Streaming Sketching: Mathematical Theory and Practical Algorithms

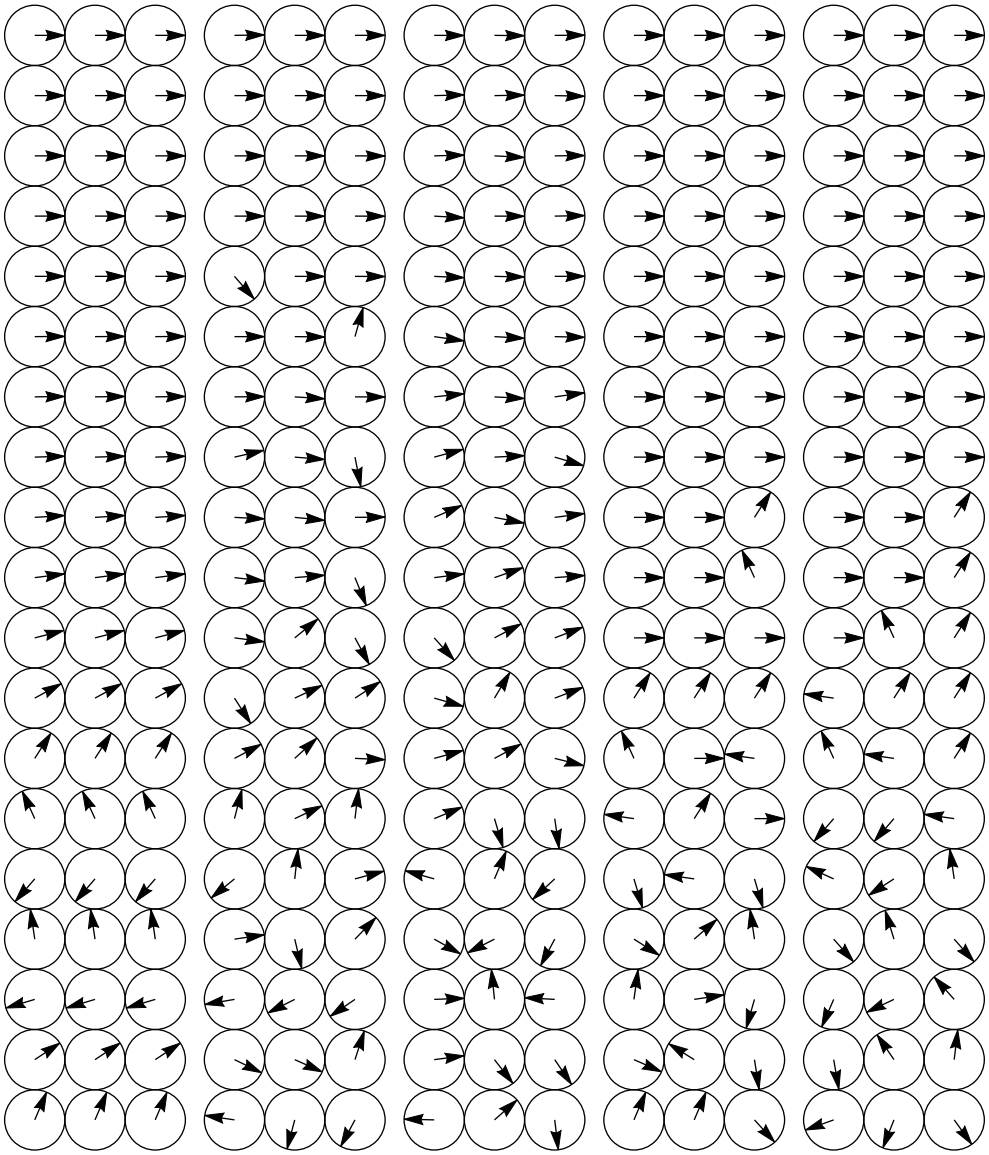
by

Dingyu Wang

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in the University of Michigan
2025

Doctoral Committee:

Professor Seth Pettie, Chair
Professor Nikhil Bansal
Professor Michał Dereziński
Professor Mark Rudelson
Professor David Woodruff



The river will forgive all the ripples.

Dingyu Wang
wangdy@umich.edu
ORCID iD: 0009-0000-5462-8715

© Dingyu Wang 2025

ACKNOWLEDGEMENTS

I am deeply grateful to my advisor, Seth Pettie, for his academic, mental, and philosophical guidance over the past six years. During my PhD, I was knocked down by life many times, and it was he who encouraged me again and again, lifted me up, and helped me keep moving forward.

I also want to thank my committee members, Nikhil Bansal, Michał Dereziński, Mark Rudelson, and David Woodruff. Their feedbacks on my proposal broadened my horizon and helped me make my research more comprehensive. Prof. Rudelson has taught me so many important tools and concepts in probability theory that have been shaping my research. He has also pointed me to the right direction many times when I was lost in the complicated probabilistic structures of the subjects in my research.

Thanks to the theory fellows in the department and the ones I met at conferences and workshops. I have been gaining new inspirations during conversations and talks.

Thanks to my dear friends, who taught me how to drive my stick-shift car, who have farmed with me without growing anything edible, who taught me how to build a snowman, who woke up in the middle of the night to rescue me (when my apartment's outer door was frozen solid), who found my glasses that were flung off by inertia and wind when we rode the pendulum, who drove me to pick up my crashed car, who played music with me, who made games with me, who cooked me Zongzi, and who went all the way south with me until there was no road left. Without my dear friends, I would have long since fallen into the depths of despair.

Finally, I want to thank my parents for their unconditional love and support, which has given me the freedom to wander in the abstract world.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS ii

LIST OF FIGURES vi

LIST OF TABLES viii

LIST OF PROGRAMS ix

REMARKS ON NOTATIONS x

ABSTRACT xii

CHAPTER

1 Introduction 1

1.1 Computation over Streams 1
1.2 Scope and Background 2
1.3 Tractability of Queries 3
1.4 Universality of Sketches 11
1.5 `COUNT-DISTINCT` Sketches 14
1.6 Sampling 20
1.7 Approximate Counting 26
1.8 Road Map 27

2 Modeling Streaming Computation 29

2.1 Modeling Streams 29
2.2 Modeling Computation 31
2.3 Modeling Queries 35
2.4 Homomorphic Sketching System 37

3 Infinitely Divisible Sketches . . . 39

3.1 Prelude: Infinite Divisibility of Sketches . . . 39
3.2 Overview . . . 43
3.3 Preliminaries: Lévy Processes . . . 45
3.4 Estimating f-Moments with Lévy Processes . . . 47
3.5 G-Sampling with Subordinators . . . 56
3.6 LévyPCSA and LévyHyperLogLog . . . 71
3.7 Previous Sketches and Lévy Processes . . . 76
3.8 Tractability and Lévy Processes . . . 81
3.9 Summary . . . 87
3.10 Generalization . . . 89

4 Harmonic Decomposition of Queries . . . 95

4.1 Prelude: Probabilistic Counting with Deletions . . . 96
4.2 Background . . . 99
4.3 Technical Introduction & Related Work . . . 106
4.4 Implicit Level Selection and Smoothed Subsampling . . . 114
4.5 SymmetricPoissonTower . . . 116
4.6 Analysis of Combined Estimators . . . 124
4.7 Proof of Theorem 4.1 . . . 131
4.8 Summary . . . 132
4.9 Example Functions . . . 133
4.10 Mathematical Lemmas . . . 135

5 Cardinality Estimation . . . 139

5.1 Prelude: The Structure of Cardinality Sketches . . . 140
5.2 The Fisher-Shannon Number . . . 145
5.3 Fish Numbers of PCSA and LogLog . . . 162
5.4 A Sharp Lower Bound on Linearizable Sketches . . . 173
5.5 Fishmonger . . . 191
5.6 Interlude: Non-Commutative Sketching . . . 197
5.7 Beyond Fisher-Shannon: the Sequential Setting . . . 201
5.8 The τ-GRA Statistics . . . 232

6 Generalized Cardinality Estimation . . . 249

6.1 Preliminaries: Commutative Monoids 249
6.2 `COUNT-DISTINCT` over Abelian Groups 251
6.3 `COUNT-DISTINCT` over Commutative Monoids 254
6.4 Applications 257
6.5 A Statistical Analysis of the Finite Field Random Projection Approach . . . 259

7 Approximate Counting 277

7.1 Multi-dimensional Approximate Counting 278
7.2 Symmetric Approximate Counting 299

8 Fraud Detection for Streaming Sketches 308

8.1 Technical Overview 310
8.2 Derivation of the $k/1$-Chain 313
8.3 Concentration of the Time Average of $\log Z_n$ 318
8.4 Summary of Fraud Detection 328

BIBLIOGRAPHY 330

LIST OF FIGURES

FIGURE

1.1 Streaming sketches in the Lévy-Khintchine perspective 10
1.2 The compound Poisson process with characteristic exponent g_{np} 11
1.3 Harmonic decomposition: Another way to approach universality that is "dual" to sampling 15
1.4 Diagrams of `COUNT-DISTINCT` sketches 19
1.5 Relative variance for τ-GRA estimators 21

2.1 Diagram of homomorphic sketching of f-moments 36

3.1 Lévy-Tower with $m = 3$ and $\mathbf{x} = (1, 0, \ldots, 0)$ (source of the frontispiece image) . 50
3.2 Level functions for F_0-sampler and F_1-sampler 66
3.3 Level functions for samplers induced by well-known Lévy processes 67
3.4 Visualization of universal sampling by ParetoSampler 69
3.5 L_0 with Lévy-Tower 81
3.6 g_{np} with Lévy-Tower 83
3.7 The flat stochastic sampling circuit corresponding to the generic G-Sampler (Algorithm 1.4). 92
3.8 Circuit diagram for G-Edge-Sampler (Algorithm 3.4) with $G(a, b) = \log(1+\sqrt{a}+\sqrt{b}) + 2(1 - e^{-(a+b)})$ 92

4.1 PCSA with the finite field random projection trick 98
4.2 Probability distribution of S_p as the cell size p varies 99
4.3 Diagram of the f-moment estimation process 105
4.4 Summary of tractability and universality 113
4.5 Gamma function $\Gamma(x)$ over $(-1, 0) \cup (0, 1)$ 119

5.1 Classic cardinality sketches as covering structures 143
5.2 Pipeline of cardinality sketches 144

5.3 Analogy between cardinality sketches and rulers: Why scale-invariant? 156
5.4 Analogy between cardinality sketches and rulers: Weak scale-invariance 156
5.5 Entropy and normalized Fisher information number for q-LogLog sketches for $\lambda \in [2^{16}, 2^{24}]$ 157
5.6 Analogy between cardinality sketches and rulers: Smoothing 158
5.7 The empirical relative error $(\hat{\lambda}/\lambda)$ distribution (for $\lambda \in [2^{16}, 2^{24}]$) of q-LogLog . . 159
5.8 The cell partition used by q-PCSA and q-LogLog 174
5.9 A classification of sketching algorithms for cardinality estimation 175
5.10 $\dot{H}(t), \dot{I}(t)$ and $\dot{H}(t)/\dot{I}(t)$ 178
5.11 HyperBitBit Experiments with cardinality $\lambda = 400000$ 200
5.12 HyperBitBit Experiments with terminating condition $(L, \text{HammingWeight}(S_0)) = (12, 31)$ 200
5.13 Visualization of Curtain 204
5.14 Visualization of SecondCurtain 207
5.15 Experiments for Martingale estimators: Comparison of error distributions. . . . 231
5.16 Visualization of remaining areas 234

6.1 PCSA, G-PoissonTower, and collapsed M-PoissonTower 256

7.1 Comparison between per-entry error and Euclidean error 279
7.2 Comparison between the set of estimates with and without compressing the relative vector V 283

8.1 Visualization of the coupling between continuous 1-Min and poissonized 1-Min processes 315
8.2 Two ways of simulating a Poisson point process. 316
8.3 Visualization of the backwards jumping times of 1-Min 317
8.4 Backwards observation process, stepping from t_n to t_{n+1} 318

LIST OF TABLES

TABLE

1.1 Sampling words from Shakespeare's plays 22

2.1 Common M-turnstile models 30

3.1 Intuition for scale-invariance 41
3.2 Common one-dimensional Lévy processes. 46
3.3 ParetoSampler for the experiment in Figure 3.4 70
3.4 HyperLogLog and PCSA as Lévy-Tower 79

4.1 Comparison between the sketches and estimators constructed in Chapter 3 and Chapter 4 96
4.2 Relative error of PCSA over finite fields using LogLog-like estimator 97

5.1 Classic cardinality sketches as covering structures 142
5.2 Summary of cardinality sketches 147
5.3 Memory-variance products of mergeable and non-mergeable sketches 203
5.4 Experiments of Martingale sketches 231
5.5 Summary of estimators for PCSA and LogLog 236

6.1 Asymptotic constants of F-PCSA as $m \to \infty$ 263

LIST OF PROGRAMS

PROGRAM

1.1 F_1-sampler 23
1.2 F_0-sampler 23
1.3 $F_{1/2}$-sampler 24
1.4 G-sampler 24

3.1 ParetoSampler 68
3.2 (G, k)-Sampler-WOR 70
3.3 Stable-HyperLogLog 74
3.4 G-Edge-Sampler for $G(a, b) = \log(1 + \sqrt{a} + \sqrt{b}) + 2(1 - e^{-(a+b)})$ 93

5.1 The cardinality sketch identified by a covering structure $\mathcal{C}$ 144

7.1 Morris counter 277
7.2 d-dimensional approximate counter 285

REMARKS ON NOTATIONS

Sets For two sets A and B, I write $A \subset B$ to mean A is a subset of B, proper or not. $\emptyset$ denotes the empty set. $\mathbb{N} = \{0, 1, 2 \ldots\}$ and $\mathbb{R}_+ = [0, \infty)$, inclusive at zero, while $\mathbb{Z}_+ = \{1, 2, 3, \ldots\}$, exclusive at zero. $\mathbb{T}$ denotes the complex unit circle, often identified as an additive group $[0, 2\pi)$. $\mathbb{F}_{p^k}$ denotes the finite field of order p^k.

For $k \in \mathbb{N}$, $[k]$ is short for $\{1, \ldots, k\}$, and $[0] = \emptyset$. $\mathbb{1}[\cdot]$ is the indicator function. For any event/statement E, $\mathbb{1}[E] = 1$ if E is true and 0 otherwise. An operation on an empty set returns the identity of the operation. Thus for example, for real numbers a_k and sets A_k,

$$\sum_{k=1}^{0} a_k = 0, \quad \prod_{k=1}^{0} a_k = 1, \quad \bigcup_{k=1}^{0} A_k = \emptyset, \quad \bigcap_{k=1}^{0} A_k = (\text{universe}).$$

Operations between elements apply through sets. For example, for $x \in \mathbb{R}$ and $A, B \subset \mathbb{R}$,

$$x + A = \{x + z : z \in A\}, \qquad xA = \{xz : z \in A\}$$
$$A + B = \{a + b : a \in A, b \in B\}, \qquad AB = \{ab : a \in A, b \in B\}$$

For $a, b \in \mathbb{R}$, I write $a \pm b$ to indicate the interval $(a - b, a + b)$. Thus for example, an error range written as $t(1 \pm \epsilon)$ denotes the interval $(t - t\epsilon, t + t\epsilon)$. This will usually be used together with the big-O notation. For example, for real sequences $\{g(m)\}_{m \in \mathbb{N}}$ and $\{h(m)\}_{m \in \mathbb{N}}$,

$$g(m) \in t(a, b) + O(h(m)), \quad \text{as } m \to \infty$$

means there exists $c \in \mathbb{R}_+$ and $N \in \mathbb{N}$ such that for any $m \geq N$

$$g(m) \in t(a, b) + (-c|h(m)|, c|h(m)|) = (at - c|h(m)|, bt + c|h(m)|).$$

Probability For two random variables X, Y, I write $X \sim Y$ if they are identically distributed. Typically, a sequence of i.i.d. copies of random variable X will be written as $X^* = (X^{(1)}, X^{(2)}, \ldots)$. I say a random variable is *fresh* or *freshly drawn* if it is independent

with any other randomness in the context. Occasionally I use *anonymous random variables* that are identified just by their distributions. Anonymous random variables are all freshly drawn. Commonly used ones are the following.

- Poisson(λ) denotes a fresh Poisson random variable with rate λ.
- Uniform$(0, 1)$ denotes a fresh uniform random variable over $[0, 1)$.
- Exp(λ) denotes a fresh exponential random variable with rate λ.
- Normal(μ, σ^2) denotes a fresh normal random variable with mean μ and variance σ^2.

Thus for example, the equation

$$\min(\text{Exp}(a), \text{Exp}(b)) \sim \text{Exp}(a + b)$$

means the minimum of two freshly drawn exponential random variables with rates a and b distributes identically with an exponential random variable with rate $a + b$.

$\mathbb{P}(A)$ denotes the probability of the event A. $\mathbb{E}X$ and $\mathbb{V}X$ denote the mean and variance of the random variable X. When X is a complex-valued random variable, $\mathbb{V}X = \mathbb{E}(X - \mathbb{E}X)\overline{(X - \mathbb{E}X)}$.

Vectors and sequences For any dimension d, I use $e_1, \ldots, e_d$ to denote the natural basis of $\mathbb{R}^d$. For any $x \in \mathbb{R}^d$, $|x| = \sqrt{\sum_{j=1}^d x_j^2}$ is the Euclidean length.

For any monoid M, a vector $\mathbf{x} \in M^n$ is expanded as $(\mathbf{x}(1), \ldots, \mathbf{x}(n))$. I write $\text{supp}(\mathbf{x})$ for the set of indices v that $\mathbf{x}(v) \neq 0_M$, aka the support, and $\|\mathbf{x}\|_0 = |\text{supp}(\mathbf{x})|$, aka the support size. For Euclidean spaces, for any $x \in \mathbb{R}^d$, I write $\|\mathbf{x}\|_p$ for $(\sum_{v\in[n]} |\mathbf{x}(v)|^p)^{1/p}$, aka p-norm, and $\|\mathbf{x}\|_\infty$ for $\max_{v\in[n]} |\mathbf{x}(v)|$, aka max-norm.

Statistics For any real-valued quantity x, we write $\widehat{x}$ as an *anonymous estimator* of x, who is either a generic estimator or a specific estimator defined in the context. This notation can be applied to expressions. For example, $\widehat{x + y}$ is an anonymous estimator of $x + y$.

ABSTRACT

A streaming sketch is a data structure that can approximately answer queries of a data stream, whose size is *much* smaller than the stream length or the domain size. This thesis studies a wide range of topics on streaming sketching including approximate counting, cardinality estimation, the tractability of queries, the universality of sketches, non-mergeable sketches, as well as sampling. My research goes in two directions.

Mathematical theory For this direction, the simplicity and generality of the *theory* is the primary goal, and the produced algorithms can be existential or more like mathematical prototypes. Notable progress includes the following.

- I explicitly connect Lévy processes to streaming algorithms, revealing a deep connection between infinite divisibility and scale-invariance. This connection sheds new light on the important tractability problem and also pins down the hash pattern one needs to G-sample over incremental streams with exactly correct sampling probabilities.
- I propose a new universal sketching scheme by decomposing general streaming queries into easy-to-estimate "harmonic queries." The decomposition can be obtained from the Lévy-Khintchine representation theorem or the Fourier transform.
- I propose a formal measurement of the goodness of a cardinality sketch, which sets up a tension between the Fisher information number and Shannon's entropy. I also generalize cardinality sketches to work over any (finite) commutative monoid, using the harmonic decomposition technique together with the semigroup decomposition of commutative monoids.

Practical algorithms For this direction, the simplicity and practicality of the *algorithm* is the primary goal, and the associated analysis is usually complicated and computation-heavy. Small-scale experiments are often included to confirm the analysis.

- I analyze various classes of sketches and estimators for the cardinality query. Many recipes are constructed for practitioners to pick and construct cardinality sketches and estimators in different contexts with different performance emphases.

- I study the *sequential* behaviors of cardinality sketches and construct simple fraud detectors.
- I study two variants of the Morris counter, one tracks a multi-dimensional count vector and the other tracks a count that can be both incremented and decremented.

Fortunately, some of the algorithms I construct turn out to possess simplicity in both theory and practice. Notable examples are the $F_{1/2}$-sampler, stable-HyperLogLog, and the d-dimensional approximate counter.

CHAPTER 1

Introduction

> For practical purposes, algorithm PCSA is quite satisfactory. It consumes only a few operations per element scanned (may be 20 or 30 assembly language instructions), has good accuracy described at length in the previous sections, and may be used to gather statistics on files *on the fly* (therefore eliminating the additional cost of disk accesses). Philippe Flajolet and G. Nigel Martin [83], 1985

1.1 Computation over Streams

In the classic random-access machine (RAM) model, the input of a program lies directly in the memory, whose contents can be accessed in unit time [57]. The RAM model captures important real-world scenarios such as factorizing integers or solving (small) linear programs. However, if one is designing a program that parses the entire disk, the input (the files on the disk) is literally not in the memory and the RAM poorly models this scenario: from the perspective of a CPU, accessing the disk takes an eternity. An algorithm designer may use the *streaming model* [83, 5] in such scenarios. The streaming model assumes a "working memory" that can be arbitrarily accessed and a separate "input stream." For example, in the scenario above, the working memory models the main memory (or the cache blocks) near the CPU and the input stream models a sequential scanning of the disk. We call algorithms and data structures in the streaming model *streaming algorithms* and *streaming sketches*, respectively, which are the central subjects I will study in this thesis.

The incentives to study streaming algorithms and sketches are significant. The streaming model, which was initially used to describe the "disk parsing" scenario in the 1980s [83], became more and more *useful* in the 21st century. The world is digitally connected now, and we are no longer talking about a single CPU scanning a local disk in the same room. We are talking about thousands of computation units processing data sent from billions of devices

around the world [95]. In some cases, the incoming data, such as the metadata of the internet traffic, is not stored anywhere [116]. The RAM model becomes completely invalid in such cases since the input will be literally *gone* and a re-access is impossible. On the other hand, the streaming model, which only assumes a small working memory, perfectly models such scenarios where streams are processed "on the fly." Apart from computer systems, streaming computations naturally arise in other areas like processing real-time sensor data [124] and analyzing DNA sequences [171].

Despite staying very close to production, streaming algorithms are not only an empirical science. In fact, many of the leaps forward in this field have come from serious theoretical investigations [83, 5, 34, 82, 61, 23]. It turns out that there are actual underlying theories determining the capacity and behaviors of such data structures. The theory of streaming interacts closely with other fields like probability theory, statistics, information theory, communication complexity, and harmonic analysis.

In this thesis, I will continue the theoretical investigation of the subject of streaming, developing new theory for understanding the streaming computation regime. At the same time I will also frequently ground myself to practice, designing and analyzing more implementable algorithms with empirical experiments.

1.2 Scope and Background

Let me now locate the academic scope of this thesis. The subjects of this thesis are *algorithms* and *data structures* over streams. There are multiple sub-fields that assume the data is presented in an on-line fashion, though with distinct emphases.

- Dynamic algorithms [76] emphasize that the *update time* to maintain its data structure should be significantly smaller than that of recomputing it from scratch.
- Online algorithms [111] emphasize that decisions have to be made as the stream flows in and the *utility* needs to be competitive against the decisions that are made with the knowledge of the whole input stream in the beginning.
- Streaming algorithms [128, 83] emphasize the *space* of the working memory being very small relative to the stream length and the domain size, and consider update time secondarily.

In this thesis, I will focus on *streaming algorithms*. Streaming algorithms are very diverse, intersecting with many specific areas such as data embedding/dimension reduction [106],

compressed sensing [70], randomized linear algebra [165], large-scale graph/geometric problems [20, 126], etc. This thesis will focus on a rather classic area—*estimation over streams*. This line of research is established by the landmark work of Flajolet and Martin [83] in 1983 on counting the number of distinct elements in a stream and the work of Alon, Matias, and Szegedy [5] in 1996 on the analysis of the space complexity for frequency moment estimation. Unlike other areas of streaming, streaming estimation focuses on the task of approximately answering *statistical queries* about the stream whose answers are mostly one-dimensional numbers. This distinguishes this area from, e.g., compressed sensing where the goal is to reconstruct an approximation of the whole stream.

The scope of this thesis can be further specified in terms of the models for streaming estimation. First, I will primarily consider *distributed streams* and *one-pass* algorithms. Second, I will assume *random oracles* (full independence hash functions are provided externally) and pay attention to the *statistical efficiency* of algorithms. There is another line of research [5, 96, 110, 23, 19] that develops techniques to replace random oracles with explicitly constructed hash functions, where one only needs to assume the random *seed* is uniformly random. Depending on whether random oracles are assumed, algorithms can be divided into two categories.

Standard model Hash functions are explicitly constructed and stored.

Random oracle model Hash functions are provided externally and are uniformly random.

The two models are highly related with some distinct emphases. A more detailed discussion on the choices of modeling is in Chapter 2. Most of the algorithms in this thesis are in the random oracle model and when I discuss prior standard model algorithms, the hash functions will be replaced with random oracles by default for a level playground.

Next, I am going to introduce a few topics about streaming algorithms, together with overviews of the literature and my research progress on them.

1.3 Tractability of Queries

1.3.1 Motivation and Definition

The "computation power" of a streaming algorithm is very limited because the size of the working memory is *much* smaller than the length of the input stream and the domain size.

Therefore, streaming algorithms are not expected to compute hard problems, but only expected to answer some queries about the stream. Exact answers of the queries are often impossible to obtain in the streaming model and we are satisfied if an approximation of the correct answer can be returned.

Definition 1.1 (approximation)**.** We say $\hat{y}$ is a $(1\pm\epsilon)$-*approximation* of $y \in \mathbb{R}$ if $\hat{y} \in (1\pm\epsilon)y$ with probability 2/3.[1] We say $\hat{y}$ is a *decent approximation* of y, if $\hat{y}$ $(1 \pm 1/3)$-approximates y.

Unfortunately, for many important queries, even a decent approximation is hopeless. To demonstrate this, let us consider a concrete streaming model.

Definition 1.2 (insertion-only model)**.** Fix a domain $[n]$.[2] The *insertion-only model* assumes the input stream is a sequence of elements $(a_1, a_2, \ldots, a_N)$ where each a_i is an element in the domain $[n]$.

The insertion-only model is the earliest streaming model which captures scenarios like parsing network logs or database records. In this model the common query `MAX-FREQUENCY` (Query 1) requires $\Omega(n)$ space to approximate decently. This is proved by Alon, Matias, and Szegedy [5].

Query 1 (`MAX-FREQUENCY`)**.** Return the frequency of the most frequent element in the stream.

In other words, `MAX-FREQUENCY` is impossible to approximate without having a working memory essentially as large as the domain size or the input stream. This seminal work by Alon et al. [5] starts a long line of research that tries to answer the following fundamental question.

With limited working memory, what queries are approximable over a stream?

Definition 1.3 (tractability [23])**.** With any streaming model specified, a query is said to be *tractable* (in the specified model) if it can be $(1 \pm \epsilon)$-approximated with $O_\epsilon(\operatorname{poly}\log(n))$ space.[3]

[1] It is standard to reduce the failure probability to δ by maintaining $O(\log \delta^{-1})$ i.i.d. copies of $\hat{y}$ and taking the median [96, 19].

[2] $[n]$ denotes the set $\{1, 2, \ldots, n\}$ where n is a natural number.

[3] Some authors (e.g., [19]) define tractable queries to take $O_\epsilon(n^{o(1)})$ space.

Tractable queries can be approximately answered over streams using a feasible amount of working memory, even for domains as large as the index space of IPv6, which has size $n = 2^{128}$. Knowledge of tractability is instructive in real-world system design. For example, suppose a system designer needs to implement the following common queries.

Query 2 (`COUNT-ODD`)**.** Return the number of elements that appear an odd number of times in the stream.

Query 3 (`COUNT-EVEN`)**.** Return the number of elements that appear a positive and even number of times in the stream.

Query 4 (`COUNT-SINGLETON`)**.** Return the number of elements that appear exactly once in the stream.

Query 5 (`COUNT-DISTINCT`)**.** Return the number of elements that appear at least once in the stream.

The system designer may learn from the streaming literature that `COUNT-ODD` and `COUNT-DISTINCT` are tractable while `COUNT-EVEN` and `COUNT-SINGLETON` are not.[4] With this knowledge, he can quickly implement algorithms for answering `COUNT-ODD` and `COUNT-DISTINCT` based on their constructive upper bounds and he will be *aware* that `COUNT-EVEN` and `COUNT-SINGLETON` are in general impossible to track with a poly-log sized working memory. Thus, he has to either relax the error guarantee or make additional assumptions in order to answer `COUNT-EVEN` and `COUNT-SINGLETON`. For example, he may relax the multiplicative error guarantee to an *additive* error guarantee—a simple, sampling-based, poly-log space sketch can approximate `COUNT-EVEN` and `COUNT-SINGLETON` with error proportional to the total number of distinct elements in the stream.

1.3.2 A Brief Survey on Tractability

`COUNT-DISTINCT` In 1983,[5] Flajolet and Martin [83] proposed the first streaming sketch, Probabilistic Counting with Stochastic Averaging (PCSA), which estimates `COUNT-DISTINCT` using $O(\epsilon^{-2} \log n)$ bits. In 2003, Durand and Flajolet [73] invented the LogLog sketch, improving the space to $O(\epsilon^{-2} \log \log n)$ bits. With more deliberate memory compression, the

[4] `COUNT-ODD` and `COUNT-DISTINCT` can be decently approximated with $O(\log(n))$ space via PCSA based sketches [83] (see Chapters 5 and 6). `COUNT-EVEN` and `COUNT-SINGLETON` require $\Omega(n)$ space, reducing from the hardness of INDEX_n. See [39, Theorem 12].

[5] When surveying the history, I will anchor a work in the year of its first appearance. The citation entry on the other hand often points to its journal version that appeared a few years later.

space needed to get a $(1 \pm \epsilon)$-estimate of COUNT-DISTINCT was reduced to $O(\epsilon^{-2} + \log n)$ bits by Kane, Nelson, and Woodruff in 2010 [110] (assuming the standard model), which matches the lower bound $\Omega(\epsilon^{-2})$ proved by Woodruff [164] and the lower bound $\Omega(\log n)$ proved by Alon et al. [5]. With random oracles, it was observed by Durand in her PhD thesis [72] (2004) that the LogLog sketch only has entropy $O(\epsilon^{-2})$. Therefore in the random oracle model the space can be reduced to $O(\epsilon^{-2})$ with a proper encoding, plus the overhead to store the estimate. Note that to store an $(1 \pm \epsilon)$-approximation of a number less than n, it only requires $O(\log \log_{1+\epsilon} n) = O(\log \log n + \log \epsilon^{-1})$ bits.

FREQUENCY-MOMENT Alon et al. [5] defined the important FREQUENCY-MOMENT query.

Query 6 (FREQUENCY-MOMENT)**.** Consider an insertion-only model with domain $[n]$. Let $\mathbf{x}(v)$ be the number of times element v is inserted in the stream. The pth FREQUENCY-MOMENT is defined as $\sum_{v=1}^{n} \mathbf{x}(v)^p$, which is also denoted as F_p. F_p is well-defined for all $p \in \mathbb{R}_+$ where $F_0 = \sum_{v=1}^{n} \mathbb{1}[\mathbf{x}(v) > 0]$, aka COUNT-DISTINCT.

In 1996, Alon et al. [5] systematically studied the tractability of F_p with $p \in \mathbb{N}$ in the standard model. In fact, this work [5] was the first to work in the standard model. The F_1 case is trivial since it is just a linear sum. For F_0, they modified Flajolet and Martin's PCSA algorithm so that the random oracle can be replaced by a linear hash function. For F_2, they invented the celebrated AMS[6] sketch which can be considered a 4-wise independent version of Johnson-Lindenstrauss projection [106]. The AMS sketch $(1 \pm \epsilon)$-approximates F_2 using $O(\epsilon^{-2} \log n)$ bits of space.[7] In general they show that F_p needs $\Omega(n^{1-5/p})$ bits of space to approximate decently. This implies F_p are intractable for $p = 6, 7, \ldots$ and so is MAX-FREQUENCY.

In 2000, Indyk [96] showed that by replacing the Rademacher random variables with p-stable random variables, the AMS sketch can be modified to estimate F_p-moments using $O(\epsilon^{-2} \log n)$ bits of space.[8] Of course, p-stable random variables only exist for $p \in (0, 2]$ and thus Indyk proved that F_p is tractable for all *fractional* $p \in (0, 2]$.

In 2002, Bar-Yossef, Jayram, Kumar, and Sivakumar [8] proved that estimating the F_p-moment requires $\Omega(n^{1-2/p})$ bits, which is $\Omega(\text{poly}(n))$ space for any real $p > 2$. Their work completes the tractability landscape for FREQUENCY-MOMENT—

[6] AMS for Alon-Matias-Szegedy.

[7] For this bound and later on, the stream length N is assumed to be $O(\text{poly}(n))$.

[8] Here a random oracle is assumed to provide uniformly random hash values. Without assuming the random oracle, Indyk [96] used Nisan's pseudorandom generator [134], which incurs one more log factor in the space complexity. Later Kane, Nelson, and Woodruff [109] and Kane, Nelson, Porat, and Woodruff [108] improved the derandomization and achieved $O(\epsilon^{-2} \log n)$ bits in the standard model.

F_p is tractable if and only if $p \in [0, 2]$.

Since this thesis focuses on the polylog-space regime, I will not cover work in the polynomial regime, e.g., those on the bounds for large frequency moments ($F_p, p > 2$) [5, 22, 98, 33, 142, 87, 58, 24].

`FUNCTION-MOMENT` In the conclusion of the seminal work [5], Alon et al. asked about the tractability of a further generalized version of `FREQUENCY-MOMENT`—`FUNCTION-MOMENT`.

Query 7 (`FUNCTION-MOMENT`)**.** Let $f\colon \mathbb{N} \to \mathbb{R}_+$ be a function with $f(0) = 0$. With the setup in Query 6, the `FUNCTION-MOMENT` parametrized by f is defined as $f(\mathbf{x}) = \sum_{j=1}^{n} f(\mathbf{x}(i))$, which is referred to as the *f-moment*.

Remark 1.1. `FREQUENCY-MOMENT` generalizes `COUNT-DISTINCT` which is simply F_0. `FUNCTION-MOMENT` generalizes `FREQUENCY-MOMENT` where F_p is simply the f-moment with $f(x) = |x|^p$ and F_0 is the f-moment with $f(x) = \mathbb{1}[x > 0]$.

Unlike for the `FREQUENCY-MOMENT` where the fine-grained space complexity is pinned down for each F_p, the line of research [23, 18, 19] on characterizing the tractability of f-moments usually pays less attention to the log factors, only to whether the space is poly-log in n.

In 2010, Braverman and Ostrovsky [23] managed to characterize all tractable functions that are *increasing*. Braverman and Chestnut [18] then in 2015 characterized all tractable functions that are *decreasing* on $[1, \infty)$.[9] In 2016, Braverman, Chestnut, Woodruff, and Yang [19] further extended the characterization to all functions, except for a class of "nearly periodic" functions. For these works, the lower bounds are reduced from carefully designed communication complexity instances and the upper bounds come from *sampling*. The most powerful sampling technique for estimating f-moments is Indyk-Woodruff *layering* [99, 25], which maintains heavy hitters on geometrically spaced subsampled levels. The generic problem of characterizing the tractability of f-moments is considered to be a central topic in streaming [5, 23], which is usually just called *the tractability problem*. It is, unfortunately, not fully understood yet.

One evidence that the layering method does not cover all tractable functions is given in [19], where Braverman et al. defined $g_{np}\colon \mathbb{N} \to \mathbb{R}_+$.[10]

$$g_{np}(0) = 0 \quad \text{and} \quad g_{np}(x) = 2^{-\tau(|x|)}, \quad \text{where} \quad \tau(x) = \max\{j \in \mathbb{N} : 2^j | x\}. \tag{1.1}$$

[9] Recall that $f(0) = 0$.

[10] In other words, $\tau(x)$ is the least significant bit position in the binary representation of $x > 0$.

For functions like g_{np}, the entries with large g_{np}-values are not necessarily heavy hitters, so sampling heavy hitters is generally not helpful. However, Braverman et al. presented an *ad hoc* sketch for $(1 \pm \epsilon)$-approximating g_{np}-moments with $O(\epsilon^{-8} \log^{15} n)$ space.

`SYMMETRIC-NORM` In 2017, Błasiok, Braverman, Chestnut, Krauthgamer, and Yang [14] further extend the layering method to estimate *symmetric norms*[11] of a dynamic vector $\mathbf{x}$. Such heavy hitter based methods are proved to be optimal for all symmetric norms up to polylog factors. Nevertheless, this result does not seem to improve our understanding of the tractability of f-moments, as the unclassified ones from [23] do not correspond to norms.

