furthermore, even for the $\ell_i > \log(k/s)$, we have $2^{O(\ell_i)} \cdot (\ell_i/\log(k/s))^{\ell_i} = 2^{O(\ell_i)} \cdot (s/k)^2 \cdot (\ell_i/\log(k/s))^{\ell_i}$, so that the $(s/k)^q$ term is valid. Taking derivatives shows that the above is maximized for $q = s^2/(ek) < \ell/2$, which gives a summand of $2^{O(\ell)} \cdot \ell^\ell$. Thus, we have that the above moment is at most $(\varepsilon/2)^\ell$ when $k = C'\ell/\varepsilon^2$ for sufficiently large C'. The claim then follows since $\mathbf{Pr}_{h,\sigma}[|Z| > \varepsilon] < \varepsilon^{-\ell} \cdot \mathbf{E}_{h,\sigma}[Z^\ell]$ by Markov's inequality, and we set $\ell = \log(1/\delta)$. ■

Graph Construction

In this section, we analyze the construction in Figure 5-1(b) when the hashing is done randomly. Our analysis for this construction is quite similar to that of Figure 5-1(c).

The distribution over matrices S is such that each column of S has exactly $s = \Theta(\varepsilon^{-1}\log(1/\delta))$ of its entries, chosen at random, each randomly set to $\pm 1/\sqrt{s}$. All other entries in S are 0. That is, we pick a random bipartite graph with d vertices in the left vertex set and k in the right, where every vertex on the left has degree s. The matrix S is the incidence matrix of this graph, divided by $\sqrt{s}$ and with random sign flips.

We realize the distribution over S via two hash functions $h : [d] \times [k] \to \{0, 1\}$ and $\sigma : [d] \times [s] \to \{-1, 1\}$. The function σ is drawn from from a $2\log(1/\delta)$-wise independent family. The function h has the property that for any i, exactly s distinct $r \in [k]$ have $h(i, r) = 1$; in particular, we pick d seeds $\log(1/\delta)$-wise independently to determine h_i for $i = 1, \ldots, d$, and where each h_i is drawn from a γ-almost $2\log(1/\delta)$-wise independent family of permutations on $[d]$ for $\gamma = (\varepsilon s/(d^2k))^{\Theta(\log(1/\delta))}$. The seed length required for any one such permutation is $O(\log(1/\delta)\log d + \log(1/\gamma)) = O(\log(1/\delta)\log d)$ [80], and thus we can pick d such seeds $2\log(1/\delta)$-wise independently using total seed length $O(\log^2(1/\delta)\log d)$. We then let $h(i, r) = 1$ iff some $j \in [s]$ has $h_i(j) = r$.

If $y = Sx$, then we have

$$\|y\|_2^2 = \|x\|_2^2 + \frac{1}{s} \cdot \sum_{i \neq j \in [d], r \in [s]} x_i x_j \sigma(i, r)\sigma(j, r)h(i, r)h(j, r),$$

Define

$$Z = \frac{1}{s} \cdot \sum_r \sum_{i \neq j} x_i x_j \sigma(i, r)\sigma(j, r)h(i, r)h(j, r) = \frac{1}{s} \cdot \sum_r Z_r.$$

We would like to show that $\mathbf{Pr}_{h,\sigma}[|Z| > \varepsilon] < \delta$, which we show by via the Markov bound $\mathbf{Pr}_{h,\sigma}[|Z| > \varepsilon] < \varepsilon^{-\ell} \cdot \mathbf{E}_{h,\sigma}[Z^\ell]$ for some sufficiently large even integer ℓ. We furthermore note $\mathbf{E}_{h,\sigma}[Z^\ell] < \mathbf{E}_\sigma[Y^\ell]$ where

$$Y = \frac{1}{s} \cdot \sum_r \sum_{i \neq j} x_i x_j \sigma(i, r)\sigma(j, r)\delta_{i,r}\delta_{j,r} = \frac{1}{s} \cdot \sum_r Y_r,$$

where the $\delta_{i,r}$ are independent 0/1 random variables each with mean s/k. This is because, when expanding Z^ℓ into monomials, the expectation over h (after taking the expectation over σ) only term-by-term increases by replacing the random variables $h(i, r)$ with $\delta_{i,r}$. We analyze moments of the Y_r to then obtain an upper bound on the $\mathbf{E}_\sigma[Y^\ell]$ for $\ell = \log(1/\delta)$, which in turns provides an upper bound on $\mathbf{E}_{h,\sigma}[Z^\ell]$.

We carry out our analysis assuming h is perfectly random, then describe in Remark 102 how to relax this assumption by using γ-almost $2\log(1/\delta)$-wise permutations as discussed above.

Lemma 100. *For $t > 1$ an integer,* $\mathbf{E}_\sigma[Y_r^t] \le 2^{O(t)} \cdot \begin{cases} (s/k)^2 & t < \log(k/s) \\ (t/\log(k/s))^t & \text{otherwise} \end{cases}$.

Proof. We have

$$\mathbf{E}_\sigma[Y_r^t] = \sum_{i_1 \neq j_1, \ldots, i_t \neq j_t} \left(\prod_{u=1}^t x_{i_u} x_{j_u}\right) \cdot \mathbf{E}\left[\prod_{u=1}^t \sigma(i_u, r)\sigma(j_u, r)\right] \cdot \mathbf{E}\left[\prod_{u=1}^t \delta_{i_u,r}\delta_{j_u,r}\right]. \tag{5.13}$$

Define $\mathcal{G}_t$ as the set of isomorphism classes of directed multigraphs with t edges having distinct labels in $[t]$ and no self-loops, with between 2 and t vertices (inclusive), and where every vertex has an even and positive sum of in- and out-degrees. Let f map variable sequences to their corresponding graph isomorphism class. For a graph G, let v be its number of vertices, and let d_u be the sum of in- and out-degrees of vertex u. Then,

$$\begin{aligned}
\mathbf{E}_\sigma[Y_r^t] &= \sum_{G \in \mathcal{G}_t} \sum_{\substack{i_1 \neq j_1, \ldots, i_t \neq j_t \\ f((i_u, j_u))_{u=1}^t = G}} \left(\prod_u x_{i_u} x_{u_u}\right) \cdot \mathbf{E}\left[\prod_u \eta_{i_u, j_u, r_u, r'_u}\right] \\
&\le 2^{O(t)} \cdot \sum_{G \in \mathcal{G}_t} \sum_{\substack{i_1 \neq j_1, \ldots, i_t \neq j_t \\ f((i_u, j_u))_{u=1}^t = G}} \left(\prod_u x_{i_u} x_{j_u}\right) \cdot \left(\frac{s}{k}\right)^v \\
&\le 2^{O(t)} \cdot \sum_{G \in \mathcal{G}_t} \left(\frac{s}{k}\right)^v \cdot v! \cdot \frac{1}{\binom{t}{d_1/2, \ldots, d_v/2}} \\
&\le 2^{O(t)} \cdot \sum_{G \in \mathcal{G}'_t} \left(\frac{s}{k}\right)^v \cdot \frac{1}{\binom{t}{d_1/2, \ldots, d_v/2}} \\
&\le 2^{O(t)} \cdot \sum_v \left(\frac{s}{k}\right)^v \cdot \frac{1}{t^t} \cdot \left(\sum_G \prod_{u=1}^v \sqrt{d_u}^{d_u}\right),
\end{aligned} \tag{5.14}$$

where $\mathcal{G}'_t$ is the set of all isomorphism classes of directed multigraphs as in $\mathcal{G}_t$, but in which vertices are labeled as well, with distinct labels in $[v]$. The summation over G in Eq. (5.14) is over the $G \in \mathcal{G}'_t$ with v vertices. We bound this summation. We start with a graph with zero edges with vertices labeled $1, \ldots, v$ then consider how our summation increases as we build the graph edge by edge. Initially set $S_0 = 1$. We will think each S_i as a sum of several terms, where each term corresponds to some graph obtained by a sequence of i edge additions, so that the summation in Eq. (5.14) is bounded by S_t. When we add the $(i+1)$st edge, we have

$$S_{i+1}/S_i \le \left(\sum_{u \neq w} \sqrt{d_u} \cdot \sqrt{d_w}\right) \le \left(\sum_{u=1}^v \sqrt{d_u}\right)^2 \le 2tv,$$

with the last inequality following by Cauchy-Schwarz. It thus follows that the summation in Eq. (5.14) is at most $(2tv)^t$, implying

$$\mathbf{E}_\sigma[Y_r^t] \le 2^{O(t)} \cdot \sum_v \left(\frac{s}{k}\right)^v \cdot v^t.$$

The above is maximized for $v = \max\{2, t/\ln(k/s)\}$ (recall $v \ge 2$), giving our lemma. ■

Theorem 101. $\mathbf{Pr}_{h,\sigma}[|Z| > \varepsilon] < \delta$.