Query 8 (`SYMMETRIC-NORM`)**.** Let l be a symmetric norm over $\mathbb{R}^n$. With the setup in Query 6, the `SYMMETRIC-NORM` parametrized by l is defined as $l(\mathbf{x})$.

1.3.3 My Contribution

My contribution to the tractability problem is a new generic construction of streaming sketches that are based on *Lévy processes*, by which the *Lévy-Khintchine representation theorem* can imply a new kind of sufficient condition for tractability. Informally speaking, the result can be summarized as follows.

One may aggregate elements projected using Lévy processes with characteristic exponent f to track the f-moment of a stream.

It turns out that it suffices to sample Lévy processes at geometrically spaced times for the purpose of f-moment estimation. The induced sketch is called Lévy-Tower.

Theorem 1.1 (Lévy-Tower, Pettie and **Wang** [138], Section 3.4.2)**.** *Let $f\colon \mathbb{R}^d \to \mathbb{C}$ be any Lévy-Khintchine representable function*

$$f(z) = \frac{1}{2}\langle z, Az\rangle - i\langle \gamma, z\rangle + \int_{\mathbb{R}^d} \left(1 + i\langle z, s\rangle \mathbb{1}\left[|s| < 1\right] - e^{i\langle z,s\rangle}\right) \nu(ds),$$

where A is a covariance matrix, $\gamma \in \mathbb{R}^d$, and ν is a positive measure such that $\int_{\mathbb{R}^d} \min(|x|^2, 1)\, \nu(dx) < \infty$. There exists a sketch of $O(\epsilon^{-2} \log n)$ words such that for any input stream $\mathbf{x} \in (\mathbb{R}^d)^n$ with $|f(\mathbf{x})| \in [1, \mathrm{poly}(n)]$, it returns a $(1 \pm \epsilon)$-estimate of $f(\mathbf{x})$.[12]

[11] A norm on $\mathbb{R}^d$ is symmetric if it is "invariant under sign-flips and coordinate-permutations" ([14]).

[12] Here $f(\mathbf{x})$ can in general be complex valued. We say $\hat{y}$ is a $(1 \pm \epsilon)$-estimate of $f(\mathbf{x})$ if $\mathbb{P}(|\hat{y} - f(\mathbf{x})| < \epsilon |f(\mathbf{x})|) > 2/3$ where $|\cdot|$ is the modulus. See REMARKS ON NOTATIONS in the front of the thesis for choices of mathematical symbols and conventions.

Remark 1.2. Here f is a function over $\mathbb{R}^d$, meaning each element $v \in [n]$ can have a value $\mathbf{x}(v) \in \mathbb{R}^d$ to be aggregated over streams. This is a model that is more general than the insertion-only model in which $\mathbf{x}(v)$ is a natural number. Functions over $\mathbb{R}^d$ can be used to describe *relational queries*, i.e., queries about elements over multiple streams. This is the primary reason to study such multi-variate function moments. I will talk more about modeling streams in Section 2.1.

Remark 1.3. This theorem describes a new *sufficient* condition for a function moment to be tractable over streams. For integer streams, there are known tractable functions that are not Lévy-Khintchine representable. There are workarounds though. See Section 3.4.

When the function f is the characteristic exponent of a *stable* processes, it is possible to *emulate* Indyk's sketch to estimate the f-moment.

Theorem 1.2 (Lévy-Stable, Pettie and **Wang** [138], Section 3.4.3)**.** *Let f be the characteristic exponent of any d-dimensional p-stable process X. Let the Lévy-Stable sketch be parameterized by X. Given any input vector $\mathbf{x} \in (\mathbb{R}^d)^n$ and $\mathbf{x}' \in \mathbb{R}^n$ such that $f(\mathbf{x}) = \sum_{v\in[n]} |\mathbf{x}'(v)|^p$, Lévy-Stable with input $\mathbf{x}$ and Indyk's p-stable sketch with input $\mathbf{x}'$ distribute identically.*

Theorem 1.2 serves as an illuminating example showing how the Lévy-Khintchine theorem helps to understand streaming sketching. Previously, only two classes of stable moments are considered: one dimensional F_α-moments by Indyk [96], and multidimensional $F_{p,q}$-moments by Ganguly et al. [88], which are sketched by *algorithmic tricks* of combining stable random variables. The Lévy-Stable sketch extends such tricks to *all stable processes* in a systematic way. For example, we can now estimate, using any estimator of Indyk's sketch [96, 118], the f-moment of an $\mathbb{R}^3$-turnstile stream with

$$f(x) = \int_{\mathbb{S}_2} \frac{|\langle \xi, x\rangle|}{|\xi_1|^2 + |\xi_2| + |\xi_3|^{1/2}}\, d\xi,$$

where $\mathbb{S}_2$ is the unit $\mathbb{R}^3$-sphere.

At a high level, if one considers Indyk's p-stable sketch [96] as a generalization of the AMS sketch [5], where the sub-Gaussian projection is replaced with the p-stable projection, the new Lévy-Tower and Lévy-Stable sketches are a further generalization which replaces Indyk's p-stable projection with a generic projection by sampling Lévy processes at different times. This new theory induces a uniform construction for a wide range of classic sketches. See Fig. 1.1 for a summary. See Section 3.7 for a detailed discussion. For the sketches in the figure, the AMS sketch [5], Indyk's sketch [96], and Flajolet and Martin's PCSA sketch [83] have been discussed before. $F_{p,q}$ is considered by Ganguly, Bansal, and Dube [88]. The

sketch for `COUNT-DISTINCT` with both insertions and deletions (this version of the query is denoted as L_0) is given by Kane et al. [110].

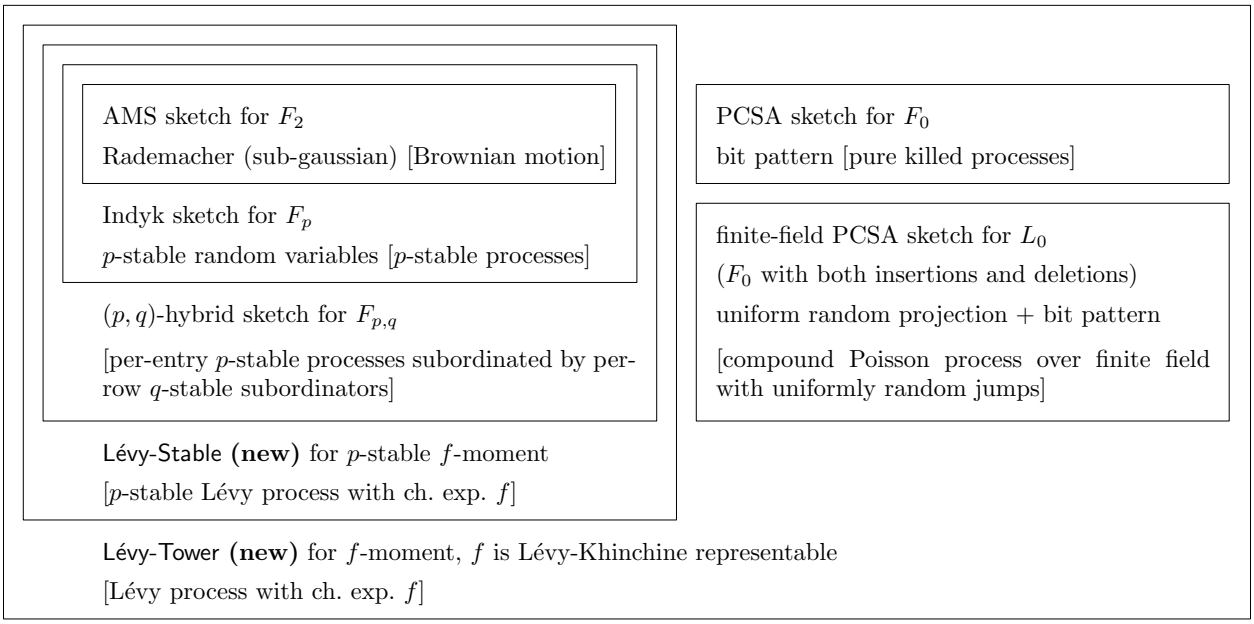
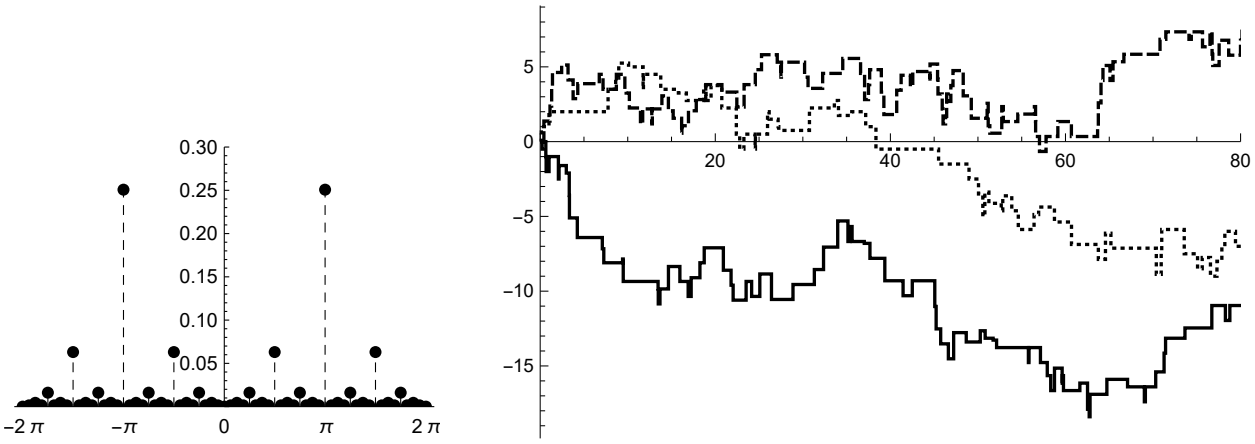


Figure 1.1: Streaming sketches in the Lévy-Khintchine perspective

The main technical contribution of Theorem 1.1 is the implication of the tractability of a large class of queries that are unknown to be tractable before. Specifically, it implies the tractability of the set of nearly periodic functions in [19] that are Lévy-Khintchine representable, which include the example g_{np}. This approach also, in some sense, *explains* the tractability of g_{np} as it, indeed, corresponds to a Lévy process, in particular, a compound Poisson process. See Fig. 1.2. For another concrete example, the *golden harmonic moment* ($f(x) = 1 - \cos(\frac{1+\sqrt{5}}{2} \cdot 2\pi x)$, see Lemma 4.3) is Lévy-Khintchine representable but can not be estimated by previous methods.

Notably, the idea of constructing streaming sketches from Lévy processes is highly extendable. Informally speaking, the set of Lévy processes over an LCA (locally compact and abelian) group G can be used as streaming sketches over streams with elements' states lying in the *Pontryagin dual group* Γ. See Section 3.10.



(a) jump distribution (b) three sample runs

Figure 1.2: The compound Poisson process with characteristic exponent g_{np}

1.4 Universality of Sketches

1.4.1 Motivation and Definition

Would not it be nice if there were a streaming sketch that were able to answer a wide range of queries? Perhaps surprisingly, Braverman and Ostrovsky in [23] observed that their sketch for tracking increasing function moments is actually *universal* for the target class of functions.

Definition 1.4 (universality)**.** Fix any error parameter $\epsilon > 0$. We say a streaming sketch is $(\epsilon, \mathcal{Q})$-*universal* where $\mathcal{Q}$ is a class of queries, if after reading the stream, it can return a $(1 \pm \epsilon)$-approximation of any query in $\mathcal{Q}$.

Remark 1.4. Note that the $(1\pm\epsilon)$-approximation is only required for each query in $\mathcal{Q}$, one at a time. Specifically, it does *not* require all queries to be $(1\pm\epsilon)$-approximated simultaneously, which is a notable difference compared to the convention in other fields like data embedding.

The larger the class $\mathcal{Q}$ is, the more universal the sketch is considered to be. A naive way to build a $\mathcal{Q}$-universal sketch is to compile together a stack of q-sketches for all queries $q \in \mathcal{Q}$. This is wasteful when $\mathcal{Q}$ becomes large. Efficient universal sketches are thus desirable in practice as they provide a wide range of information with feasible space [26]. The ultimate question is

Is there a "truly universal" sketch that can approximate ***all tractable queries****?*

Technically, only *statistical queries* are considered when discussing universal sketches. A statistical query is insensitive to the identities of the elements and is a function of the collection of state values (in the insertion-only model, the state value of an element is the number of times it appears in the stream). For example, a non-statistical query like whether element r appeared in the stream can be surely answered by a single bit which is set to one upon r is inserted. However, there are no poly-log space universal sketches for such non-statistical queries since if all the memberships can be checked by one single sketch, the sketch requires $\Omega(n)$ space.[13]

1.4.2 A Brief Survey on Universality

I will focus on the class of `FUNCTION-MOMENT` queries. In this part I am going to assume the streams have both insertions and deletions, aka the *turnstile* model. The turnstile model is more general than the insertion-only model and thus algorithms over turnstile streams can be applied to insertion-only streams.

Remark 1.5. By convention, I will use F_p to denote frequency moments in the insertion-only model and L_p to denote frequency moments in the turnstile model. Note that the elements' value $\mathbf{x}(v)$ can be negative in the turnstile model and L_p is defined as $\sum_{v=1}^{n} |\mathbf{x}(v)|^p$.[14]

The universality question is closely related to the tractability question discussed earlier, and asks for powerful sketches that can track as many tractable f-moments (Query 6) as possible. The existing universal sketches for f-moments all use the idea of *sampling* [23, 18, 19]. Different sampling techniques are developed to sample a set of indices $\{r_1, \ldots, r_k\} \subset [n]$ and estimate their frequencies $\widehat{\mathbf{x}(r_j)}$s and weights w_js. Based on the samples, the estimator of the f-moment is

$$\widehat{\sum_{r\in[n]} f(\mathbf{x}(r))} \coloneqq \sum_{j=1}^{k} w_j f(\widehat{\mathbf{x}(r_j)}) \qquad \text{[sampling-based estimator]} \quad (1.2)$$

This is a universal method as the samples are maintained without the knowledge of the target function f. By plugging in different fs in the estimator one gets estimates of different f-moments of the stream. Let me go over two main sampling techniques in the literature. Note that these are also the techniques used in [23, 18, 19] to prove sufficient conditions for

[13] There are only n different membership queries and thus a universal sketch for these can be modified to answer them *simultaneously*, with one log factor overhead. This is because one may reduce the failure rate of each single estimate to δ with a $\log \delta^{-1}$ factor overhead, by taking the median of i.i.d. estimates [96, 19].

[14] In this thesis, $L_p = \sum_{v=1}^{n} |\mathbf{x}(v)|^p$ refers to the pth frequency moment of a turnstile stream $\mathbf{x}$. For the p-norm, I will use $\|\mathbf{x}\|_p = (\sum_{v=1}^{n} |\mathbf{x}(v)|^p)^{1/p}$.

tractability.

L_0-sampling. A simple universal sketch is to sample m elements $\mathbf{x}(v_1), \ldots, \mathbf{x}(v_m)$ uniformly from $\mathrm{supp}(\mathbf{x})$ and then estimate the f-moment by the empirical mean $V_f = \frac{\|\mathbf{x}\|_0}{m} \sum_{j\in[m]} f(\mathbf{x}(v_j))$. The sampling can be done with L_0-samplers (see [107], $O(\log^2 n)$ bits on average to get one sample) and the support size $\|\mathbf{x}\|_0$ can be estimated by L_0-sketches [59, 110]. This scheme is natural but the estimates are poor if f varies a lot. Specifically, assuming $\|\mathbf{x}\|_\infty \leq M$, for functions f such that $f(x) > 0$ when $x \neq 0$, the number m of samples needs to be $\Omega(\epsilon^{-2} \max_{j\in[M]} f(j)/\min_{j\in[M]} f(j))$ for V_f to be a $(1 \pm \epsilon)$-approximation of the f-moment; see, Chestnut [39]. For functions decreasing on $[1, M]$, it is proved by Braverman et al. in [18] that all tractable moments can be estimated using this scheme. However, functions like $f(x) = |x|^p$ for $p \in (0, 2]$ will take $\Omega(M^p)$ number of L_0-samples to approximate.

L_2-heavy hitters. A more powerful sketch, used in [23, 25, 19], is to use variations on Indyk and Woodruff's [98] method of collecting L_2-heavy hitters using CountSketch [34] at different subsampled levels, and estimate f-moments based on the f-values of the "recursive" heavy hitters. This technique is powerful enough to estimate many f-moments universally. Nevertheless, as we have discussed in the tractability part, it fails to track the class of *nearly periodic* functions [19].

1.4.3 My Contribution

In Chapter 4, I will present a new universal sketching scheme that is almost "dual" to the sampling-based approach. In contrast to the existing sampling-based methods, my approach does not use *any* explicit samples but relies on the *harmonic structure* of the target function f. The idea is to estimate all the γ-moments of the stream where γ is a homomorphism to the unit complex circle $\mathbb{T}$, and then assemble them into an estimator of the f-moment. The harmonic estimator of the f-moment is

$$\widehat{\sum_{r\in[n]} f(\mathbf{x}(r))} \coloneqq \int \widehat{\sum_{r\in[n]} \gamma(\mathbf{x}(r))}\, \mu(d\gamma), \qquad \text{[harmonic estimator]} \quad (1.3)$$

where μ weights the decomposition with $f = \int \gamma\, \mu(d\gamma)$. Because γ is *homomorphic*, the estimates of γ-moments $\widehat{\sum_{r\in[n]} \gamma(\mathbf{x}(r))}$ can be obtained by aggregating the streams at different sub-sampling levels. The harmonic approach is thus a new universal sketching scheme as the γ-moments can be estimated without the knowledge of the target function f. After

knowing f, one simply decomposes f into harmonic components and plugs in its estimated harmonics to the estimator in Eq. (1.3) to get an estimate of the f-moment. It turns out that the estimate is decently accurate when μ is a *positive measure*.[15] See Fig. 1.3.

Theorem 1.3 (informal, **Wang** [158], Chapter 4)**.** *There exists an $O(\epsilon^{-2}\log^2(n\epsilon^{-1}))$-bit space sketch (SymmetricPoissonTower) such that for any $f(x) = cx^2 + \int_0^\infty (1-\cos(\gamma x))\,\nu(d\gamma)$ with $c \in \mathbb{R}_+$, ν being a positive measure, if the f-moment is within $[1/\operatorname{poly}(n), \operatorname{poly}(n)]$, then a $(1\pm\epsilon)$-approximation of the f-moment can be returned with probability $2/3$.*

The new harmonic approach is very appealing both practically and theoretically. For practical purposes, the harmonic approach is more space efficient as it does not need the data structures for obtaining informative samples. For theoretical purposes, the harmonic approach can track one-dimensional, symmetric, Lévy-Khintchine representable functions, naturally handling functions like g_{np} and the golden harmonic function ($f(x) = 1-\cos(\frac{1+\sqrt{5}}{2}\cdot 2\pi x)$, see Lemma 4.3). A provably tractable f-moment that is provably inestimable by the harmonic approach has not been found yet. Therefore, this approach is a new candidate for the ultimate truly universal sketch.

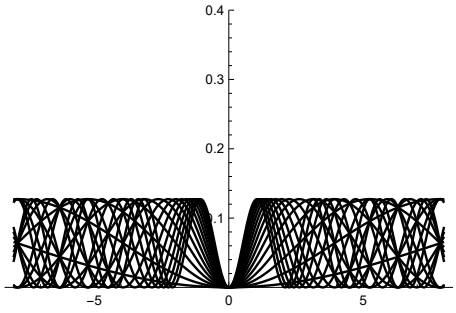
Similar to the Lévy-Tower, this idea of harmonic decomposition of queries can be extended to handle streams over general LCA groups. See Chapter 6.

1.5 `COUNT-DISTINCT` Sketches

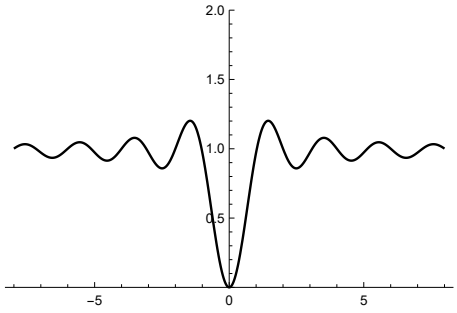
It is natural to pursue optimality for any algorithm designer. Note that in the streaming computation regime, the primary performance measure is the *space complexity* rather than the time complexity.[16] If a query Q is practically in demand, we are motivated to find space-optimal (up to constants) algorithms that answer it. However, if a query is *omnipresent* in every device, in every router, and in every database, such as `COUNT-DISTINCT`, we are motivated to search for the *best* algorithm that is optimal even in the constants. Indeed, even after the optimal-up-to-constants algorithm is found by Kane et al. [110] and Błasiok

[15] Note that here the presentation of this decomposition is considerably simplified. For streaming f-moment estimation, one needs to guarantee $f(0) = 0$ so that the value of the f-moment will not depend on the size of the universe. As γ is a homomorphism to $\mathbb{T}$, we have $\gamma(0) = 1$ and γ-moment should be defined as $\sum_{r\in[n]}(1-\gamma(\mathbf{x}(r)))$ instead. Accordingly, the decomposition is actually $f = \int(1-\gamma)\,\mu(d\gamma)$.

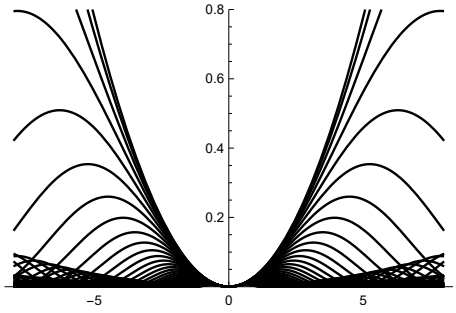
[16] The rationale is that the primary purpose of using streaming algorithms is to run the program near the CPU. The smaller space it requires, the closer to CPU it can be put. For example, if it can be fit in a local computer, then one does not need to spend time to fetch remote data; if it can be fit in the main memory, then one does not need to fetch/evict pages from the disk; if it can be fit in a cache block near the CPU, then one can have 100% cache hit rate.



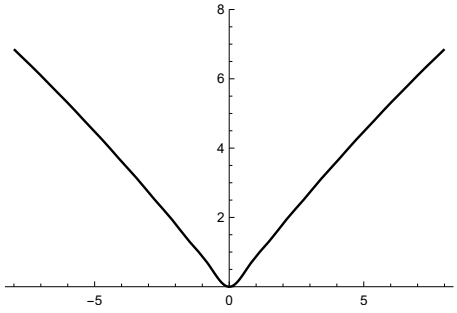
(a) Harmonic decomposition of L_0



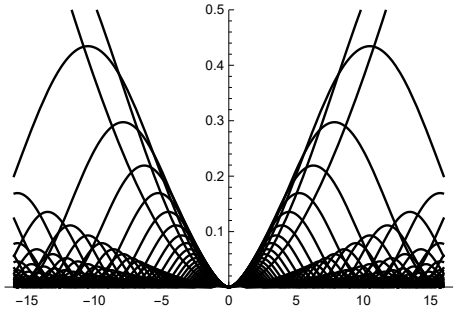
(b) Sum back $f(z) = \mathbb{1}[z \neq 0]$, $z \in \mathbb{Z}$



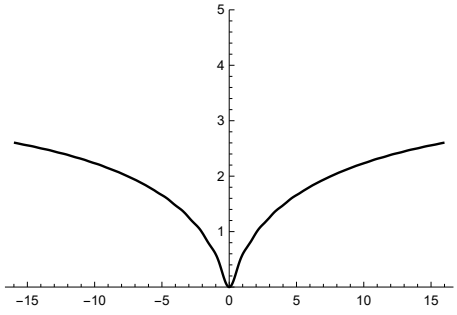
(c) Harmonic decomposition of L_1



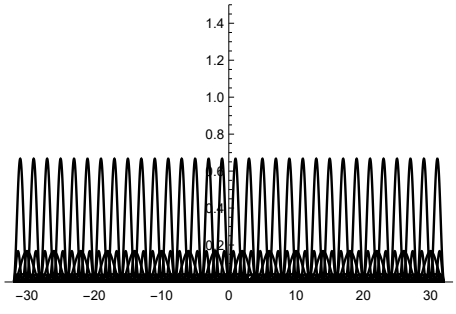
(d) Sum back to $f(z) = |z|$, $z \in \mathbb{Z}$



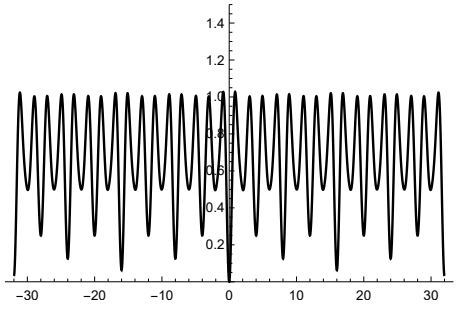
(e) Harmonic decomposition of logarithm



(f) Sum back to $f(z) = \log(1 + |z|)$, $z \in \mathbb{Z}$



(g) Harmonic decomposition of g_{np}



(h) Sum back to $g_{np}(z) = 2^{-\tau(|z|)}$, $z \in \mathbb{Z}$

Figure 1.3: Harmonic decomposition: Another way to approach universality that is "dual" to sampling. Estimates of harmonic components are combined with different weights to form estimates for different f-moments.

et al. [13], many new efforts are made and new techniques are developed to further improve the performance of `COUNT-DISTINCT` sketches [95, 117, 112, 78, 100].

In this section I am going to focus on the important `COUNT-DISTINCT` query in the insertion-only model. This problem is also known as *cardinality estimation* or F_0 *estimation*.

1.5.1 The Fisher-Shannon Number

Well, is there a limit for the performance of a cardinality sketch? To answer this question one needs to formally defining the "goodness" of a `COUNT-DISTINCT` algorithm. Note that such notions of absolute optimality are not naturally available for all queries. This is similar to the point estimation problem in statistics, where it is subtle to claim an estimator is *absolutely optimal* as there can often be another estimator that has better accuracy for *some* of the input. However, I will demonstrate that there is a natural definition of absolute optimality of the `COUNT-DISTINCT` sketch, which connects two important information measures—the Fisher information number (I) and Shannon's entropy (H).

Definition 1.5 (Fisher-Shannon number, informal)**.** After reading a stream $\mathbf{x}$ with $\|\mathbf{x}\|_0 = \lambda$, let the final state of a mergeable `COUNT-DISTINCT` sketch $\mathcal{C}$ be (a random variable) S_λ. The Fisher-Shannon number of $\mathcal{C}$ is defined as the ratio of the "average" Shannon entropy and the "average" relative Fisher information number—

$$\mathsf{Fish}(\mathcal{C}) = \lim_{T\to\infty} \frac{\int_{-T}^{T} H(S_{2^r})\,dr}{\int_{-T}^{T} 2^{2r} I(S_{2^r})\,dr}.$$

The Fisher-Shannon number $\mathsf{Fish}(\mathcal{C})$ characterizes the *memory-variance product* of an optimally implemented $\mathcal{C}$, where

- memory is compressed down to its entropy;
- a statistically efficient estimator is used;
- unbounded computing power and random oracles are assumed.

The Fisher-Shannon number provides a fair way to compare sketches that does not rely on their nominal representations. For example, we can now compare the important PCSA [83] and HyperLogLog [82] sketches.

Theorem 1.4 (Pettie and **Wang** [137], Section 5.3)**.** *The following statements are true.*

1. *PCSA has Fisher-Shannon number* $H_0/I_0 \approx 1.98016$ *where*

$$H_0 = \frac{1}{\log 2} + \sum_{k=1}^{\infty} \frac{1}{k} \log_2(1 + 1/k), \quad I_0 = \frac{\pi^2}{6}.$$

2. *HyperLogLog has Fisher-Shannon number*

$$\frac{\frac{1}{2\log 2} + \sum_{k=1}^{\infty} \frac{1}{k} \log_2\left(\frac{k+2}{k+1}\right)}{\pi^2/6 - 5/4} \approx 3.04366$$

The theorem above confirms the empirical observation by Lang [117] that PCSA has a better performance than HyperLogLog after compressing the memory and using statistically efficient estimators. We prove that H_0/I_0 is optimal for a large class of sketches that we call *linearizable*, which is a subclass of the mergeable sketches.

Theorem 1.5 (Pettie and **Wang** [137], Section 5.4)**.** *If a* `COUNT-DISTINCT` *sketch is linearizable, then its* Fish *number is at least* H_0/I_0*.*

We also show the lower bound above is algorithmically reachable. We give a new sketch called Fishmonger whose space, at all times, is $O(\log^2 \log n) + (1 + o(1))(H_0/I_0)b \approx 1.98b$ bits and whose relative standard error[17], at all times, is $1/\sqrt{b}$, with probability $1 - 1/\operatorname{poly}(b)$.

Most known `COUNT-DISTINCT` sketches are linearizable, including PCSA [83], HyperLogLog [82], and k-MinCount [90]. Some sketches are non-linearizable (e.g., AdaptiveSampling [81]) but none of them is known to have a lower Fisher-Shannon number than H_0/I_0. It is thus quite likely that H_0/I_0 is the universal lower bound, but whether this is true remains an open problem.

1.5.2 Beyond Fisher-Shannon: the Sequential Setting

Data streams are often processed in a *distributed manner* (see the MapReduce model [66]), where multiple processors are working simultaneously on the updates. Each processor only sketches a fraction of the data stream and their results need to be merged before inferring information of the whole stream. As I primarily focus on distributed streams, I have been assuming so far that only the *final state* is observed and not caring about how the updates are collected and processed in the distributed system. Nevertheless, there are situations where the updates are fed into a single machine sequentially. Even in a distributed system, cases

[17] Let $\hat{y}$ be an estimator of y. The relative standard error is defined as $|y|^{-1}\sqrt{\mathbb{E}|\hat{y} - y|^2}$.

could be that a single server is collecting local data for a long time before communicating with another server, and during such time the sketch states can be considered sequentially observed. So a concrete question is how well can we answer `COUNT-DISTINCT` if, apart from the final state, all previous sketch states are observed.

Cohen [49] and Ting [151] independently invented the *martingale estimator*.

Definition 1.6 (martingale estimator)**.** Let $S_0, S_1, S_2, \ldots$ be the sequence of sketch states where S_λ is the state after λ distinct insertions. For each $\lambda \in \mathbb{N}$, define

$$E_\lambda = \sum_{j=1}^{\lambda} \frac{\mathbb{1}\left[S_j \neq S_{j-1}\right]}{\mathbb{P}(S_j \neq S_{j-1} | S_{j-1})}.$$

E_λ is called the *martingale estimator* and $\mathbb{P}(S_\lambda \neq S_{\lambda-1}|S_{\lambda-1})$ is called the *remaining area*, of the sketch $\mathcal{C}$, at time $\lambda - 1$.

The martingale estimator E_λ is clearly unbiased since $(E_\lambda - \lambda)_{\lambda\in\mathbb{N}}$ is a martingale. I analyze the variance for a generic sketch $\mathcal{C}$ as the number m of i.i.d. sub-sketches goes to infinity. Let us denote the sketch that consists of m copies of $\mathcal{C}$ as $\mathcal{C}^m$.

Theorem 1.6 (Pettie, **Wang**, and Yin [140], Section 5.7)**.** *Let $E_{m,\lambda}$ and $P_{m,\lambda}$ be the martingale estimator and remaining area for $\mathcal{C}^m$ after λ insertions per sub-sketch. Define $\kappa = \lim_{\lambda\to\infty} \lim_{m\to\infty} \lambda P_{m,\lambda}$. If κ exists, then*

$$\lim_{\lambda\to\infty} \lim_{m\to\infty} m\lambda^{-2}\mathbb{V}E_{m,\lambda} = \frac{1}{2\kappa}.$$

My contribution to this topic is the asymptotic theorem above which provides a simple recipe to compute the asymptotic relative variance, making analysis of complicated sketches possible. To demonstrate its power, in Section 5.7.1 we define a class of Curtain sketches, which interpolate between PCSA and HyperLogLog and allow *correlations* between sub-sketches, providing a wide range of choices for a sketch designer to trade off among update time, memory-variance product, query time, and simplicity for implementation. See Fig. 1.4.

One may compute the constant κ for PCSA and HyperLogLog, which are $\frac{1}{\log 2}$ and $\frac{1}{2\log 2}$ respectively. The memory-variance products are thus $H_0/2 \approx 1.63$ and $\frac{1}{2\log 2} + \sum_{k=1}^{\infty} \frac{1}{k}\log_2(\frac{k+2}{k+1})) \approx 1.96$ respectively. Note that both are lower than $H_0/I_0 \approx 1.98$, which implies that one may get memory-variance product lower than H_0/I_0 (and simpler estimators) *after sacrificing mergeability*. A similar lower bound can be proved for the linearizable sketches in this sequential setting.

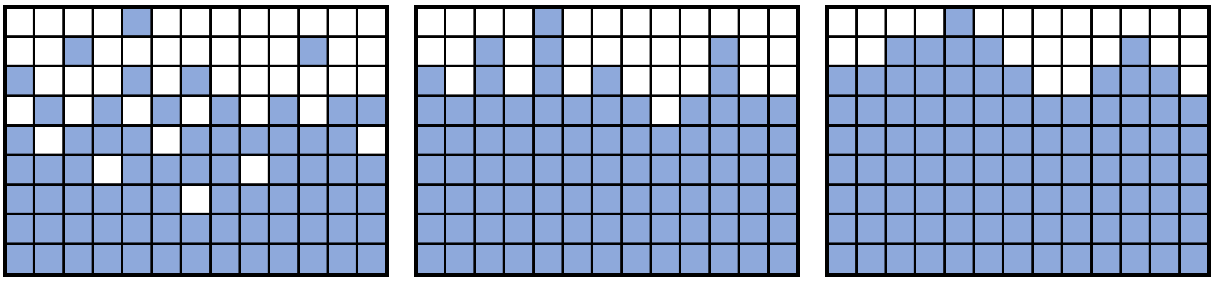


Figure 1.4: Diagrams of COUNT-DISTINCT sketches. Each new insertion will independently mark a cell with probability $2^{-\text{height of the cell}}$. Left: PCSA, storing every cell independently. Middle: HyperLogLog, storing only the profile; Right: Curtain **(new)**, storing only the *Lipschitz profile* (adjacent columns differ by at most one).

Theorem 1.7 (Pettie, **Wang**, and Yin [140], Section 5.7.4)**.** *Let $\mathcal{C}$ be a linearizable sketch such that $\kappa = \lim_{\lambda\to\infty}\lim_{m\to\infty}\lambda P_{m,\lambda}$ exists. Let $S_{m,\lambda}$ be the sketch state for $\mathcal{C}^m$ after λ insertions per sub-sketch and $E_{m,\lambda}$ be the martingale estimator. Then*

$$\lim_{\lambda\to\infty}\lim_{m\to\infty} H(S_{m,\lambda})\lambda^{-2}\mathbb{V}E_{m,\lambda} \geq H_0/2.$$

This lower bound can also be achieved algorithmically using Martingale Fishmonger.

1.5.3 A Useful Statistic: Generalized Remaining Areas

Let us now switch back to the usual distributed/mergeable setting where only the final state is observed. While in general one may use the maximum likelihood estimator, its computation often requires a time-consuming numerical procedure which results in slow query time (and complicated implementation) in practice. In Section 5.8 I study an explicit estimation method [160] for COUNT-DISTINCT that generalized many prior estimators [83, 73, 82]. It is based on statistics that generalize the *remaining area* statistic. Recall that the remaining area statistic is also used in the martingale estimator (Definition 1.6).

Definition 1.7 (generalized remaining area, informal)**.** Fix any sketch and let it be represented as a board of cells as in Fig. 1.4. Let $\mathcal{B}$ be the set of all cells and A be the current state which is a subset of $\mathcal{B}$. The *τ-generalized remaining area* is defined as

$$\tau\text{-GRA}(A) = \sum_{C\in\mathcal{B}, C\not\subset A} |C|^\tau.$$

Definition 1.8 (τ-GRA estimator, informal)**.** Let S_λ be the state at cardinality λ. Let

$S_\lambda^{(1)}, \dots S_\lambda^{(m)}$ be m sub-sketches that are i.i.d. copies of S_λ. The τ-GRA estimator of the product sketch $(S_\lambda^{(1)}, \dots S_\lambda^{(m)})$ is defined as

$$E_{m,\lambda}^* = \frac{E_{m,\lambda}}{\mathbb{E}E_{m,1}}, \text{ where } E_{m,\lambda} = \left(\frac{1}{m}\sum_{j=1}^{m} \tau\text{-GRA}(S_\lambda^{(j)})\right)^{-\tau^{-1}}.$$

We analyze the mean and variance of the τ-GRA estimator in the regime where both λ and m are large.

Theorem 1.8 (Pettie and **Wang** [160], Section 5.8)**.** *For any $\lambda > 0$, E_λ^* is asymptotically unbiased with relative variance $(m^{-1} + O(m^{-2}))\tau^{-2}(\mathbb{E}(\tau\text{-GRA}(S_1)))^{-2}\mathbb{V}(\tau\text{-GRA}(S_1))$ as $m \to \infty$.*

The class of τ-GRA estimators includes classic estimators LogLog [73] (0-GRA), HyperLogLog [82] (1-GRA), and Lang's coupon collector for PCSA [117] (0-GRA), where 0-GRA estimator can be considered the limit of τ-GRA estimator as $\tau \to 0$. See Fig. 1.5.

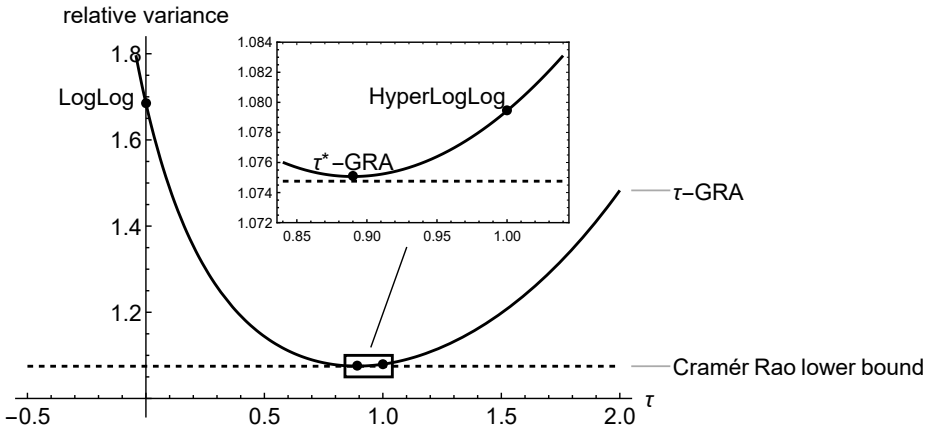
The τ-GRA estimation framework not only improves the constants (and remains quickly computable, unlike MLE) for classic sketches, but more importantly provides an automatic, base-line method to devise estimators for "scale-invariant" sketches. For example, a subclass of the complicated Curtain sketches that used to be only analyzable in the sequential setting (Section 5.7.1) can now be estimated using τ-GRA with tractable asymptotic relative variance. Furthermore, statistics similar to negative τ-GRA are used in the estimation of harmonic moments. See Section 4.5.1.

1.6 Sampling

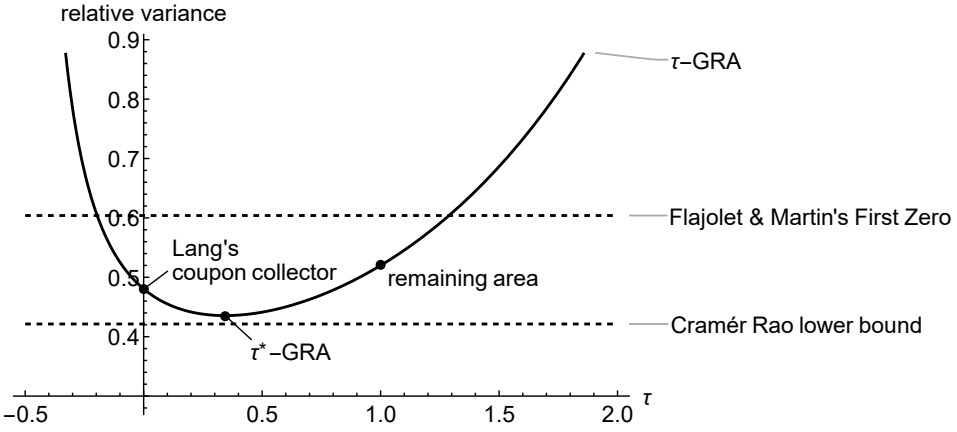
So far the topic has been *estimation*. There is another important area of streaming—*sampling*.

Query 9 (SAMPLE)**.** Consider the insertion-only model with domain $[n]$ and let $\mathbf{x}(v)$ be the number of times element v is inserted in the stream. A SAMPLE query is parameterized by a function $G\colon \mathbb{N} \to \mathbb{R}_+$ with $G(0) = 0$. It asks for a *G-sample* $v_* \in [n]$ such that for any $v \in [n]$,

$$\mathbb{P}(v = v_*) = \frac{G(\mathbf{x}(v))}{\sum_{u=1}^{n} G(\mathbf{x}(v))}(1 \pm \epsilon) \pm \eta,$$



(a) Relative variance of estimators for the LogLog sketch. The τ-GRA estimator attains minimum variance at $\tau^* = 0.88989$, which comes within 0.02% of the Cramér-Rao lower bound. As a comparison, HyperLogLog is 0.4% over the bound.



(b) Relative variance of estimators for the PCSA sketch. The τ-GRA estimator attains minimum variance at $\tau^* = 0.343557$, which comes within 3% of the Cramér-Rao lower bound.

Figure 1.5: Relative variance for τ-GRA estimators

where ϵ is the multiplicative error and η is the additive error. A streaming algorithm that can return v_* with $(\epsilon, \eta) = (0, 0)$ is called a *G-sampler*, otherwise, it is an *(ϵ, η)-approximate G-sampler*.

Remark 1.6. This definition is simplified from the G-sampling definition in Jayaram's thesis [101], where the failure probability δ is included in the additive error η—one just outputs an arbitrary index upon failure.

With different functions G, G-samples reveal different information about the stream. See Table 1.1 for an example.

F_0-SAMPLES	COUNT	F_1-SAMPLES	COUNT	$F_{1/2}$-SAMPLES	COUNT
lends	(13)	methought	(43)	way	(562)
vurther	(1)	that	(11121)	lay	(278)
wormy	(1)	iago	(362)	waking	(25)
pepperd	(2)	break	(257)	perfectly	(4)
scuffling	(1)	would	(2295)	most	(1182)
observing	(5)	ay	(764)	lock'd	(27)
padua	(41)	bona	(21)	slew	(37)
bidst	(5)	end	(345)	plead	(48)
preventions	(1)	friends	(480)	rememb'rest	(4)
engrossd	(2)	my	(12480)	grease	(6)

Table 1.1: Sampling words from Shakespeare's plays. F_p-samples are returned by G-samplers with $G(x) = x^p$. One may observe that the F_0-samples are rather uncommon words as those are sampled uniformly from all distinct words (and we know Shakespeare has a large vocabulary). On the other hand, the F_1-samples are much more common as the probability of a word gets sampled is proportional to their counts in the text. Finally, $F_{1/2}$-samples are sampled with probability proportional to the square root of their occurrences. Thus $F_{1/2}$ samples are more common in comparison to the F_0-samples, and at the same time, they are more diverse than the F_1-samples, which are useful for, e.g., generating quizzes for a Shakespeare contest.

A detailed survey on the streaming sampling literature is deferred to Section 3.5. Long story short, prior samplers [102, 104, 52] all have non-zero errors in their sampling probabilities, except for the following two special cases—`SAMPLE-UNIFORM` and `SAMPLE-DISTINCT`. My contribution for this topic is an elegant construction of a large class of G-samplers with *exactly correct* sampling probabilities (Pettie and **Wang** [139], Section 3.5.3).

Query 10 (`SAMPLE-UNIFORM`, aka F_1-sampling)**.** Return an element uniformly at random over all insertions in the stream.

Query 11 (`SAMPLE-DISTINCT`, aka F_0-sampling)**.** Return an element uniformly at random over the set of elements appeared in the stream.

An F_1-sampler is presented in Algorithm 1.1, which can be considered a min-based reservoir sampler [155]. A F_0-sampler is presented in Algorithm 1.2, which is folk-lore, equivalent to Cohen's 1-Min sketch [48]. The two algorithms are both straightforward. For generic G, the key ingredient to get correct G-samples is the construction of a class of *level functions* ℓ_G, with which one can "mix" the F_1-sampler and F_0-sampler in the correct way to form a G-sampler. See Algorithm 1.3 for a $F_{1/2}$-sampler constructed in this way. The "mixing" is non-trivial though, which uses the inverse error function erf^{-1}.[18]

Algorithm 1.1: F_1-sampler. Sample an element uniformly from all insertions.

Sketch : $(v_*, h_*) \in ([n] \cup \{\perp\}) \times (\mathbb{R}_+ \cup \{\infty\})$, initialized as $(\perp, \infty)$
Hash function: $H \colon [n] \to \text{Uniform}(0, 1)$

Insert (v)**:**
  `// upon inserting` $v \in [n]$
  $h \leftarrow \text{Exp}(1)$ `// h is freshly sampled`
  if $h < h_*$ **then**
    $(v_*, h_*) \leftarrow (v, h)$
  end
end

Sample ( )**:**
  `// upon sample`
  return v_*
end

Algorithm 1.2: F_0-sampler. Sample an element uniformly from $\text{supp}(\mathbf{x})$.

Sketch : $(v_*, h_*) \in ([n] \cup \{\perp\}) \times (\mathbb{R}_+ \cup \{\infty\})$, initialized as $(\perp, \infty)$
Hash function: $H \colon [n] \to \text{Uniform}(0, 1)$

Insert (v)**:**
  `// upon inserting` $v \in [n]$
  $h \leftarrow H(v)$ `// h's randomness is indexed by` v
  if $h < h_*$ **then**
    $(v_*, h_*) \leftarrow (v, h)$
  end
end

Sample ( )**:**
  `// upon sample`
  return v_*
end

[18] erf^{-1} is available as `boost::math::erf_inv` in C++, and `scipy.special.erfinv` in Python.

Algorithm 1.3: $F_{1/2}$-sampler. **(new)**. Sample an element v_* with prob. $\propto \sqrt{\mathbf{x}(v_*)}$.

Sketch : $(v_*, h_*) \in ([n] \cup \{\perp\}) \times (\mathbb{R}_+ \cup \{\infty\})$, initialized as $(\perp, \infty)$
Hash function: $H\colon [n] \to \text{Uniform}(0,1)$

Insert (v)**:**
  `// upon inserting` $v \in [n]$
  $h \leftarrow \sqrt{2\text{Exp}(1)} \cdot \text{erf}^{-1}(H(v))$ `// mixing fresh and indexed randomness`
  if $h < h_*$ **then**
    $(v_*, h_*) \leftarrow (v, h)$
  end
end

Sample $(\,)$**:**
  `// upon sample`
  return v_*
end

Algorithm 1.4: G-sampler. **(new)**. Sample an element v_* with prob. $\propto G(\mathbf{x}(v_*))$.

Sketch : $(v_*, h_*) \in ([n] \cup \{\perp\}) \times (\mathbb{R}_+ \cup \{\infty\})$, initialized as $(\perp, \infty)$
Hash function: $H\colon [n] \to \text{Uniform}(0,1)$
Level function: $\ell_G\colon \mathbb{R}_+ \times [0,1] \to \mathbb{R}_+$

Insert (v)**:**
  `// upon inserting` $v \in [n]$
  $h \leftarrow \ell_G(\text{Exp}(1), H(v))$ `// mixing fresh and indexed randomness`
  if $h < h_*$ **then**
    $(v_*, h_*) \leftarrow (v, h)$
  end
end

Sample $(\,)$**:**
  `// upon sample`
  return v_*
end

1.6.1 Level Functions, G-Sampling, and More

Level functions ℓ_G are constructed carefully so that the generic Algorithm 1.4 correctly returns a G-sample. I construct such ℓ_G for all G that is the Laplace exponent of a non-negative Lévy process, aka a subordinator.

Definition 1.9 (Lévy induced level function)**.** Let $G\colon \mathbb{R}_+ \to \mathbb{R}_+$ be the Laplace exponent of a subordinator $X = (X_t)_{t\geq 0}$, i.e., for any $t, z \in \mathbb{R}_+$,

$$\mathbb{E}e^{-zX_t} = e^{-tG(z)}.$$

The *induced level function* $\ell_G\colon \mathbb{R}_+ \times [0,1] \to \mathbb{R}_+$ is defined to be

$$\ell_G(a,b) = \inf\{t \mid \mathbb{P}(X_t \geq a) \geq b\}.$$

Lemma 1.1 (level functions)**.** *For any Laplace exponent G, ℓ_G satisfies:*

2D-monotonicity. *for any $a, a' \in \mathbb{R}_+$ and $b, b' \in [0,1]$, $a \leq a'$ and $b \leq b'$ implies $\ell_G(a,b) \leq \ell_G(a',b')$;*

G-transformation. *if $Y \sim \mathrm{Exp}(\lambda)$ and $U \sim \mathrm{Uniform}(0,1)$, then $\ell_G(Y,U) \sim \mathrm{Exp}(G(\lambda))$.*

The two properties above are sufficient for Algorithm 1.4 to have exactly correct sampling probabilities.

Theorem 1.9 (Lévy-Min-Sampler, Section 3.5.3)**.** *Let $G : \mathbb{R}_+ \to \mathbb{R}_+$ be any function of the form*

$$G(z) = c\mathbb{1}[z \neq 0] + \gamma_0 z + \int_0^\infty (1 - e^{-zs})\,\nu(ds),$$

where $c, \gamma_0, \geq 0$, and ν is any positive measure such that $\int_0^\infty \min(x,1)\,\nu(dx) < \infty$. There is a min-based sketch (Algorithm 1.4) storing only a single pair $(v_*, h_*) \in [n] \times \mathbb{R}_+$, where $v_* \in [n]$ is the sampled index and $h_* \in \mathbb{R}_+$ is the minimum hash value. For any given input vector $\mathbf{x} \in \mathbb{R}_+^n$ it is guaranteed that

- *For any $u \in [n]$, $\mathbb{P}(v_* = u) = G(\mathbf{x}(u))/G(\mathbf{x})$.*
- *$h_* \sim \mathrm{Exp}(G(\mathbf{x}))$.*

The space required is therefore only 2 words.

The level functions are initially invented for G-sampling, but it turns out to be a very convenient tool. In Sections 3.5 and 3.10.2, I will demonstrate how ℓ_G can be used to construct universal samplers, samplers with or without replacements, and more general sampling networks.

1.7 Approximate Counting

Finally, let us turn our attention to the most classic streaming problem—*approximate counting*, i.e., to $(1 \pm \epsilon)$-approximately answer `COUNT`. This problem was first introduced by Morris [128].

Query 12 (`COUNT`)**.** Return the total number of insertions over the stream.

In the framework of f-moment in Section 1.3, `COUNT` corresponds to the f-moment with $f(x) = x$, which is *linear*. Thus approximate counting is trivial in this perspective, which only needs an $O(\log n)$-bit counter to perfectly answer ($\epsilon = 0$), assuming the stream length is $O(\text{poly}(n))$. Nevertheless, this is not optimal when a non-zero relative error ϵ is allowed.

In 1978, Morris [128] invented the classic *approximate counter* which can count up to n with $\log_2 \log_2 n + O(\log_2 \epsilon^{-1})$ bits. In the original paper [128], Morris analyzed the mean and variance of the estimates. Flajolet [80] analyzed the mean and variance of the *index*[19] of the Morris counter, from which a quantified space bound can be derived with Chebyshev's bound. Nelson and Yu [133] analyzed the *tail* of the index, obtaining a sharper dependence on the failure probability. Specifically, for increments up to n and $\epsilon, \delta > 0$, if an estimate $\hat{n}$ is produced with $\mathbb{P}(|\hat{n} - n| < \epsilon n) \geq 1 - \delta$ then the Morris counter needs $O(\log \log n + \log \epsilon^{-1} + \log \log \delta^{-1})$ space, and this is proved to be optimal [133].[20]

The common scenario in real-world applications is that many different counters are maintained simultaneously. For example, the original motivation of Morris is to count the number of each trigram in texts [123, 133] where $d = 26^3$ different trigrams are counted simultaneously. Aden-Ali, Han, Nelson, and Yu [2] showed that, essentially, d independent counters cannot be compressed as long as each counter is required to report an estimate with an error *relative to itself*. However, as I will show in the thesis, the space *can be compressed* if the error metric is the d-dimensional Euclidean norm. While the per-entry relative error

[19] Roughly speaking, Morris counter stores an index v to represent a count around 2^v. Therefore, to count up to n, the index increases to $\log n$ and thus it takes $\log \log n$ to store the index.

[20] Technically, one needs to keep a separate deterministic counter for small $n = O(a)$ to obtain the optimal failure rate [133].

requirement considered in [2] is reasonable when an approximation for *each individual entry* is needed, we remark that in applications where an approximation of the *whole count vector* is needed, the Euclidean error metric is more natural.

Definition 1.10 ((Euclidean) d-dimensional counting)**.** Let $n \geq 1$ be the maximum stream length, $d \geq 1$ be the dimension, and $\epsilon > 0$ be the relative error. An (n, d, ϵ)-*counter* is a randomized data structure that, for any input sequence $(w_1, \ldots, w_k) \in [d]^k$ with $k \leq n$, returns an estimate $\hat{x}$ of $x = \sum_{j=1}^{k} e_{w_j}$ with $\mathbb{E}|\hat{x} - x|^2 < \epsilon^2 |x|^2$, where e_{w_j} is the unit vector with 1 in the w_jth entry.[21]

The main contribution here, presented in Chapter 7, is the construction of a simple and optimal (n, d, ϵ)-counter, with a matching lower bound.

Theorem 1.10 (informal, **Wang** [159])**.** *The optimal space of an (n, d, ϵ)-counter is $\log_2 \log_2 n + \Omega(d \log_2 \epsilon^{-1})$ bits.*

1.8 Road Map

The main body of the thesis consists of several of my published and unpublished works. Let me quickly draw the road map. More details and related works will be discussed in the corresponding sections.

- In Chapter 2, I will describe the scope of this thesis by specifying the choices I made when modeling streams, computation, and queries.
- In Chapter 3, I will present systematic ways to construct sketches/samplers from Lévy processes, connecting the design of streaming algorithms with the Lévy-Khintchine representation theorem. With this theory, one may now *deduce* very specific sketches for different estimation tasks with a common, simple estimator. One can also deduce very specific samplers for different weighting functions with precisely correct sampling probabilities. This chapter is based on the following two works.
  - Sketching, Moment Estimation, and the Lévy-Khintchine Representation Theorem (ITCS25), with Seth Pettie
  - Universal Perfect Samplers for Incremental Streams (SODA25), with Seth Pettie

[21] See REMARKS ON NOTATIONS in the front of the thesis for choices of mathematical symbols and conventions.

- In Chapter 4, I will present another sketching scheme that is *complementary* to the approach in Chapter 3. Instead of constructing task-specific sketches, I will start with the most classic subsample-and-aggregate sketch and study how to extract information from it for different queries. Now the sketch itself is classic and simple, but for each statistical query, a specific estimator will be derived based on the query's harmonic structure, revealing a new path towards universality. This chapter is based on the following extended abstract.
  - Harmonic Decomposition in Data Sketches (STOC25)
- In Chapter 5, I will focus on the query of `COUNT-DISTINCT`, aka, the cardinality estimation problem. I will formally study `COUNT-DISTINCT` sketches (caring about leading constants), investigating their theoretical limit, constructing diverse sketching schemes, and analyzing diverse estimation schemes. This chapter is based on the following works.
  - Information Theoretic Limits of Cardinality Estimation: Fisher meets Shannon (STOC21), with Seth Pettie
  - Non-Mergeable Sketching for Cardinality Estimation (ICALP21), with Seth Pettie and Longhui Yin
  - Better Cardinality Estimators for HyperLogLog, PCSA, and Beyond (PODS23), with Seth Pettie
- In Chapter 6, I will construct cardinality sketches over M-turnstile streams for any (finite) commutative monoid M. (I am going to define the M-turnstile model shortly in Section 2.1.) The construction is based on the theory and techniques developed in Chapters 3 to 5. This chapter is meant to demonstrate how streaming sketches can be generalized to generic spaces.
- In Chapter 7, I will study two variants of the classic approximate counting problem—multi-dimensional counting and counting with deletions. The multi-dimensional part is based on the following work.
  - Multi-dimensional Approximate Counting (SOSA25)
- In Chapter 8, I will study the *sequential behavior* of cardinality sketches, from which a delicate fraud detection scheme is constructed.

Without further ado, let us now enter the world of streaming.

CHAPTER 2

Modeling Streaming Computation

All models are wrong, but some are useful. George E. P. Box [17], 1976

Mathematical investigation is only possible once a specific model is chosen as a precisely defined abstract representation of the real-world scenario and, hopefully, the insights and solutions obtained in the abstract model can map back to useful programs and structures on real-world hardware. In this section, I will discuss the modeling assumptions I make in this thesis and explain the rationale for those choices.

2.1 Modeling Streams

A *stream* in general describes a collection of dynamic *updates* to a *domain* of *elements*.

Domain I will assume the domain is countably infinite, usually identified by $\mathbb{N}$. This reflects one natural characteristic of streams—they can go on forever. When accounting for space, I will assume the stream only contains elements in the prefix $[n]$, and in such cases the space can be expressed as a function of n. This assumption requires the algorithm to have an "infinite version" which can handle unbounded domains and emphasizes that the algorithm should be *uniform* for all domain sizes n, instead of having different algorithms for different n.

States and updates I will model the state space of an element as a *commutative monoid* $(M, +)$, which naturally induces the rules for initialization and updates. Let the state of element $v \in \mathbb{N}$ be $\mathbf{x}(v)$.

- Every element has initial state $\mathbf{x}(v) = 0_M$, i.e., the identity of M.

- Each update in the stream is a pair (v, y) where $v \in \mathbb{N}$ is an element in the domain and $y \in M$ is the change of its state. Thus after receiving (v, y), $\mathbf{x}(v) \leftarrow \mathbf{x}(v) + y$.

Let $\mathbf{x} = (\mathbf{x}(1), \mathbf{x}(2), \ldots)$ be the *state vector*. The stream can be distributed—a *set* of updates $\{(v_1, y_1), \ldots, (v_k, y_k)\}$ can be collected in arbitrary order with the same final state vector $\mathbf{x}$, because M is commutative and associative. We assume $\mathbf{x}$ only contains an arbitrary but finite number of non-zero entries.

Later on, when we study more specific settings, a streaming model with element state space M will be called M-***turnstile***. This notation is convenient when discussing "cross model" algorithms. The M-turnstile model characterizes the *algorithmic functionality* (e.g., incremental-only vs. incremental/decremental) by the *mathematical structure* of $(M, +)$. For example, if M is idempotent, then duplicated updates are ignored; if M is a group, then "deletions" are allowed; if M is a continuous space, then fractional updates are allowed; if M is multi-dimensional, then attribute-wise updates are allowed. Specifically, the usual insertion-only model (Definition 1.2) corresponds to $\mathbb{N}$-turnstile, where the element state $\mathbf{x}(v)$ indicates its number of occurrences in the stream. See Table 2.1.

Model Name	M-turnstile Framework	Notes
insertion-only	$\mathbb{N}$-turnstile model	common in the literature
(original) turnstile model	$\mathbb{Z}$-turnstile model	common in the literature
boolean model	$\mathbb{F}_{2^k}$-turnstile model	k is a positive integer
modulo model	$\mathbb{Z}/p\mathbb{Z}$-turnstile model	p is a positive integer
finite field model	F-turnstile model	F is a finite field
invertible model	G-turnstile model	G is an abelian group
base model	M-turnstile model	M is a commutative monoid

Table 2.1: Common M-turnstile models. The name "M-turnstile" comes from the *turnstile model* used in the literature [130, 97] to denote the streaming model that allows both insertions and deletions.

Remark 2.1. Such algebraic definitions like M-turnstile serve as the basis of developing "cross model" algorithms. Many streaming algorithms actually only require a few properties of the underlying state space, which means the same algorithms will work as well over other state spaces that satisfy those properties. This idea is especially useful when extending algorithms to handle multi-dimensional streams where, e.g., each element has a state in $\mathbb{Z}^d$ (Section 3.4). I will also present a cardinality sketch that works in the *base model*, which can be applied to arbitrary M-turnstile streams (Chapter 6).

Remark 2.2. In the literature, streams of elements with both insertions and deletions are called *turnstile streams* [130], which correspond to $\mathbb{Z}$-turnstile streams in the framework

above. While M-turnstile is very expressive a template for modeling streams with various element state spaces, it does not capture more specific stream settings such as the following.

- Strict turnstile model [130, 3]. If the stream contain actual insertions and deletions, there is a natural invariant that $\mathbf{x}(v) \geq 0$—a record cannot occur a (-1) number of times. A strict turnstile model allows both insertions and deletions but requires the frequency of each element to be non-negative at all times.
- Bounded deletion model [103]. For many queries, algorithms in the $\mathbb{N}$-turnstile model require less space than those in the $\mathbb{Z}$-turnstile model. Jayaram and Woodruff [103] propose the bounded deletion model which assumes only positive updates are made to a $1/\alpha$-fraction of elements, and negative/positive updates can be made to other elements. This model smoothly interpolates between $\mathbb{N}$-turnstile ($\alpha = 1$) and $\mathbb{Z}$-turnstile ($\alpha = \infty$), providing tailored algorithms for streams of different α parameters.

2.2 Modeling Computation

Relative to the previous literature, my assumptions on streams are general and weak, which assumes nothing but the existence of identity (to initialize streams) and commutativity/associativity (to merge streams). However, I will make *strong* assumptions on the algorithms in the thesis, which I now discuss and defend.

Homomorphic sketching I will assume the sketch is a *homomorphic image* of the input stream. More specifically, the *memory state space* $(\Omega, \uplus)$ is also a commutative monoid, where $\uplus$ is the merge operation. Each element v corresponds to a *randomized* homomorphism $\Psi^{(v)} \colon M \to \Omega$. Recall that $\mathbf{x}(v)$ is the state of the element $v \in \mathbb{N}$ and $\mathbf{x}$ is the state vector. The sketch stores

$$\begin{aligned}\Psi^*(\mathbf{x}) &= \Psi^{(1)}(\mathbf{x}(1)) \uplus \Psi^{(2)}(\mathbf{x}(2)) \uplus \cdots \\ &= \biguplus_{v=1}^{\infty} \Psi^{(v)}(\mathbf{x}(v)),\end{aligned}$$

where $\biguplus_{v=1}^{\infty} \Psi^{(v)}(\mathbf{x}(v))$ is well-defined as we have assumed $\mathbf{x}$ has a finite support. This naturally induces the update and merge operation.

- Given the current sketch $\Psi^*(\mathbf{x})$, upon an update (v, y) with $v \in \mathbb{N}, y \in M$, the updated state vector is $\mathbf{x}' = \mathbf{x} + ye_v$, where e_v is the vector with vth entry being one and others

being zero.[1] Then new sketch state can be computed as

$$\underbrace{\Psi^*(\overbrace{\mathbf{x}'}^{\text{new stream state}})}_{\text{new sketch state}} = \Psi^*(\underbrace{\mathbf{x}}_{\text{previous stream state}} + \overbrace{ye_v}^{\text{update to stream}}) = \underbrace{\Psi^*(\mathbf{x})}_{\text{previous sketch state}} \uplus \overbrace{\Psi^{(v)}(y)}^{\text{update to sketch}}.$$

- Suppose now we have two streams, one with state vector $\mathbf{x}$ and the other with $\mathbf{y}$. The sketches for the two are $\Psi^*(\mathbf{x})$ and $\Psi^*(\mathbf{y})$. The merged sketch state can be computed as

$$\underbrace{\Psi^*(\overbrace{\mathbf{x}+\mathbf{y}}^{\text{merged stream}})}_{\text{merged sketch}} = \underbrace{\Psi^*(\overbrace{\mathbf{x}}^{\text{stream 1}})}_{\text{sketch 1}} \uplus \underbrace{\Psi^*(\overbrace{\mathbf{y}}^{\text{stream 2}})}_{\text{sketch 2}}.$$

Both update and merge depend crucially on the fact that $\Psi^* = \biguplus(\Psi^{(1)}, \Psi^{(2)}, \ldots)$ is homomorphic from $M^{\mathbb{N}}$ to Ω. It is by construction that update and merge defined in this way can be computed in distributed systems, where the aggregated final sketch is insensitive to the order of association. Thus from now on we will not distinguish the stream and the state vector—both are denoted as $\mathbf{x}$.

The fact that homomorphic sketches are native to distributed streams is the primary reason for me to make this assumption. In fact, most existing sketches are homomorphic (either taking minimum or taking weighted sum). Practically, contemporary parallel computing frameworks explicitly ask engineers to write commutative and associative functions for fast and robust distributed computing.

> The combining function should be **commutative** and **associative**, as the function is not necessarily invoked exactly once on all values with a given key. Because the input data (including the value collection) may be distributed across multiple workers, the combining function might be called multiple times to perform partial combining on subsets of the value collection.[2]
>
> Apache Beam Programming Guide, 4.2.4. Combine

An important consequence of the existence of homomorphisms from M to Ω is that the structure of $(M, +)$ is also projected to the structure of $(\Omega, \uplus)$.[3] For example, if M is

[1] For M is a commutative monoid, $x \in M$ and $k \in \mathbb{N}$, we define $kx = \underbrace{x + \cdots + x}_{k}$.

[2] https://beam.apache.org/documentation/programming-guide/#combine

[3] Here, I assume Ω contains only *reachable states*. The algebraic structure of M will be projected to all reachable states. In general, of course, one may equip Ω with arbitrary structures that is hidden from M, which is pointless in algorithm design.

idempotent, then Ω is necessarily idempotent and if M is a group, then Ω is necessarily a group. This is the underlying reason that sketches in the same model are algebraically similar.

Algorithms in Chapter 3 and Chapter 4 are all homomorphic sketches. In Section 5.2, we will prove lower bounds for homomorphic `COUNT-DISTINCT` sketches. Nevertheless, homomorphic sketches are oblivious to the intermediate states of the stream and thus are insensitive to sequential behaviors. In Section 5.7, I will analyze a few non-mergeable algorithms.

Are generic sketches significantly more powerful than homomorphic sketches? Not necessarily. In fact, Li, Nguyễn, and Woodruff [119] show that in the $\mathbb{Z}$-turnstile model, essentially any tractable queries can be answered by linear (and thus homomorphic) sketches as well. For `COUNT-DISTINCT`, it is observed by Cohen [49] and Ting [151] that a `COUNT-DISTINCT` algorithm can be made more accurate using the sequential information. Nevertheless, the gain is in the constants, reducing about 20 to 50% of the variance depending on the structure of the sketch.

Random oracle I will assume the sketch has access to a *random oracle* that, upon a query on an element v, returns an indexed random variable $X^{(v)}$, usually considered to be $\mathrm{Uniform}(0,1)$. In a practical implementation, this random oracle is replaced by a suitable hash function that is provided externally. For a homomorphic sketch, $X^{(v)}$ is used to seed the map $\Psi^{(v)}$ so that $\Psi^{(1)}, \Psi^{(2)}, \dots$ is an independent sequence, which naturally pins down a *unique* distribution of the final sketch state $\Psi^*(\mathbf{x})$ for each input stream $\mathbf{x}$. Thus by combining homomorphic sketching with random oracles, a statistical structure emerges—the sketch is a *parametric distribution family* parametrized by $\mathbf{x}$ and the estimation target is some statistical query $q(\mathbf{x})$. This enables algorithm designers to be *statistically aware* in the sense that every algorithm can have a well-defined error (e.g., relative mean squared error with the worst input), instead of having only an error bound.

Of course, access to random oracles is not necessary in general to design streaming algorithms. Many important sketches have been successfully partially derandomized [5, 110] by using hash functions with limited independence. In contrast to assuming random oracles where we need to hope the externally provided hash values are i.i.d., the standard model streaming algorithms only need to assume the random seed for the hash functions is truly random, which is, indeed, a widely accepted assumption. It is thus considered to be the *standard model* [5, 110, 101] in which hash functions are explicitly constructed and stored in the working memory. On the other hand, assuming access to random oracles is called the

random oracle model in the literature [10, 101].

There are both large bodies of works in the standard model and random oracle model, shining together. Notably in the standard model, optimal algorithms for frequency moment estimation have been constructed [96, 98, 110] and there are also powerful universal sketching schemes [23, 19, 13]. In the random oracle model, one big success is the HyperLogLog sketch [82], which is a statistically improved version of the LogLog sketch [73]. The HyperLogLog has been further tuned by Google [95] and is widely implemented in databases, web servers, and routers.

Personally, I assume random oracles for most of my work during my PhD because my first few projects focus on developing and analyzing estimators for `COUNT-DISTINCT` sketches, caring about constants. For my later works on more generic sampling and sketching, I pay less attention to the leading constants, though they are often small. For these works, I assume random oracles mostly for technical ease that I can apply classic probability theorems about independent sequences to sketching. As these works have interesting results in space complexity (not only improving constants), it is well motivated to transform these new algorithms to the standard model. This direction is left to future work.

So to defend the choice of assuming random oracles in this thesis, let me list a few reasons why results in the random oracle model are valuable in themselves.

Granularity As we discussed before, given a sketch and an input stream, in the random oracle model there is a well-defined and *unique* distribution of the final sketch state. This level of granularity allows algorithm designers to make finer algorithmic choices, resulting in better algorithms in practice. On the other hand, an algorithm using k-wise independent hash functions can have many possible final distributions depending on the implementation of the hash functions, making a constant-aware analysis difficult.[4] One prominent example is the development from LogLog [73] to HyperLogLog [82] where both sketches are exactly the same except that LogLog uses geometric mean for estimation and HyperLogLog uses harmonic mean. This difference can only be told if random oracles are assumed so that there is a unique final distribution for the error to be measured.

Reality Empirically, experiments on real-world data with real-world hash functions agree with the prediction in the random oracle model [83, 95]. The random oracle model is

[4]For example, (normalized) sum of pairwisely independent and identically distributed random variables can converge to a Gaussian distribution, a chi-squared distribution, or even *converge to 0 in probability*, depending on their joint distribution [15]. In contrast, with i.i.d. random variables, the sum has to go to a Gaussian by the central limit theorem.

thus *useful* in the perspective of predicting an algorithm's real-world performance.

> The only assumption made is that records can be hashed in a suitably pseudo-uniform manner. This does not however appear to be a severe limitation since empirical studies on large industrial files reveal that careful implementations of standard hashing techniques do achieve practically uniformity of hashed values. Philippe Flajolet and G. Nigel Martin [83], 1985

Another reason is that, in practice, the hash function is often provided by the system and there are usually a large number of sketches sharing the same hash function in the context. In fact, in order for the sketches to be mergeable, they *have to* share one single hash function. Thus even if a non-trivial amount of memory is required for a good enough hash function, the space is amortized across all the sketches using it. In this perspective, the random oracle model assumes there is an infinite number of sketches sharing one hash function and thus zero space *per sketch* for hashing is used. The standard model on the other hand assumes there is only one sketch that is responsible for storing the entire hash function, which is usually not the case.

> Here we follow [83, 82] and consider the hash representation to be provided by the platform, which is consistent with practice where hash functions are reused and shared by multiple sketches. Edith Cohen [50], 2017

2.3 Modeling Queries

I will consider two types of queries—statistical queries and sampling queries. A statistical query q is a function from $M^{\mathbb{N}}$ to $\mathbb{C}$. A sampling query S is a randomized function from M^n to the index space $\mathbb{N}$.

Statistical queries To answer a statistical query q, one needs to equip a homomorphic sketch $\Psi^*\colon M^{\mathbb{N}} \to \Omega$ with an *estimator* $g\colon \Omega \to \mathbb{C}$ such that $g \circ \Psi^*$ approximates q. The most common type of statistical query is *f-moment*.

Definition 2.1 (f-moment)**.** Given a commutative monoid M, a state vector $\mathbf{x} \in M^{\mathbb{N}}$, and a function $f\colon M \to \mathbb{C}$ with $f(0_M) = 0$, the f-moment of the stream $\mathbf{x}$ is defined as

$$f(\mathbf{x}) = \sum_{v=1}^{\infty} f(\mathbf{x}(v)).$$

Remark 2.3. f-moment is well-defined since we assumed stream $\mathbf{x}$ has a finite support and $f(0_M) = 0$. The infinite summation on the right hand side is in fact always a finite sum.

Remark 2.4. In the f-moment case, I slightly abuse notation and write f for both the function on entries and the induced function on the whole vector. Later on, I will use different notations when I discuss more generic queries. For example, I write $f^*(\mathbf{x}) = \sum_{v=1}^{\infty} f^{(v)}(\mathbf{x}(v))$ where $f^{(v)}$ can be different from each other. I write $\Phi(\mathbf{x})$ for a generic query Φ that is imposed on the whole vector.