Proof. We have

$$\begin{aligned}
\mathbf{E}_\sigma[Y^\ell] &= \frac{1}{s^\ell} \cdot \sum_{\substack{r_1 < \ldots < r_q \\ \ell_1, \ldots, \ell_q \\ \forall i\ \ell_i > 1 \\ \sum_i \ell_i = \ell}} \binom{\ell}{\ell_1, \ldots, \ell_q} \cdot \sum_{i=1}^{q} \mathbf{E}_\sigma[Y_{r_i}^{\ell_i}] \\
&\le \frac{1}{s^\ell} \cdot 2^{O(\ell)} \cdot \sum_{q=1}^{\ell/2} \binom{k}{q} \cdot \ell^\ell \cdot \left(\frac{s}{k}\right)^{2q} \qquad (5.15) \\
&\le \frac{1}{s^\ell} \cdot 2^{O(\ell)} \cdot \sum_{q=1}^{\ell/2} \ell^\ell \cdot \left(\frac{s^2}{qk}\right)^q
\end{aligned}$$

The above expression is then identical to that in the proof of Theorem 99, and thus it is at most $(\varepsilon/2)^\ell$. We then set set $\ell = \log(1/\delta)$ an even integer so that, by Markov's inequality,

$$\mathbf{Pr}_{h,\sigma}[|Z| > \varepsilon] < \varepsilon^{-\ell} \cdot \mathbf{E}_{h,\sigma}[Z^\ell] \le \varepsilon^{-\ell} \cdot \mathbf{E}_\sigma[Y^\ell] < 2^{-\ell} = \delta.$$

■

Remark 102. As mentioned in Section 5.1.3, we can specify h via d hash functions h_i chosen $\log(1/\delta)$-wise independently where each h_i is drawn at random from a γ-almost $2\log(1/\delta)$-wise independent family of permutations, and where the seeds used to generate the h_i are drawn $\log(1/\delta)$-wise independently. Here, $\gamma = (\varepsilon s/(d^2 k))^{\Theta(\log(1/\delta))}$. In general, a γ-almost ℓ-wise independent family of permutations from $[d]$ onto itself is a family of permutations $\mathcal{F}$ where the image of any fixed ℓ elements in $[d]$ has statistical distance at most γ when choosing a random $f \in \mathcal{F}$ when compared with choosing a uniformly random permutation f. Now, there are $(kd^2)^\ell$ monomials in the expansion of Z^ℓ. In each such monomial, the coefficient of the $\mathbf{E}[\prod_u h(i_u, r_u)h(j_u, r_u)]$ term is at most $s^{-\ell}$. In the end, we want $\mathbf{E}_{h,\sigma}[Z^\ell] < O(\varepsilon)^\ell$ to apply Markov's inequality. Thus, we want $(kd^2/s)^\ell \cdot \gamma < O(\varepsilon)^\ell$.

Remark 103. It is worth noting that if one wants distortion $1 \pm \varepsilon_i$ with probability $1-\delta_i$ simultaneously for all i in some set S, our proofs of Theorem 99 and Theorem 101 reveal that it suffices to set $s = C \cdot \sup_{i \in S} \varepsilon_i^{-1} \log(1/\delta_i)$ and $k = C \cdot \sup_{i \in S} \varepsilon_i^{-2} \log(1/\delta_i)$ in both our constructions Figure 5-1(b) and Figure 5-1(c).

5.1.4 Tightness of Analyses

In this section we show that sparsity $\Omega(\varepsilon^{-1}\log(1/\delta))$ is required in Figure 5-1(b) and Figure 5-1(c), even if the hash functions used are completely random. We also show that sparsity $\tilde{\Omega}(\varepsilon^{-1}\log^2(1/\delta))$ is required in the DKS construction (Figure 5-1(a)), nearly matching the upper bounds of [18, 75]. Interestingly, all three of our proofs of (near-)tightness of analyses for these three constructions use the same hard input vectors. In particular, if $s = o(1/\varepsilon)$, then we show that a vector with $t = \lfloor 1/(s\varepsilon)\rfloor$ entries each of value $1/\sqrt{t}$ incurs large distortion with large probability. If $s = \Omega(1/\varepsilon)$ but is still not sufficiently large, we show that the vector $(1/\sqrt{2}, 1/\sqrt{2}, 0, \ldots, 0)$ incurs large distortion with large probability (in fact, for the DKS scheme in Figure 5-1(a) one can even take the vector $(1, 0, \ldots, 0)$).

Near-tightness for DKS Construction

The main theorem of this section is the following.

Theorem 104. *The DKS construction requires sparsity* $\Omega(\varepsilon^{-1}\cdot\lceil\log^2(1/\delta)/\log^2(1/\varepsilon)\rceil)$ *to achieve distortion* $1 \pm \varepsilon$ *with success probability* $1 - \delta$.

Before proving Theorem 104, we recall the DKS construction (Figure 5-1(a)). First, we replicate each coordinate s times while preserving the ℓ_2 norm. That is, we produce the vector $\tilde{x} = (x_1, \ldots, x_1, x_2, \ldots, x_2, \ldots, x_d, \ldots, x_d)/\sqrt{s}$, where each x_i is replicated s times. Then, pick a random $k \times ds$ embedding matrix A for $k = C\varepsilon^{-2}\log(1/\delta)$ where each column has exactly one non-zero entry, in a location defined by some random function $h : [ds] \to [k]$, and where this non-zero entry is ± 1, determined by some random function $\sigma : [ds] \to \{-1, 1\}$. The value $C > 0$ is some fixed constant. The final embedding is A applied to $\tilde{x}$. We are now ready to prove Theorem 104. The proof is similar to that of Theorem 107.

Our proof will use the following standard fact.

Fact 105 ([94, Proposition B.3])**.** *For all* $t, n \in \mathbb{R}$ *with* $n \geq 1$ *and* $|t| \leq n$,

$$e^t(1 - t^2/n) \leq (1 + t/n)^n \leq e^t.$$

Proof (of Theorem 104). First suppose $s \leq 1/(2\varepsilon)$. Consider a vector with $t = \lfloor 1/(s\varepsilon)\rfloor$ non-zero coordinates each of value $1/\sqrt{t}$. If there is exactly one pair $\{i, j\}$ that collides under h, and furthermore the signs agree under σ, the ℓ_2 norm squared of our embedded vector will be $(st - 2)/(st) + 4/(st)$. Since $1/(st) \geq \varepsilon$, this quantity is at least $1 + 2\varepsilon$. The event of exactly one pair $\{i, j\}$ colliding occurs with probability

$$\begin{aligned}\binom{st}{2}\cdot\frac{1}{k}\cdot\prod_{i=0}^{st-2}(1 - i/k) &\geq \Omega\left(\frac{1}{\log(1/\delta)}\right)\cdot(1 - \varepsilon/2)^{1/\varepsilon}\\ &= \Omega(1/\log(1/\delta)),\end{aligned}$$

which is much larger than $\delta/2$ for δ smaller than some constant. Now, given a collision, the colliding items have the same sign with probability 1/2.

We next consider the case $1/(2\varepsilon) < s \le 4/\varepsilon$. Consider the vector $x = (1, 0, \ldots, 0)$. If there are exactly three pairs $\{i_1, j_1\}, \ldots, \{i_3, j_3\}$ that collide under h in three distinct target coordinates, and furthermore the signs agree under σ, the ℓ_2 norm squared of our embedded vector will be $(s-6)/(s) + 12/(s) > 1 + 3\varepsilon/2$. The event of three pairs colliding occurs with probability

$$\binom{s}{2}\binom{s-2}{2}\binom{s-4}{2} \cdot \frac{1}{3!} \cdot \frac{1}{k^3} \cdot \prod_{i=0}^{s-4}(1 - i/k) \ge \Omega\left(\frac{1}{\log^3(1/\delta)}\right) \cdot (1 - \varepsilon/8)^{4/\varepsilon}$$
$$= \Omega(1/\log^3(1/\delta)),$$

which is much larger than $\delta/2$ for δ smaller than some constant. Now, given a collision, the colliding items have the same sign with probability 1/8.

We lastly consider the case $4/\varepsilon < s \le 2c\varepsilon^{-1}\log^2(1/\delta)/\log^2(1/\varepsilon)$ for some constant $c > 0$ (depending on C) to be determined later. First note this case only exists when $\delta = O(\varepsilon)$. Define $x = (1, 0, \ldots, 0)$. Suppose there exists an integer q so that

1. $q^2/s \ge 4\varepsilon$
2. $q/s < \varepsilon$
3. $(s/(qk))^q(1 - 1/k)^s > \delta^{1/3}$.

First we show it is possible to satisfy the above conditions simultaneously for our range of s. We set $q = 2\sqrt{\varepsilon s}$, satisfying item 1 trivially, and item 2 since $s > 4/\varepsilon$. For item 3, Fact 105 gives

$$(s/(qk))^q \cdot (1 - 1/k)^s \ge \left(\frac{s}{qk}\right)^q \cdot e^{-s/k} \cdot \left(1 - \frac{s}{k^2}\right).$$

The $e^{-s/k} \cdot (1 - (s/k^2))$ term is at least $\delta^{1/6}$ by the settings of s, k, and the $(s/(qk))^q$ term is also at least $\delta^{1/6}$ for c sufficiently small.

Now, consider the event $\mathcal{E}$ that exactly q of the s copies of x_1 are hashed to 1 by h, and to $+1$ by σ. If $\mathcal{E}$ occurs, then coordinate 1 in the target vector contributes $q^2/s \ge 4\varepsilon$ to ℓ_2^2 in the target vector by item 1 above, whereas these coordinates only contribute $q/s < \varepsilon$ to $\|x\|_2^2$ by item 2 above, thus causing error at least 3ε. Furthermore, the $s - q$ coordinates which do not hash to 1 are being hashed to a vector of length $k - 1 = \omega(1/\varepsilon^2)$ with random signs, and thus these coordinates have their ℓ_2^2 contribution preserved up to $1 \pm o(\varepsilon)$ with constant probability by Chebyshev's inequality. It thus just remains to show that $\mathbf{Pr}[\mathcal{E}] \gg \delta$. We have

$$\begin{aligned}\mathbf{Pr}[\mathcal{E}] &= \binom{s}{q} \cdot k^{-q} \cdot \left(1 - \frac{1}{k}\right)^{s-q} \cdot 1/2^q \\ &\geq \left(\frac{s}{qk}\right)^q \cdot \left(1 - \frac{1}{k}\right)^s \cdot \frac{1}{2^q} \\ &> \delta^{1/3} \cdot \frac{1}{2^q}.\end{aligned}$$

The 2^{-q} term is $\omega(\delta^{1/3})$ and thus overall $\mathbf{Pr}[\mathcal{E}] = \omega(\delta^{2/3}) \gg \delta$. ■

We will next show tightness of analysis for our constructions in Figure 5-1(b) and Figure 5-1(c). The arguments given there apply with very minor modification to the DKS construction as well, and thus $\Omega(\varepsilon^{-1}\log(1/\delta))$ is also a lower bound on the sparsity of the DKS construction.

Tightness of Figure 5-1(b) analysis

Theorem 106. *For δ smaller than a constant depending on C for $k = C\varepsilon^{-2}\log(1/\delta)$, the scheme of Section 5.1.3 requires $s = \Omega(\varepsilon^{-1}\log(1/\delta))$ to obtain distortion $1 \pm \varepsilon$ with probability $1 - \delta$.*

Proof. First suppose $s \leq 1/(2\varepsilon)$. We consider a vector with $t = \lfloor 1/(s\varepsilon) \rfloor$ non-zero coordinates each of value $1/\sqrt{t}$. If there is exactly one set i, j, r with $i \neq j$ such that $S_{r,i}, S_{r,j}$ are both non-zero for the embedding matrix S (i.e., there is exactly one collision), then the total error is $2/(ts) \geq 2\varepsilon$. It just remains to show that this happens with probability larger than δ. The probability of this occurring is

$$\begin{aligned}s^2 \cdot \binom{t}{2} \cdot \frac{1}{k} \cdot \frac{k-s}{k-1} \cdots \frac{k-2s+2}{k-s+1} \cdot \left(\frac{(k-2s+1)!}{(k-ts+1)!}\right) \cdot \left(\frac{(k-s)!}{k!}\right)^{t-2} \\ \geq \frac{s^2t^2}{2k} \cdot \left(\frac{k-st}{k}\right)^{st} \\ \geq \frac{s^2t^2}{2k} \cdot \left(1 - \frac{s^2t^2}{k}\right) \\ = \Omega(1/\log(1/\delta)).\end{aligned}$$

Now consider the case $1/(2\varepsilon) < s < c \cdot \varepsilon^{-1}\log(1/\delta)$ for some small constant c. Consider the vector $(1/\sqrt{2}, 1/\sqrt{2}, 0, \ldots, 0)$. Suppose there are exactly $2s\varepsilon$ collisions, i.e. $2s\varepsilon$ distinct values of r such that $S_{r,i}, S_{j,r}$ are both non-zero (to avoid tedium we disregard floors and ceilings and just assume $s\varepsilon$ is an integer). Also, suppose that in each colliding row r we have $\sigma(1, r) = \sigma(2, r)$. Then, the total error would be 2ε. It just remains to show that this happens with probability larger than δ. The probability of signs agreeing in exactly $2\varepsilon s$ chunks is $2^{-2\varepsilon s} > 2^{-2c\log(1/\delta)}$, which is larger than $\sqrt{\delta}$ for $c < 1/4$. The probability of exactly $2\varepsilon s$ collisions is

$$\binom{s}{2\varepsilon s} \cdot \left(\prod_{i=0}^{2\varepsilon s-1} \frac{s-i}{k-i}\right) \cdot \left(\prod_{i=0}^{s-2\varepsilon s-1} \frac{k-i-s}{k-i-2\varepsilon s}\right)$$

$$\geq \left(\frac{1}{2\varepsilon}\right)^{2\varepsilon s} \cdot \left(\frac{(1-2\varepsilon)s}{k}\right)^{2\varepsilon s} \cdot \left(1 - \frac{s}{k-s}\right)^{s-2\varepsilon s} \tag{5.16}$$

$$\geq \left(\frac{s}{4\varepsilon k}\right)^{2\varepsilon s} \cdot \left(1 - \frac{2s}{k}\right)^{s}. \tag{5.17}$$

It suffices for the right hand side to be at least $\sqrt{\delta}$ since h is independent of σ, and thus the total probability of error larger than 2ε would be greater than $\sqrt{\delta}^2 = \delta$. Taking natural logarithms, it suffices to have

$$2\varepsilon s \ln\left(\frac{4\varepsilon k}{s}\right) - s \ln\left(1 - \frac{2s}{k}\right) \leq \ln(1/\delta)/2.$$

Writing $s = q/\varepsilon$ and $a = 4C\log(1/\delta)$, the left hand side is $2q\ln(a/q) + \Theta(s^2/k)$. Taking a derivative shows $2q\ln(a/q)$ is monotonically increasing for $q < a/e$. Thus as long as $q < ca$ for a sufficiently small constant c, $2q\ln(a/q) < \ln(1/\delta)/4$. Also, the $\Theta(s^2/k)$ term is at most $\ln(1/\delta)/4$ for c sufficiently small. ■

Tightness of Figure 5-1(c) analysis

Theorem 107. *For δ smaller than a constant depending on C for $k = C\varepsilon^{-2}\log(1/\delta)$, the scheme of Section 5.1.3 requires $s = \Omega(\varepsilon^{-1}\log(1/\delta))$ to obtain distortion $1 \pm \varepsilon$ with probability $1 - \delta$.*

Proof. First suppose $s \leq 1/(2\varepsilon)$. Consider a vector with $t = \lfloor 1/(s\varepsilon) \rfloor$ non-zero coordinates each of value $1/\sqrt{t}$. If there is exactly one set i, j, r with $i \neq j$ such that $h(i,r) = h(j,r)$ (i.e. exactly one collision), then the total error is $2/(ts) \geq 2\varepsilon$. It just remains to show that this happens with probability larger than δ.

The probability of exactly one collision is

$$s\left[\frac{t! \cdot \binom{k/s}{t}}{(k/s)^t}\right]^{s-1} \binom{t}{2} \binom{k}{s} \left[\frac{(t-2)!\binom{k/s-1}{t-2}}{(k/s)^t}\right]$$

$$\geq s\left(1 - \frac{st}{k}\right)^{t(s-1)} \binom{t}{2} \left(\frac{s}{k}\right) \left(1 - \frac{st}{k}\right)^{t-2} \tag{5.18}$$

$$= \frac{s^2 t(t-1)}{2k} \cdot \left(1 - \frac{st}{k}\right)^{st-2}$$

$$\geq \frac{s^2 t(t-1)}{2k} \cdot \left(1 - \frac{s^2t^2}{k}\right)$$

$$= \Omega(1/\log(1/\delta)),$$

which is larger than δ for δ smaller than a universal constant.

Now consider $1/(2\varepsilon) < s < c\cdot\varepsilon^{-1}\log(1/\delta)$ for some small constant c. Consider the vector $x = (1/\sqrt{2}, 1/\sqrt{2}, 0, \ldots, 0)$. Suppose there are exactly $2s\varepsilon$ collisions, i.e. $2s\varepsilon$ distinct values of r such that $h(1,r) = h(2,r)$ (to avoid tedium we disregard floors and ceilings and just assume $s\varepsilon$ is an integer). Also, suppose that in each colliding chunk r we have $\sigma(1,r) = \sigma(2,r)$. Then, the total error would be 2ε. It just remains to show that this happens with probability larger than δ. The probability of signs agreeing in exactly $2\varepsilon s$ chunks is $2^{-2\varepsilon s} > 2^{-2c\log(1/\delta)}$, which is larger than $\sqrt{\delta}$ for $c < 1/4$. The probability of exactly $2\varepsilon s$ collisions is

$$\binom{s}{2\varepsilon s}\left(\frac{s}{k}\right)^{2\varepsilon s}\left(1-\frac{s}{k}\right)^{(1-2\varepsilon)s} \geq \left(\frac{s}{2\varepsilon k}\right)^{2\varepsilon s}\left(1-\frac{s}{k}\right)^{(1-2\varepsilon)s}$$

The above is at most $\sqrt{\delta}$, by the analysis following Eq. (5.17). Since h is independent of σ, the total probability of having error larger than 2ε is greater than $\sqrt{\delta}^2 = \delta$. ■

5.2 Numerical Linear Algebra Applications

The works of [30, 113] gave algorithms to solve various approximate numerical linear algebra problems given small memory and a only one or few passes over an input matrix. They considered models where one only sees a row or column at a time of some matrix $A \in \mathbb{R}^{d\times n}$. Another update model considered was the turnstile streaming model. In this model, the matrix A starts off as 0. One then sees a sequence of m updates $(i_1, j_1, v_1), \ldots, (i_m, j_m, v_m)$, where each update (i, j, v) triggers the change $A_{i,j} \leftarrow A_{i,j} + v$. The goal in all these models is to compute some functions of A at the end of seeing all rows, columns, or turnstile updates. The algorithm should use little memory (much less than what is required to store A explicitly). Both works [30, 113] solved problems such as approximate linear regression and best rank-r approximation by reducing to the problem of sketches for approximate matrix products. Before delving further, first we give a definition.