The f-moment is a *symmetric* query in the sense that the answer is insensitive to re-indexing the domain. For symmetric queries, it is natural to use i.i.d. maps $\Phi^{(v)}$ for all indices $v \in \mathbb{N}$. In such cases, I will simply write $\Psi(\mathbf{x}) = \biguplus_{v=1}^{\infty} \Psi^{(v)}(\mathbf{x}(v))$. See Fig. 2.1 for a summarization of the setup so far.

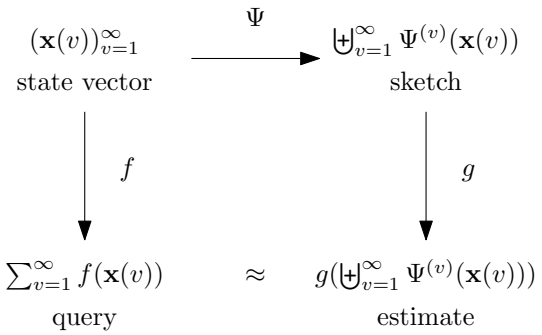


Figure 2.1: Diagram of homomorphic sketching of f-moments. For each element $v \in [n]$, $\mathbf{x}(v) \in M$ where $(M, +)$ is a commutative monoid and "+" defines the update/merge operation for the element state. Ψ is a random homomorphism from $(M, +)$ to $(\Omega, \uplus)$ where Ω is the memory space of the sketch and "$\uplus$" defines the update/merge operation for the sketch. $\Psi^{(v)}$s are i.i.d. copies of Ψ. $f : M \to \mathbb{C}$ is the estimation target. $g : \Omega \to \mathbb{C}$ is the estimator.

Let me briefly showcase the expressivity of the f-moment problem in a generic M-turnstile model.

- Counting the number of distinct elements over an insertion-stream corresponds to the f-moment estimation problem with $f(x) = \mathbb{1}[x \neq 0]$ in the $\mathbb{N}$-turnstile or $(\{0, 1\}, \mathrm{OR})$-turnstile model.
- If two streams $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}$ are present and one wants to know the number of elements that are non-zero in both streams (i.e., the union size). This corresponds to the cardinality estimation problem of the product stream $\mathbf{x} = ((\mathbf{x}^{(1)}(v), \mathbf{x}^{(2)}(v)))_{v=1}^{n}$ in the $\mathbb{N} \times \mathbb{N}$-turnstile model.

- Suppose now each entry $\mathbf{x}(v)$ is a point in $\mathbb{R}^2$, dynamically updated in the $\mathbb{R}^2$-turnstile model. Let $f(x,y) = \sqrt{x^2+y^2}$. Then the f-moment of this $\mathbb{R}^2$-stream is equal to $\sum_{v=1}^{n} |\mathbf{x}(v)|$, i.e., the sum of Euclidean distances of all points away from the origin.
- Continuing with the $\mathbb{R}^2$-turnstile example, set $f(x,y) = \sqrt{(x-x_0)^2+(y-y_0)^2} - \sqrt{x_0^2+y_0^2}$ for some point $z = (x_0, y_0) \in \mathbb{R}^2$. Then the f-moment now is $\sum_{v=1}^{n}(|\mathbf{x}(v) - z| - |z|)$, which will be considerably different based on the position of z.

Sampling queries To implement a sampling query S, one needs to equip the sketch Ψ^* with a *selector* $s : \Omega \to \mathbb{N}$ such that $s \circ \Psi^*(\mathbf{x}) \sim S(\mathbf{x})$ for any input stream $\mathbf{x}$. The composition $s \circ \Psi^*$ is called a *sampler*. In the literature, if $s \circ \Psi^* \sim S$, then $s \circ \Psi^*$ is called *truly perfect* [104]. Many approximate samplers [52, 102] for various sampling queries have been constructed in the literature, which only requires $s \circ \Psi^*$ to be statistically close to S. The most common type of sampling query is *G-sampling*.

Definition 2.2 (G-sampling)**.** Given a commutative monoid M, a state vector $\mathbf{x} \in M^{\mathbb{N}}$, and a function $G\colon M \to \mathbb{R}_+$ with $G(0_M) = 0$, the G-sampling of the stream $\mathbf{x}$ is a randomized choice function S_G such that for all $\mathbf{x} \in M^{\mathbb{N}}$ and $v \in \mathbb{N}$

$$\mathbb{P}(S_G(\mathbf{x}) = v) = \frac{G(\mathbf{x}(v))}{\sum_{r=1}^{\infty} G(\mathbf{x}(r))}.$$

Remark 2.5. Similar to the f-moment, G-sampling is well-defined since $\mathbf{x}$ has finite support and $G(0_M) = 0$. The assumption that G is non-negative ensures the formula above defines a discrete probability measure.

2.4 Homomorphic Sketching System

Finally, let me summarize the assumptions above to the following sketching system. For simplicity, I will assume here that the elements' maps are i.i.d., as most of the time I will consider symmetric queries such as f-moment estimation and G-sampling.

Definition 2.3 (homomorphic sketching system)**.** A *homomorphic sketching system* is a triplet $((M,+),(\Omega,\uplus),(\Psi,\mu_\Psi))$ where

- $(M,+)$ is the *element state space*, which is a commutative monoid,
- $(\Omega,\uplus)$ is the *sketch state space*, which is also a commutative monoid.

- (Ψ, μ_Ψ) is the *random-morphism*, which is a random homomorphism from $(M, +)$ to $(\Omega, \uplus)$ with distribution μ_Ψ.

Let $\mathbf{x} \in M^n$ be the *state vector* where n is the size of the *universe* $[n] = \{1, \ldots, n\}$. For any *element* $v \in [n]$, let $\Psi^{(v)}$ be an i.i.d. copy of Ψ indexed by v. The *current sketch state* is $\biguplus_{v \in [n]} \Psi^{(v)}(\mathbf{x}(u))$.

CHAPTER 3

Infinitely Divisible Sketches

> It seems to me that my chosen field, computer science—information processing—is a field where theory and practice come together more than in any other discipline, because of the nature of computing machines.
>
> Donald E. Knuth [114], 1991

In this chapter, I am going to present a wide range of sketches for various estimation/sampling queries in various streaming models. All of them share a common property—they are *infinitely divisible* (Definitions 3.1 and 3.2). Before jumping into the technical results, let me first discuss how infinite divisibility naturally arises from two properties of streaming sketching—allowing fractional elements and being scale-invariant.

3.1 Prelude: Infinite Divisibility of Sketches

3.1.1 Allowing Fractional Elements

Back in 1983, when Flajolet and Martin were analyzing the behavior of their PCSA sketch, they imagined[1] *poissonizing* the stream, which replaces each insertion with Poisson(1) insertions. They take advantage of the following important property of the Poisson distribution.

Fact (poissonized bins-and-balls)**.** Let there be m bins and Poisson(λ) balls. If each ball is independently put into a uniformly random bin, then the number of balls in the bins are *i.i.d.* Poisson(λ/m).

[1]They call this trick poissonization-depoissonization, i.e., one first poissonizes the stream and obtain a precise analysis (poissonization), and then one argues that the "distance" between the poissonized sketch and the true sketch is small (depoissonization).

In comparison with the usual bins-and-balls where the bins' contents are correlated, the bins in the poissonized version are independent, which simplifies the analysis significantly. The PCSA sketch is essentially a bins-and-balls experiment with non-uniform assignment distribution and thus its analysis can be simplified if one assumes the stream is poissonized.

Unintendedly, this trick naturally allows a cardinality sketch to count *fractional elements* in the streams, e.g., some of elements are considered "minor" and only count as a half in the total cardinality. This is called *weighted cardinality estimation* by Cohen [53] and has applications in databases. Indeed, in a poissonized stream, each fresh insertion in the stream is simulated by Poisson(1) actual insertions to the sketch. Therefore, a "half insertion" can be naturally simulated by Poisson(1/2) actual insertions to the sketch. By the property of Poisson distribution, the final distribution of the sketch will not depend on how the weights are distributed, as

$$\mathrm{Poisson}(x_1) + \cdots + \mathrm{Poisson}(x_k) \sim \mathrm{Poisson}(y_1) + \cdots + \mathrm{Poisson}(y_l)$$

if and only if

$$x_1 + \cdots + x_k \sim y_1 + \cdots + y_l.$$

Let us now generalize the property of "allowing fractional elements" to a generic homomorphic sketching system (M, Ω, Ψ) (Definition 2.3). Let the current sketch state be $S \in \Omega$. By definition, if an element v with value y is inserted, the state should be updated to

$$S \leftarrow S \uplus \Psi^{(v)}(y). \qquad \text{(on an insertion)}$$

A half-element can in general be defined in a descriptive way—insertions of two half-elements with the same value y and a single element with value y should update the sketch in an identical way. Similarly one may define $\frac{1}{k}$-elements where inserting k $\frac{1}{k}$-elements has the same effect of inserting one ordinary element. Let us rephrase this mathematically.

Definition 3.1 (allow fractional elements)**.** A (M, Ω, Ψ) sketch *allows fractional elements* if for any $k \in \mathbb{Z}_+$, there exists a random homomorphism $\Psi^{(1/k)}$ such that for any $y \in M$,

$$\Psi(y) \sim \Psi^{(1,1/k)}(y) \uplus \Psi^{(2,1/k)}(y) \uplus \cdots \uplus \Psi^{(k,1/k)}(y),$$

where $\Psi^{(j,1/k)}$s are i.i.d. copies of $\Psi^{(1/k)}$.

Now, upon inserting a half-element v with value y, the state can be updated to

$$S \leftarrow S \uplus \Psi^{(v,1/2)}(y). \qquad \text{(on a } \textit{half}\text{-insertion)}$$

This clearly captures the descriptive definition. For example, let u, v be half-elements and w be an ordinary element. Then by construction we have

$$\Psi^{(u,1/2)}(y) \uplus \Psi^{(v,1/2)}(y) \sim \Psi^{(w)}(y).$$
(two half-insertions are equivalent to one ordinary insertion)

Note that, Definition 3.1 is exactly the mathematical definition of a random object being infinitely divisible. See [113].

Definition 3.2 (infinite divisible sketches)**.** A (M, Ω, Ψ) sketch is infinite divisible if it allows fractional elements (Definition 3.1).

3.1.2 Scale-invariance

Allowing fractional elements may seem like a small feature for streaming sketches which is not generally necessary. In this part, I will argue that a much more important property of sketches—being scale-invariant—will also imply infinite divisibility.

Scale-invariance, roughly speaking, is the property that a streaming sketch should behave *relatively identically* when the support size is scaled up. See Table 3.1 for intuition. Descrip-

Stream	State Vector	Answering F_0	Answering F_2
A	$(\underbrace{1,\ldots,1}_{100},\underbrace{2,\ldots,2}_{30},\underbrace{3,\ldots,3}_{50},0,\ldots)$	180 ± 20	670 ± 80
B	$(\underbrace{1,\ldots,1}_{1000},\underbrace{2,\ldots,2}_{300},\underbrace{3,\ldots,3}_{500},0,\ldots)$	1800 ± 200	6700 ± 800

Table 3.1: Intuition for scale-invariance. Stream B has the identical frequency distribution with stream A except for having a ten times larger support. A "reasonable" streaming sketch should have "relatively invariant" final distributions with input A and B, with the same relative error distribution for answering *all* queries. For example, if the sketch estimates F_0 of A to be between 180 ± 20, it should estimate F_0 of B to be between 1800 ± 200—exactly the same distribution with the prior case except for multiplying both the average and deviation by ten.

tively, one wants a scale-invariant sketch to have the same relative error for answering *any* one query at *all* scales. This is, however, a property that is subtle to define formally. For

simplicity, I define a homomorphic sketch to be *base-2 scale-invariant*, if the sketch state is *isomorphic* to the sketch over a stream with double support size.

Definition 3.3 (base-2 scale-invariance)**.** Let (M, Ω, Ψ) be a homomorphic sketch. For any input stream if the current state is S, then after the support is doubled, i.e., after the input stream is repeated exactly in a disjoint domain, the state becomes $S \uplus S'$, where S' is an i.i.d. copy of S. We say the sketch is base-2 scale-invariant if there exists an automorphism $\phi : \Omega \to \Omega$ for any input stream,

$$S \uplus S' \sim \phi(S). \qquad \text{(base-2 scale-invariance)}$$

The base-2 scale-invariance captures the descriptive definition of scale-invariance, if the estimator is *eigen* to ϕ.

Definition 3.4 (eigen-estimator)**.** Let (M, Ω, Ψ) be a base-2 scale-invariant sketch with automorphism ϕ. A function $g : \Omega \to \mathbb{R}$ is an *eigen-estimator* if

$$g \circ \phi = 2g.$$

Indeed, after doubling support, the estimator is also exactly doubled in distribution, so do the average and deviation.

$$\begin{aligned} \overbrace{g(S \uplus S')}^{\text{doubling support}} &\sim g(\phi(S)) && \text{(base-2 scale-invariance)} \\ &= \underbrace{2g(S)}_{\text{doubling estimate}} && \text{(eigen-estimator)} \end{aligned}$$

In fact, most existing sketches essentially satisfy the base-2 scale-invariance and their estimators are eigen as well. Let us now see, why base-2 scale-invariance will imply the infinite divisibility of the sketch. Let Ψ be base-2 scale-invariant. Then for any value $y \in M$, one has

$$\begin{aligned} \Psi(y) &\sim \phi^{-1}(\Psi^{\mathtt{0}}(y) \uplus \Psi^{\mathtt{1}}(y)) && \text{(base-2 scale-invariance)} \\ &\sim \phi^{-1}(\Psi^{\mathtt{0}}(y)) \uplus \phi^{-1}(\Psi^{\mathtt{1}}(y)) && (\phi \text{ is a morphism}) \end{aligned}$$

repeat the above steps for $\Psi^{\mathtt{0}}$ and $\Psi^{\mathtt{1}}$

$$\begin{aligned} &\sim \phi^{-1}(\phi^{-1}(\Psi^{\mathtt{00}}(y)) \uplus \phi^{-1}(\Psi^{\mathtt{01}}(y))) \uplus \phi^{-1}(\phi^{-1}(\Psi^{\mathtt{10}}(y)) \uplus \phi^{-1}(\Psi^{\mathtt{11}}(y))) \\ &\sim \phi^{-2}(\Psi^{\mathtt{00}}(y)) \uplus \phi^{-2}(\Psi^{\mathtt{01}}(y)) \uplus \phi^{-2}(\Psi^{\mathtt{10}}(y)) \uplus \phi^{-2}(\Psi^{\mathtt{11}}(y)) \end{aligned}$$

$$\begin{aligned}
&\dots && \text{(repeat } k \text{ times)}\\
&\sim \biguplus_{w\in\{0,1\}^k} \phi^{-k}(\Psi^w(y)),
\end{aligned}$$

where k can be arbitrarily large. Note that $\phi^{-k}(\Psi^w(y))$ for $w \in \{0,1\}^k$ are 2^k i.i.d. random states that merge to $\Psi(y)$, and therefore $\Psi(y)$ is infinitely divisible.[2]

The observation above implies an interesting way to handle half-elements for a base-2 scale-invariant sketch. Let us say a half-element v with value $y \in M$ is inserted to a sketch with current state S. Then one only needs to update

$$S \leftarrow S \uplus \phi^{-1}(\Psi^{(v)}(y)).$$

Two half-elements u, v will have the same effect as an ordinary element since

$$\phi^{-1}(\Psi^{(u)}(y)) \uplus \phi^{-1}(\Psi^{(v)}(y)) = \phi^{-1}(\Psi^{(u)}(y) \uplus \Psi^{(v)}(y)) \sim \Psi^{(u)}(y),$$

by the fact that ϕ is a morphism and Ψ is base-2 scale-invariant.

Remark 3.1. There has been a series of definitions of "scale-invariance" during my investigation of cardinality estimation in order to make certain concepts well-defined. In [137] (Section 5.2), a sketch is scale-invariant if the entropy and normalized Fisher information are invariant at different scales. In [140] (Section 5.7), a sketch is scale-invariant if the normalized remaining area has a limit when $\lambda \to \infty$ (i.e., asymptotically scale-invariant). In [160] (Section 5.8), a sketch is scale-invariant if the distributions of the normalized τ-generalized remaining area are invariant at different scales. The base-2 scale-invariance defined in Definition 3.3 is a strong version which requires the sketch to be *isomorphic* at different scales. All the previous definitions in [137, 140, 160] of scale-invariance can be implied by the base-2 scale-invariance above in Definition 3.3.

3.2 Overview

In Section 3.1 above, I demonstrate that the infinite divisibility of the sketch Ψ is a hidden but important requirement for any reasonable (scale-invariant) streaming sketch. For the

[2]There is a technical subtlety here that in Definition 3.1, one needs $\Psi(y)$ to be the convolution of r i.i.d. random variables for any $r \in \mathbb{Z}_+$, while from base-2 scale-invariance, one only has this property for $r = 2^k$ being a power of 2. It is routine to show that they are equivalent in common settings of M and Ω, which mimics the procedure of representing $1/r$ as a binary fraction. Alternatively, one may require base-r scale-invariance here which will directly induce such a convolution.

rest of this chapter, I will go in the opposite direction, start with infinitely divisible random objects, more specifically the *Lévy processes*, and construct a wide range of new sketches.

In addition to new sketches, this Lévy perspective also reconstructs many classic sketches, providing a unified view of sketch design. For example, from an algorithmic point of view, the poissonization trick in the analysis of PCSA [83] seems to be irrelevant with the trick of using p-stable random variables in Indyk's L_p-stable sketches [96]. However, the two tricks are actually deeply connected in terms of infinite divisibility—the first trick can be reconstructed by sampling the Poisson counting process and the second trick can be reconstructed by sampling the p-stable process.

Lévy processes have been studied since the early 20th century, and have been used to model phenomena in various fields, e.g., physics (how does a gas particle move?) and finance (how does the stock price change?). The construction in this chapter shows that all Lévy processes also have *algorithmic* interpretations in the context of streaming sketches, and the fundamental *Lévy-Khintchine representation theorem* can now be used to understand streaming computation.

Organization We first review Lévy processes and the Lévy-Khintchine representation theorem in Section 3.3. We prove the two main theorems in Section 3.4 and Section 3.5 that connect the Lévy-Khintchine theorem to streaming sketches. Specifically, in Section 3.4 we construct the Lévy-Tower sketch, which transforms any generic Lévy process X into an f_X-moment sketch, where f_X is the characteristic exponent of X. In Section 3.5 we construct the Lévy-Min-Sampler, which can be parameterized to sample elements according to any weight function G, where G is the Laplace exponent of a subordinator.

In Section 3.6, we use the G-transformation technique to reduce the problem of G-moment estimation to cardinality estimation; this leads to the LévyPCSA, LévyHyperLogLog, and Stable-HyperLogLog sketches. In Section 3.7, we discuss the connection between previous sketches and Lévy processes in a greater detail. In Section 3.8, we discuss the problem of characterizing the set of *tractable* functions, and describe the *Fourier-Hahn-Lévy* method for expanding the range of the Lévy-Tower beyond Lévy-Khintchinerepresentable functions. We conclude in Section 3.9 with some conjectures.

3.3 Preliminaries: Lévy Processes

I use Ken-Iti Sato's text [113] as my reference for the theory of Lévy processes. See REMARKS ON NOTATIONS in the front of the thesis for choices of mathematical symbols and conventions.

3.3.1 Lévy Processes on $\mathbb{R}^d$

Definition 3.5 (Lévy processes [113, page 3])**.** A random process $X = (X_t)_{t\in\mathbb{R}_+}$ on $\mathbb{R}^d$ is a *Lévy process* if

(A) $X_{t+s} - X_t \sim X_s$ for all $t, s \in \mathbb{R}_+$ (stationary increments);

(B) for $0 \le t_1 < t_2 \ldots < t_k$, $X_{t_1}, X_{t_2} - X_{t_1}, \ldots, X_{t_k} - X_{t_{k-1}}$ are mutually independent (independent increments);

(C) $X_0 = 0$ almost surely and $\lim_{t\searrow 0} \mathbb{P}(|X_t| > \epsilon) = 0$ for any $\epsilon > 0$ (stochastic continuity).

A process that satisfies (A) and (B) is called *memoryless*, i.e., $(X_t)_{t>t^*}$ is independent of $(X_t)_{t<t^*}$. The primary way to study Lévy processes is through their *characteristic functions*.

Theorem 3.1 (Lévy-Khintchine representation [113, page 37])**.** *Any Lévy process $X = (X_t)_{t\in\mathbb{R}_+}$ on $\mathbb{R}^d$ can be identified by a triplet (A, ν, γ) where A is a covariance matrix, ν is a measure on $\mathbb{R}$ such that*

$$\nu(\{0\}) = 0 \quad \textit{and} \quad \int_{\mathbb{R}^d} \min(|s|^2, 1)\, \nu(ds) < \infty, \tag{3.1}$$

and $\gamma \in \mathbb{R}^d$. The identification is through the characteristic function. For any $t \in \mathbb{R}_+$ and $z \in \mathbb{R}^d$,

$$\mathbb{E}e^{i\langle X_t, z\rangle} = \exp\left(-t\left(\frac{1}{2}\langle z, Az\rangle - i\langle \gamma, z\rangle + \int_{\mathbb{R}^d} (1 + i\langle z, s\rangle \mathbb{1}\left[|s| < 1\right] - e^{i\langle z,s\rangle})\, \nu(ds)\right)\right). \tag{3.2}$$

Conversely, any triplet (A, ν, γ) where A is a covariance matrix, ν satisfies (3.1), and $\gamma \in \mathbb{R}^d$ corresponds to a Lévy process satisfying (3.2).

Remark 3.2. In the one-dimensional case, the matrix A can be identified by the usual variance σ^2. Then we have for $z \in \mathbb{R}$,

$$\mathbb{E}e^{izX_t} = \exp\left(-t\left(\frac{1}{2}\sigma^2 z^2 - i\gamma z + \int_{\mathbb{R}} (1 + izs\mathbb{1}\left[|s| < 1\right] - e^{izs})\, \nu(ds)\right)\right). \tag{3.3}$$

We call the exponent in (3.2) the *characteristic exponent*, denoted by $f_X(z) = -\log \mathbb{E}e^{i\langle X_1, z\rangle}$. We note some properties of the characteristic exponent f.

Lemma 3.1. *Let X be any Lévy process on $\mathbb{R}^d$ and f be its characteristic exponent.*

- $\mathfrak{Re} f \geq 0$. $\mathfrak{Re} f = 0$ *if and only if X is a deterministic drift, i.e., $X_t = \gamma t$ for all $t \in \mathbb{R}_+$.*
- *For any $x \in \mathbb{R}^d$, $f(-x) = \overline{f(x)}$, the complex conjugate of $f(x)$.*

We list some common one-dimensional Lévy processes in Table 3.2 that will be frequently used later.

Process	Characteristic Exponent	Notes
Linear drift	$i\gamma z$	$\gamma \in \mathbb{R}_+$
α-stable	$\lvert z\rvert^\alpha$	$\alpha \in (0, 2]$
Poisson	$1 - e^{iz}$	
Compound Poisson	$\int_{-\infty}^{\infty}(1 - e^{izs})\,\nu(ds)$	$\nu(\mathbb{R}) < \infty$
Symmetric Compound Poisson	$\int_0^{\infty}(1 - \cos(zs))\,\nu(ds)$	$\nu(\mathbb{R}) < \infty$

Table 3.2: Common one-dimensional Lévy processes.

3.3.2 Subordinators (Lévy Processes on $[0, \infty]$)

One-dimensional, non-negative Lévy processes are called *subordinators*. Subordinators are of special use in the study of Lévy processes and also in streaming sketches. They can be characterized via their *Laplace transforms*.

Theorem 3.2 (Lévy-Khintchine for subordinators [113, page 138])**.** *Any subordinator $X = (X_t)_{t\in\mathbb{R}_+}$ on $[0, \infty]$ can be identified by a triplet (c, ν, γ_0) where $c, \gamma_0 \geq 0$, and $\int_0^\infty \min(s, 1)\,\nu(ds) < \infty$, such that for any $t, z \in \mathbb{R}_+$,*

$$\mathbb{E}e^{-zX_t} = \exp\left(-t\left(c\mathbb{1}\left[z \neq 0\right] + \gamma_0 z + \int_0^\infty (1 - e^{-zs})\,\nu(ds)\right)\right). \tag{3.4}$$

Conversely, given any $c, \gamma_0 \geq 0$, and measure ν such that $\int_0^\infty \min(s, 1)\,\nu(ds) < \infty$, there is a corresponding subordinator satisfying (3.4).

The exponent in (3.4) is called the *Laplace exponent* (Lap. exp.), denoted as $G_X(z) = -\log \mathbb{E}e^{-zX_1}$. Subordinators are necessarily non-decreasing. The parameter c is called the

kill rate. Ken-Iti's text [113] only considers processes with $c = 0$. However, we do want to consider killed processes (those with $c > 0$) here because, as we will see later, popular sketches like PCSA and HyperLogLog are induced by killed processes. There is a standard way to "kill" a process.

Lemma 3.2 (kill a Lévy process; see [169])**.** *Given a subordinator X with Laplace exponent $G : \mathbb{R}_+ \to \mathbb{R}_+$ and $c > 0$, let Y be an independent* $\mathrm{Exp}(c)$ *random variable and define the process X' as*

$$X'_t = \begin{cases} X_t, & t < Y, \\ \infty, & t \geq Y. \end{cases}$$

Then for $z \in \mathbb{R}$,

$$\mathbb{E}e^{-zX'_t} = e^{-t(G(z)+c\mathbb{1}[z\neq 0])},$$

where we define $e^{-z\infty} = \mathbb{1}[z = 0]$.

As its name suggests, subordinators subordinate Lévy processes.

Theorem 3.3 (subordination [113, page 198])**.** *Let $X = (X_t)$ be a Lévy process on $\mathbb{R}^d$ and $Z = (Z_t)_{t\in\mathbb{R}_+}$ be a subordinator with $c = 0$. Then $(X_{Z_t})_{t\in\mathbb{R}_+}$ is a Lévy process on $\mathbb{R}^d$ such that for any $t \in \mathbb{R}_+$ and $z \in \mathbb{R}^d$,*

$$\mathbb{E}e^{i\langle X_{Z_t}, z\rangle} = \exp(-tG_Z(f_X(z))),$$

where f_X is the characteristic exponent of X and G_Z is the Laplace exponent of Z.

3.4 Estimating f-Moments with Lévy Processes

I will first present how Lévy processes can be used for estimating f-moments over streams.

Problem 1 (f-moment estimation in the $\mathbb{R}^d$-turnstile model)**.** Fix $d \in \mathbb{Z}_+$, let $f : \mathbb{R}^d \to \mathbb{C}$ be a function with $f(0) = 0$, and let $\mathbf{x} \in (\mathbb{R}^d)^n$ be the vector subject to streaming updates to its coordinates. The estimation target is the *f-moment* $f(\mathbf{x})$, where

$$f(\mathbf{x}) \stackrel{\text{def}}{=} \sum_{v\in[n]} f(\mathbf{x}(v)).$$

Literatures on the this problem have been surveyed in Section 1.3. Here let me give a brief summary.

3.4.1 Summary of literature

f-moment estimation has been mainly studied in the one-dimensional streams, i.e., in the $\mathbb{R}$- and $\mathbb{Z}$-turnstile models. Alon, Matias, and Szegedy [5] estimate the L_2-moment ($f(x) = |x|^2$) with a data structure now commonly known as the AMS sketch. Indyk [96] designed a class of sketches for estimating the L_p moments, with $p \in (0, 2]$. When $p > 2$, Bar-Yossef, Jayram, Kumar, and Sivakumar [8] proved that estimating the L_p-moment requires $\Omega(\text{poly}(n))$ space. The L_0-moment ($f(x) = \mathbb{1}[x \neq 0]$) has been estimated in *two* distinct ways. Cormode, Data, Indyk, and Muthukrishnan [59] approximate L_0 by L_α with very small $\alpha > 0$. Kane, Nelson, and Woodruff [110] project each element onto $\mathbb{Z}_p$, $p > \epsilon^{-1} \log n$ being a random prime, which effectively reduces L_0-estimation to cardinality estimation (F_0-estimation).

In the $\mathbb{R}^d$-turnstile model, the $L_{p,q}$ hybrid moment is defined by $f(x_1, \ldots, x_d) = (\sum_{j=1}^d |x_j|^p)^q$. Ganguly, Bansal, and Dube [88] approximate $L_{p,q}$ for $p \in [0, 2], q \in [0, 1]$.[3]

Apart from the frequency moments L_p and and the hybrid moment $L_{p,q}$, a large class of other f-moments can be universally estimated through the subsample-and-CountSketch techniques developed in [23, 25, 19].

3.4.2 Lévy-Tower

Now I demonstrate how Lévy processes can be used to store information over streams. Given any Lévy process $X = (X_t)_{t \in \mathbb{R}_+}$ on $\mathbb{R}^d$, by the Lévy-Khintchine representation theorem (Theorem 3.1), there exists a function $f_X : \mathbb{R}^d \to \mathbb{C}$ such that for any $t \in \mathbb{R}_+$ and $z \in \mathbb{R}^d$, $\mathbb{E}e^{i\langle z, X_t\rangle} = e^{-tf_X(z)}$. The *Lévy projection at time t* induced by X is defined as

$$\Psi_t(x) = \langle x, X_t \rangle.$$

Ψ_t is clearly homomorphic from $\mathbb{R}^d$ to $\mathbb{R}$ and t is the time at which the process is sampled. By Definition 2.3, after reading a stream $\mathbf{x} \in (\mathbb{R}^d)^n$, the current sketch state is

$$C_t = \biguplus_{v=1}^{n} \Psi_t^{(v)}(\mathbf{x}(v)) \tag{3.5}$$

[3]For consistency, we call the hybrid moment $L_{p,q}$ to emphasize that it is in the turnstile model. It has been called the $F_{p,q}$ hybrid moment in [88] and in Section 1.3.

$$= \sum_{v=1}^{n} \langle \mathbf{x}(v), X_t^{(v)} \rangle. \qquad \text{(merge "}\uplus\text{" is addition in } (\mathbb{R}, +))$$

Here the $X_t^{(v)}$ are i.i.d. copies of X_t. To see how information about the stream $\mathbf{x}$ can be maintained in the sketch C_t, note that

$$\begin{aligned}
\mathbb{E}e^{iC_t} &= \mathbb{E}e^{i\sum_{v\in[n]}\left\langle X_t^{(v)}, \mathbf{x}(v)\right\rangle} && (3.6)\\
&= \prod_{v\in[n]} \mathbb{E}e^{i\left\langle X_t^{(v)}, \mathbf{x}(v)\right\rangle} && \text{(by independence)}\\
&= \prod_{v\in[n]} e^{-tf_X(\mathbf{x}(v))} && \text{(by Lévy-Khintchine)}\\
&= e^{-tf_X(\mathbf{x})}, && \text{(recall that } f(\mathbf{x}) = \textstyle\sum_{v\in[n]} f(\mathbf{x}(v)))
\end{aligned}$$

where the f_X-moment $f_X(\mathbf{x})$ lies right in the exponent. $f_X(\mathbf{x})$ can be recovered by choosing a suitable time $t \approx \Theta(1/|f(\mathbf{x})|)$. Of course, we do not always know a suitable t, and therefore maintain samples for many t, evenly spaced on a logarithmic scale. This observation lets us to convert any Lévy process with characteristic f to a streaming sketch that estimates the f-moment.

An interesting twist here is that, in the above computation one in fact only needs to store the C_t *modulo* 2π as $e^{i2\pi} = 1$. The the sketch state space of C_t can be wrapped in $(\mathbb{T}, +)$. Thus while nearly all sketches in the turnstile model is *linear*, i.e., maps from $\mathbb{R}^d$ to $\mathbb{R}$, the sketch I am going to construct here this is only *additive*, $\mathbb{R}^d$ to $\mathbb{T}$. The sketch can be visualized by a grid of clocks (Fig. 3.1).

Now let me define this generic sketch.

Definition 3.6 ((f, m)-Lévy-Tower)**.** Let f be the characteristic exponent of a Lévy process X on $\mathbb{R}^d$ and let $m \in \mathbb{Z}_+$ be the number sub-sketches. An (f, m)-*Lévy-Tower* is an infinite vector $S = (S_k^{(j)})_{k\in\mathbb{Z}, j\in[m]} \subset \mathbb{T}^{\mathbb{Z}}$, initialized as all zero. For any element $v \in [n]$ and $y \in \mathbb{R}^d$, a vector update $\mathbf{x}(v) \leftarrow \mathbf{x}(v) + y$ is effected by:

Update(v, y) : For each $k \in \mathbb{Z}$ and $j \in [m]$, $S_k^{(j)} \leftarrow S_k^{(j)} + \left\langle y, X_{2^{-k}}^{(v,j)} \right\rangle \bmod 2\pi$, where $X^{(v,j)} = (X_t^{(v,j)})_{t\in\mathbb{R}_+}$ is an i.i.d. copy of the Lévy process X with characteristic exponent f.

Remark 3.3. If $|f(\mathbf{x})|$ is in the range $[1, \mathrm{poly}(n)]$, then it suffices to store levels $k \in [0, O(\log n)]$. Therefore a Lévy-Tower takes $O(m \log n)$ words of space and each word stores a number in $[0, 2\pi)$.

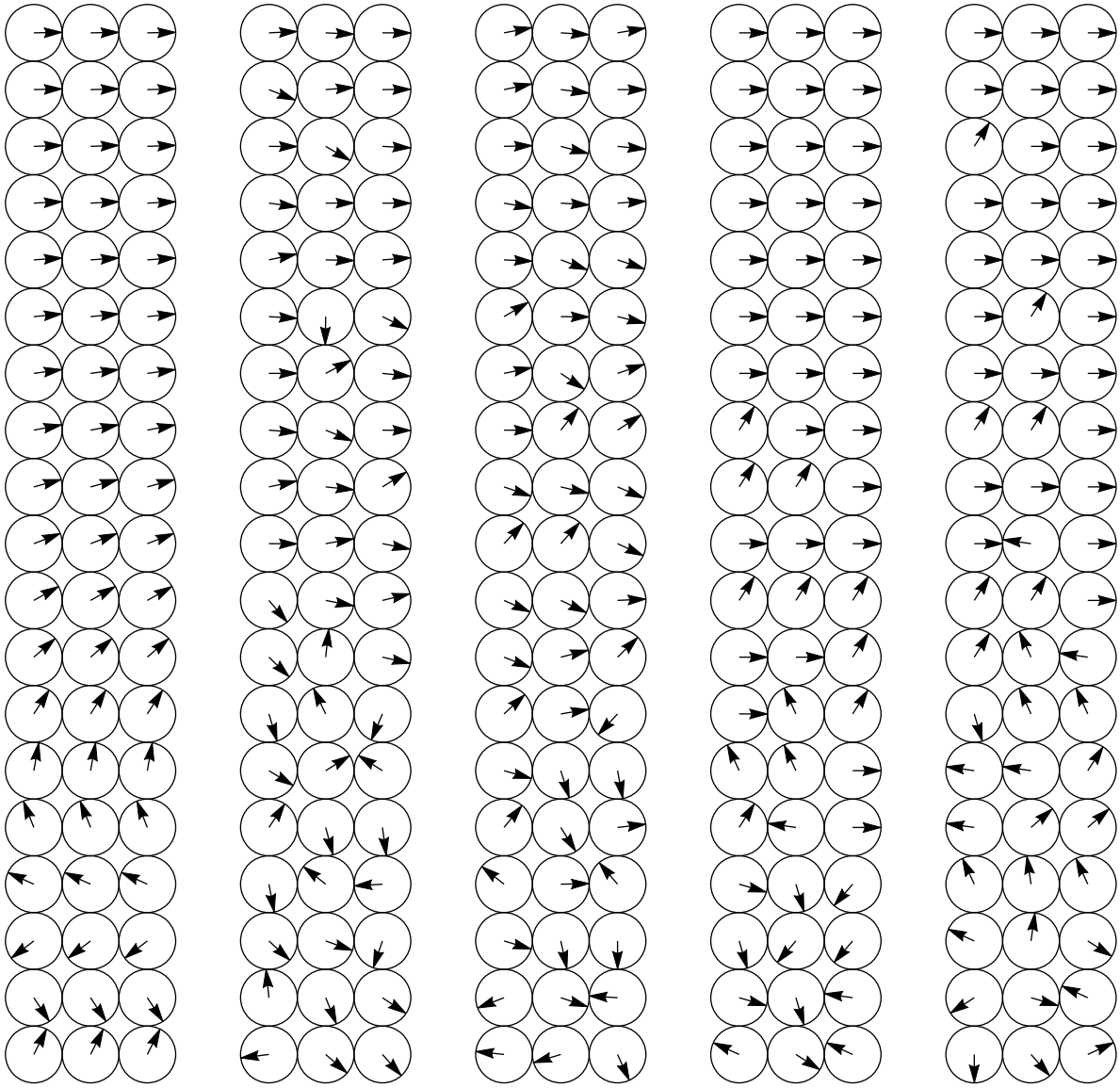


Figure 3.1: Lévy-Tower with $m = 3$ and $\mathbf{x} = (1, 0, \ldots, 0)$ (source of the frontispiece image). From left to right: linear drift, Cauchy process, Wiener process/Brownian motion, Poisson process with rate 1, Poisson process with rate 2. Different Lévy processes have different "sensitivities" for a target function-moment. For example, linear drift is only sensitive to the sum of the vector and insensitive to how the values are distributed. The Cauchy process is only sensitive to the L_1-moment, while the Wiener process/Brownian motion is only sensitive to the L_2-moment. Poisson processes are sensitive to the *support size* of $\mathbf{x}$ and at the same time leaking information about other f-moments.