Definition 108. *Distribution $\mathcal{D}$ over $\mathbb{R}^{k\times d}$ has (ε,δ)-*JL moments *if for $\ell = \log(1/\delta)$ and $\forall x \in S^{d-1}$,*

$$\mathbf{E}_{S\sim\mathcal{D}}\left[\left|\|Sx\|_2^2 - 1\right|^\ell\right] \leq (\varepsilon/2)^\ell.$$

Now, the following theorem is a generalization of [30, Theorem 2.1]. The theorem states that any distribution with JL moments also provides a sketch for approximate matrix products. A similar statement was made in [113, Lemma 6], but that statement was slightly weaker in its parameters because it resorted to a union bound, which we avoid by using Minkowski's inequality.

Theorem 109. *Given $0 < \varepsilon, \delta < 1/2$, let $\mathcal{D}$ be any distribution over matrices with d columns with the (ε,δ)-JL moment property. Then for A, B any real matrices with d*

rows and $\|A\|_F = \|B\|_F = 1$,

$$\mathbf{Pr}_{S \sim \mathcal{D}} \left[\|A^T S^T S B - A^T B\|_F > 3\varepsilon/2 \right] < \delta.$$

Proof. Let $x, y \in \mathbb{R}^d$ each have ℓ_2 norm 1. Then

$$\langle Sx, Sy \rangle = \frac{\|Sx\|_2^2 + \|Sy\|_2^2 - \|S(x-y)\|_2^2}{2}$$

so that

$$\begin{aligned}
&\mathbf{E}\left[|\langle Sx, Sy\rangle - \langle x, y\rangle|^\ell \right] \\
&\quad = \frac{1}{2^\ell} \cdot \left(\mathbf{E}\left[\left| (\|Sx\|_2^2 - 1) + (\|Sy\|_2^2 - 1) - (\|S(x-y)\|_2^2 - \|x-y\|_2^2) \right|^\ell \right] \right) \\
&\quad \le \frac{3^\ell}{2^\ell} \cdot \max\left\{ \mathbf{E}\left[\left| \|Sx\|_2^2 - 1 \right|^\ell \right], \mathbf{E}\left[\left| \|Sy\|_2^2 - 1 \right|^\ell \right], \mathbf{E}\left[\left| \|S(x-y)\|_2^2 - \|x-y\|_2^2 \right|^\ell \right] \right\} \\
&\quad \le \left(\frac{3\varepsilon}{4} \right)^\ell
\end{aligned}$$

with the middle inequality following by Minkowski's inequality. Now, if A has n columns and B has m columns, label the columns of A as $x_1, \ldots, x_n \in \mathbb{R}^d$ and the columns of B as $y_1, \ldots, y_m \in \mathbb{R}^d$. Define the random variable $X_{i,j} = 1/(\|x_i\|_2 \|y_j\|_2) \cdot (\langle Sx_i, Sy_j\rangle - \langle x_i, y_j\rangle)$. Then $\|A^T S^T S B - A^T B\|_F^2 = \sum_{i \ne j} \|x_i\|_2^2 \cdot \|y_j\|_2^2 \cdot X_{i,j}^2$. Then again by Minkowski's inequality,

$$\begin{aligned}
\mathbf{E}\left[\left(\|A^T S^S B - A^T B\|_F^2 \right)^{\ell/2} \right] &= \mathbf{E}\left[\left| \sum_{i \ne j} \|x_i\|_2^2 \cdot \|y_j\|_2^2 \cdot X_{i,j}^2 \right|^{\ell/2} \right] \\
&\le \left(\sum_{i \ne j} \|x_i\|_2^2 \cdot \|y_j\|_2^2 \cdot \mathbf{E}[|X_{i,j}|^\ell]^{2/\ell} \right)^{\ell/2} \\
&\le \left(\sum_{i \ne j} \|x_i\|_2^2 \cdot \|y_j\|_2^2 \cdot (3\varepsilon/4)^2 \right)^{\ell/2} \\
&\le (3\varepsilon/4)^\ell \cdot (\|A\|_F^2 \cdot \|B\|_F^2)^{\ell/2} \\
&= (3\varepsilon/4)^\ell.
\end{aligned}$$

For $\ell = \log(1/\delta)$,

$$\mathbf{Pr}\left[\|A^T S^S B - A^T B\|_F > 3\varepsilon/\,2 \right] < (2\varepsilon/3)^{-\ell} \cdot \mathbf{E}\left[\|A^T S^S B - A^T B\|_F^\ell \right] \le \delta.$$

■

Remark 110. Often when one constructs a JL distribution $\mathcal{D}$ over $k \times d$ matrices, it is shown that

$$\forall x \in S^{d-1}\ \forall \varepsilon > 1/\sqrt{k}\ \mathbf{Pr}_{S\sim\mathcal{D}}\left[\left|\|Sx\|_2^2 - 1\right| > \varepsilon\right] < e^{-\Theta(\varepsilon^2 k)}$$

Any such distribution automatically satisfies the $(\varepsilon, e^{-\Theta(\varepsilon^2 k)})$-JL moment property for any $\varepsilon > 1/\sqrt{k}$ by converting the tail bound into a moment bound via integration by parts.

Now we arrive at the main point of this section. Several algorithms for approximate linear regression and best rank-r approximation in [30] simply maintain SA as A is updated, where S comes from the JL distribution with $\Omega(\log(1/\delta))$-wise independent $\pm 1/\sqrt{k}$ entries. In fact though, their analyses of their algorithms only use the fact that this distribution satisfies the approximate matrix product sketch guarantees of Theorem 109. Due to Theorem 109 though, we know that *any* distribution satisfying the (ε, δ)-JL moment condition gives an approximate matrix product sketch. Thus, random Bernoulli matrices may be replaced with our sparse JL distributions in this work. We now state some of the algorithmic results given in [30] and describe how our constructions provide improvements in the update time (the time to process new columns, rows, or turnstile updates).

As in [30], when stating our results we will ignore the space and time complexities of storing and evaluating the hash functions in our JL distributions. We discuss this issue later in Remark 113.

5.2.1 Linear regression

In this problem we have a $A \in \mathbb{R}^{d\times n}$ and $b \in \mathbb{R}^d$. We would like to compute a vector $\tilde{x}$ such that $\|A\tilde{x} - b\|_F \le (1+\varepsilon)\cdot \min_{x^*} \|Ax^* - b\|_F$ with probability $1-\delta$. In [30], it is assumed that the entries of A, b require $O(\log(nd))$ bits of precision to store precisely. Both A, b receive turnstile updates.

Theorem 3.2 of [30] proves that such an $\tilde{x}$ can be computed with probability $1-\delta$ from SA and Sb, where S is drawn from a distribution that simultaneously satisfies both the $(1/2, \eta^{-r}\delta)$ and $(\sqrt{\varepsilon/r}, \delta)$-JL moment properties for some fixed constant $\eta > 1$, and where $\mathrm{rank}(A) \le r \le n$. Thus due to Remark 103, we have the following.

Theorem 111. *There is a one-pass streaming algorithm for linear regression in the turnstile model where one maintains a sketch of size $O(n^2\varepsilon^{-1}\log(1/\delta)\log(nd))$. Processing each update requires $O(n + \sqrt{n/\varepsilon}\cdot\log(1/\delta))$ arithmetic operations and hash function evaluations.*

Theorem 111 improves upon the update complexity of $O(n\varepsilon^{-1}\log(1/\delta))$ in [30].

5.2.2 Low rank approximation

In this problem, we have an $A \in \mathbb{R}^{d\times n}$ of rank ρ with entries that require precision $O(\log(nd))$ to store. We would like to compute the best rank-r approximation A_r to A. We define $\Delta_r \stackrel{\text{def}}{=} \|A - A_r\|_F$ as the error of A_r. We relax the problem by only requiring that we compute a matrix A'_r such that $\|A - A'_r\|_F \le (1+\varepsilon)\Delta_r$ with probability $1-\delta$ over the randomness of the algorithm.

Two-pass algorithm: Theorem 4.4 of [30] gives a 2-pass algorithm where in the first pass, one maintains SA where S is drawn from a distribution that simultaneously satisfies both the $(1/2, \eta^{-r}\delta)$ and $(\sqrt{\varepsilon/r}, \delta)$-JL moment properties. It is also assumed that $\rho \ge 2r+1$. The first pass is thus sped up again as in Theorem 111.

One-pass algorithm for column/row-wise updates: Theorem 4.5 of [30] gives a one-pass algorithm in the case that A is seen either one whole column or row at a time. The algorithm maintains both SA and SAA^T where S is drawn from a distribution that simultaneously satisfies both the $(1/2, \eta^{-r}\delta)$ and $(\sqrt{\varepsilon/r}, \delta)$-JL moment properties. This implies the following.

Theorem 112. *There is a one-pass streaming algorithm for approximate low rank approximation with row/column-wise updates where one maintains a sketch of size $O(r\varepsilon^{-1}(n+d)\log(1/\delta)\log(nd))$. Processing each update requires $O(r+\sqrt{r/\varepsilon}\cdot\log(1/\delta))$ amortized arithmetic operations and hash function evaluations per entry of A.*

This improves the amortized update complexity of $O(r\varepsilon^{-1}\log(1/\delta))$ in [30].

Three-pass algorithm for row-wise updates: Theorem 4.6 of [30] gives a three-pass algorithm using less space in the case that A is seen one row at a time. Again, the first pass simply maintains SA where S is drawn from a distribution that satisfies both the $(1/2, \eta^{-r}\delta)$ and $(\sqrt{\varepsilon/r}, \delta)$-JL moment properties. This pass is sped up using our sparser JL distribution.

One-pass algorithm in the turnstile model, bi-criteria: Theorem 4.7 of [30] gives a one-pass algorithm under turnstile updates where SA and RA^T are maintained in the stream. S is drawn from a distribution satisfying both the $(1/2, \eta^{-r\log(1/\delta)/\varepsilon}\delta)$ and $(\varepsilon/\sqrt{r\log(1/\delta)}, \delta)$-JL moment properties. R is drawn from a distribution satisfying both the $(1/2, \eta^{-r}\delta)$ and $(\sqrt{\varepsilon/r}, \delta)$-JL moment properties. Theorem 4.7 of [30] then shows how to compute a matrix of rank $O(r\varepsilon^{-1}\log(1/\delta))$ which achieves the desired error guarantee given SA and RA^T.

One-pass algorithm in the turnstile model: Theorem 4.9 of [30] gives a one-pass algorithm under turnstile updates where SA and RA^T are maintained in the stream. S is drawn from a distribution satisfying both the $(1/2, \eta^{-r\log(1/\delta)/\varepsilon^2}\delta)$ and $(\varepsilon\sqrt{\varepsilon/(r\log(1/\delta))}, \delta)$-JL moment properties. R is drawn from a distribution satisfying both the $(1/2, \eta^{-r}\delta)$ and $(\sqrt{\varepsilon/r}, \delta)$-JL moment properties. Theorem 4.9 of [30] then shows how to compute a matrix of rank r which achieves the desired error guarantee given SA and RA^T.

Remark 113. In the algorithms above, we counted the number of hash function evaluations that must be performed. We use our construction in Figure 5-1(c), which uses $2\log(1/\delta)$-wise independent hash functions. Standard constructions of t-wise independent hash functions over universes with elements fitting in a machine word require $O(t)$ time to evaluate [24]. In our case, this would blow up our update time by factors such as n or r, which could be large. Instead, we use fast multipoint evaluation of polynomials. The standard construction [24] of our desired hash functions mapping some domain $[z]$ onto itself for z a power of 2 takes a degree-$(t-1)$ polynomial p with random coefficients in $\mathbb{F}_z$. The hash function evaluation at some point y is then the evaluation $p(y)$ over $\mathbb{F}_z$. Theorem 55 states that p can be evaluated at t points in total time $\tilde{O}(t)$. We note that in the theorems above, we are always required to evaluate some t-wise independent hash function on many more than t points per stream update. Thus, we can group these evaluation points into groups of size t then perform fast multipoint evaluation for each group. We borrow this idea from [77], which used it to give a fast algorithm for moment estimation in data streams.

5.3 A proof of the Hanson-Wright inequality

In this section we provide a proof of the Hanson-Wright inequality. First, we reminder the reader of its statement.

Theorem 89 (restatement). *There is a universal constant $C > 0$ such that for $z = (z_1, \ldots, z_n)$ a vector of i.i.d. Bernoulli ± 1 random variables, $B \in \mathbb{R}^{n\times n}$ symmetric, and integer $\ell \geq 2$,*

$$\mathbf{E}\left[\left|z^T Bz - \mathrm{trace}(B)\right|^{\ell}\right] \leq C^{\ell} \cdot \max\left\{\sqrt{\ell} \cdot \|B\|_F, \ell \cdot \|B\|_2\right\}^{\ell}.$$

First, we need the following lemma.

Lemma 114. *If X, Y are independent with $\mathbf{E}[Y] = 0$ and if $\ell \geq 2$, then $\mathbf{E}[|X|^{\ell}] \leq \mathbf{E}[|X - Y|^{\ell}]$.*

Proof. Consider the function $f(y) = |X - y|^{\ell}$. Since $f^{(2)}$ is nonnegative on $\mathbb{R}$, the claim follows by Taylor's theorem since $|X - Y|^{\ell} \geq |X|^{\ell} - \ell Y(\mathrm{sgn}(X) \cdot X)^{\ell-1}$. ■

Proof (of Theorem 89). Without loss of generality we can assume $\mathrm{tr}(B) = 0$. This is because if one considers $B' = B - (\mathrm{tr}(B)/n) \cdot I$, then $z^T Bz - \mathrm{tr}(B) = z^T B' z$, and we have $\|B'\|_F \leq \|B\|_F$ and $\|B'\|_2 \leq 2\|B\|_2$. We now start by proving our theorem for ℓ a power of 2 by induction on ℓ. For $\ell = 2$, $\mathbf{E}[(z^T Bz)^2] = 4\sum_{i<j} B_{i,j}^2$ and $\|B\|_F^2 = \sum_i B_{i,i}^2 + 2\sum_{i<j} B_{i,j}^2$. Thus $\mathbf{E}[(z^T Bz)^2] \leq 2\|B\|_F^2$. Next we assume the statement of our Theorem for $\ell/2$ and attempt to prove it for ℓ.

We note that by Lemma 114,

$$\mathbf{E}[|z^T Bz|^{\ell}] \leq \mathbf{E}[|z^T Bz - y^T By|^{\ell}] = \mathbf{E}[|(z+y)^T B(z-y)|^{\ell}],$$

where $y \in \{-1, 1\}^n$ is random and independent of z. Notice that if we swap z_i with y_i then $z + y$ remains constant as does $|z_j - y_j|$ and that $z_i - y_i$ is replaced by its negation. Consider averaging over all such swaps. Let $\xi_i = ((z+y)^T B)_i$ and $\eta_i = z_i - y_i$. Let

ν_i be 1 if we did not swap and -1 if we did. Then $(z+y)^T B(z-y) = \sum_i \xi_i \eta_i \nu_i$. Averaging over all swaps,

$$\mathbf{E}_z[|(z+y)^T B(z-y)|^\ell] \le \left(\sum_i \xi_i^2 \eta_i^2\right)^{\ell/2} \cdot \ell^{\ell/2} \le 2^\ell \ell^{\ell/2} \cdot \left(\sum_i \xi_i^2\right)^{\ell/2}.$$

The first inequality is by Lemma 45, and the second uses that $|\eta_i| \le 2$. Note that

$$\sum_i \xi_i^2 = \|B(z+y)\|_2^2 \le 2\|Bz\|_2^2 + 2\|By\|_2^2,$$

and hence

$$\mathbf{E}[|z^T Bz|^\ell] \le 2^\ell \sqrt{\ell}^\ell \mathbf{E}[(2\|Bz\|_2^2 + 2\|By\|_2^2)^{\ell/2}] \le 4^\ell \sqrt{\ell}^\ell \mathbf{E}[(\|Bz\|_2^2)^{\ell/2}],$$

with the final inequality using Minkowski's inequality (namely that $|\mathbf{E}[|X+Y|^p]|^{1/p} \le |\mathbf{E}[|X|^p]|^{1/p} + |\mathbf{E}[|Y|^p]|^{1/p}$ for any random variables X, Y and any $1 \le p < \infty$).

Next note $\|Bz\|_2^2 = \langle Bz, Bz\rangle = z^T B^2 z$. Let $A = B^2 - \frac{\mathrm{tr}(B^2)}{n} I$. Then $\mathrm{tr}(A) = 0$. Also, $\|A\|_F \le \|B\|_F \|B\|_2$ and $\|A\|_2 \le \|B\|_2^2$. The former holds since

$$\|A\|_F^2 = \left(\sum_i \lambda_i^4\right) - \left(\sum_i \lambda_i^2\right)^2 \Big/ n \le \sum_i \lambda_i^4 \le \|B\|_F^2 \|B\|_2^2,$$

where B has eigenvalues $\lambda_1, \ldots, \lambda_n$. The latter holds since the eigenvalues of A are $\lambda_i^2 - (\sum_{j=1}^n \lambda_j^2)/n$ for each $i \in [n]$. The largest eigenvalue of A is thus at most that of B^2, and since $\lambda_i^2 \ge 0$, the smallest eigenvalue of A cannot be smaller than $-\|B\|_2^2$.

We then have

$$\mathbf{E}[(\|Bz\|_2^2)^{\ell/2}] = \mathbf{E}\left[\left|\|B\|_F^2 + z^T Az\right|^{\ell/2}\right] \le 2^\ell \max\left\{\|B\|_F^\ell, \mathbf{E}[|z^T Az|^{\ell/2}]\right\}.$$

Hence employing the inductive hypothesis on A we have that

$$\begin{aligned}
\mathbf{E}[|z^T Bz|^\ell] &\le 8^\ell \max\left\{\sqrt{\ell}\|B\|_F, C^{\ell/2}\ell^{3/4}\|A\|_F, C^{\ell/2}\ell\sqrt{\|A\|_2}\right\}^\ell \\
&\le 8^\ell C^{\ell/2} \max\left\{\sqrt{\ell}\|B\|_F, \ell^{3/4}\sqrt{\|B\|_F\|B\|_2}, \ell\|B\|_2\right\}^\ell \\
&= 8^\ell C^{\ell/2} \max\left\{\sqrt{\ell}\|B\|_F, \ell\|B\|_2\right\}^\ell,
\end{aligned}$$

with the final equality holding since the middle term above is the geometric mean of the other two, and thus is dominated by at least one of them. This proves our hypothesis as long as $C \ge 64$.