Remark 3.4. The construction is actually standard in sketch design [80, 5, 96, 110, 23]: subsample, randomly project, and then sum, but with an important and powerful tweak that the two steps "subsample and random project" are replaced by a single step: sampling at time 2^{-k} of the Lévy process whose characteristic exponent is f. Consequently, such sketches can be used to estimate f-moments.

We characterize the distribution of the sketch as follows.

Lemma 3.3. *Fix a frequency vector $\mathbf{x} = (\mathbf{x}(v))_{v\in[n]} \in (\mathbb{R}^d)^n$. For any $k \in \mathbb{Z}$, we have*

$$S_k \sim \sum_{v\in[n]} \left\langle X^{(v)}_{2^{-k}}, \mathbf{x}(v) \right\rangle$$
$$\mathbb{E}e^{iS_k} = e^{-2^{-k} f(\mathbf{x})}.$$

Proof. Note that S_k is simply $C_{2^{-k}}$, i.e., Lévy projection at time $t = 2^{-k}$. Thus the former claim follows from Eq. (3.5) and the latter claim follows from Eq. (3.6). □

Note that the f-moment lies exactly in the exponent of $\mathbb{E}e^{iS_k}$. We now formally present the estimation method.

Estimation of Lévy-Tower Recall from previous discussion that the estimator must pick a suitable t and infer an estimate of the f-moment from the Lévy projection C_t. In the Lévy-Tower, we store $S_k = C_{2^{-k}}$ for all $k \in \mathbb{Z}$. We first prove the concentration of the empirical mean at each level.

Lemma 3.4. *For any $t \in \mathbb{R}_+$,*

$$\mathbb{P}\left(\left| \frac{1}{m} \sum_{j=1}^{m} e^{iC_t^{(j)}} - e^{-tf(\mathbf{x})} \right| > \eta \right) \le \frac{2t|f(\mathbf{x})|}{m\eta^2}.$$

Proof. By Chebyshev's inequality, we have

$$\mathbb{P}\left(\left| \frac{1}{m} \sum_{j=1}^{m} e^{iC_t^{(j)}} - e^{-tf(\mathbf{x})} \right| > \eta \right) \le \frac{\mathbb{V}e^{iC_t}}{m\eta^2}.$$

Furthermore, since $\mathbb{E}e^{iC_t} = e^{-tf(\mathbf{x})}$, we have

$$\mathbb{V}e^{iC_t} = \mathbb{E}|e^{iC_t}|^2 - |\mathbb{E}e^{iC_t}|^2 = 1 - e^{-2t\mathfrak{Re}f(\mathbf{x})} \le 2t\mathfrak{Re}f(\mathbf{x}) \le 2t|f(\mathbf{x})|.$$

□

We then prove that the empirical mean has small deviation for all $k \geq \log_2 |f(\mathbf{x})|$ simultaneously.

Lemma 3.5. *With probability at least* 99/100*, for any* $k \geq \log_2 |f(\mathbf{x})|$*, we have*

$$\left| \frac{1}{m} \sum_{j=1}^{m} e^{iS_k^{(j)}} - e^{-2^{-k} f(\mathbf{x})} \right| \leq O(1/\sqrt{m}).$$

Proof. By the union bound and Lemma 3.4,

$$\begin{aligned}
&\mathbb{P}\left(\exists k \geq \log_2 |f(\mathbf{x})|, \left| \frac{1}{m} \sum_{j=1}^{m} e^{iS_k^{(j)}} - e^{-2^{-k} f(\mathbf{x})} \right| > \eta \right) \\
&\leq \sum_{k=\log_2 |f(\mathbf{x})|}^{\infty} \frac{2 \cdot 2^{-k} |f(\mathbf{x})|}{m\eta^2} \\
&= O\left(\frac{1}{m\eta^2} \right).
\end{aligned}$$

Thus it suffices to choose $\eta = O(1/\sqrt{m})$ for the probability above at most 1/100. □

We consider the complex logarithm defined on $\mathbb{C}$ except for the negative real line. Note the following facts.

Lemma 3.6. *Let* $z \in \mathbb{C}$ *with* $|z| \leq 1$ *and* $\mathfrak{Re}(z) \geq 0$*, then* $|1 - e^{-z}| \in (|z|/2, |z|]$.

Lemma 3.7. *Let* $x, y \in \{z \in \mathbb{C} : |1 - z| \leq 0.85\}$*, then* $|\log x - \log y| < 7|x - y|$.

The following theorem formally proves that f-moment can be inferred from the Lévy tower, proving the claim in Theorem 1.1.

Theorem 3.4. *With probability at least 99/100, the following procedure returns an estimate* $\widehat{f(\mathbf{x})}$ *such that* $|\widehat{f(\mathbf{x})} - f(\mathbf{x})| \leq O(|f(\mathbf{x})|/\sqrt{m})$*. Let* K *be the maximum level and assume* $K \geq \log_2 |f(\mathbf{x})| \geq 0$*. Enumerate* $k = K, K-1, \ldots, 0$ *in decreasing order, and let* $Y_k = \frac{1}{m} \sum_{j=1}^{m} e^{iS_k^{(j)}}$*. If* $|1 - Y_k| > 0.2$*, then return* $-2^k \log Y_k$ *and* HALT.

Proof. By Lemma 3.5, with probability at least 99/100, we have $\left| Y_k - e^{-tf(\mathbf{x})} \right| \leq O(1/\sqrt{m})$ for all $k \in [\log_2 |f(\mathbf{x})|, K]$. We assume this event happens. For $k \in [\log_2 |f(\mathbf{x})|, K]$, we have $|2^{-k} f(\mathbf{x})| \leq 1$ and by Lemma 3.1 we know $\mathfrak{Re}(f(\mathbf{x})) \geq 0$. Thus we have by Lemma 3.6

$$|1 - Y_k| = |1 - e^{-2^{-k} f(\mathbf{x})}| + O(|e^{-2^{-k} f(\mathbf{x})} - Y_k|) \in |2^{-k} f(\mathbf{x})|(1/2, 1] + O(1/\sqrt{m}). \tag{3.7}$$

Let m be large enough that $O(1/\sqrt{m}) < 0.01$. Let W be the index selected by the procedure in the statement and we must have $W \geq \log_2 |f(\mathbf{x})|$ as $|1 - Y_{\lceil \log_2 |f(\mathbf{x})| \rceil}| > 0.24$ by Eq. (3.7). Since the times are base 2, one has

$$2^{-W} \in [0.19/|f(\mathbf{x})|, 0.84/|f(\mathbf{x})|]. \tag{3.8}$$

$2^{-W}|f(\mathbf{x})|$ cannot be smaller than 0.19 as otherwise Eq. (3.7) implies $|1 - Y_w| < 0.2$, contradicting the selection procedure. Also, $2^{-W}|f(\mathbf{x})|$ cannot be bigger than 0.84 because otherwise one will have $2^{-W-1}|f(\mathbf{x})| \geq 0.42$ and Eq. (3.7) implies $|1 - Y_{w-1}| > 0.2$, contradicting the stated procedure (should have picked some index $\leq W - 1$ instead). Note that $|1 - e^{-2^{-W} f(\mathbf{x})}| \leq |2^{-W} f(\mathbf{x})| \leq 0.84$ by Eq. (3.8) and by Eq. (3.7) we have $|1 - Y_W| \leq 0.85$. Apply Lemma 3.7 to Y_W and $e^{-2^{-W} f(\mathbf{x})}$, we have

$$|\log Y_W - (-2^{-W} f(\mathbf{x}))| \leq 7|Y_W - e^{-2^{-W} f(\mathbf{x})}| = O(1/\sqrt{m}).$$

We conclude that

$$|-2^W \log Y_W - f(\mathbf{x})| \leq O(2^W/\sqrt{m}) = O(|f(\mathbf{x})|/\sqrt{m}).$$

□

This proves the claim in Theorem 1.1.

3.4.3 Lévy-Stable

There is an important special case of the Lévy-Tower sketch above when the Lévy process X is *stable*. Recall A d-dimensional Lévy process X is called α-*stable* if for any $t \in \mathbb{R}_+$, $X_t \sim t^{1/\alpha} X_1$. If the Lévy process X in $C_t = \sum_{v \in [n]} \left\langle \mathbf{x}(v), X_t^{(v)} \right\rangle$ is α-stable, then we have

$$C_t = \sum_{v \in [n]} \left\langle \mathbf{x}(v), X_t^{(v)} \right\rangle \sim \sum_{v \in [n]} \left\langle \mathbf{x}(v), t^{1/\alpha} X_1^{(v)} \right\rangle \sim t^{1/\alpha} C_1.$$

Thus the Lévy-Tower induced by a stable process is *self-similar*: all registers in the tower are identically distributed with a proper normalization. Due to this self-similarity, it suffices to consider the projection only at time 1 and there is no need for using multiple time scales. Now let the characteristic exponent be $f_X(z)$.

Lemma 3.8. *$C_1 \sim f(\mathbf{x})^{1/\alpha} Y_\alpha$ where Y_α is a* one-*dimensional α-stable random variable with*

$\mathbb{E}e^{izY_\alpha} = e^{-|z|^\alpha}$.

Proof. We have the characteristic function

$$\mathbb{E}e^{izC_1} = \mathbb{E}e^{iz\sum_{v\in[n]}\left\langle \mathbf{x}(v), X_1^{(v)}\right\rangle} = \prod_{v\in[n]} \mathbb{E}e^{i\left\langle \mathbf{x}(v), zX_1^{(v)}\right\rangle} \qquad \text{(by independence)}$$

Since $X^{(v)}$ is α-stable, we have $zX_1^{(v)} \sim X_{z^\alpha}^{(v)}$

$$\begin{aligned}
&= \prod_{v\in[n]} \mathbb{E}e^{i\left\langle \mathbf{x}(v), X_{z^\alpha}^{(v)}\right\rangle} \\
&= \prod_{v\in[n]} e^{-z^\alpha f(\mathbf{x}(v))} && \text{(by Lévy-Khintchine)} \\
&= e^{-z^\alpha f(\mathbf{x})} = e^{-(zf(\mathbf{x})^{1/\alpha})^\alpha} \\
&= \mathbb{E}e^{izf(\mathbf{x})^{1/\alpha}Y_\alpha}, && \text{(by Lévy-Khintchine)}
\end{aligned}$$

Since the characteristic function uniquely identifies the distribution, we know $C_1 \sim f(\mathbf{x})^{1/\alpha}Y_\alpha$. □

Note that a register in Indyk's [96] L_α-stable sketch distributes as $(\sum_{v\in[n]} |\mathbf{x}(v)|^\alpha)^{1/\alpha}Y_\alpha$. Thus m i.i.d. copies of C_1 *emulate* the distribution of Indyk's L_α-sketch where the L_α-moment is replaced by the f-moment. Such function moments can be estimated by Indyk's estimator [96] or Li's estimators [118] with the same error guarantee.

Note that such stable projection takes less space than the generic projection considered in Section 3.4 since in general C_t needs to be stored for many t while stable projection only requires us to store C_1.

Definition 3.7 (stable moments)**.** An f-moment is called a *stable moment* if f is the characteristic exponent of a stable Lévy process.

The class of stable moments, i.e., the characteristic exponent of stable processes are characterized by the following theorem.

Theorem 3.5 (Lévy-Khintchine for stable processes [113, page 86])**.** *Let X be a Lévy process on $\mathbb{R}^d$ and $x \in \mathbb{R}^d$. X is 2-stable if and only if $\mathbb{E}e^{i\langle x, X_t\rangle} = e^{-\frac{1}{2}t\langle x, Ax\rangle}$ for some covariance matrix A. If X is symmetric, then it is α-stable for $\alpha \in (0, 2)$ if and only if there is a finite,*

positive measure μ *on the sphere* $\mathbb{S}_{d-1} = \{x \in \mathbb{R}^d : |x| = 1\}$ *such that*

$$\mathbb{E}e^{i\langle x, X_t\rangle} = \exp\left(-t\int_{\mathbb{S}_{d-1}} |\langle x, \xi\rangle|^\alpha \, \mu(d\xi)\right).$$

Remark 3.5. For simplicity we only consider symmetric processes here when $\alpha \in (0, 2)$ so that the characteristic exponent $f_X(x_1, \ldots, x_d) = f_X(|x_1|, \ldots, |x_d|)$ for any $x \in \mathbb{R}^d$. See [113] for the full characterization.

We now give the formal definition of the Lévy-Stable sketch that can be considered a specialized Lévy-Tower for estimating stable moments.

Definition 3.8 ((f, m)-Lévy-Stable)**.** Let f be the characteristic exponent of an α-stable Lévy process X on $\mathbb{R}^d$ and $m \in \mathbb{Z}_+$. An (f, m)-*Lévy-Stable* sketch is a vector of m registers $T = (T_1, \ldots, T_m) \in \mathbb{R}^m$, initialized as all zero. For any element $v \in [n]$ and $y \in \mathbb{R}^d$, the vector update $\mathbf{x}(v) \leftarrow \mathbf{x}(v) + y$ is effected by:

Update(v, y) : For each $k \in [m]$, $T_k \leftarrow T_k + \left\langle y, X_k^{(v)} - X_{k-1}^{(v)} \right\rangle$, where $X^{(v)} = (X_t^{(v)})_{t\in\mathbb{R}_+}$ is an independent copy of the Lévy process X with characteristic exponent f, indexed by v.

Let us now prove the claim in Chapter 1.

Theorem 1.2 (Restated)**.** *Let* f *be the characteristic exponent of any* d*-dimensional* p*-stable process* X*. Let the* Lévy-Stable *sketch be parameterized by* X*. Given any input vector* $\mathbf{x} \in (\mathbb{R}^d)^n$ *and* $\mathbf{x}' \in \mathbb{R}^n$ *such that* $f(\mathbf{x}) = \sum_{v\in[n]} |\mathbf{x}'(v)|^p$*,* Lévy-Stable *with input* $\mathbf{x}$ *and Indyk's* p*-stable sketch with input* $\mathbf{x}'$ *distribute identically.*

Proof of Theorem 1.2. Since X is Lévy, $X_1, X_2 - X_1, \ldots, X_m - X_{m-1}$ are i.i.d., and it suffices to look at the first register $C_1 = \sum_{v\in[n]} \left\langle \mathbf{x}(v), X_1^{(v)} \right\rangle$. The emulation follows from Lemma 3.8. □

Remark 3.6. We remark here on the differences between the Lévy-Tower sketch and the Lévy-Stable sketch.

- Lévy-Tower estimates generic function moment f_X where X is the characteristic exponent of *any* Lévy process. Lévy-Tower stores projections at multiple times of the process and each projection is wrapped around within $[0, 2\pi)$.

- Lévy-Stable estimates function moment f_X where X is the characteristic exponent of a *stable* Lévy process. Lévy-Stable stores projections at unit time of the process and each projection is in $\mathbb{R}$. Lévy-Stable generalizes the stable sketches of Indyk [96] and Ganguly et al. [88].

3.5 G-Sampling with Subordinators

Perhaps surprisingly, the connection between Lévy-Khintchine processes and streaming sketches goes beyond the linear case. We now demonstrate that the *min-based* sketches/samplers are closely related to *non-negative* Lévy-Khintchine processes, also known as subordinators. In this section, I will show how the simple $F_{1/2}$-sampler in Algorithm 1.3 is discovered, step by step.

Let us first consider a *generic min sketch*. To index the fresh randomness associated with an update, we suppose updates occur at times $k = 1, 2, \ldots$. The min-based samplers are based on the following

Definition 3.9 (generic min sketch)**.** A *generic min sketch* is a pair $(v_*, h_*) \in ([n] \cup \{\perp\}) \times (\mathbb{R}_+ \cup \{\infty\})$ initialized as $(\perp, \infty)$. It is associated with a certain hash function $H : [n] \times \mathbb{R}_+ \times \mathbb{Z}_+ \to \mathbb{R}_+$ that uses the hash function provided by the random oracle. A vector update $\mathbf{x}(v) \leftarrow \mathbf{x}(v) + \Delta$ is handled as follows.

Update(v, Δ) **(issued at time** k**):** If $H(v, \Delta, k) < h_*$, then set $(v_*, h_*) \leftarrow (v, H(v, \Delta, k))$.

Assume for simplicity that $\Delta = 1$ in all updates. Suppose $\mathbf{x}(v) = w$, being incremented at times $k_1, \ldots, k_w$. The minimum hash value produced by v is thus $Z_v = \min(H(v, 1, k_1), \ldots, H(v, 1, k_w))$. The observation is that, no matter what the weight function G is, the probability that v gets sampled should only depend on w, rather than the insertion timestamps $(k_1, \ldots, k_w)$. Thus, we want the random sequence

$$H(v, 1, 1), H(v, 1, 2), H(v, 1, 3), \ldots \tag{3.9}$$

to be *exchangeable*.[4] De Finetti's theorem[5] then implies that there is a non-negative random

[4] A finite random vector $(Y_1, \ldots, Y_k)$ is *exchangeable* if its distribution remains unchanged after reordering the coordinates. An infinite random sequence is exchangeable if any finite prefix is exchangeable. See [28, Definition 2.1] for a recent treatment.

[5] This is a classic result by De Finetti from 1931 on exchangeable random sequences. We use the version described in [28, Theorem 2.2].

process $X = (X_t)_{t\in\mathbb{R}_+}$ such that

$$(H(v,1,k))_{k\in\mathbb{Z}_+} = (\{\inf\{t \in \mathbb{R}_+ : X_t > E_k\}\})_{k\in\mathbb{Z}_+},$$

where the $E_k \sim \text{Exp}(1)$ are i.i.d. standard exponential random variables. In addition, in order to sample with the correct probability, we need the minimum hash value $\min_{j\in[w]}(H(v,1,k_j))$ to distribute as an exponential random variable (also observed in [102]), which will be true if X is a Lévy process.[6] Note that X needs to be non-negative in De Finetti's theorem. Now suppose X is a subordinator (non-negative Lévy process) with Laplace exponent $G_X : \mathbb{R}_+ \to \mathbb{R}_+$. Then for $z > 0$,

$$\begin{aligned}
&\mathbb{P}(\min(H(v,1,k_1),\ldots,H(v,1,k_w)) > z)\\
&= \mathbb{P}(\min(H(v,1,1),\ldots,H(v,1,w)) > z) && ((3.9) \text{ is exchangeable})\\
&= \mathbb{P}(\min(\inf\{t \in \mathbb{R}_+ : X_t > E_1\},\ldots,\inf\{t \in \mathbb{R}_+ : X_t > E_w\}) > z) && (\text{De Finetti's theorem})\\
&= \mathbb{P}(\inf\{t \in \mathbb{R}_+ : X_t > \text{Exp}(w)\} > z)\\
&= \mathbb{P}(X_z \leq \text{Exp}(w))\\
&= \mathbb{E}e^{-wX_z} = e^{-zG_X(w)}. && (\text{Lévy-Khintchine})
\end{aligned}$$

Note that $\mathbb{P}(\text{Exp}(\lambda) > z) = e^{-z\lambda}$. Therefore we know that

$$\min(H(v,1,k_1),\ldots,H(v,1,k_w)) \sim \text{Exp}(G_X(w)) = \text{Exp}(G_X(\mathbf{x}(v))).$$

I.e., the minimum hash value produced by an element v is *exactly* an exponential random variable with rate $G(\mathbf{x}(v))$. This is the underlying intuition for the Lévy-Min-Sampler studied in Section 3.5. The precise simulation of $\text{Exp}(G_X(w))$ can be used to emulate existing cardinality estimators with the *cardinality* of $\mathbf{x}$ ($\|\mathbf{x}\|_0 = |\text{supp}(\mathbf{x})|$) replaced by the *G-moment* $G(\mathbf{x})$. This is the underlying intuition for the LévyPCSA and LévyHyperLogLog sketches studied in Section 3.6.

3.5.1 The Raw Lévy-Min-Sampler

Based on the discussion above, we now consider hash functions induced by subordinators.

[6] As we will see later, the reason that X is not *necessarily* Lévy is because the whole vector (3.9) is only observed once through a subset min. Here one can also construct the sketch from the infinite divisibility point of view.

Theorem 3.6 (raw Lévy-Min-Sampler)**.** *For $u \in [n], \Delta \in \mathbb{R}_+$, and $k \in \mathbb{Z}_+$, define*

$$H(u, \Delta, k) = \inf \left\{ t \in \mathbb{R}_+ : X_t^{(u)} > Z_k / \Delta \right\},$$

where the $X^{(u)} = (X_t^{(u)})_{t \in \mathbb{R}_+}$ are i.i.d. subordinators with Laplace exponent $G : \mathbb{R}_+ \to \mathbb{R}_+$ and the $Z_k \sim \mathrm{Exp}(1)$ are i.i.d. standard exponential random variables. Then the min sketch induced by H is a G-sampler. I.e., the resulting state (v_*, h_*) satisfies the following.

- *For any $u \in [n]$, $\mathbb{P}(v_* = u) = G(\mathbf{x}(u))/G(\mathbf{x})$.*
- *$h_* \sim \mathrm{Exp}(G(\mathbf{x}))$.*

Remark 3.7. The hash values $(H(u, 1, k))_{k \in \mathbb{N}}$ in the raw Lévy-Min-Sampler form a min-wise infinitely divisible, exchangeable sequence [28].

The proof of Theorem 3.6 makes use of the following properties of exponential random variables.

Lemma 3.9 (properties of exponential random variables)**.** *Let $Y_1, \ldots, Y_k$ be independent random variables where $Y_j \sim \mathrm{Exp}(\lambda_j)$ for $j \in [k]$. Then*

$$\min\{Y_1, \ldots, Y_k\} \sim \mathrm{Exp}(\lambda_1 + \ldots + \lambda_k),$$
$$\mathbb{P}\left(Y_u < \min_{j \neq u} Y_j \right) = \frac{\lambda_u}{\sum_{j \in [k]} \lambda_j}.$$

Proof of Theorem 3.6. Suppose element u is updated at times (k_j) with increments (Δ_j), where $\sum_j \Delta_j = \mathbf{x}(u)$. Observe that for any $z \in \mathbb{R}_+$,

$$\begin{aligned}
\mathbb{P}\left(\min_j \{H(u, \Delta_j, k_j)\} \geq z \right) &= \mathbb{P}\left(\inf \left\{ t \in \mathbb{R}_+ : X_t^{(u)} > \min_j \{Z_{k_j}/\Delta_j\} \right\} \geq z \right) \\
&= \mathbb{P}\left(X_z^{(u)} \leq \min_j \{Z_{k_j}/\Delta_j\} \right).
\end{aligned}$$

Since the Z_{k_j} are i.i.d. Exp(1) random variables, $\min_j\{Z_{k_j}/\Delta\} \sim \mathrm{Exp}(\sum_j \Delta_j) = \mathrm{Exp}(\mathbf{x}(u))$. Continuing,

$$\begin{aligned}
&= \mathbb{P}(X_z^{(u)} \leq \mathrm{Exp}(\mathbf{x}(u))) \\
&= \mathbb{E}\mathbb{P}\left(X_z^{(u)} \leq \mathrm{Exp}(\mathbf{x}(u)) \mid X_z^{(u)} \right) \\
&= \mathbb{E} e^{-\mathbf{x}(u) X_z^{(u)}},
\end{aligned}$$

and since $X^{(u)}$ has Laplace exponent G, this is equal to

$$= e^{-zG(\mathbf{x}(u))}.$$

By the CDF of the exponential distribution,

$$\min_j\{H(u, \Delta_j, k_j)\} \sim \text{Exp}(G(\mathbf{x}(u))). \tag{3.10}$$

□

3.5.2 Stable-Min-Sampler

For $G(x) = x^\alpha$, a much simpler simulation is available.

Theorem 3.7 (Stable-Min-Sampler)**.** *Let $\alpha \in (0, 1)$. For $u \in [n]$ and $k \in \mathbb{Z}_+$, define*

$$H(u, \Delta, k) = \left(\frac{Z_k}{W^{(u)}\Delta}\right)^\alpha,$$

where $W^{(u)}$s are i.i.d. one-sided α-stable random variables and the $Z_k \sim \text{Exp}(1)$ are i.i.d. standard exponential random variables. Then the min sketch induced by H is an F_α-sampler. I.e., the resulting state (v_*, h_*) satisfies the following.

- *For any $u \in [n]$, $\mathbb{P}(v_* = u) = \mathbf{x}(u)^\alpha / \sum_{v\in[n]} \mathbf{x}(v)^\alpha$.*
- *$h_* \sim \text{Exp}\left(\sum_{v\in[n]} \mathbf{x}(v)^\alpha\right)$.*

Proof. Note that

$$\mathbb{P}((\text{Exp}(\mathbf{x}(u))/W^{(u)})^\alpha \geq z) = \mathbb{P}(\text{Exp}(\mathbf{x}(u)) \geq z^{1/\alpha} W^{(u)}) = \mathbb{E}e^{-z^{1/\alpha}\mathbf{x}(u)W^{(u)}} = e^{-z\mathbf{x}(u)^\alpha}.$$

□

3.5.3 Lévy-Min-Sampler

For a generic weight function G, the raw Lévy-Min-Sampler can be simplified since we only need $\min_j\{H(u, \Delta_j, k_j)\}$ to be distributed as $\text{Exp}(G(\mathbf{x}(u)))$, which can be simulated through *level functions*. Level functions are similar to the *score* functions used by Cohen [51], which map the indexed hash values and fresh randomness into one number. We now construct level functions from subordinators.

Definition 3.10 (level function induced by G)**.** Given a subordinator $X = (X_t)_{t\in\mathbb{R}_+}$ with Laplace exponent $G : \mathbb{R}_+ \to \mathbb{R}_+$, the *induced level function* is

$$\ell_G(x, y) = \inf\{z : \mathbb{P}(X_z \geq x) \geq y\},$$

for any $x, y \in \mathbb{R}_+$.

Level functions constructed above have two desirable properties for algorithm design—*2D-monotonicity* and *G-transformation*.

Lemma 3.10 (level functions)**.** *For any Laplace exponent G, the induced level function ℓ_G satisfy:*

2D-monotonicity *for any $a, a' \in \mathbb{R}_+$ and $b, b' \in [0, 1]$, $a \leq a'$ and $b \leq b'$ implies $\ell_G(a, b) \leq \ell_G(a', b')$;*

G-transformation *if $Y \sim \text{Exp}(\lambda)$ and $U \sim \text{Uniform}(0, 1)$, then $\ell_G(Y, U) \sim \text{Exp}(G(\lambda))$.*

Proof. Recall that $G \in \mathcal{G}$, $Y \sim \text{Exp}(\lambda)$, and $U \sim \text{Uniform}(0, 1)$. We argue that $\ell_G(Y, U)$ (Definition 1.9) is monotonic in both arguments and analyze its distribution.

By Lévy-Khintchine, G has a corresponding non-negative Lévy process $X = (X_t)_{t\geq 0}$ for which $\mathbb{E}e^{-zX_t} = e^{-tG(z)}$. Note that since Lévy processes are memoryless, a non-negative Lévy process is also *non-decreasing*. Therefore $\mathbb{P}(X_t \geq a)$ is increasing in t and decreasing in a, and as a consequence, $\ell_G(a, b) = \inf\{t \mid \mathbb{P}(X_t \geq a) \geq b\}$ is 2D-monotonic. We now analyze the distribution of $\ell_G(Y, U)$. For any $w > 0$, we have

$$\mathbb{P}(\ell_G(Y, U) \geq w) = \mathbb{P}(\inf\{t : \mathbb{P}(X_t \geq Y) \geq U\} \geq w) \qquad \text{definition of } \ell_G$$

Note that by the definition of Lévy process, $\mathbb{P}(X_t \geq Y)$ is a continuous function in t and therefore $\inf\{t : \mathbb{P}(X_t \geq Y) \geq U\} \geq w$ is equivalent to $\mathbb{P}(X_w \geq Y) \leq U$. Thus, this is equal to

$$\begin{aligned}
&= \mathbb{P}(\mathbb{P}(X_w \geq Y) \leq U) \\
&= 1 - \mathbb{P}(X_w \geq Y) && U \sim \text{Uniform}(0, 1) \\
&= \mathbb{P}(X_w < Y) \\
&= \mathbb{E}(\mathbb{P}(X_w < Y) \mid X_w) \\
&= \mathbb{E}(e^{-\lambda X_w} \mid X_w) && Y \sim \text{Exp}(\lambda) \\
&= \mathbb{E}e^{-\lambda X_w}
\end{aligned}$$

$$= e^{-wG(\lambda)} \qquad \text{by Lévy-Khintchine (Theorem 3.1).}$$

Since the CDF of $\text{Exp}(\lambda)$ is $1 - e^{-\lambda x}$, we conclude that $\ell_G(Y, U) \sim \text{Exp}(G(\lambda))$. □

The final Lévy-Min-Sampler using level functions is given by Theorem 3.8, which implements the G-sampler in Algorithm 1.4. The following theorem proves the correctness of it as claimed in Theorem 1.9.

Theorem 3.8 (Lévy-Min-Sampler/G-Sampler)**.** *Let $H : [n] \to [0, 1]$ be a uniformly random hash function. Let $G : \mathbb{R}_+ \to \mathbb{R}_+$ be the Laplace exponent of any subordinator. The Lévy-Min-Sampler is a pair (v_*, h_*) initialized as $(\perp, \infty)$. A vector update $\mathbf{x}(v) \leftarrow \mathbf{x}(v) + \Delta$ at time k is effected by:*

Update(v, Δ) **(issued at time k):** *If $\ell_G(\frac{Z_k}{\Delta}, H(v)) < h_*$, then $(v_*, h_*) \leftarrow (v, \ell_G(\frac{Z_k}{\Delta}, H(v)))$,*

where the $Z_k \sim \text{Exp}(1)$ are i.i.d. standard exponential random variables. When the input vector is $\mathbf{x} \in \mathbb{R}^n_+$, the resulting sampler state satisfies the following.

- *For any $u \in [n]$, $\mathbb{P}(v_* = u) = G(\mathbf{x}(u))/G(\mathbf{x})$.*
- *$h_* \sim \text{Exp}(G(\mathbf{x}))$.*

Proof of Theorem 3.8. Let $\mathbf{x} \in \mathbb{R}^n_+$ be the final vector after all updates, and (v_*, h_*) the final memory state of G-Sampler. For $v \in [n]$, let h_v be the smallest value produced by the k updates $(v, \Delta_i)_{i \in [k]}$ to index v, so $\mathbf{x}(v) = \sum_{i=1}^k \Delta_i$. Let $Y_1, \ldots, Y_k$ be the i.i.d. $\text{Exp}(1)$ random variables generated during those updates. Then

$$h_v \sim \min\big\{\ell_G(Y_1/\Delta_1, H(v)), \ldots \ell_G(Y_k/\Delta_k, H(v))\big\}$$

and by the 2D-monotonicity property of Lemma 3.10, this is equal to

$$= \ell_G\big(\min\big\{Y_1/\Delta_1, \ldots, Y_k/\Delta_k\big\}, H(v)\big).$$

Note that $\min\{Y_1/\Delta_1, \ldots, Y_k/\Delta_k\} \sim \text{Exp}(\sum_i \Delta_i) = \text{Exp}(\mathbf{x}(v))$ and $H(v) \sim \text{Uniform}(0, 1)$. By the G-transformation property of Lemma 3.10, this is distributed as

$$\sim \text{Exp}(G(\mathbf{x}(v))).$$

By properties of the exponential distribution, we have

$$h_* = \min_{v \in [n]} h_v \sim \min_{v \in [n]} \text{Exp}(G(\mathbf{x}(v))) \sim \text{Exp}(G(\mathbf{x}))$$

and the probability that $v_* = v$ is sampled is exactly $G(\mathbf{x}(v))/G(\mathbf{x})$. □

Remark 3.8. The proof of Theorem 3.8 shows that G-Sampler has exactly correct sampling probability, assuming the the value h_* can be stored in a word. On a discrete computer, where the value h_* can only be stored discretely, there is no hope to have h_* to distribute as $\mathrm{Exp}(G(\mathbf{x}))$ perfectly. Nevertheless, a correct sample can still be returned on discrete computers, because one does not have to compute the exact $\ell_G(x, y)$ values but just correctly *compare them*. We now discuss such a scheme. Let $\|\mathbf{x}\|_\infty = \mathrm{poly}(n)$. Supposing that we only stored h_* to $O(\log n)$ bits of precision, we may not be able to correctly ascertain whether $h < h_*$ in the G-Sampler (Algorithm 1.4). This event occurs with probability $1/\mathrm{poly}(n)$,[7] which induces an additive error in the sampling probability, i.e., $(\epsilon, \eta) = (0, 1/\mathrm{poly}(n))$. There are ways to achieve $(\epsilon, \eta) = (0, 0)$ in the random oracle model. Suppose update (v, Δ) is issued at (integer) time k. We could store (v, k) and generate Y from k via the random oracle. Thus, in a stream of m updates, the space would be $O(\log(n+m))$ bits. Another option is to generate more precise estimates of Y (and $H(v)$) on the fly. Rather than store a tuple (v, h), $h = \ell_G(Y/\Delta, H(v))$, we store (v, R, Δ), where $R \sim \mathrm{Uniform}(0, 1)$ is dynamically generated to the precision necessary to determine whether $h < h_*$ in Algorithm 1.4. Specifically, $Y = -\log R \sim \mathrm{Exp}(1)$ is derived from R, and if we cannot determine if $h < h_*$, where $h = \ell_G(Y/\Delta, H(v)), h_* = \ell_G(Y_*/\Delta_*, H(v_*))$, we append additional random bits to R, R_* until the outcome of the comparison is certain.

Remark 3.9. Let me discuss about the related work here. Cohen [52] developed *approximate* G-samplers for soft concave sublinear functions, which are a subset of Laplace exponents. Jayaram and Woodruff [104] have developed G-samplers with precisely correct sampling probabilities for functions $G : \mathbb{N} \to \mathbb{R}$ such that $\max_{z\in\mathbb{N}}(G(z) - G(z-1)) < \infty$ with $O(\frac{\sum_{v\in[n]}\mathbf{x}(v)}{G(\mathbf{x})}\log n)$ bits of space. For G being a Laplace exponent, the factor $\frac{\sum_{v\in[n]}\mathbf{x}(v)}{G(\mathbf{x})}$ in [104] can be $\Omega(\mathrm{poly}(n))$ when $\sum_{v\in[n]}\mathbf{x}(v) \gg G(\mathbf{x})$. The significance of Theorem 3.8 is that the sampling probability is *precisely correct* and the space usage is only two words.