To prove our statement for general ℓ, set $\ell' = 2^{\lceil \log_2 \ell \rceil}$. Then by the power mean inequality and our results for ℓ' a power of 2,

$$\mathbf{E}[|z^T Bz|^\ell] \le (\mathbf{E}[|z^T Bz|^{\ell'}])^{\ell/\ell'} \le 128^\ell \max\left\{\sqrt{\ell}\|B\|_F, \ell\|B\|_2\right\}^\ell.$$

■

5.4 Derandomizing the JL lemma

The main theorem of this section is the following.

Theorem 115. *For any $0 < \varepsilon, \delta < 1/2$ and integer $d > 0$, there exists a JL distribution with optimal target dimension $k = O(\varepsilon^{-2}\log(1/\delta))$ such that a random matrix can be sampled from this distribution using $O(\log d + \log(1/\varepsilon)\log(1/\delta) + \log(1/\delta)\log\log(1/\delta))$ uniform random bits. Given $x \in \mathbb{R}^d$ and the random seed specifying a matrix, we can compute the embedding of x in $d^{O(1)}$ time.*

The main idea in our proof of Theorem 115 is to gradually reduce the dimension. In particular, note that in Eq. (5.2) in the proof of Theorem 90, we could embed into the optimal target dimension $k = O(\varepsilon^{-2}\log(1/\delta))$ and set $\ell = \log(1/\delta)$ to obtain a JL-distribution, but then we would need the entries of the sign matrix to be $2\ell = 2\log(1/\delta)$-wise independent. The seed length required would then be $O(\log(1/\delta)\log d)$. On the other hand, we could instead embed into suboptimal dimension $k = O(\varepsilon^{-2}\delta^{-1})$ then set $\ell = 2$, requiring only 4-wise independent entries and thus seed length only $O(\log d)$.

We exploit the above tradeoff between target dimension and independence required by gradually reducing the dimension. In particular, consider values $\varepsilon', \delta' > 0$ which we will pick later. Define $t_j = \delta'^{-1/2^j}$. We embed $\mathbb{R}^d$ into $\mathbb{R}^{k_1}$ for $k_1 = \varepsilon'^{-2}t_1$. We then embed $\mathbb{R}^{k_{j-1}}$ into $\mathbb{R}^{k_j}$ for $k_j = \varepsilon'^{-2}t_j$ until the point $j = j^* = O(\log(1/\delta')/\log\log(1/\delta'))$ where $t_{j^*} = O(\log^3(1/\delta'))$. We then embed $\mathbb{R}^{k_{j^*}}$ into $\mathbb{R}^k$ for $k = O(\varepsilon'^{-2}\log(1/\delta'))$.

The embeddings into each k_j are performed by picking a Bernoulli matrix with r_j-wise independent entries, as in Theorem 90. To achieve error probability δ' of having distortion larger than $(1+\varepsilon')$ in the jth step, Eq. (5.2) in the proof of Theorem 90 tells us we need $r_j = O(\log_{t_j}(1/\delta'))$. Thus, in the first embedding into $\mathbb{R}^{k_1}$ we need $O(\log d)$ random bits. Then in the future embeddings except the last, we need $O(2^j \cdot (\log(1/\varepsilon') + 2^{-j}\log(1/\delta')))$ random bits to embed into $\mathbb{R}^{k_j}$. In the final embedding we require $O(\log(1/\delta') \cdot (\log(1/\varepsilon') + \log\log(1/\delta'))$ random bits. Thus, in total, we have used $O(\log d + \log(1/\varepsilon')\log(1/\delta') + \log(1/\delta')\log\log(1/\delta'))$ bits of seed to achieve error probability $O(\delta' \cdot j^*)$ of distortion $(1+\varepsilon')^{j^*}$. Theorem 115 thus follows by applying this argument with error probability $\delta' = \delta/(j^*+1)$ and distortion parameter $\varepsilon' = \Theta(\varepsilon/j^*)$.

Remark 116. One may worry that along the way we are embedding into potentially very large dimension (e.g. $1/\delta$ may be $2^{\Omega(d)}$), so that our overall running time could be exponentially large. However, we can simply start the above iterative embedding at the level j where $\varepsilon^{-2}t_j < d$.

Chapter 6

Conclusion

In this thesis we presented several efficient solutions for estimating the number of distinct elements, the Hamming norm, moments, and entropy for a vector being updated in a turnstile data stream. We also presented a more time-efficient algorithm for performing dimensionality reduction in ℓ_2 in a data stream, by designing a family of sparse embedding matrices. Several natural questions remain open from the work described here.

The distinct elements algorithm in Chapter 2 requires multiplication to achieve constant update time. For example, the final algorithm in Figure 2-2 performs least significant bit computations, which uses multiplication to achieve a constant time algorithm in the word RAM [20].

OPEN: Is there an algorithm for simultaneously achieving $O(\varepsilon^{-2} + \log d)$ space and $O(1)$ update and reporting times for F_0-estimation which only uses $\mathbf{AC^0}$ operations?

In the work [72], Jayram and Woodruff showed that F_0-estimation with success probability $1 - \delta$ requires $\Omega(\varepsilon^{-2} \log 1/\delta + \log d)$ space. Naïvely using our algorithm to achieve such success probability would give space $O((\varepsilon^{-2} + \log d) \log(1/\delta))$, although some slight improvement to this upper bound is known to be possible [72].

OPEN: Is there an algorithm for F_0 estimation with success probability $1 - \delta$ using space $O(\varepsilon^{-2} \log(1/\delta) + \log d)$?

Our algorithm for ℓ_0 estimation requires space $O(\varepsilon^{-2} \log d(\log 1/\varepsilon + \log \log mM))$, whereas the best known lower bound is $\Omega(\varepsilon^{-2} \log d + \log \log mM)$.

OPEN: Is there an algorithm for ℓ_0 estimation using space $O(\varepsilon^{-2} \log d + \log \log mM)$?

Regarding F_p estimation for $0 < p < 2$, we now understand the space requirements, however there is still room for improvement in the update and reporting times. It is known that optimal space and $O(1)$ update/reporting times are simultaneously achievable for the case $p = 2$ [29, 118].

OPEN: Is there an algorithm for F_p estimation for all $0 < p < 2$ which simultaneously achieves optimal space and $O(1)$ update and reporting times?

In our reduction from entropy estimation to moment estimation, although we were able to achieve near-optimal dependence on ε in the space, our dependence on $\log mM$ is suboptimal. Furthermore, the polylogarithmic factors in our bound are large.

OPEN: Is there an algorithm for empirical entropy estimation using $O(\varepsilon^{-2}\log mM + \log\log d)$ space in the turnstile model?

Regarding the Johnson-Lindenstrauss transform, perhaps the most tantalizing open question is whether we can embed sparse vectors in (near)-linear time.

OPEN: Devise a JL-distribution over $\mathbb{R}^{k\times d}$ for optimal k such that for any matrix S in the support of the distribution and $x \in \mathbb{R}^d$, we can compute Sx in time $\tilde{O}(\|x\|_0 + k)$.

Two perhaps easier open problems are the following.

OPEN: Determine the best column sparsity achievable in a JL distribution.

OPEN: Devise a JL-distribution over $\mathbb{R}^{k\times d}$ for optimal k such that for any matrix S in the support of the distribution and $x \in \mathbb{R}^d$, we can compute Sx in time $\tilde{O}(d)$.

Also, we saw in Section 5.4 that it is possible to devise a JL-distribution over $k \times d$ matrices for $k = O(\varepsilon^{-2}\log(1/\delta))$ such that a matrix can be drawn from the distribution using only $O(\log d + \log(1/\varepsilon)\log(1/\delta) + \log(1/\delta)\log\log(1/\delta))$ uniform random bits. As we saw in Section 1.5.2, a JL-distribution exists which can be sampled from using $O(\log(d/\delta))$ random bits.

OPEN: Give an explicit JL-distribution which can be sampled from using $s = O(\log(d/\delta))$ random bits. Two interesting degrees of explicitness to consider are as follows: (1) given a seed representing some matrix S and given some $x \in \mathbb{R}^d$, there should be a deterministic algorithm to compute Sx in $d^{O(1)}$ time, and the more stringent (2) given a seed and $(i, j) \in [k] \times [d]$, one should be able to compute the (i, j) entry of the embedding matrix specified by s in $s^{O(1)}$ time.

Bibliography

[1] Swarup Acharya, Phillip B. Gibbons, Viswanath Poosala, and Sridhar Ramaswamy. The Aqua approximate query answering system. In *SIGMOD Conference*, pages 574–576, 1999.

[2] Dimitris Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *J. Comput. Syst. Sci.*, 66(4):671–687, 2003.

[3] Nir Ailon and Bernard Chazelle. Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform. In *Proceedings of the 38th ACM Symposium on Theory of Computing (STOC)*, pages 557–563, 2006.

[4] Nir Ailon and Edo Liberty. Fast dimension reduction using Rademacher series on dual BCH codes. *Discrete Comput. Geom.*, 42(4):615–630, 2009.

[5] Nir Ailon and Edo Liberty. Almost optimal unrestricted fast Johnson-Lindenstrauss transform. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 185–191, 2011.

[6] Aditya Akella, Ashwin Bharambe, Mike Reiter, and Srinivasan Seshan. Detecting DDoS attacks on ISP networks. In *ACM SIGMOD/PODS Workshop on Management and Processing of Data Streams (MPDS)*, 2003.

[7] Noga Alon, Phillip B. Gibbons, Yossi Matias, and Mario Szegedy. Tracking join and self-join sizes in limited storage. *J. Comput. Syst. Sci.*, 64(3):719–747, 2002.

[8] Noga Alon, Yossi Matias, and Mario Szegedy. The Space Complexity of Approximating the Frequency Moments. *J. Comput. Syst. Sci. (see also STOC 1996)*, 58(1):137–147, 1999.

[9] Rosa I. Arriaga and Santosh Vempala. An algorithmic theory of learning: Robust concepts and random projection. *Machine Learning*, 63(2):161–182, 2006.

[10] Ziv Bar-Yossef, T. S. Jayram, Robert Krauthgamer, and Ravi Kumar. The sketching complexity of pattern matching. In *8th International Workshop on Randomization and Computation (RANDOM)*, pages 261–272, 2004.

[11] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. In *Proceedings of the 43rd Symposium on Foundations of Computer Science (FOCS)*, pages 209–218, 2002.

[12] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, D. Sivakumar, and Luca Trevisan. Counting distinct elements in a data stream. In *Randomization and Approximation Techniques, 6th International Workshop (RANDOM)*, pages 1–10, 2002.

[13] Ziv Bar-Yossef, Ravi Kumar, and D. Sivakumar. Reductions in streaming algorithms, with an application to counting triangles in graphs. In *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 623–632, 2002.

[14] Bonnie Berger. The fourth moment method. *SIAM J. Comput.*, 26(4):1188–1207, 1997.

[15] Kevin S. Beyer, Peter J. Haas, Berthold Reinwald, Yannis Sismanis, and Rainer Gemulla. On synopses for distinct-value estimation under multiset operations. In *SIGMOD Conference*, pages 199–210, 2007.

[16] Lakshminath Bhuvanagiri and Sumit Ganguly. Estimating entropy over data streams. In *Proceedings of the 14th Annual European Symposium on Algorithms (ESA)*, pages 148–159, 2006.

[17] Daniel K. Blandford and Guy E. Blelloch. Compact dictionaries for variable-length keys and data with applications. *ACM Transactions on Algorithms*, 4(2), 2008.

[18] Vladimir Braverman, Rafail Ostrovsky, and Yuval Rabani. Rademacher chaos, random Eulerian graphs and the sparse Johnson-Lindenstrauss transform. *CoRR*, abs/1011.2590, 2010.

[19] Andrei Z. Broder. On the resemblence and containment of documents. In *Compression and Complexity of Sequences*, 1997.

[20] Andrej Brodnik. Computation of the least significant set bit. In *Proceedings of the 2nd Electrotechnical and Computer Science Conference (ERK)*, 1993.

[21] Joshua Brody and Amit Chakrabarti. A multi-round communication lower bound for gap hamming and some consequences. In *Proceedings of the 23rd Annual IEEE Conference on Computational Complexity (CCC)*, pages 358–368, 2009.

[22] Paul Brown, Peter J. Haas, Jussi Myllymaki, Hamid Pirahesh, Berthold Reinwald, and Yannis Sismanis. Toward automated large-scale information integration and discovery. In *Data Management in a Connected World*, pages 161–180, 2005.

[23] CAIDA analysis of Code Red, 2001. http://www.caida.org/analysis/security/code-red/.

[24] J. Lawrence Carter and Mark N. Wegman. Universal classes of hash functions. *J. Comput. Syst. Sci.*, 18(2):143–154, 1979.

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

[26] Amit Chakrabarti, Khanh Do Ba, and S. Muthukrishnan. Estimating Entropy and Entropy Norm on Data Streams. In *Proceedings of the 23rd Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 196–205, 2006.

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

[28] John M. Chambers, Colin L. Mallows, and B. W. Stuck. A method for simulating stable random variables. *J. Amer. Statist. Assoc.*, 71:340–344, 1976.

[29] Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. *Theor. Comput. Sci.*, 312(1):3–15, 2004.

[30] Kenneth L. Clarkson and David P. Woodruff. Numerical linear algebra in the streaming model. In *Proceedings of the 41st ACM Symposium on Theory of Computing (STOC)*, pages 205–214, 2009.

[31] Edith Cohen. Size-estimation framework with applications to transitive closure and reachability. *J. Comput. Syst. Sci.*, 55(3):441–453, 1997.

[32] Graham Cormode, Mayur Datar, Piotr Indyk, and S. Muthukrishnan. Comparing data streams using Hamming norms (how to zero in). *IEEE Trans. Knowl. Data Eng.*, 15(3):529–540, 2003.

[33] Graham Cormode, Piotr Indyk, Nick Koudas, and S. Muthukrishnan. Fast mining of massive tabular data via approximate distance computations. In *ICDE*, pages 605–, 2002.

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

[35] Anirban Dasgupta, Ravi Kumar, and Tamás Sarlós. A sparse Johnson-Lindenstrauss transform. In *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC)*, pages 341–350, 2010.

[36] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Struct. Algorithms*, 22(1):60–65, 2003.

[37] Tamraparni Dasu, Theodore Johnson, S. Muthukrishnan, and Vladislav Shkapenyuk. Mining database structure; or, how to build a data quality browser. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, pages 240–251, 2002.

[38] Ilias Diakonikolas, Daniel M. Kane, and Jelani Nelson. Bounded independence fools degree-2 threshold functions. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 11–20, 2010.

[39] Devdatt P. Dubhashi and Alessandro Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, 2009.

[40] Devdatt P. Dubhashi and Desh Ranjan. Balls and bins: A study in negative dependence. *Random Struct. Algorithms*, 13(2):99–124, 1998.

[41] Marianne Durand and Philippe Flajolet. Loglog counting of large cardinalities (extended abstract). In *11th Annual European Symposium on Algorithms (ESA)*, pages 605–617, 2003.

[42] Lars Engebretsen, Piotr Indyk, and Ryan O'Donnell. Derandomized dimensionality reduction with applications. In *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 705–712, 2002.

[43] Funda Ergün, Hossein Jowhari, and Mert Saglam. Periodicity in streams. In *14th International Workshop on Randomization and Computation (RANDOM)*, pages 545–559, 2010.

[44] Cristian Estan, George Varghese, and Michael E. Fisk. Bitmap algorithms for counting active flows on high-speed links. *IEEE/ACM Trans. Netw.*, 14(5):925–937, 2006.

[45] Joan Feigenbaum, Sampath Kannan, Martin Strauss, and Mahesh Viswanathan. An approximate L1-difference algorithm for massive data streams. *SIAM J. Comput. (see also FOCS 1999)*, 32(1):131–151, 2002.

[46] Dan Feldman, Morteza Monemizadeh, Christian Sohler, and David P. Woodruff. Coresets and sketches for high dimensional subspace problems. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 630–649, 2010.

[47] Sheldon J. Finkelstein, Mario Schkolnick, and Paolo Tiberio. Physical database design for relational databases. *ACM Trans. Database Syst.*, 13(1):91–128, 1988.

[48] Philippe Flajolet, Eric Fusy, Olivier Gandouet, and Frédéric Meunier. Hyperloglog: The analysis of a near-optimal cardinality estimation algorithm. *Discrete Mathematics and Theoretical Computer Science (DMTCS)*, AH:127–146, 2007.

[49] Philippe Flajolet and G. Nigel Martin. Probabilistic counting algorithms for data base applications. *J. Comput. Syst. Sci. (see also FOCS 1983)*, 31(2):182–209, 1985.

[50] Peter Frankl and Hiroshi Maehara. The Johnson-Lindenstrauss lemma and the sphericity of some graphs. *J. Comb. Theory. Ser. B*, 44(3):355–362, 1988.

[51] Michael L. Fredman, János Komlós, and Endre Szemerédi. Storing a sparse table with O(1) worst case access time. *J. ACM*, 31(3):538–544, 1984.

[52] Michael L. Fredman and Dan E. Willard. Surpassing the information theoretic bound with fusion trees. *J. Comput. Syst. Sci.*, 47(3):424–436, 1993.

[53] Sumit Ganguly. Estimating frequency moments of data streams using random linear combinations. In *Proceedings of the 8th International Workshop on Randomization and Computation (RANDOM)*, pages 369–380, 2004.

[54] Sumit Ganguly. Counting distinct items over update streams. *Theor. Comput. Sci.*, 378(3):211–222, 2007.

[55] Sumit Ganguly and Graham Cormode. On estimating frequency moments of data streams. In *Proceedings of the 11th International Workshop on Randomization and Computation (RANDOM)*, pages 479–493, 2007.

[56] Sumit Ganguly, Abhayendra N. Singh, and Satyam Shankar. Finding frequent items over general update streams. In *Proceedings of the 20th International Conference on Scientific and Statistical Database Management (SSDBM)*, pages 204–221, 2008.

[57] Phillip B. Gibbons. Distinct sampling for highly-accurate answers to distinct values queries and event reports. In *VLDB*, pages 541–550, 2001.

[58] Phillip B. Gibbons and Srikanta Tirthapura. Estimating simple functions on the union of data streams. In *Proceedings of the 13th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 281–291, 2001.

[59] Anna C. Gilbert, Sudipto Guha, Piotr Indyk, Yannis Kotidis, S. Muthukrishnan, and Martin Strauss. Fast, small-space algorithms for approximate histogram maintenance. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 389–398, 2002.

[60] Sudipto Guha, Andrew McGregor, and Suresh Venkatasubramanian. Streaming and sublinear approximation of entropy and information distances. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 733–742, 2006.

[61] Uffe Haagerup. The best constants in the Khintchine inequality. *Studia Math.*, 70(3):231–283, 1982.