We demonstrate by examples how the Lévy-Min-Sampler generalizes previous sketches and leads to new sketches. First, recall that by Lévy-Khintchine (Theorem 3.2), a function $G : \mathbb{R}_+ \to \mathbb{R}_+$ is the Laplace exponent of some subordinator if and only if it is generated by

[7] To see this, consider two independent random variables $Y_1 \sim \mathrm{Exp}(\lambda_1)$ and $Y_2 \sim \mathrm{Exp}(\lambda_2)$. By the properties of exponential random variables, conditioning on $Y_1 < Y_2$, then $|Y_1 - Y_2| \sim \mathrm{Exp}(\lambda_2)$; conditioning on $Y_1 > Y_2$, then $|Y_1 - Y_2| \sim \mathrm{Exp}(\lambda_1)$. This suggests $\mathbb{P}(|Y_1 - Y_2| \geq z) \geq e^{-\max(\lambda_1,\lambda_2)z}$. Thus if both λ_1 and λ_2 are $O(\mathrm{poly}(n))$, then with $|Y_1 - Y_2| = \Omega(1/\mathrm{poly}(n))$ with probability $e^{-O(1/\mathrm{poly}(n))} \geq 1 - O(1/\mathrm{poly}(n))$. Now let $\lambda_1 = \mathbf{x}(v'_*)$ and $\lambda_1 = \mathbf{x}(v_*)$, where v'_* is the sample selected by the $O(\log n)$-bit sketch and v_* is the sample selected by the infinite precision sketch (the first $O(\log n)$ bits are the same with the former one). If $v'_* \neq v_*$ then it implies $|Y_2 - Y_1| < O(1/\mathrm{poly}(n))$, which happens only with probability $O(1/\mathrm{poly}(n))$.

a triplet (c, ν, γ_0) where $c, \gamma_0 \geq 0$, and $\int_0^\infty \min(s, 1)\, \nu(ds) < \infty$.

$$G(x) = c\mathbb{1}\left[z \neq 0\right] + \gamma_0 z + \int_0^\infty (1 - e^{-zs})\, \nu(ds).$$

Example 1 (F_0-sampler $\mapsto$ the Min sketch [48])**.** For F_0-sampling, we have $G(x) = \mathbb{1}\left[x > 0\right]$, which is generated by $(1, 0, 0)$. This corresponds to a "pure-killed process" X, where, given a kill time $Y \sim \text{Exp}(1)$, $X_t = 0$ if $t < Y$ and $X_t = \infty$ if $t \geq Y$. The induced level function (Definition 3.10) is,

$$\ell(x, y) = \inf\{z : \mathbb{P}(X_z \geq x) \geq y\}$$

since $x > 0$, $\mathbb{P}(X_z \geq x) = \mathbb{P}(z \geq Y) = 1 - e^{-z}$, regardless of x, this is equal to

$$\begin{aligned} &= \inf\{z : 1 - e^{-z} \geq y\} \\ &= -\log(1 - y). \end{aligned}$$

Thus the resulting sketch stores $-\log(1 - \eta)$, where η is the minimum hash value, thereby essentially reproducing Cohen's [48] Min sketch.

Example 2 (F_1-sampler $\mapsto$ min-based reservoir sampling [155])**.** For F_1-sampling, we have $G(x) = x$, which is generated by $(0, 1, 0)$. This corresponds to a deterministic drift process X, where $X_t = t$ for $t \in \mathbb{R}_+$. The induced level function is,

$$\ell(x, y) = \inf\{z : \mathbb{P}(X_z \geq x) \geq y\}$$

and since $X_z = z$, $\mathbb{P}(X_z \geq x) = \mathbb{1}\left[z \geq x\right]$,

$$\begin{aligned} &= \inf\{z : \mathbb{1}\left[z \geq x\right] \geq y\} \\ &= x. && \text{(note that } y \in (0, 1)\text{)} \end{aligned}$$

Thus the resulting sketch stores the minimum random value sampled freshly at each insertion, reproducing the reservoir sampler ([155]) with the choice of replacement simulated by taking the min.

Next, we demonstrate a "non-trivial" application: the construction of the $F_{1/2}$-sampler.

Example 3 ($F_{1/2}$-sampler)**.** For $F_{1/2}$, we have $G(x) = \sqrt{x}$, which corresponds to the 1/2-

stable process. The induced level function is

$$\ell(x, y) = \inf\{z : \mathbb{P}(X_z \geq x) \geq y\}$$

and since X is 1/2-stable, we have $X_z \sim z^2 X_1$

$$= \inf\{z : \mathbb{P}(z^2 X_1 \geq x) \geq y\}.$$

It is known that the 1/2-stable X_1 distributes identically with $1/Z^2$ where Z is a standard Gaussian [113, page 29]. Thus, we have $\mathbb{P}(X_1 \geq r) = \mathbb{P}(|Z| \leq \sqrt{1/r}) = \operatorname{erf}\left(\sqrt{\frac{1}{2r}}\right)$, where $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-x^2}\, dx$ is the *Gauss error function*. As $\mathbb{P}(z^2 X_1 \geq x) = \mathbb{P}(X_1 \geq x/z^2) = \operatorname{erf}(z/\sqrt{2x})$, this is equal to

$$= \sqrt{2x} \cdot \operatorname{erf}^{-1}(y).$$

Thus to sample with weight function $G(x) = \sqrt{x}$, upon an $\mathsf{Update}(v, \Delta)$, one just needs to compute $h = \sqrt{2Y/\Delta} \cdot \operatorname{erf}^{-1}(H(v))$ where $Y \sim \text{Exp}(1)$ is freshly sampled and $H(v) \sim \text{Uniform}(0, 1)$ is the hash value of v, and replace (v_*, h_*) with (v, h) if $h < h_*$.[8] The example justifies the correctness of the $F_{1/2}$-sampler in Algorithm 1.3.

The $F_{1/2}$-moment is a special case where the 1/2-stable distribution has a clean expression. Penson and Górska [136] give explicit formulas for one-sided k/l-stable distributions. For a generic $\alpha \in (0, 1)$ whose level function is difficult to compute, it is more convenient to use the Stable-Min-Sampler (Theorem 3.7).

Previously, approximate "soft cap"-samplers were used by Cohen to estimate cap-statistics [51]. The weight function for a soft cap sampler is parameterized by $\tau > 0$, where $G_\tau(z) = 1 - e^{-\tau z}$. We now compute the level functions needed to build soft cap samplers with precisely correct sampling probabilities.

Example 4 ("soft cap"-sampler)**.** The weight function is $G_\tau(z) = 1 - e^{-\tau z}$. By Lévy-Khintchine, G_τ corresponds to a unit-rate Poisson counting process $(X_t)_{t \geq 0}$ with jump size τ, where $X_t/\tau \sim \text{Poisson}(t)$. By Definition 1.9,

$$\ell_{G_\tau}(a, b) = \inf\{z \mid \mathbb{P}(X_z \geq a) \geq b\}$$

[8] The inverse error function erf^{-1} is available, e.g., as `scipy.special.erfinv` in `Python`.

Since $X_z/\tau \sim \text{Poisson}(z)$, $\mathbb{P}(X_z \geq a) = e^{-z} \sum_{j=\lceil a\tau \rceil}^{\infty} \frac{z^j}{j!}$. Continuing,

$$= \inf \left\{ z \;\middle|\; e^{-z} \sum_{j=\lceil a\tau \rceil}^{\infty} \frac{z^j}{j!} \geq b \right\}.$$

Thus $\ell_{G_\tau}(a,b) = w_{\lceil a\tau \rceil, b}$ where $w_{k,b}$ is the unique solution to the equation $e^{-w} \sum_{j=k}^{\infty} \frac{w^j}{j!} = b$. Note that for any $k \in \mathbb{Z}_+$, the function $g(w) = e^{-w} \sum_{j=k}^{\infty} \frac{w^j}{j!} = \mathbb{P}(\text{Poisson}(w) \geq k)$ is increasing from 0 to ∞ as w increases from 0 to ∞, by a simple coupling argument. Therefore, $w_{k,b}$, as the solution of $g(w) = b$, can be computed with a binary search.

Example 5 (log-sampler)**.** Consider the weight function $G(z) = \log(1+z)$. By Lévy-Khintchine, the corresponding process is a Gamma process $(X_t)_{t \geq 0}$, where $X_t \sim \text{Gamma}(t,1)$. The PDF of $\text{Gamma}(t,1)$ is $f(r) = r^{t-1}e^{-r}/\Gamma(t)$. By Definition 1.9,

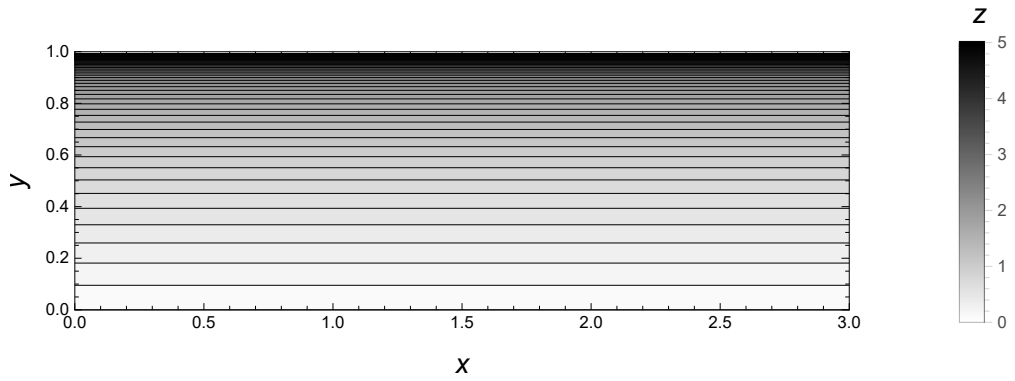
$$\ell_{G_\tau}(a,b) = \inf\{z \;\mid\; \mathbb{P}(X_z \geq a) \geq b\}$$

Since $X_z \sim \text{Gamma}(z,1)$, $\mathbb{P}(X_z \geq a) = \Gamma(z)^{-1} \int_a^\infty r^{z-1}e^{-r}\,dr$. Continuing,

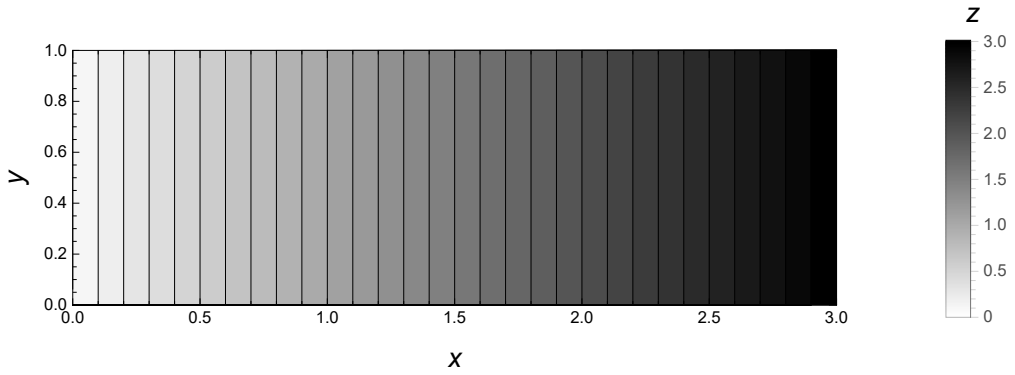
$$= \inf \left\{ z \;\middle|\; \Gamma(z)^{-1} \int_a^\infty r^{z-1}e^{-r}\,dr \geq b \right\}.$$

Thus $\ell_G(a,b) = w'_{a,b}$ which is the unique solution of the equation $\Gamma(w)^{-1} \int_a^\infty r^{w-1}e^{-r}\,dr = b$. Once again the left hand side is monotonic in w and therefore $w'_{a,b}$ can be found with a binary search.

See Fig. 3.2 and Fig. 3.3 for contour plots of the level functions used in the examples. As discussed in Remark 3.8, there is no need to compute the level functions exactly, which is impossible to do so in the practical finite-precision model anyway. We only need to evaluate level functions to tell which index v has the smallest $\ell_G(x,y)$ value. Such samples are still truly perfect, even though the final value h_* is not a perfect $\text{Exp}(G(\mathbf{x}))$ random variable. For a generic G, in practice one may pre-compute the level function of G on a geometrically spaced lattice and cache it as a read-only table. Such a table can be shared and read simultaneously by an unbounded number of G-samplers for different applications and therefore the amortized space overhead is typically small.

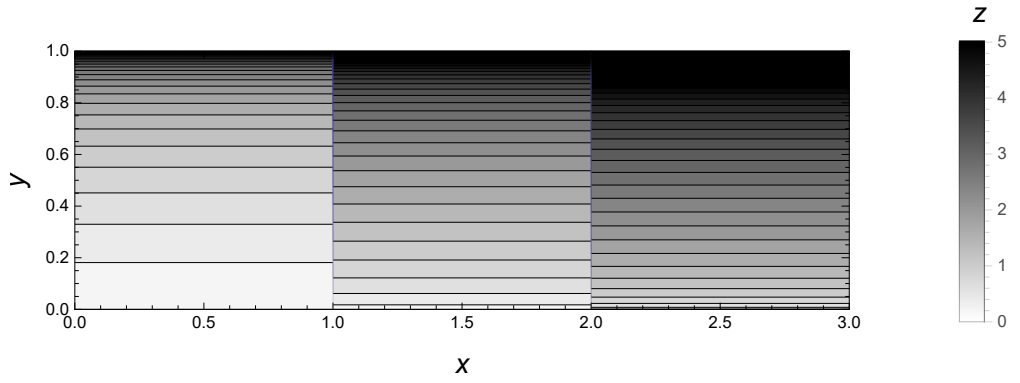


(a) Level function for F_0-sampler (G-sampler with $G(z) = \mathbb{1}[z > 0]$)

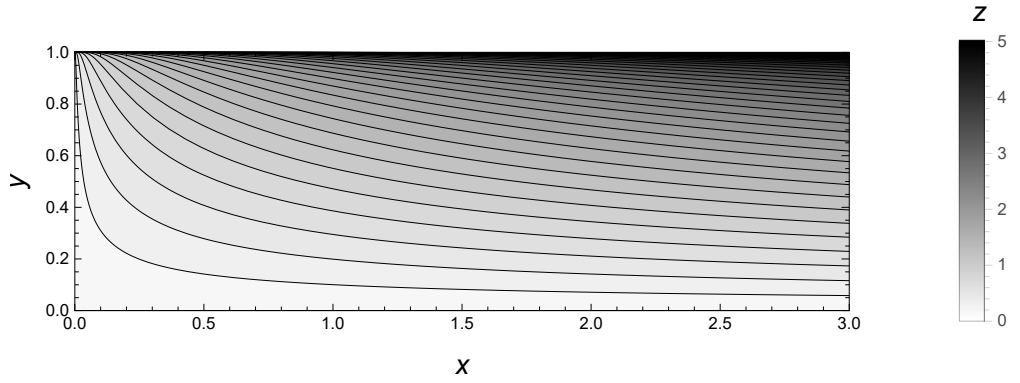


(b) Level function for F_1-sampler (G-sampler with $G(z) = z$)

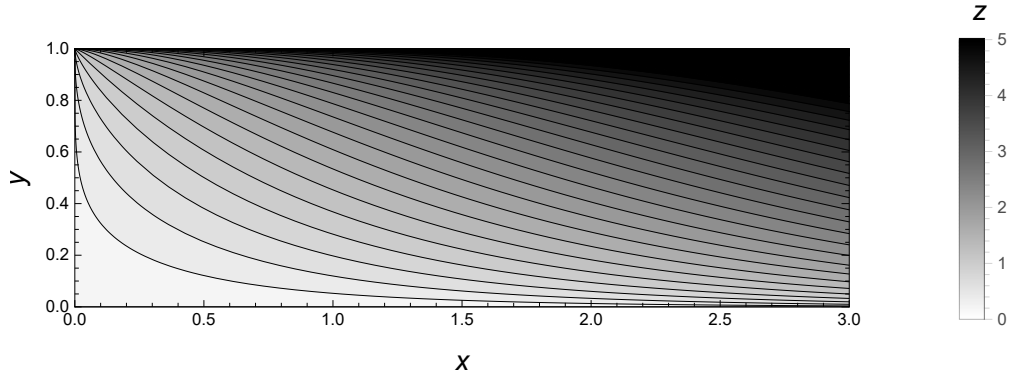
Figure 3.2: Level functions for F_0-sampler and F_1-sampler



(a) Level function for $(1 - e^{-z})$-sampler, induced by a Poisson counting process



(b) Level function for $z^{1/2}$-sampler (a.k.a $F_{1/2}$-sampler), induced by a 1/2-stable process



(c) Level function for $\log(1 + z)$-sampler, induced by a Gamma process

Figure 3.3: Level functions for samplers induced by well-known Lévy processes

3.5.4 Universal Sampler

Since $\ell_G(a, b)$ is increasing in both arguments, among all updates $\{(v_i, \Delta_i)\}$ to the Lévy-Min-Sampler, the stored sample must correspond to a point on the (minimum) *Pareto frontier* of $\{(Y_i/\Delta_i, H(v_i))\}$. Thus, it is possible to produce a G-sample for any $G \in \mathcal{G}$ simply by storing the Pareto frontier. (This observation was also used by Cohen [51] in her approximate samplers.) The size of the Pareto frontier is a random variable that is less than $\log n + 1$ in expectation and $O(\log n)$ with high probability.

Theorem 3.9. *Suppose* ParetoSampler *processes a stream of* $\mathrm{poly}(n)$ *updates to* $\mathbf{x}$. *The maximum space used is* $O(\log n)$ *words with probability* $1 - 1/\mathrm{poly}(n)$. *At any time, given a* $G \in \mathcal{G}$, *it can produce a* $v_* \in [n]$ *such that* $\mathbb{P}(v_* = v) = G(\mathbf{x}(v))/G(\mathbf{x})$.

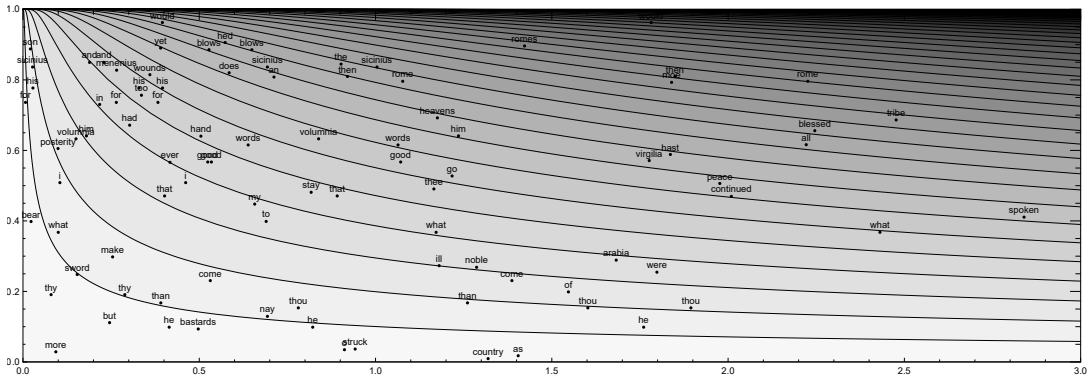
Algorithm 3.1: ParetoSampler. The function Pareto(L) returns the (minimum) Pareto frontier of the tuples L w.r.t. their first two coordinates.

Sketch : $S \subset \mathbb{R}_+ \times \mathbb{R}_+ \times [n]$, initialized as $\emptyset$
Hash function: $H : [n] \to \mathrm{Uniform}(0, 1)$
Result: Sample an element u with prob. $G(\mathbf{x}(u))/\sum_{v\in[n]} G(\mathbf{x}(v))$ upon **Sample** (G)
Update (v, Δ)**:**
  `// upon update` $\mathbf{x}(v) \leftarrow \mathbf{x}(v) + \Delta$
  $S \leftarrow \mathsf{Pareto}(S \cup \{(\mathrm{Exp}(\Delta), H(v), v)\})$ `//` $\mathrm{Exp}(\Delta)$ `is freshly sampled`
end
Sample (G)**:**
  `// upon` G`-sample`
  $(a_*, b_*, v_*) \leftarrow \mathrm{argmin}_{(a,b,v)\in S}\{\ell_G(a, b)\}$
  return v_*
end

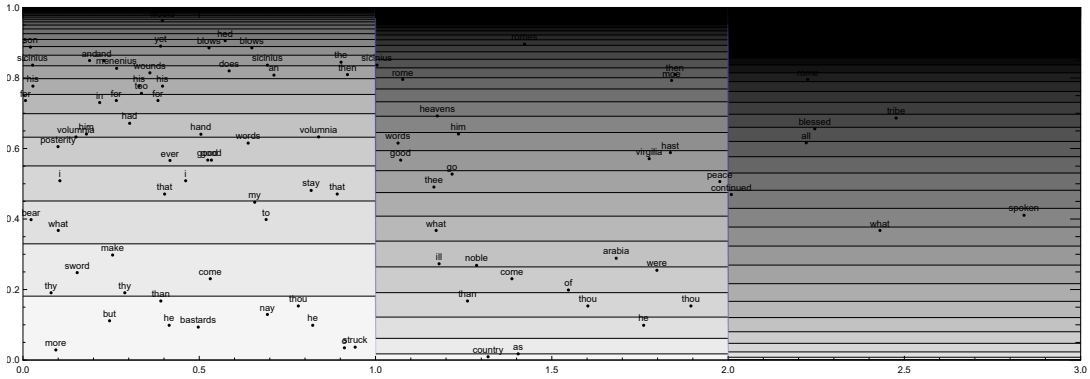
3.5.5 Sampling Without Replacement

We can take k independent copies of the Lévy-Min-Sampler or ParetoSampler sketches to sample k indices from the $(G(\mathbf{x}(v))/G(\mathbf{x}))_{v\in[n]}$ distribution *with* replacement. A small change to these algorithms will sample k indices *without* replacement. See Cohen, Pagh, and Woodruff [54] for an extensive discussion of why WOR (without replacement) samplers are often more desirable in practice. The algorithm (G, k)-Sampler-WOR(Algorithm 3.2) samples k (distinct) indices without replacement.

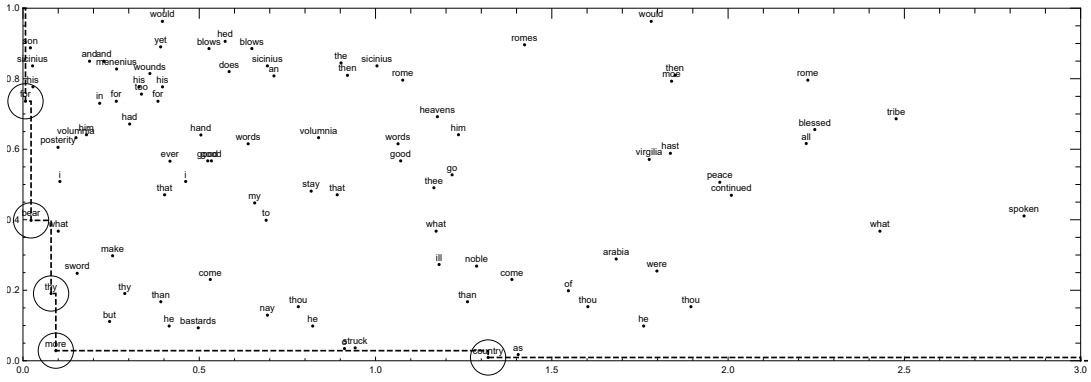
In a similar fashion, one can define a ParetoSampler without replacement that maintains the minimum k-Pareto frontier, defined by discarding any tuple (a, b, v) if there is another



(a) $F_{1/2}$-sampling



(b) $(1 - e^{-z})$-sampling



(c) universal sampling

Figure 3.4: Visualization of universal sampling by ParetoSampler. A section of Shakespeare's play is fed to the sketch, where each insertion of word w corresponds to a point $(\mathrm{Exp}(1), H(w))$ in the board. For a specific G, one may maintain the insertion with the minimum value of $\ell_G(\mathrm{Exp}(1), H(w))$. One observes that the level functions (contour plots above) for $F_{1/2}$ and $(1 - e^{-z})$ are very different. One may store the whole pareto frontier from which a G-sample can be returned for any Laplace exponent G, *without knowing* G *beforehand during sketching*. See Table 3.3.

Word	Location	F_0	F_1	$F_{1/2}$	Log	SoftCap
`for`	(0.007165, 0.7361)	1.332	**0.007165**	0.09457	0.2910	1.332
`bear`	(0.02294, 0.3982)	0.5079	0.02294	0.07904	0.1523	0.5078
`thy`	(0.07974, 0.1908)	0.2117	0.07975	0.06820	0.1007	0.2117
`more`	(0.09321, 0.02866)	0.02908	0.09321	**0.01097**	**0.01532**	**0.02908**
`country`	(1.319, 0.009352)	**0.009396**	1.319	0.01346	0.06580	0.1434

Table 3.3: ParetoSampler for the experiment in Fig. 3.4. The Pareto frontier contains five words with their locations. One may evaluate different level functions on the locations to obtain G-samples for different G. Several functions G are considered above and the entries below are the ℓ_G-values. For each weight function G, the one with the smallest ℓ_G-value (appears in bold) is returned as a correct G-sample.

Algorithm 3.2: (G, k)-Sampler-WOR. The function k-Min(L) takes a list $L \subset [n] \times \mathbb{R}_+$, discards any $(v, h) \in L$ if there is a $(v, h') \in L$ with $h' < h$, then returns the k elements with the smallest second coordinate.

Sketch : $S \subset [n] \times \mathbb{R}_+$, initialized as $\emptyset$
Hash function: $H : [n] \to \text{Uniform}(0, 1)$
Result: Sample k elements u with prob. $\propto G(\mathbf{x}(u))$ *without replacement*
Update (v, Δ)**:**
  `// upon update` $\mathbf{x}(v) \leftarrow \mathbf{x}(v) + \Delta$
  $S \leftarrow k$-Min$(S \cup \{(v, \ell_G(\text{Exp}(\Delta), H(v)))\})$ `//` $\text{Exp}(\Delta)$ `is freshly sampled`
end
Sample ( )**:**
  `// upon sample`
  return $\{v : (v, \cdot) \in S\}$
end

(a', b, v) with $a' < a$, then retaining only those tuples that are dominated by at most $k-1$ other tuples.

Theorem 3.10. *Consider a stream of* $\mathrm{poly}(n)$ *incremental updates to a vector* $\mathbf{x} \in \mathbb{R}_+^n$. *The* (G,k)-*Sampler-WOR occupies* $2k$ *words of memory, and can report an ordered tuple* $(v_*^1, \ldots, v_*^k) \in [n]^k$ *such that*

$$\mathbb{P}((v_*^1, \ldots, v_*^k) = (v^1, \ldots, v^k)) = \prod_{i=1}^{k} \frac{G(\mathbf{x}(v^i))}{G(\mathbf{x}) - \sum_{j=1}^{i-1} G(\mathbf{x}(v^j))}. \tag{3.11}$$

The k-*ParetoSampler occupies* $O(k \log n)$ *words w.h.p. and for any* $G \in \mathcal{G}$ *at query time, can report a tuple* $(v_*^1, \ldots, v_*^k) \in [n]^k$ *distributed according to Eq.* (3.11).

Proof. The proof of Theorem 3.8 shows that $h_v \sim \mathrm{Exp}(G(\mathbf{x}(v)))$ and if v_*^1 minimizes $h_{v_*^1}$, that $h_{v_*^1} \sim \mathrm{Exp}(G(\mathbf{x}))$. It follows that $\mathbb{P}(v_*^1 = v) = G(\mathbf{x}(v))/G(\mathbf{x})$. By the memoryless property of the exponential distribution, for any $v \neq v_*^1$, $h_v - h_{v_*^1} \sim \mathrm{Exp}(G(\mathbf{x}(v)))$, hence $h_{v_*^2} - h_{v_*^1} \sim \mathrm{Exp}(G(\mathbf{x}) - G(\mathbf{x}(v_*^1)))$ and $\mathbb{P}(v_*^2 = v \mid v_*^1, v \neq v_*^1) = G(\mathbf{x}(v))/(G(\mathbf{x}) - G(\mathbf{x}(v_*^1)))$. The distribution of $v_*^3, \ldots, v_*^k$ is analyzed in the same way.

By the 2D-monotonicity property, the k-Pareto frontier contains all the points that would be returned by (G,k)-Sampler-WOR, hence the output distribution of k-ParetoSampler-WOR is identical. The analysis of the space bound follows the same lines, except that X_i is the indicator for the event that h_{v_i} is among the k-smallest elements of $\{h_{v_1}, \ldots, h_{v_i}\}$, so $\mathbb{E}(X_i) = \min\{k/i, 1\}$, $\mathbb{E}(|S|) < kH_n$, and by a Chernoff bound, $|S| = O(k \log n)$ with high probability. □

3.6 LévyPCSA and LévyHyperLogLog

The G-transformation technique in Section 3.5.3 can be used to reduce G-*moment* estimation in the $\mathbb{R}_+$-turnstile to the intensively studied cardinality estimation (F_0-moment) problem. I will prove the following two emulation theorems in this section. Recall that A cardinality sketch is *poissonized* if each actual insertion is simulated by Poisson(1) insertions.

Theorem 3.11 (PCSA emulation)**.** *Let vectors* $\mathbf{x}, \mathbf{x}' \in \mathbb{R}_+^n$ *be such that* $G(\mathbf{x}) = \|\mathbf{x}'\|_0$. *Then LévyPCSA (parameterized by* G*) on input* $\mathbf{x}$ *and poissonized PCSA with input* $\mathbf{x}'$ *distribute identically.*

Theorem 3.12 (HyperLogLog emulation, Section 3.6)**.** *Let vectors* $\mathbf{x}, \mathbf{x}' \in \mathbb{R}_+^n$ *be such that* $G(\mathbf{x}) = \|\mathbf{x}'\|_0$*. Then* LévyHyperLogLog *(parameterized by* G*) with input* $\mathbf{x}$ *and poissonized* HyperLogLog *with input* $\mathbf{x}'$ *distribute identically.*

The significance of the two emulation theorems is that, since the final distribution of the sketches are the same as the classic sketches, every analysis, estimator, or optimization developed over the years can be applied to LévyPCSA, and LévyHyperLogLog for free, such as the near-optimal τ-GRA estimators (Section 5.8) for PCSA and HyperLogLog, and maximum likelihood estimation of PCSA ([117] and Section 5.3). In particular, the Fishmonger sketch (Section 5.5) (an entropy-compressed version of PCSA with maximum likelihood estimation) can estimate $G(\mathbf{x})$ with relative variance $1/m$ using $(1 + o(1))m(H_0/I_0) + O(\log^2 \log n) \approx 1.98m$ bits.[9]

3.6.1 Lévy-Activation and G-Cells

Ting [151] and I (Section 5.1.1) analyze cardinality sketches in a uniform fashion by viewing them as a set $\mathcal{C}$ of *cells* which begin *inactive* (0) and can make a one-time transition to *active* (1). The probability that an update activates a cell is proportional to its *size*. The *state* of the sketch is a function of the set of active cells. See Sections 5.1.1 and 5.4.1 for more details.

Definition 3.11 (activation of cells)**.** Let $H : [n] \times \mathcal{C} \to [0, 1]$ be a uniformly random hash function. Let $s(c)$ be the size of cell $c \in \mathcal{C}$. Initially $c \leftarrow 0$ for all $c \in \mathcal{C}$. Upon an update with $\Delta > 0$,

Update(v, Δ) **:** For each $c \in \mathcal{C}$, $c \leftarrow c \vee \mathbb{1}\left[H(v, c) < s(c)\right]$.

Assume the total cardinality is $\lambda \in \mathbb{N}$. A cell with size s will be activated with probability $1-(1-s)^\lambda \to 1-e^{-s\lambda}$ as $\lambda \to \infty$ and $s\lambda = O(1)$. All upper bounds for cardinality estimation in the random oracle model can be applied to the G-moment estimation if cells with activation probability $1 - e^{-sG(\mathbf{x})}$ can be *simulated*. We call the simulated cells *G-cells*.

Definition 3.12 (Lévy-activation of G-cells)**.** Let X be a subordinator and G be the corresponding Laplace exponent. Upon an update with $\Delta > 0$,

Update(v, Δ)**:** For each $c \in \mathcal{C}$, $c \leftarrow c \vee \mathbb{1}\left[X_{s(c)}^{(v,c)} > Y/\Delta\right]$, where $X_{s(c)}^{(v,c)} \sim X_{s(c)}$ is hashed with key (v, c) and $Y \sim \text{Exp}(1)$ is a freshly sampled standard exponential random variable.

[9] By definition $H_0 = \frac{1}{\log 2} + \sum_{k=1}^{\infty} \frac{1}{k} \log_2 (1 + 1/k)$ and $I_0 = \pi^2/6$. See Section 5.3.

We now prove that the G-cells simulate the ordinary cells in Definition 3.11 asymptotically with the cardinality λ replaced by the G-moment.

Lemma 3.11. *Given any input vector $\mathbf{x} \in \mathbb{R}_+^n$, the activation of G-cells are independent and for each G-cell with size s,*

$$\mathbb{P}(c = 1) = 1 - e^{-sG(\mathbf{x})}.$$

Proof. The independence is given by construction. Note that $\mathbb{P}\left(X_s^{(v)} \leq \operatorname{Exp}(\mathbf{x}(v))\right) = \mathbb{E}e^{-X_s^{(v)}\mathbf{x}(v)} = e^{-sG(\mathbf{x}(v))}$ by Lévy-Khintchine. And thus $\mathbb{P}\left(\forall v \in [n].\, X_s^{(v)} \leq \operatorname{Exp}(\mathbf{x}(v))\right) = \prod_{v\in[n]} e^{-sG(\mathbf{x}(v))} = e^{-sG(\mathbf{x})}$. □

The Lévy activation lemma enables essentially all cardinality estimation techniques to be applied to G-moment estimation, by replacing cells with G-cells.

We first consider the *LévyPCSA* sketch, i.e., the sketch that replaces every cell in PCSA with a G-cell with the same size. Theorem 3.11 follows from Lemma 3.11.

The most efficient mergeable cardinality sketch is Fishmonger (Section 5.5), which is an entropy-compressed version of PCSA with an optimum estimator. It is optimal among all "linearizable" mergeable sketches Section 5.4. Substituting cells with G-cells in Fishmonger yields *LévyFishmonger*.

Corollary 3.1. *The LévyFishmonger sketch provides an asymptotically unbiased estimate of the G-moment with relative variance $1/m$. It uses $(1 + o(1))(H_0/I_0)m + O(\log\log G(\mathbf{x}))^2 \approx 1.98m + O(\log\log G(\mathbf{x}))^2$ bits. Here $H_0 = \frac{1}{\log 2} + \sum_{k=1}^{\infty} \frac{1}{k}\log_2(1 + 1/k)$, $I_0 = \pi^2/6$, and $H_0/I_0 \approx 1.98$.*

Although (compressed) PCSA has the best size-accuracy tradeoff ([117, 150] and Section 5.3), HyperLogLog is already widely deployed in industry. For these reasons the LévyHyperLogLog sketch will be more attractive in some real world applications. Let $H : [n] \to \mathbb{N}$ be a hash function where $\mathbb{P}(H(v) \geq k) = 2^{-k}$ for $k \in \mathbb{N}$. One HyperLogLog subsketch [82] stores a single number M such that upon insertion v, $M = \max(M, H(v))$. With Poisson(λ) distinct insertions, $\mathbb{P}(M \geq k) = 1 - e^{-\lambda 2^{-k}}$. We now show how to emulate M with λ replaced by $G(\mathbf{x})$, proving Theorem 3.12.

Lemma 3.12 (LévyHyperLogLog)**.** *Let G be the Laplace exponent and X be the corresponding subordinator. Let M be an $\mathbb{N}$-register initialized as 0. A vector update $\mathbf{x}(v) \leftarrow \mathbf{x}(v) + \Delta$ is effected by:*

Update(v, Δ) **:** $M \leftarrow \max\left\{M, \max\left\{k \in \mathbb{N} : X^{(u)}_{2^{-k}} > Y/\Delta\right\}\right\}$, *where* $X^{(v)} \sim X$ *is hashed with key* v *and* $Y \sim \mathrm{Exp}(1)$ *is a freshly sampled standard exponential random variable.*

Then given any input vector $\mathbf{x} \in \mathbb{R}^n_+$, $\mathbb{P}(M \geq k) = 1 - e^{-2^{-k}G(\mathbf{x})}$.

Remark 3.10. HyperLogLog is restored by taking $G(x) = \mathbb{1}[x > 0]$, with X being the "pure-killed process."

Proof. For any $u \in [n], \mathbf{x}(u) > 0$, and $j \in \mathbb{N}$,

$$\begin{aligned}&\mathbb{P}\left(\max\left\{k \in \mathbb{N} : X^{(u)}_{2^{-k}} > \mathrm{Exp}(\mathbf{x}(u))\right\} < j\right)\\ &= \mathbb{P}\left(X^{(u)}_{2^{-j}} \leq \mathrm{Exp}(\lambda)\right) \;=\; \mathbb{E}e^{-\mathbf{x}(u)X^{(u)}_{2^{-j}}} \;=\; e^{-2^{-j}G(\mathbf{x}(u))}. \qquad \text{(by Lévy-Khintchine)}\end{aligned}$$

Thus

$$\mathbb{P}(M < j) = \prod_{u\in[n]} \mathbb{P}\left(\max\left\{k \in \mathbb{N} : X^{(u)}_{2^{-k}} > \mathrm{Exp}(\mathbf{x}(u))\right\} < j\right) \;=\; e^{-2^{-j}G(\mathbf{x})}.$$

□

3.6.2 Stable-HyperLogLog

For α-stable processes, it is sufficient to hash every element to a single one-sided α-stable random variable, instead of an α-stable sample path. Lemma 3.13 establishes the correctness of Algorithm 3.3.

Algorithm 3.3: Stable-HyperLogLog

Sketch : $M[1], \ldots, M[m]$, initialized as $-\infty$
Hash function: $H : [n] \times [m] \to$ one sided α-stable, $\alpha \in (0, 1)$
Result: Estimate $\sum_{v\in[n]} \mathbf{x}(v)^\alpha$
`// upon update` $\mathbf{x}(v) \leftarrow \mathbf{x}(v) + \Delta$
Update (v, Δ)
  for $j \in [m]$ **do**
    $w \leftarrow \lfloor \alpha(\log_2 H(v, j) - \log_2 \mathrm{Exp}(\Delta))\rfloor$ `// Exp(Δ) is freshly sampled`
    $M[j] \leftarrow \max(M[j], w)$
  end
`// upon query`
Query ( )
  return $C_m m(\sum_{j=1}^m 2^{-M[j]})^{-1}$ `// HyperLogLog's estimator;` $C_m \to 0.7213$

Lemma 3.13 (Stable-HyperLogLog)**.** *Fix $\alpha \in (0,1)$. Let M be an $\mathbb{N}$-register initialized as 0. A vector update $\mathbf{x}(v) \leftarrow \mathbf{x}(v) + \Delta$, $\Delta > 0$, is effected by:*

$$M \leftarrow \max\left\{M, \left\lfloor \alpha(\log_2 Z^{(v)} - \log_2 Y + \log_2 \Delta)\right\rfloor\right\},$$

where the $Z^{(v)}$ are i.i.d. one-sided α-stable random variables and $Y \sim \mathrm{Exp}(1)$ is a freshly sampled standard exponential random variable. For any input vector $\mathbf{x} \in \mathbb{R}_+^n$, $\mathbb{P}(M \geq k) = 1 - e^{2^{-k}\sum_{v\in[n]}\mathbf{x}(v)^\alpha}$.

Remark 3.11. For comparison, note that at $\mathsf{Update}(v, \Delta)$, for any $\Delta > 0$, HyperLogLog updates as

$$M \leftarrow \max(M, \lfloor -\log_2 W^{(v)} \rfloor),$$

where $W^{(v)} \sim \mathrm{Exp}(1)$. HyperLogLog can be considered the limiting case of Stable-HyperLogLog, since

$$\alpha(\log_2 Z^{(v)} - \log_2 Y + \log_2 \Delta) \to -\log_2 W^{(v)}$$

in distribution as $\alpha \to 0$. It is known that (α-stable)$^\alpha \to 1/\mathrm{Exp}(1)$ in distribution as $\alpha \to 0$ [64].

Proof. For any $u \in [n], \mathbf{x}(u) > 0$ and $j \in \mathbb{N}$,

$$\begin{aligned}
&\mathbb{P}\left(\lfloor \alpha(\log_2 Z^{(v)} - \log_2 \mathrm{Exp}(\mathbf{x}(u)))\rfloor < j\right) \\
&= \mathbb{P}\left(\max\left\{k \in \mathbb{N} : Z^{(u)} > 2^{k/\alpha}\mathrm{Exp}(\mathbf{x}(u))\right\} < j\right) \\
&= \mathbb{P}\left(Z^{(v)} \leq 2^{j/\alpha}\mathrm{Exp}(\mathbf{x}(u))\right) \\
&= \mathbb{E}e^{-\mathbf{x}(u)2^{-j/\alpha}Z^{(v)}}.
\end{aligned}$$

By Lévy-Khintchine, $\mathbb{E}e^{-\lambda Z^{(v)}} = e^{-\lambda^\alpha}$ since $Z^{(v)}$ is α-stable, so this is equal to

$$= e^{-2^{-j}\mathbf{x}(u)^\alpha}.$$

□

3.7 Previous Sketches and Lévy Processes

In this section I will discuss in greater detail how previous sketches can be viewed as sampling Lévy processes. The guiding question is: what will happen if the input vector is replicated a large number of times on disjoint supports?

3.7.1 Single-Level Aggregation and the Central Limit Theorem

We start from the AMS sketch by Alon, Matias, and Szegedy [5], where a cell Q stores

$$Q = \sum_{v \in [n]} \mathbf{x}(v)\xi_v.$$

Here $\mathbf{x} \in \mathbb{R}^n$ is updated in an $\mathbb{R}$-stream and the $\xi_v \in \{-1, 1\}$ are i.i.d. Rademacher random variables.[10] Now suppose the input vector is repeated w times on disjoint supports, i.e., $\mathbf{x}(w) \in \mathbb{R}^{nw}$ with $\mathbf{x}(w)(v) = \mathbf{x}(v \bmod n)$. Then the final state of the sketch for $\mathbf{x}(w)$ would be

$$M_w = Q_1 + Q_2 + \cdots + Q_w,$$

where the (Q_j) are i.i.d. copies of Q. Note that $\mathbb{E}Q = 0$ and $\mathbb{V}Q = \sum_{v \in [n]} \mathbf{x}(v)^2$. Thus as $w \to \infty$ the normalized final state $M_w/\sqrt{w}$ converges to a centered Gaussian random variable with variance $\sum_{v \in [n]} \mathbf{x}(v)^2$ in distribution, by the central limit theorem. Thus, in the regime as $w \to \infty$, the AMS sketch eventually becomes the same as a cell in the Lévy-Tower $\sum_{v \in [n]} \left\langle X_1^{(v)}, \mathbf{x}(v) \right\rangle$, where the $(X^{(v)})$ are i.i.d. one dimensional Wiener processes/Brownian motion and $X_1^{(v)}$ is the value of the process at time $t = 1$. Of course, since Wiener processes/Brownian motions are self-similar, one does not need to sample the Lévy processes at exponentially spaced intervals; samples from m independent processes at time $t = 1$ suffice to approximate $\sum_{v \in [n]} \mathbf{x}(v)^2$ with $O(1/m)$ relative variance. The characteristic exponent of the Wiener process/Brownian motion is the target function for the L_2-moment: $f(x) = |x|^2$.

Another illuminating example is Ganguly's [87] L_k-moment estimator for $k \geq 3$. Ganguly randomly projects the elements and stores $S = \sum_{v \in [n]} \mathbf{x}(v)Z^{(v)}$ where each $Z^{(v)}$ is a random root of $x^k = 1$ on the complex unit circle. The statistic S^k seems to be a good estimator for the F_k-moment since $\mathbb{E}S^k = \sum_{v \in [n]} \mathbf{x}(v)^k$. However, since this random projection has

[10] Again, in [5]'s analysis, 4-wise independence hashing suffices to guarantee a good enough estimate but we need to assume they are i.i.d. here to talk about the exact distribution of the final state.

finite variance, the normalized sum $S/\sqrt{\sum_{v\in[n]}|\mathbf{x}(v)|^2}$ will converge to a complex Gaussian as the input vector is duplicated on disjoint supports. In the limit, the only information remaining in the sketch pertains to the L_2-moment. Indeed, one in fact needs the number of i.i.d. registers to grow polynomially in the support-size for it to work for F_k [87].

By the generalized central limit theorem, for whatever random variable Q that is produced by the current input $\mathbf{x}$, if for some sequences $(a_w)_{w\in\mathbb{N}}$ and $(b_w)_{w\in\mathbb{N}}$, $(M_w - b_w)/a_w$ converges to some non-degenerate random variable Y as $w\to\infty$, then Y has to be α-stable for $\alpha\in(0,2]$. If in addition, $\mathbb{V}Q<\infty$, then Y is Gaussian. Non-Gaussian stable distributions are discussed later in Section 3.7.3.

3.7.2 Multi-Level Subsampling and the Poisson Limit Theorem

Another important sketching technique is *subsampling*. The idea is to devise a sketch that works for $\Theta(m)$ elements and then solve the generic case by subsampling the stream at rates 2^{-k} for $k\in\mathbb{N}$, one of which reduces it down to $\Theta(m)$ elements. Without loss of generality, suppose now we have m non-zero elements with distinct values $\mathbf{x}(1),\ldots,\mathbf{x}(m)$. Once again, let $\mathbf{x}(w)\in\mathbf{R}^{nw}$ be $\mathbf{x}$ repeated w times on disjoint supports. To obtain a $\Theta(m)$-size set of subsamples, one needs to subsample $\mathbf{x}(w)$ with rate $1/w$. Let $Y_{w,j}$ be the indicator that the jth copy of $\mathbf{x}(1)$ is sampled where $\mathbb{E}Y_{w,j}=1/w$. The number of elements with value $\mathbf{x}(1)$ in the subsampled set is

$$\sum_{j=1}^{w} Y_{w,j} \to \text{Poisson}(1) \text{ in distribution as } w\to\infty,$$

where we have invoked the Poisson limit theorem (see, e.g., [74, Theorem 3.6.1]) because $\mathbb{E}\sum_{j=1}^{w}Y_{w,j}=1$ and $\max_{j=1}^{w}\mathbb{P}(Y_{w,j}\neq 0)=\frac{1}{w}\to 0$ as $w\to\infty$. Similarly, the number of elements with value $\mathbf{x}(j)$ is also Poisson(1) and the occurrences of different values are independent. It is well known that such a limiting distribution can be simulated algorithmically by duplicating every element a Poisson(1) number of times (see [83, 82, 137, 160]). On the other hand, such limit distributions can be equivalently simulated by hashing each element to a Poisson process and then sampling at different times. Thus, all sketches based on subsampling can be simulated by the Lévy-Tower (without mapping to the complex unit circle $\mathbb{T}$) with the corresponding (compound) Poisson processes.

3.7.3 Stable Random Variables and Stable Processes

Whenever Q has a finite variance, $(Q_1 + Q_2 + \ldots + Q_w)/\sqrt{w}$ goes to Gaussian as $w \to \infty$. Nevertheless, with infinite variance, Q can lie in the domain of attraction of an α-stable distribution for any $\alpha \in (0, 2)$. Indeed, the use of stable random variables is another sketching technique that directly corresponds to Lévy processes: α-stable processes. Similar to the Gaussian case (Section 3.7.1), α-stable processes are self-similar so there is no need to store the whole tower with exponentially spaced sample times 2^{-k}. Rather, it suffices to sample m independent processes at time $t = 1$.

Indyk [96] uses one-dimensional α-stable random variables for $\alpha \in (0, 2]$ to estimate the L_α-moment, and indeed, the characteristic exponent of the α-stable process is $f(x) = |x|^\alpha$. Ganguly, Bansal, and Dube [88] consider the higher dimensional case, where each element has d attributes. We now show how to reconstruct [88]'s sketch from the perspective of Lévy processes. The target function is

$$f : \mathbb{R}^d \to \mathbb{C}, \quad f(x) = \left(\sum_{j=1}^{d} |x_j|^p \right)^q,$$

where $p \in [0, 2], q \in (0, 1]$. Such f-moments are called $L_{p,q}$ hybrid moments in [88]. By the subordination theorem [113, page 197], this function is the characteristic exponent of a vector of d independent p-stable processes that are subordinated by a common q-stable subordinator. That is,

$$X_t = (X^{[1]}_{Z_t}, \ldots, X^{[d]}_{Z_t}),$$

where Z is a q-stable subordinator and the $(X^{[j]})$ are i.i.d. p-stable processes. Note that since $X^{[j]}$ is p-stable, we have $X^{[j]}_{Z_t} \sim Z_t^{1/p} X^{[j]}_1$. Thus, we sample X at time 1 and get

$$X_1 \sim Z_1^{1/p} \left(X_1^{[1]}, \ldots, X_1^{[d]} \right).$$

Take the inner product and we have, for $x \in \mathbb{R}^d$,

$$\langle x, X_1 \rangle = Z_1^{1/p} \sum_{j=1}^{d} x_j X_1^{[j]},$$

which is exactly the random projection defined in [88].

3.7.4 HyperLogLog, PCSA, and Pure Killed Processes

Cardinality sketches like HyperLogLog [82] and PCSA [83] that only allow increments also correspond to Lévy processes in a surprisingly natural way. We consider the number system $\mathbb{R} \cup \{\infty\}$, where for any $x \in \mathbb{R}$, $x + \infty = \infty + x = \infty$ and $\infty + \infty = \infty$. Such definitions extend to multiplication with natural numbers where $0 \cdot \infty = 0$ and $k \cdot \infty = \infty$ for any $k \in \mathbb{Z}_+$.

Definition 3.13 (pure killed processes)**.** A *pure killed process* $X = (X_t)_{t \in \mathbb{R}_+}$ with *kill rate* $c > 0$ is a Lévy process over $\mathbb{R} \cup \{\infty\}$ which can be simulated as follows.

- Sample a *kill time* $Y \sim \text{Exp}(c)$.
- $X_t = 0$ if $t < Y$ and $X_t = \infty$ otherwise.

In particular, we have $\mathbb{P}(X_t = \infty) = \mathbb{P}(Y \leq t) = 1 - e^{-ct}$.

Assume the insertion stream is $v_1, \ldots, v_T$, where each v_j is an element in the universe $[n]$. Let $X = (X_t)_{t \in \mathbb{R}_+}$ be a pure killed process with unit kill rate ($c = 1$). Similar to the turnstile case in Section 3.4, one stores the sum at time t,

$$C_t = \sum_{j=1}^{T} X_t^{(v_j)} = \sum_{v \in [n]} \mathbf{x}(v) \cdot X_t^{(v)},$$

where $X^{(v)}$s are i.i.d. copies of X. It is straightforward to see that S_k perfectly simulates a poissonized PCSA cell [83, 137] (Table 3.4).

Bit operation in cardinality sketches	Jumps of a pure killed process
$0 \vee 0 = 0$	$0 + 0 = 0$
$0 \vee 1 = 1$	$0 + \infty = \infty$
$1 \vee 0 = 1$	$\infty + 0 = \infty$
$1 \vee 1 = 1$	$\infty + \infty = \infty$

Table 3.4: HyperLogLog and PCSA as Lévy-Tower. HyperLogLog and PCSA can be considered Lévy-Tower sketches with pure killed processes where the bit-or operation $(\{0, 1\}, \vee)$ is simulated with the extended real numbers $(\{0, \infty\}, +)$. Note that, of course, such a reinterpretation does not upgrade HyperLogLog and PCSA to work over turnstile streams. The resulting Lévy-Tower is still incremental-only since $\infty - \infty$ is not defined.

In fact, one computes

$$\mathbb{E}e^{iC_t} = \prod_{v \in [n]} \mathbb{E}e^{i\mathbf{x}(v) \cdot X_t^{(v)}} = \left(\mathbb{E}e^{iX_t}\right)^{\|\mathbf{x}\|_0} \tag{3.12}$$

since $0 + \cdots + 0 = 0$ and $\infty + \cdots + \infty = \infty$. By Definition 3.13, $X_t = 0$ with probability e^{-t}, in which case $e^{iX_t} = 1$, and $X_t = \infty$ with probability $1 - e^{-t}$, in which case $e^{iX_t} = 0$.[11] Thus we have $\mathbb{E}e^{iX_t} = e^{-t}$. Inserting this back to Eq. (3.12), we have

$$\mathbb{E}e^{iC_t} = e^{-t\|\mathbf{x}\|_0}.$$

This coincides with the observation in Section 3.4, where the estimation target $\|\mathbf{x}\|_0$ lies right in the exponent of $\mathbb{E}e^{iC_t}$.

We note that HyperLogLog and PCSA both correspond to the pure killed process, with different couplings of cells on different levels. In particular, HyperLogLog stores $(C_1, C_{1/2}, C_{1/4}, \ldots)$ where $C_{2^{-j}}$s are sampled from *a single process* (exactly as in the Lévy-Tower) while PCSA stores $(C_1, C_{1/2}, C_{1/4}, \ldots)$ where $C_{2^{-j}}$s are independent. Since a pure killed process will remain at ∞ after its first jump, for HyperLogLog, $(C_1, C_{1/2}, C_{1/4}, \ldots)$ will always be in the form of a prefix of ones (∞s) followed by all zeros. Therefore, HyperLogLog only needs to store the length of the prefix of ones.

3.7.5 Uniform Random Projection and the Integral Lévy-Tower

Indyk's L_p-stable sketches [96] are able to estimate the L_p-moment for $p \in (0, 2]$ but do not handle the L_0-moment. Cormode, Datar, Indyk, and Muthukrishnan [59] approximate the L_0-moment by the L_α-moment for very small $\alpha > 0$. This scheme needs a bound on $\|\mathbf{x}\|_\infty$, as a single unbounded update can raise the estimate to infinity. Kane, Nelson, and Woodruff [110] take another approach: subsample the stream to a suitable level, randomly project each element within the field $\mathbb{Z}_p$ for a random prime $p > \epsilon^{-1} \log(\|\mathbf{x}\|_\infty)$, and then use an (insertion-only) F_0 estimator. The random projection approach [110] also requires a bound on $\|\mathbf{x}\|_\infty$. As an L_0-moment sketch it has a certain failure probability (e.g., $p|\mathbf{x}(v)$ for many v) but it perfectly estimates a related quantity: the $f_{L_0,p}$-moment for $f_{L_0,p}(x) = \mathbb{1}[p \nmid x]$.

We show how the Lévy-Tower reconstructs [110]'s trick by *pure computation*. The target function $f(k) = \mathbb{1}[p \nmid k]$ for $k \in \mathbb{Z}$ can be written as

$$f(x) = \frac{1}{p} \sum_{j=0}^{p-1} (1 - \cos(2\pi j x/p)).$$

The Lévy process with characteristic exponent f is the compound Poisson process with jumps uniformly chosen from $-2\pi(p-1)/p, \ldots, 0, \ldots, 2\pi(p-1)/p$. See Fig. 3.5. The corresponding

[11] Here we take the Cesàro limit $e^{i\infty} = \lim_{T\to\infty} \frac{1}{T} \int_0^T e^{it}\, dt = 0$.

Lévy-Tower reconstructs [110]'s sketch: subsample through Poisson processes and uniformly project over $\mathbb{Z}_p$ with the uniform random jumps. It is straightforward to see that the resulting Lévy-Tower is *integral* in the sense that $S_k \in \{2j\pi/p : j = 0, \ldots, p\}$ for any k and S_k can be identified by a $\mathbb{Z}_p$-value.

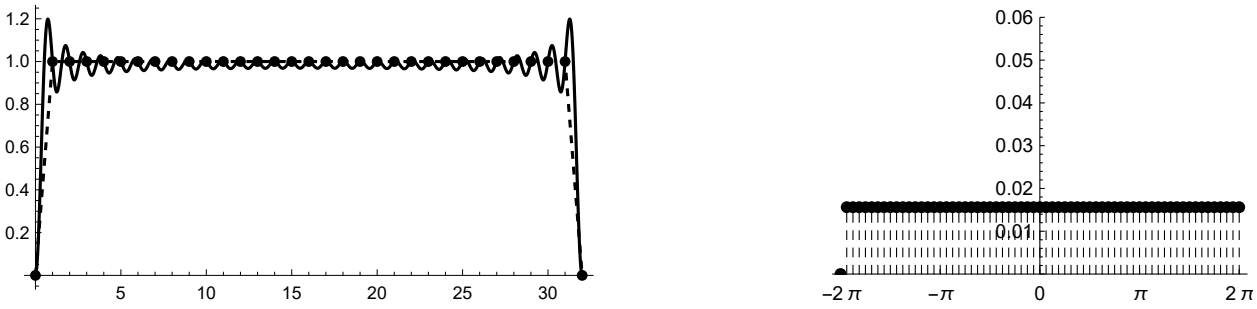


Figure 3.5: L_0 with Lévy-Tower. Left: $f_{L_0,32}(x) = \sum_{j=1}^{31} \frac{1}{32}(1-\cos(2\pi jx/32))$. The black dots mark the values at integer xs. Right: The jump distribution of the compound Poisson process X with characteristic exponent $f_{L_0,32}$. The subsampling and uniform random projection tricks used in [110] are recovered from computing the corresponding Lévy process.

3.8 Tractability and Lévy Processes

A function $f : \mathbb{Z} \to \mathbb{R}$ is *tractable* [19] if the f-moment of any input stream $\mathbf{x} \in \{-M, \ldots, M\}^n$, where $M = O(\text{poly}(n))$, can be $(1 \pm \epsilon)$-approximated with $O(\text{poly}(\epsilon^{-1}, \log n))$ space. Monotonic functions are fully characterized in [23] and all functions except for a class of nearly periodic functions are characterized in [19]. The Lévy-Tower tracks every function f that is the characteristic exponent of a Lévy process, and such exponents are characterized by the Lévy-Khintchine representation theorem. In this section, we investigate the set of tractable functions implied by Lévy-Khintchine.

We start a comparison between one-dimensional real characteristic exponents and existing communication complexity lower bounds. Note that by Lévy-Khintchine, such exponents can be written as $f(x) = \frac{1}{2}\sigma^2x^2 + 2\int_0^\infty (1 - \cos(xs))\,\nu(ds)$.

Lemma 3.14. *Let $f(x) = \frac{1}{2}\sigma^2x^2 + 2\int_0^\infty (1 - \cos(xs))\,\nu(ds)$. The following statements are true.*

- *$f(x) \geq 0$ for any $x \in \mathbb{R}$. [Commentary: functions with both positive and negative values require $\Omega(\text{poly}(n))$-size sketches for constant factor approximations [39].]*

- *For any $z \in \mathbb{Z}$ and $x \in \mathbb{R}$, $f(zx) \leq z^2 f(x)$. [Commentary: functions increasing faster than quadratic require $\Omega(\text{poly}(n))$-size sketches for constant factor approximations [5, 8].]*

- *For any $z \in \mathbb{R}_+$, $f(z) = 0$ if and only if $f(x+z) = f(x), \forall x \in \mathbb{R}$, i.e., f is periodic with period z. [Commentary: functions that have zeros other than the origin require $\Omega(\text{poly}(n))$-size sketches, unless the function is periodic with period $\min\{z > 0 : f(z) = 0\}$ [39].]*

Proof. The first statement is obvious. For the second statement, it suffices to prove that with $y = xs$,

$$(1 - \cos(zy)) \leq z^2(1 - \cos(y)),$$

for any $y \in \mathbb{R}, z \in \mathbb{Z}$. When $z = 0$, the statement trivially holds. Without loss of generality, assume $z \in \mathbb{Z}_+$ and let $g(y) = z^2(1 - \cos(y)) - (1 - \cos(zy))$. Check that $g'(y) = z^2 \sin(y) - z \sin(zy)$ and $g''(y) = z^2 \cos(y) - z^2 \cos(zy)$. Clearly for $y \in [0, \pi/z]$, we have $\cos(zy) \leq \cos(y)$ and thus $g''(y) \geq 0$. Note $g'(0) = 0$ and therefore $g(y) \geq 0$ for $y \in [0, \pi/z]$. Now consider $y \in [\pi/z, \pi]$, $z^2(1 - \cos(y)) \geq z^2(1 - \cos(\pi/z)) \geq 2 \geq 1 - \cos(zy)$, which implies $g(y) \geq 0$ on $[0, \pi]$. Thus we have $g(y) \geq 0$ for $[\pi, 2\pi]$ too by symmetry. The statement is thus proved since $g(y)$ is periodic in 2π.

For the third statement, $f(z) = 0$ necessarily implies that $\sigma = 0$ and the measure ν concentrates on $2\pi j/z$ for $j \in \mathbb{Z}$. Thus $f(x) = \sum_{j \in \mathbb{Z}} (1 - \cos(2\pi j x/z)) \nu(\{2\pi j/z\})$ is periodic with period z. □

Braverman, Chestnut, Woodruff, and Yang [19] characterize the tractability of all symmetric functions over $\mathbb{Z}$, except for a class of nearly periodic functions. Specifically, the following example is given in [19, §5] which cannot be solved in the L_2-heavy hitter-based framework, but can be estimated using *ad hoc* algorithmic tricks.

Definition 3.14 (g_{np}, a nearly periodic function [19, §5])**.** For $x \in \mathbb{N}$, $g_{np}(x) = 2^{-\tau(x)}$, where $\tau(x) = \max\{j \in \mathbb{N} : 2^j | x\}$, i.e., the position of the first "1" in the binary representation of x.

The reason that g_{np} cannot be tracked by finding L_2-heavy hitters is that $g_{np}(x)$ can occasionally becomes polynomially small in x. In particular, for $k \in \mathbb{N}$, $g_{np}(2^k) = 2^{-k}$. We now demonstrate how the function g_{np} can be estimated *directly* by the Lévy-Tower.

3.8.1 Estimating g_{np} with the Lévy-Tower

Since it is assumed the element values are at most poly(n), it suffices to consider a 2^w-periodic version of g_{np} where $w = O(\log(n))$.

Definition 3.15 (2^w-periodic g_{np})**.** For $x \in \mathbb{N}$,

$$g_{np,w}(x) = \begin{cases} g_{np}(x), & 2^w \nmid x \\ 0, & 2^w \mid x. \end{cases}$$

We show that, indeed, g_{np} corresponds to a Lévy process, specifically a compound Poisson process.

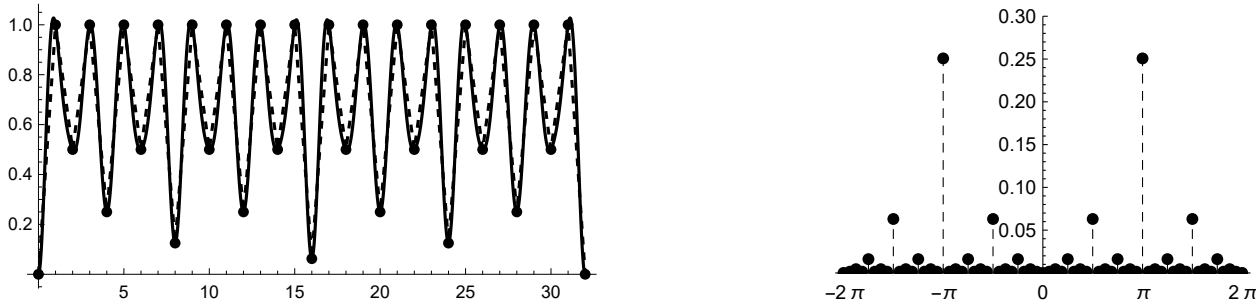


Figure 3.6: g_{np} with Lévy-Tower. Left: $g_{np,5}(x) = \sum_{j=1}^{31} \frac{2^{2\tau(j)+1}+1}{1536}(1 - \cos(2\pi jx/32))$. The black dots mark the values at integer xs. Right: The jump distribution of the compound Poisson process X with characteristic exponent $g_{np,5}$. Such nearly periodic functions do not fit in the L_2-heavy-hitter based framework in [19]. Nevertheless, one may compute the corresponding Lévy process and apply the Lévy-Tower.

Lemma 3.15. *Fix $w \in \mathbb{N}$. For any $x \in \mathbb{N}$,*

$$g_{np,w}(x) = \sum_{j=1}^{2^w-1} \frac{2^{2\tau(j)+1}+1}{3 \cdot 2^{2w-1}}(1 - \cos(2\pi jx/2^w)).$$

The jump rate is

$$\sum_{j=1}^{2^w-1} \frac{2^{2\tau(j)+1}+1}{3 \cdot 2^{2w-1}} = \frac{2^{2w}-1}{3 \cdot 2^{2w-1}}.$$

Remark 3.12. Here we analyze the periodic version $g_{np,w}$ which is useful to save bits. Only $O(w)$ bits are needed to store a cell in the Lévy-Tower for $g_{np,w}$. Note that the Lévy-Tower for g_{np} will have real-valued cells. See Section 4.9.1 for a direct analysis on g_{np}.

Proof of Lemma 3.15. This proof is by induction on w. It is true for $w = 1$ since

$$\begin{aligned} g_{np,1}(x) &= \frac{2+1}{3 \cdot 2}(1 - \cos(\pi x)) \\ &= \mathbb{1}\left[2 \nmid x\right]. \end{aligned}$$

Observe that for x such that $2 \mid x$,

$$\begin{aligned} g_{np,w+1}(x) &= \sum_{j=1}^{2^{w+1}-1} \frac{2^{2\tau(j)+1}+1}{3 \cdot 2^{2w+1}}(1 - \cos(2\pi jx/2^{w+1})) \\ &= \sum_{j=1}^{2^{w}-1} \frac{2^{2\tau(j)+1}+1}{3 \cdot 2^{2w+1}}(1 - \cos(2\pi jx/2^{w+1})) \\ &\quad + \sum_{j=2^w+1}^{2^{w+1}-1} \frac{2^{2\tau(j)+1}+1}{3 \cdot 2^{2w+1}}(1 - \cos(2\pi jx/2^{w+1})) \end{aligned}$$

Note that $\tau(j + 2^w) = \tau(j)$ for $j \in [2^w + 1, 2^{w+1} - 1]$ and $\cos(2\pi(j + 2^w)x/2^{w+1}) = \cos(2\pi jx/2^{w+1})$ since $2 \mid x$. Continuing,

$$= \frac{1}{4} g_{np,w}(x/2) + \frac{1}{4} g_{np,w}(x/2) = \frac{1}{2} g_{np,w}(x/2).$$

If $2 \nmid x$, then $\phi : j \mapsto jx$ is an automorphism of $\mathbb{Z}_{2^{w+1}}$. Thus

$$g_{np,w+1}(x) = \sum_{j=1}^{2^{w+1}-1} \frac{2^{2\tau(\phi^{-1}(j))+1}+1}{3 \cdot 2^{2w+1}}(1 - \cos(2\pi j/2^{w+1})).$$

Note that $2^r \mid jx$ if and only if $2^r \mid j$ since x is odd. This implies $\tau(\phi^{-1}(j)) = \tau(j)$. Continuing,

$$\begin{aligned} &= \sum_{j=1}^{2^{w+1}-1} \frac{2^{2\tau(j)+1}+1}{3 \cdot 2^{2w+1}}(1 - \cos(2\pi j/2^{w+1})) \\ &= g_{np,w+1}(1). \end{aligned}$$

Thus it suffices to prove $g_{np,w+1}(1) = 1$. Note that

$$g_{np,w+1}(1) = \frac{1}{2^w} \sum_{k=0}^{2^w-1} g_{np,w+1}(2k+1)$$

$$= \frac{1}{2^w} \sum_{j=1}^{2^{w+1}-1} \frac{2^{2\tau(j)+1}+1}{3 \cdot 2^{2w+1}} \sum_{k=0}^{2^w-1} (1 - \cos(2\pi j(2k+1)/2^{w+1})),$$

where

$$\sum_{k=0}^{2^w-1} (1 - \cos(2\pi j(2k+1)/2^{w+1})) = \sum_{k=0}^{2^w-1} (1 - \cos(2\pi jk/2^w + 2\pi j/2^{w+1}))$$

For $j \neq 2^w$ we see $\sum_{k=0}^{2^w-1} \cos(2\pi jk/2^w + 2\pi j/2^{w+1}) = 0$ by symmetry. If $j = 2^w$, we have $\cos(2\pi jk/2^w + 2\pi j/2^{w+1}) = -1$. Continuing, this is equal to

$$= \begin{cases} 2^{w+1} & \text{if } j = 2^w, \\ 2^w & \text{otherwise.} \end{cases}$$

We then have,

$$\begin{aligned} g_{np,w+1}(1) &= \sum_{j=1}^{2^{w+1}-1} \frac{2^{2\tau(j)+1}+1}{3 \cdot 2^{2w+1}} (1 + \mathbb{1}\left[j = 2^w\right]) \\ &= \sum_{j=1}^{2^w-1} \frac{2^{2\tau(j)+1}+1}{3 \cdot 2^{2w+1}} + \frac{2^{2\tau(2^w)+1}+1}{3 \cdot 2^{2w}} + \sum_{j=2^w+1}^{2^{w+1}-1} \frac{2^{2\tau(j)+1}+1}{3 \cdot 2^{2w+1}}. \end{aligned}$$

Noting that $\frac{2^{2\tau(2^{w-1})+1}+1}{3 \cdot 2^{2w+1}} = \frac{2^{2w-1}+1}{3 \cdot 2^{2w+1}} = \frac{1}{4}\frac{2^{2w+1}+4}{3 \cdot 2^{2w+1}}$ and $g_{np,w}(1) = \sum_{j=1}^{2^w-1} \frac{2^{2\tau(j)+1}+1}{3 \cdot 2^{2w-1}} (1 + \mathbb{1}\left[j = 2^{w-1}\right])$, this is equal to

$$\begin{aligned} &= \frac{1}{4} g_{np,w}(1) + \frac{3}{4}\frac{2^{2w+1}}{3 \cdot 2^{2w}} + \frac{1}{4} g_{np,w}(1) \\ &= 1, \end{aligned}$$

where the last line follows from the inductive hypothesis $g_{np,w}(1) = 1$. This proves the formula for $g_{np,w}$. Finally, we note that

$$1 = g_{np,w}(1) = \sum_{j=1}^{2^w-1} \frac{2^{2\tau(j)+1}+1}{3 \cdot 2^{2w-1}} (1 + \mathbb{1}\left[j = 2^{w-1}\right]),$$

and we conclude that

$$\sum_{j=1}^{2^w-1} \frac{2^{2\tau(j)+1}+1}{3 \cdot 2^{2w-1}} = 1 - \frac{2^{2\tau(2^{w-1})+1}+1}{3 \cdot 2^{2w-1}} = \frac{2^{2w}-1}{3 \cdot 2^{2w-1}}.$$

□

By Lévy-Khintchine, $g_{np,w}$ is the characteristic function of a compound Poisson process X with jump rate $\frac{2^{2w}-1}{3\cdot 2^{2w-1}}$ and jump J distributed as

$$\mathbb{P}(J = 2\pi j/2^w) = \mathbb{P}(J = -2\pi j/2^w) = \frac{1}{2}\frac{2^{2\tau(j)+1}+1}{2^{2w}-1}, \quad \text{for } j = 1, \dots 2^w - 1.$$

Moreover, the parameter w for the random jump induced by $g_{np,w}$ has a neat algorithmic interpretation as the max recursion depth if one implements each random jump by a recursive algorithm[12]. The case where $w \to \infty$ corresponds to the algorithm without depth limit. Let $\mathbb{B}$ be the set of all real fractions ($\in [0, 1)$) with finite binary representations. Note that the jump distribution converges pointwisely to

$$\mathbb{P}(J = 2\pi x) = \mathbb{P}(J = -2\pi x) = \frac{1}{2}2^{1-2\tau_*(x)},$$

for any $x \in \mathbb{B}$, where $\tau_*(x)$ returns the location of the first one in the binary representation. For example $\tau_*(1/4) = \tau_*((0.01)_2) = 2$. We may thus simulate such jumps by the following algorithm to handle unbounded input streams with dynamically allocated memory.

Lemma 3.16. *Let A, B be two random bit-tapes, embedded on reals with the binary expansion ($A = \sum_{j=1}^{\infty} A[j]2^{-j}, B = \sum_{j=1}^{\infty} B[j]2^{-j}$), and let $\xi \in \{-1, 1\}$ be a Rademacher random variable. Then*

$$J \sim 2\pi\,\xi \left(\sum_{j=1}^{\tau_*(B)-1} A[j]2^{-j} + 2^{-\tau_*(B)} \right).$$

3.8.2 The Fourier-Hahn-Lévy Method

The example above shows that it is possible to systematically sketch nearly periodic functions by sketching periodic functions and letting the period go to infinity. A natural conjecture is that *the class of tractable functions are precisely those with a Lévy-Khintchine representation*. This conjecture is *absolutely false*. We refute it with a simple tractable function having no Lévy-Khintchine representation, then attempt to salvage the conjecture via what we call the *Fourier-Hahn-Lévy* method.

[12](Halt with prob. 1/2; go left and repeat with prob. 1/4; and go right and repeat with prob. 1/4; see Fig. 3.6).

Example 6 (the 0-1-5 problem)**.** Consider the function f where $f(0) = 0$, $f(1) = 1$, $f(k) = 5$ for all $k \geq 2$. The f-moment is easy to estimate to within a $1 \pm \epsilon$ factor in $O(\epsilon^{-2} \operatorname{polylog}(n))$ space, using ℓ_0-sampling [60]. However, no Lévy process has its characteristic function equal to f, for otherwise we would have $f(2 \cdot 1) \leq 2^2 f(1)$ by Lemma 3.14, which implies the contradiction $5 \leq 4$.

We now extend our method to handle the *0-1-5 problem*. To be more concrete, we consider an arbitrary symmetric, real function f defined on $\{-p+1, \ldots 0, \ldots, p-1\}$ where $p > \|\mathbf{x}\|_\infty$ is the range of the values, say 2^{64}. We now describe the three-step *Fourier-Hahn-Lévy method*.

Fourier Transform. Decompose $f(0) - f(x) = \sum_{j=0}^{p-1} (1 - \cos(2\pi x j/p)) \hat{f}(j)$ where the coefficient $\hat{f}(j)$ can be computed by taking the Fourier transform of f. Note that $\hat{f}$ is real since we have assumed f to be real and symmetric.