[62] David Lee Hanson and Farroll Tim Wright. A bound on tail probabilities for quadratic forms in independent random variables. *Ann. Math. Statist.*, 42(3):1079–1083, 1971.

[63] Nicholas J. A. Harvey, Jelani Nelson, and Krzysztof Onak. Sketching and streaming entropy via approximation theory. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 489–498, 2008.

[64] Aicke Hinrichs and Jan Vybíral. Johnson-Lindenstrauss lemma for circulant matrices. *arXiv*, abs/1001.4919, 2010.

[65] Piotr Indyk. Algorithmic applications of low-distortion geometric embeddings. In *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 10–33, 2001.

[66] Piotr Indyk. Algorithms for dynamic geometric problems over data streams. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC)*, pages 373–380, 2004.

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

[68] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the 30th ACM Symposium on Theory of Computing (STOC)*, pages 604–613, 1998.

[69] Piotr Indyk and David P. Woodruff. Tight lower bounds for the distinct elements problem. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 283–, 2003.

[70] T. S. Jayram, Ravi Kumar, and D. Sivakumar. The one-way communication complexity of gap hamming distance. *Theory of Computing*, 4(1):129–135, 2008.

[71] T. S. Jayram and David P. Woodruff. The data stream space complexity of cascaded norms. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 765–774, 2009.

[72] T. S. Jayram and David P. Woodruff. Optimal bounds for Johnson-Lindenstrauss transforms and streaming problems with low error. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1–10, 2011.

[73] William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

[74] Daniel M. Kane, Raghu Meka, and Jelani Nelson. Almost optimal explicit Johnson-Lindenstrauss transformations. Manuscript, 2011.

[75] Daniel M. Kane and Jelani Nelson. A derandomized sparse Johnson-Lindenstrauss transform. *CoRR*, abs/1006.3585, 2010.

[76] Daniel M. Kane and Jelani Nelson. Sparser Johnson-Lindenstrauss transforms. *CoRR*, abs/1012.1577, 2011.

[77] Daniel M. Kane, Jelani Nelson, Ely Porat, and David P. Woodruff. Fast moment estimation in data streams in optimal space. In *Proceedings of the 43rd ACM Symposium on Theory of Computing (STOC), to appear*, 2011.

[78] Daniel M. Kane, Jelani Nelson, and David P. Woodruff. On the exact space complexity of sketching and streaming small norms. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1161–1178, 2010.

[79] Daniel M. Kane, Jelani Nelson, and David P. Woodruff. An optimal algorithm for the distinct elements problem. In *Proceedings of the 29th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*, pages 41–52, 2010.

[80] Eyal Kaplan, Moni Naor, and Omer Reingold. Derandomized constructions of k-wise (almost) independent permutations. *Algorithmica*, 55(1):113–133, 2009.

[81] Zohar Karnin, Yuval Rabani, and Amir Shpilka. Explicit dimension reduction and its applications. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity (CCC), to appear*, 2011.

[82] Felix Krahmer and Rachel Ward. New and improved Johnson-Lindenstrauss embeddings via the Restricted Isometry Property. *arXiv*, abs/1009.0744, 2010.

[83] Balachander Krishnamurthy, Subhabrata Sen, Yin Zhang, and Yan Chen. Sketch-based change detection: methods, evaluation, and applications. In *Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement (IMC)*, pages 234–247, 2003.

[84] Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[85] Anukool Lakhina, Mark Crovella, and Christophe Diot. Mining anomalies using traffic feature distributions. In *Proceedings of the ACM SIGCOMM Conference*, pages 217–228, 2005.

[86] Cornelius Lanczos. A precision approximation of the Gamma function. *J. Soc. Indus. Appl. Math.: Series B*, 1:76–85, 1964.

[87] Ping Li. Estimators and tail bounds for dimension reduction in l_p $(0 < p \leq 2)$ using stable random projections. In *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 10–19, 2008.

[88] Ping Li. Compressed counting. In *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 412–421, 2009.

[89] Jirí Matousek. On variants of the Johnson-Lindenstrauss lemma. *Random Struct. Algorithms*, 33(2):142–156, 2008.

[90] Raghu Meka and David Zuckerman. Pseudorandom generators for polynomial threshold functions. In *Proceedings of the 42nd Annual ACM Symposium on Theory of Computing (STOC), (see also CoRR abs/0910.4122)*, pages 427–436, 2010.

[91] Peter Bro Miltersen, Noam Nisan, Shmuel Safra, and Avi Wigderson. On data structures and asymmetric communication complexity. *J. Comput. Syst. Sci.*, 57(1):37–49, 1998.

[92] Morteza Monemizadeh and David P. Woodruff. 1-pass relative-error L_p sampling with applications. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1143–160, 2010.

[93] Robert Morris. Counting large numbers of events in small registers. *Commun. ACM*, 21(10):840–842, 1978.

[94] Rajeev Motwani and Prabakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[95] J. Ian Munro and Mike Paterson. Selection and sorting with limited storage. In *Proceedings of the 19th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 253–258, 1978.

[96] S. Muthukrishnan. Data Streams: Algorithms and Applications. *Foundations and Trends in Theoretical Computer Science*, 1(2):117–236, 2005.

[97] Jelani Nelson and David P. Woodruff. A near-optimal algorithm for L1-difference. *CoRR*, abs/0904.2027, 2009.

[98] Jelani Nelson and David P. Woodruff. Fast manhattan sketches in data streams. In *Proceedings of the 29th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems(PODS)*, pages 99–110, 2010.

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

[100] Krzysztof Oleszkiewicz. On p-pseudostable random variables, rosenthal spaces and l_p^n ball slicing. *Geometric Aspects of Functional Analysis*, 1807:188–210, 2003.

[101] Mark H. Overmars. *The Design of Dynamic Data Structures*. Springer, 1983.

[102] Sriram Padmanabhan, Bishwaranjan Bhattacharjee, Timothy Malkemus, Leslie Cranston, and Matthew Huras. Multi-dimensional clustering: A new data layout scheme in db2. In *SIGMOD Conference*, pages 637–641, 2003.

[103] Anna Pagh and Rasmus Pagh. Uniform hashing in constant time and optimal space. *SIAM J. Comput.*, 38(1):85–96, 2008.

[104] Christopher R. Palmer, George Siganos, Michalis Faloutsos, Christos Faloutsos, and Phillip B. Gibbons. The connectivity and fault-tolerance of the internet topology. In *NRDM Workshop*, 2001.

[105] Vipin Kumar Pang-Ning Tan, Michael Steinbach. *Introduction to Data Mining*. Addison-Wesley, 2005.

[106] George McArtney Phillips. *Interpolation and Approximation by Polynomials*. Springer-Verlag, New York, 2003.

[107] George McArtney Phillips and Peter John Taylor. *Theory and Applications of Numerical Analysis*. Academic Press, 2nd edition, 1996.

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

[109] Theodore J. Rivlin. *Chebyshev Polynomials*. John Wiley and Sons Co., 2 edition, 1990.

[110] Theodore J. Rivlin. *Chebyshev Polynomials: From Approximation Theory to Algebra and Number Theory*. John Wiley & Sons, 2nd edition, 1990.

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

[112] Walter Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, third edition, 1976.

[113] Tamás Sarlós. Improved approximation algorithms for large matrices via random projections. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 143–152, 2006.

[114] Patricia G. Selinger, Morton M. Astrahan, Donald D. Chamberlin, Raymond A. Lorie, and Thomas G. Price. Access path selection in a relational database management system. In *SIGMOD Conference*, pages 23–34, 1979.

[115] Amit Shukla, Prasad Deshpande, Jeffrey F. Naughton, and Karthikeyan Ramasamy. Storage estimation for multidimensional aggregates in the presence of hierarchies. In *Proc. VLDB*, pages 522–531, 1996.

[116] Alan Siegel. On universal classes of extremely random constant-time hash functions. *SIAM J. Computing*, 33(3):505–543, 2004.

[117] D. Sivakumar. Algorithmic derandomization via complexity theory. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 619–626, 2002.

[118] Mikkel Thorup and Yin Zhang. Tabulation based 4-universal hashing with applications to second moment estimation. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 615–624, 2004.

[119] Santosh Vempala. *The random projection method*, volume 65 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*. American Mathematical Society, 2004.

[120] Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra*. Cambridge University Press, 1999.

[121] Jan Vybíral. A variant of the Johnson-Lindenstrauss lemma for circulant matrices. *arXiv*, abs/1002.2847, 2010.

[122] Kilian Q. Weinberger, Anirban Dasgupta, John Langford, Alexander J. Smola, and Josh Attenberg. Feature hashing for large scale multitask learning. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, pages 1113–1120, 2009.

[123] Man Wah Wong. *Complex Analysis*, volume 2. World Scientific, 2008.

[124] David P. Woodruff. Optimal space lower bounds for all frequency moments. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 167–175, 2004.

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

[126] Kuai Xu, Zhi-Li Zhang, and Supratik Bhattacharyya. Profiling Internet backbone traffic: behavior models and applications. In *Proceedings of the ACM SIGCOMM Conference*, pages 169–180, 2005.

[127] Makoto Yamazato. Unimodality of infinitely divisible distribution functions of class L. *Ann. Probability*, 6:523–531, 1978.

[128] Vladimir Mikhailovich Zolotarev. *One-dimensional Stable Distributions*. Vol. 65 of Translations of Mathematical Monographs, American Mathematical Society, 1986.