Hahn Decomposition. Decompose $\hat{f} = \hat{f}_+ - \hat{f}_-$ where $\hat{f}_+$ and $\hat{f}_-$ are non-negative real functions.

Lévy Process Simulation. Define functions $f_{\widehat{+}}, f_{\widehat{-}}$ as the inverse Fourier transform of $\hat{f}_+, \hat{f}_-$ respectively. Since the Fourier transform is linear, we have $f = f_{\widehat{+}} - f_{\widehat{-}}$. By construction, $f_{\widehat{+}}, f_{\widehat{-}}$ are all characteristic exponents of some Lévy processes. Estimate those function moments separately with Lévy-Tower sketches and sum to yield an estimate of the f-moment.

Lemma 3.17. *Given any symmetric function $f : \mathbb{Z} \to \mathbb{R}_+$ with $f(0) = 0$ and any stream $\mathbf{x} \in [-p+1, p-1]^n$, the resulting estimator $\widehat{f(\mathbf{x})}$ of the Fourier-Hahn-Lévy method has error*

$$\left| \widehat{f(\mathbf{x})} - f(\mathbf{x}) \right| \leq O((|f_{\widehat{+}}(\mathbf{x})| + |f_{\widehat{-}}(\mathbf{x})|)/\sqrt{m}),$$

with probability at least 98/100.

If f is the characteristic exponent of a Lévy process, then $f = f_{\widehat{+}}$ and $f_{\widehat{-}} = 0$. This case was already considered, where we get an asymptotically unbiased estimate with multiplicative error. Clearly, whenever $|f_{\widehat{+}}(\mathbf{x})| + |f_{\widehat{-}}(\mathbf{x})| = O(|f(\mathbf{x})|)$, the Fourier-Hahn-Lévy method *still* returns an estimate with multiplicative error, which handles the *0-1-5 problem*, among others.

3.9 Summary

In this chapter we uncovered an intimate relation between streaming sketches and Lévy processes. The fundamental tool in the theory of Lévy processes—the Lévy-Khintchine rep-

resentation theorem—has thus been brought to bear on some central problems in theoretical computer science. The implications so far from the Lévy-Khintchine theorem are summarized as follows.

- With the Lévy-Khintchine theorem for generic Lévy processes, the Lévy-Tower sketch unifies the construction of all classic frequency moment sketches, broadens the class of known tractable functions, and also gives simple generalization to higher-dimensional streaming moments.
- With the Lévy-Khintchine theorem for subordinators, the Lévy-Min-Sampler presents a simple and elegant construction of incremental G-samplers with *precisely correct* sampling probabilities, producing many practical sketches (see Algorithms 1.3 and 3.3).

Is the Lévy-Khintchine representation theorem the final answer to the fundamental tractability problem [23, 19]? While this is possibly true for *continuous* functions, in Section 3.8 we show that there are tractable functions from $\mathbb{Z}$ to $\mathbb{R}$ that are not Lévy-Khintchine representable. Nevertheless, we introduced the *Fourier-Hahn-Lévy method*, which reduces an f without a Lévy-Khintchine representation to the difference of two functions $f_+^\wedge - f_-^\wedge$ that *can* be estimated by the Lévy-Tower.

Conjecture 3.1 (tractability over real streams)**.** *Assume non-zero entries have magnitudes at least* $\Omega(1/\operatorname{poly}(n))$*. If* $f\colon \mathbb{R} \to \mathbb{R}_+$ *is continuous and tractable, then* f *is Lévy-Khintchine representable.*

Conjecture 3.2 (tractability over integer streams)**.** *If* $f\colon \mathbb{Z} \to \mathbb{R}_+$ *is tractable, then* f *can be decomposed into* $f_+^\wedge - f_-^\wedge$ *where* $f_+^\wedge, f_-^\wedge$ *are Lévy-Khintchine representable and* $f_+^\wedge + f_-^\wedge \leq O(f \cdot \operatorname{poly}\log(n))$ *uniformly.*

The class $\mathcal{G}$ of functions G that can be handled by the Lévy-Min-Sampler is closed under summation and composition. We conjecture this class $\mathcal{G}$ characterizes all G-samplers in the $O(\log(n))$-bit regime.

Conjecture 3.3 (sampling over incremental streams)**.** *Assume all increments are in the range* $[1/\operatorname{poly}(n), \operatorname{poly}(n)]$*. Let* $G\colon \mathbb{R}_+ \to \mathbb{R}_+$*. If there exists a* $(0, 1/\operatorname{poly}(n))$*-approximate*[13] *G-sampler with* $O(\log n)$ *bits of space, then* $G \in \mathcal{G}$*, i.e.,* G *is the Laplace exponent of some subordinator.*

[13] 0 multiplicative error and $1/\operatorname{poly}(n)$ additive error. See Query 9.

3.10 Generalization

Before going to the next chapter, let me briefly demonstrate a few interesting generalizations of Lévy-Tower and Lévy-Min-Sampler.

3.10.1 Heterogeneous Moment Estimation

Imagine that given a turnstile stream $\mathbf{x} \in \mathbb{Z}^n$, one wants to estimate the *heterogeneous frequency moment*

$$\sum_{v \in [n]} |\mathbf{x}(v)|^{\alpha_v},$$

where $\alpha_v \in (0, 2]$ can be chosen differently for each element v. There is no easy modification of Indyk's L_p-sketch [96] to handle heterogeneous moments. One could store $C = \sum_{v \in [n]} \mathbf{x}(v) X^{(v)}$ where $X^{(v)}$ is a α_v-stable random variable, but the resulting sum is no longer a stable random variable and it is unclear how to estimate the heterogeneous moment from such a sketch. However, if $X^{(v)}$ is a α_v-stable *process*, then the aggregation C will remain as a Lévy process. The estimator of Lévy-Tower developed in Section 3.4 can, at its core, extract the characteristic exponent of a Lévy process from geometrically spaced samples. Thus the heterogeneous frequency moment can be estimated by Lévy-Tower with $C_t = \sum_{v \in [n]} \mathbf{x}(v) X_t^{(v)}$ where $X_t^{(v)}$ is sampled from of a α_v-stable process at time t. It is clear that Lévy-Tower works for this case with the same mean/variance guarantee, and the same proof. In general, one may assign any characteristic exponent $f^{(u)}$ to each element v and the Lévy-Tower sketch can estimate the *heterogeneous moment*

$$\sum_{u \in [n]} f^{(u)}(\mathbf{x}(u)).$$

Note that the Lévy-Tower stores the projection values in $\mathbb{T}$ and therefore can be generalized even for *heterogeneous streams*, where elements may have different state space. For example, consider a toy stream with domain size 3.

- Element 1 has value in $\mathbb{R}$, with statistic $f^{(1)}(x) = x^2$.
- Element 2 has value in $\mathbb{R}^2$, with statistic $f^{(2)}(x_1, x_2) = \sqrt{|x_1| + |x_2|}$.
- Element 3 has value in $\mathbb{F}_2$, the size-2 finite field, with statistic $f^{(3)}(r) = \mathbb{1}[r \neq 0]$.

The goal is to estimate the sum of the statistics

$$\begin{aligned} f^*(\mathbf{x}) &= f^{(1)}(\mathbf{x}(1)) + f^{(2)}(\mathbf{x}(2)) + f^{(2)}(\mathbf{x}(2)) \\ &= \mathbf{x}(1)^2 + \sqrt{|\mathbf{x}(2)_1| + |\mathbf{x}(2)_2|} + \mathbb{1}\left[\mathbf{x}(2) \neq 0\right]. \end{aligned}$$

It is sort of unimaginable to handle such streams with the previous sketching toolbox—one cannot even add the values of different elements as they are from different spaces. On the other hand, Lévy-Tower does not care about their spaces and the f^*-moment can be estimated by maintaining

$$C_t = \langle \mathbf{x}(1), X_t^{(1)} \rangle + \langle \mathbf{x}(2), X_t^{(2)} \rangle + \langle \mathbf{x}(3), X_t^{(3)} \rangle \mod 2\pi,$$

at geometrically spaced times t, where $X^{(v)}$, $v = 1, 2, 3$ are the corresponding Lévy processes with characteristic exponent $f^{(v)}$. Note that $X^{(v)}$ are processes over $\mathbb{R}, \mathbb{R}^2, \mathbb{F}_2$ respectively. By Lévy-Khintchine, one can simulate C_t by

$$C_t \sim t^{1/2}\mathbf{x}(1)X + t^{1/2}W(\mathbf{x}(2)_1 X_1 + \mathbf{x}(2)_2 X_2) + \mathbb{1}\left[U > e^{-t}\right]\mathbf{x}(3)Z\pi \mod 2\pi,$$

where $X \sim \text{Normal}(0, 1)$, X_1, X_2 are Cauchy random variables, W is a half-stable random variable, $U \sim \text{Uniform}(0, 1)$, Z is uniformly random over $\mathbb{F}_2$. Note that the deduced construction is clearly homomorphic for this heterogeneous stream. The random variables above are mutually independent. The Lévy-Tower sketch does not care about the elements' spaces and thus the function moment over heterogeneous streams can be estimated with the same guarantees and the same proof.

In theory, for a fixed domain $[n]$, one may use an n-dimensional Lévy processes X and maintain directly $\langle \mathbf{x}, X_t \rangle$ at geometrically space times t as the sketch. In this way, one can estimate stream statistics that are not just the sum of function-values of each element. For example, the symmetric norms studied by Błasiok et al. [14] contain many such statistics. However, this is moving away from practice as the domain size n in streaming is usually 2^{64}; it is not clear how a practical hash function can output hash values with a specific joint distribution over such a domain size. Thus I will stop here for this direction.

3.10.2 Stochastic Sampling Circuits

Let us see how the G-transformation technique (Section 3.5.3) can be used in a more general context. Just as currents and voltages can represent signals/numbers, one may consider an

exponential random variable $\sim \text{Exp}(\lambda)$ as a *signal* carrying information about its rate λ. These signals can be summed, scaled, and transformed as follows.

Summation. Given $Y_1 \sim \text{Exp}(\lambda_1), Y_2 \sim \text{Exp}(\lambda_2)$, $\min(Y_1, Y_2) \sim \text{Exp}(\lambda_1 + \lambda_2)$.

Scaling. Fix a scalar $\alpha > 0$. Given $Y \sim \text{Exp}(\lambda)$, $Y/\alpha \sim \text{Exp}(\alpha\lambda)$.

G-transformation. Given $Z \sim \text{Exp}(\lambda)$ and $Y \sim \text{Uniform}(0, 1)$, $\ell_G(Z, Y) \sim \text{Exp}(G(\lambda))$.

A *stochastic sampling circuit* is an object that uses *summation*, *scaling*, and *G-transformation* gates to sample according to functions with potentially many inputs. Such a circuit is represented by a directed acyclic graph (V, E) where V is the set of *gates* and $E \subset V \times V$ is a set of *wires*. There are four types of gates.

- An *input-gate* u receives a stream of incremental updates. Whenever it receives $\Delta > 0$, it generates, for *each* outgoing edge (u, v), a freshly sampled i.i.d. $Y \sim \text{Exp}(1)$ random variable and sends Y/Δ to v.
- A *scalar-gate* v is parameterized by a fixed $\alpha > 0$, and has a unique predecessor v' and successor v'', $(v', v), (v, v'') \in E$. Whenever v receives a $y \in \mathbb{R}_+$ from v' is sends y/α to v''.
- A *G-gate* v has a unique successor v', $(v, v') \in E$. It is initialized with a random seed $U \sim \text{Uniform}(0, 1)$. Whenever v receives a number $y \in \mathbb{R}_+$ from a predecessor u, it sends $\ell_G(y, U)$ to v', where ℓ_G is the Lévy-induced level function of $G \in \mathcal{G}$.
- An *output-gate* stores a pair (v_*, h_*), initialized as $(\perp, \infty)$. Whenever an output-gate receives a number $y \in \mathbb{R}_+$ from a predecessor with id $v \in V$, if $y < h_*$, then it sets $(v_*, h_*) \leftarrow (v, y)$.

The restriction that G-gates and scalar-gates have only one successor guarantees that the numbers received by one gate from different wires are independent. The generic G-Sampler (Algorithm 1.4) can be viewed as a *flat* stochastic sampling circuit (Fig. 3.7), where each element $v \in [n]$ has its own input-gate and G-gate. We could just as easily assign each input gate v to a G_v-gate, $G_v \in \mathcal{G}$, which would result in a *heterogeneous* sampler, where v is sampled with probability $G_v(\mathbf{x}(v))/\sum_u G_u(\mathbf{x}(u))$.

We illustrate how stochastic sampling circuits can be used to sample an edge from a graph with probability proportional to its weight. Let $H = ([n], E)$ be a fixed graph and $\mathbf{x} \in \mathbb{R}_+^n$ be a vector of vertex weights subject to incremental updates. For a fixed edge-weight

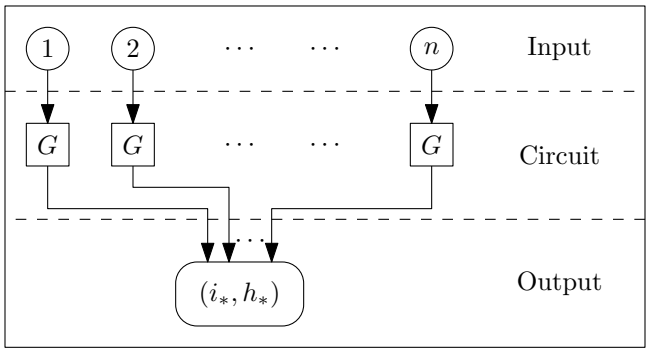


Figure 3.7: The flat stochastic sampling circuit corresponding to the generic G-Sampler (Algorithm 1.4).

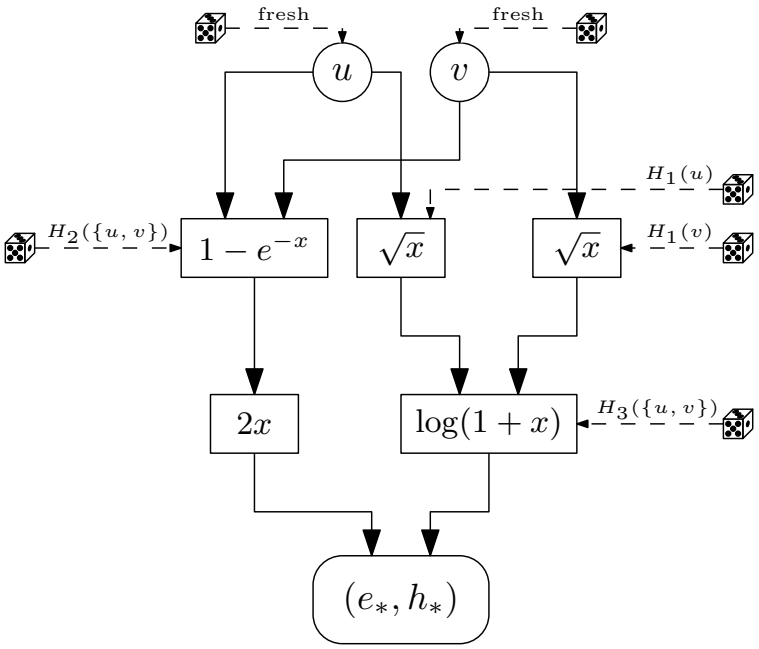


Figure 3.8: Circuit diagram for G-Edge-Sampler (Algorithm 3.4) with $G(a, b) = \log(1 + \sqrt{a} + \sqrt{b}) + 2(1 - e^{-(a+b)})$. Depicted are two of the inputs gates (u and v), the unique output gate, and all gates related to the sampling of edge $\{u, v\}$. The symbol 🎲 indicates the random seed $U \sim \text{Uniform}(0, 1)$ used by each G-gate, as well as the "fresh" exponential random variables $Y \sim \text{Exp}(1)$ generated by the input gates for each update and each output wire. The "$2x$"-gate is a deterministic scalar gate with $\alpha = 2$.

Algorithm 3.4: G-Edge-Sampler for $G(a,b) = \log(1+\sqrt{a}+\sqrt{b}) + 2(1-e^{-(a+b)})$.
Let $H = ([n], E)$ be a fixed graph that can be implicitly stored in a small space, e.g., a grid where the neighbors of a node can be inferred.

Sketch : $(e_*, h_*) \in E \times \mathbb{R}_+$, initialized as $(\perp, \infty)$
Hash function: specified below Eq. (3.13)
Level function: specified below Eq. (3.13)
Result: Sample an edge $e_* = (u_*, v_*)$ with
prob. $G(\mathbf{x}(u_*), \mathbf{x}(v_*)) / \sum_{\{u,v\}\in E} G(\mathbf{x}(u), \mathbf{x}(v))$

Update (v, Δ)**:**
  `// upon update` $\mathbf{x}(v) \leftarrow \mathbf{x}(v) + \Delta$
  for *each* $\{u, v\}$ *adjacent to* v **do**
    `// see Fig. 3.8 for how to derive the following expression`
    $h \leftarrow \min\{\ell_3(\ell_1(\mathrm{Exp}(\Delta), H_1(v)), H_3(\{u,v\})),\ \ell_2(\mathrm{Exp}(\Delta), H_2(\{u,v\}))/2\}$
    if $h < h_*$ **then**
      $(e_*, h_*) \leftarrow (\{u,v\}, h)$
    end
  end
end
Sample ( )**:**
  `// upon sample`
  return e_*
end

function $G : \mathbb{R}_+^2 \to \mathbb{R}_+$, where the weight of (u, v) is $G(\mathbf{x}(u), \mathbf{x}(v))$, we would like to select an edge (u_*, v_*) with probability exactly $G(\mathbf{x}(u_*), \mathbf{x}(v_*))/\sum_{(u,v)\in E} G(\mathbf{x}(u), \mathbf{x}(v))$. Our running example is a symmetric weight function that exhibits summation, scalar multiplication, and a variety of $\mathcal{G}$-functions.

$$G(a, b) = \log(1 + \sqrt{a} + \sqrt{b}) + 2(1 - e^{-(a+b)}). \tag{3.13}$$

The stochastic sampling circuit corresponding to G is depicted in Fig. 3.8. It uses the following level and hash functions.

- ℓ_1, level function for $G_1(x) = \sqrt{x}$. See Example 3.
- ℓ_2, level function for $G_2(x) = 1 - e^{-x}$. See Example 4.
- ℓ_3, level function for $G_3(x) = \log(1 + x)$. See Example 5.
- $H_1 : V \to (0, 1)$ uniformly at random.
- $H_2, H_3 : E \to (0, 1)$ uniformly at random.

The implementation of this circuit as a G-Edge-Sampler is given in Algorithm 3.4. Note that since G is symmetric, we can regard H as an undirected graph and let UPDATE(v, Δ) treat all edges incident to v in the same way. If G were not symmetric, the code for G-Edge-Sampler would have two **for** loops, one for outgoing edges $(v, u) \in E$, and one for incoming edges $(u, v) \in E$.

CHAPTER 4

Harmonic Decomposition of Queries

> In the early days of our field, data was intimately tied to the application programs that used it. Now we see that we want to break that tie. We want data that is independent of the application programs that use it–that is, data that is organized and structured to serve many applications and many users. What we seek is the *data base*.
>
> Charles W. Bachman [6], 1973

Before I attended the PODS23 conference, I focused on the `COUNT-DISTINCT` query over incremental streams, aka F_0. After I claimed how general and useful the τ-GRA framework (Section 5.8) is for producing statistically almost efficient estimates, Hung Ngo asked me whether this framework can also be used for turnstile streams, boosting the estimation quality of `COUNT-DISTINCT` with the presence of deletions. This question motivates me to apply the insights I have gained while studying F_0 to other more general queries in more general streaming models. Indeed, it turns out that the τ-GRA statistic in my work [160] is useful in the turnstile model as well as other streaming models in general.

In Chapter 3, the emphasis has been on *designing the sketching rule*, i.e., the distribution of the random-morphism Ψ in the homomorphic sketching system. The philosophy is to design the sketch rule specifically for one target query, so that an answer can be easily extracted from the sketch. In this chapter, the focus is switched to *designing the estimation*. The emphasis now is on finding ways to extract information from the sketch with the most classic sketching technique—subsample and aggregate (Flajolet and Martin [80]). See Table 4.1. An attractive feature of this approach is that, all queries are answered using the *same* sketch, and therefore the sketch now becomes a generic-purposed data summary that can be maintained to answer future queries without knowing them beforehand. Such sketching schemes are called *universal* [18] or *multi-objective* [52] which are often desirable in the context of data bases and network monitoring [120].

	Sketching	Estimation	Features
Chapter 3	complicated	simple	one-dimensional registers ☑ heterogeneous moments ☑ optimized for stable moments ☑
Chapter 4	simple	complicated	universal/multi-objective ☑

Table 4.1: Comparison between the sketches and estimators constructed in Chapter 3 and Chapter 4.

4.1 Prelude: Probabilistic Counting with Deletions

The inspiration for the technique developed in this chapter comes from a close study of cardinality estimation with deletions. This problem is referred as the L_0-*moment estimation* in the literature (by convention insertion-only cardinality estimation is referred as the F_0-*moment estimation*). The L_0-moment estimation has been studied in the standard model where Kane et al. [110] present a near-optimal (up to constants) algorithm. Nevertheless, it is not as well-understood as the F_0-moment estimation in the random oracle model in terms of the memory-variance trade-off (caring about statistical efficiency and constants).

4.1.1 Finite Field Random Projection

Kane, Nelson, and Woodruff [110] observed that "deletions" can be naturally done in the finite fields, since field-values have inverses, and "delete x" is equivalent to "insert $-x$." Fix a large prime p. The basic idea is to maintain subsampled registers $S_0, S_1, \ldots$, roughly as in PCSA ([83], see Section 5.1.2) such that for $j \in \mathbb{N}$

$$S_j = \sum_{v \in [n]} \mathbf{x}(v) Y_v \mathbb{1}\left[U_v < 2^{-j}\right],$$

where Y_vs are i.i.d. uniform random variables in the field $\mathbb{Z}_p$ and U_vs are i.i.d. Uniform$(0,1)$. This is clearly a homomorphic sketch and can be made base-2 scale-invariant (Definition 3.3) with poissonization. By the properties of finite fields, we have for $j \geq 0$

$$S_j \sim Y \mathbb{1}\left[\exists v \in [n], p \nmid \mathbf{x}(v) \text{ and } U_v < 2^{-j}\right],$$

where Y is uniform over $\mathbb{Z}_p$, because $Y + Y' \sim Y$ and $yY \sim Y$ if $p \nmid y$. By choosing a random large prime $p \approx \text{poly}(\epsilon^{-1} \log n)$, $\|\mathbf{x}\|_0$ can be restored with high probability [110]. I analyzed this sketch with any finite field F in the random oracle model in the technical

note [157], and use a LogLog-like estimator (based on 0-GRA), obtaining explicit constants: with m copies of the subsampling sequence $(S_j)_{j\in\mathbb{N}}$, one can get unbiased estimates with the following relative variance. See Table 4.2. With the random projection step, an empty

$\lvert F\rvert$	Relative Error
2	$1.638/\sqrt{m}$
3	$1.441/\sqrt{m}$
4	$1.387/\sqrt{m}$
2^8	$1.299/\sqrt{m}$
2^{32}	$1.298/\sqrt{m}$

Table 4.2: Relative error of PCSA over finite fields using LogLog-like estimator. When F is smaller, there are a significant fraction of "fake zeros" (see Fig. 4.1) and therefore the relative error is high (highest when $|F| = 2$). As $|F|$ goes to infinity, the chance of getting "fake zeros" goes to zero and F-PCSA estimator statistically converges to the LogLog estimator [73] for F_0. The relative error here when $|F|$ is large matches Durand and Flajolet's computation for the LogLog's relative error, which is $m^{-1/2}\sqrt{\frac{1}{12}\log^2 2 + \frac{1}{6}\pi^2} \approx 1.29806/\sqrt{m}$.

cell S_j will always be 0 (a "true zero") and a non-empty S_j is uniformly at random with a $1/|F|$ chance to be 0 (a "false zero"). See Fig. 4.1 for a visualization. Thus the constant can be improved by using HyperLogLog [82] or τ-GRA [160] when $|F|$ is large. The subtlety of optimizing constants here is that there does not seem to be a uniformly good estimation method for all finite fields F. For example, techniques in the F_0-moment estimation can be borrowed when $|F|$ is large but what should we do when $|F|$ is small? There are situations where $|F|$ is small. For example, the COUNT-ODD query (Query 2) corresponds to estimating cardinality over the finite field F with $|F| = 2$.

The random projection based approach discussed above is unsatisfying in the following aspects.

- The analysis fails if F is replaced with a finite group G that is not a field. This is because for $y \in \mathbb{Z}_+$ and Y uniform over G, $yY \not\sim Y$ whenever y divides $|G|$, where yY is defined as $\underbrace{Y + Y + \ldots + Y}_{y}$. A finite abelian group has a subgroup H of size y for every factor y and we have $yh = 0$ for any $h \in H$. Algorithmically, this means the random projection approach can count the number of elements that occur k times with $p \nmid k$ for any prime p, but *cannot* count (not by the same estimator and analysis, at least) the number of elements whose occurrence is not divisible, say, by 6.

- The random projection step makes the distribution of the sketch simple to describe. Nevertheless, from an information theoretic point of view, the distribution does not

11111111110100000000000	RRRRRRRRRR0R00000000000	10100000100100000000000
11111111011000000000000	RRRRRRRR0RR000000000000	01111010011000000000000
11111110100001000000000	RRRRRRR0R0000R000000000	00110000100001000000000
11111111111000000000000	RRRRRRRRRRR000000000000	00111000100000000000000
11111111100000000000000	RRRRRRRRR00000000000000	01100100100000000000000
11111111110100000000000	RRRRRRRRRR0R00000000000	01111000000000000000000
11111111100100000000000	RRRRRRRRR00R00000000000	11110000100100000000000
(a) PCSA	(b) PCSA over F	(c) PCSA over $\mathbb{F}_2$

Figure 4.1: PCSA with the finite field random projection trick. All three sketches are coupled in terms of what cells are occupied. In PCSA, one observes a concrete "1" if a cell is non-empty; with random projection over a finite field F, one observes a uniformly random F-valued "R" if a cell is non-empty, which have $1/|F|$ chance to be a "false zero." When $F = \mathbb{F}_2$, half of the "R" will be "false zeros," which in general incurs a higher relative variance in estimation.

need to be that nice for the cardinality estimation to be possible.

4.1.2 Probabilistic Counting over $\mathbb{Z}_3$

How should one estimate the cardinality without the random projection step? We now analyze such a sketch in the simple $\mathbb{Z}_3$-turnstile model. There are only two non-zero states in $\mathbb{Z}_3$-turnstile and therefore the state vector $\mathbf{x}$ can be identified by with the number of 1-valued elements λ_1 and the number of 2-valued elements λ_2. The vector is aggregated in a cell S_p of sub-sampling rate p, with poissonization (for infinite divisibility), i.e.,

$$S_p = \sum_{j=1}^{\lambda_1} X_j + \sum_{j=1}^{\lambda_2} 2Y_j,$$

where X_js and Y_js are i.i.d. Poisson(p) random variables. Clearly the distribution of S_p is not determined just by the cardinality $\lambda_1 + \lambda_2$ but depends on (λ_1, λ_2) in general. This moves us out of the simple point estimation framework we have stayed in when analyzing the random projection approach above: the estimation target is not the only parameter for the distribution of the final sketch state now. By direct computations, we have for $k \in \mathbb{Z}_3$, $\mathbb{P}(S_p = k \mod 3) = \frac{1}{3} + h_k(p)$, where

$$h_k(p) = \frac{2}{3} e^{-3p(\lambda_1+\lambda_2)/2} \cos(2k\pi/3 - \sqrt{3}p(\lambda_1 - \lambda_2)/2).$$

Note that $h_k(p)$s are the solutions for a damped oscillation (with linear drag) with three equally-spaced phases (as in three-phase electric power), considering p as the time. See Fig. 4.2. This harmonic-looking p.d.f. of S_p suggested to me to start trying out *harmonic statistics* like $\sin(2\pi S_p/3)$ or $\cos(4\pi S_p/3)$ and hope to have a simple closed-form expectation by maneuvering sums and products of trigonometrics around. Exploration along this direction eventually leads to a powerful estimation scheme that is able to "decode" a wide range of information about the stream from the simple subsample-and-aggregate sketch.

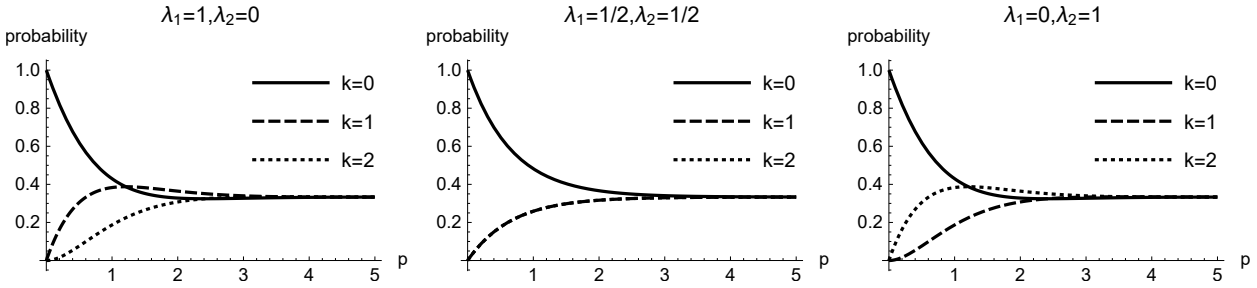


Figure 4.2: Probability distribution of S_p as the cell size p varies. p is normalized so that $\lambda_1 + \lambda_2 = 1$. Three cases are plotted: Left: all elements have value 1; Middle: half and half; Right: all elements have value 2. Observe that as p goes from 0 to ∞, S_p is initially 0 and becomes random noise eventually. When $\lambda_1 = \lambda_2$, the system corresponds a *critically damped oscillation*, which is exactly the case where the random projection step forces the sketch to be even when $\lambda_1 \neq \lambda_2$.

In this chapter, I will show how such harmonic statistics are useful in general. The theory I am going to develop here can lead to a satisfying solution to estimating cardinality over any (finite) *commutative monoid*. See Chapter 6.

4.2 Background

Let me quickly recap the basic setup. In the turnstile streaming model ([5, 59]), a *frequency vector* $\mathbf{x} = (\mathbf{x}(1), \ldots, \mathbf{x}(n)) \in \mathbb{Z}^n$ is initialized as 0^n and updated through a stream of pairs (v, Δ), $v \in [n] = \{1, 2, \ldots, n\}$ and $\Delta \in \mathbb{Z}$. Upon receiving (v, Δ), $\mathbf{x}(v) \leftarrow \mathbf{x}(v) + \Delta$. We consider the problem of estimating the *f-moment* $f(\mathbf{x})$, where $f\colon \mathbb{Z} \to \mathbb{R}_+$ is symmetric ($f(z) = f(-z)$) with $f(0) = 0$, and

$$f(\mathbf{x}) \coloneqq \sum_{v \in [n]} f(\mathbf{x}(v)). \qquad (f\text{-moment of stream } \mathbf{x})$$

Let M be the entry-wise frequency bound. We say an estimator V $(1 \pm \epsilon)$-*approximates* the f-moment, if for any $\mathbf{x} \in [-M, M]^n$,

$$\mathbb{P}\left((1-\epsilon)f(\mathbf{x}) \leq |V - f(\mathbf{x})| \leq (1+\epsilon)f(\mathbf{x})\right) \geq 2/3.$$

To simplify the presentation, we assume the usual regime where $\epsilon^{-1} \leq O(\text{poly}(n))$ and $M \leq O(\text{poly}(n))$. We define two important notions—*tractability* and *universality*—regarding the f-moment estimation problem.

Tractability of functions We say a function f is *tractable* [23] if for any fixed $\epsilon > 0$, there exists a $(1 \pm \epsilon)$-approximation algorithm of the f-moment using $O_\epsilon(\text{polylog}(n))$ space. A central problem in the streaming literature is to characterize the class of *tractable* functions [5, 23, 19].[1]

Universality of sketches We say a sketch is $\mathcal{C}$-*universal*, where $\mathcal{C}$ is a class of functions, if for any $f \in \mathcal{C}$, a $(1 \pm \epsilon)$-approximation of the f-moment can be returned by the sketch. It is highly desirable to find sketches that are $\mathcal{C}$-universal for as large a $\mathcal{C}$ as possible [18, 120].

The *ultimate* goal for f-moment sketching is thus to find a "truly universal" sketch that can $(1 \pm \epsilon)$-approximate *all tractable f-moments* with optimal log factors in the space complexity. A prerequisite of this goal is a complete characterization of tractable functions.

The tractability of classic L_p-moments (f-moment with $f(z) = |z|^p$, $p \geq 0$)[2] is well-understood. On the one hand, L_p-moments with $p \in [0, 2]$ can be $(1 \pm \epsilon)$-approximated with $O(\epsilon^{-2} \log(n))$ bits of space [5, 96, 59, 110].[3] On the other hand, Bar-Yossef, Jayram, Kumar, and Sivakumar [8] proved that estimating the L_p-moment requires $\Omega(n^{1-2/p})$ bits, which is $\Omega(\text{poly}(n))$ space for any $p > 2$. Therefore, L_p is tractable if and only if $p \in [0, 2]$. Nevertheless, classic L_p-sketches ($p \in [0, 2]$) are all *single-purposed* where the sketch design depends crucially on the parameter p. Thus despite being very efficient in space (only one log factor), classic L_p-sketches are extremely *non-universal*.

To understand the tractability of other functions, the strategy has been to fully explore the class of all approximable functions by some "powerful" sketch $\mathcal{S}$ [23, 18, 19] and then hopefully prove any function that is not approximable by $\mathcal{S}$ cannot be approximated by any

[1] Here we use the tractability notion in [23] where a function is tractable if it can be approximated with polylog-space. A slightly different definition is used in [19] where functions are defined to be tractable if the space is sub-polynomial ($n^{o(1)}$).

[2] We define $|z|^0 = \mathbb{1}[z \neq 0]$.

[3] For L_0, only $O(\epsilon^{-2} \log n(\log \epsilon^{-1} + \log\log(nM)))$ bits are needed [110].

other polylog-space sketch either. Note that this strategy actually pursues tractability and universality at the same time, since once a function class $\mathcal{C}$ is proved to be approximable by the sketch $\mathcal{S}$, $\mathcal{S}$ is automatically a $\mathcal{C}$-universal sketch. We now recap the existing techniques.

L_0-sampling. A simple universal sketch is to sample m elements $\mathbf{x}(v_1), \dots, \mathbf{x}(v_m)$ uniformly from $\mathrm{supp}(\mathbf{x})$ and then estimate the f-moment by the empirical mean $V_f = \frac{\|\mathbf{x}\|_0}{m} \sum_{j\in[m]} f(\mathbf{x}(v_j))$. The sampling can be done with L_0-samplers (see [107], $O(\log^2 n)$ bits to get one L_0-sample) and the support size $\|\mathbf{x}\|_0$ can be estimated by L_0-sketches [59, 110]. This scheme is natural but the estimates are poor if f varies a lot. Specifically, for functions f such that $f(x) > 0$ when $x \neq 0$, the number m of samples needs to be $\Omega(\epsilon^{-2} \max_{j\in[M]} f(j)/\min_{j\in[M]} f(j))$ for V_f to be a $(1\pm\epsilon)$-approximation of the f-moment; see, Chestnut [39]. For functions decreasing on $[1, M]$, it is proved in [18] that all tractable moments can be estimated using this scheme. However, functions like $f(x) = |x|^p$ for $p \in (0, 2]$ will take $\Omega(M^p)$ number of L_0-samples to approximate.[4]

L_2-heavy hitters. A more powerful sketch, used in [23, 25, 19], is to use variations on Indyk and Woodruff's [98] method of collecting L_2-heavy hitters using CountSketch [34] at different subsampling levels, and estimate f-moments based on the f-values of the "recursive" heavy hitters. This technique is powerful enough to estimate all tractable f-moments, except for a class of *nearly periodic* functions [19]. One such function defined in [19, §5] is $g_{np}\colon \mathbb{Z} \to \mathbb{R}_+$.[5]

$$g_{np}(0) = 0 \quad \text{and} \quad g_{np}(x) = 2^{-\tau(|x|)}, \quad \text{where} \quad \tau(x) = \max\{j \in \mathbb{N} : 2^j | x\}. \tag{4.1}$$

For functions like g_{np}, the entries with large g_{np}-values are not necessarily L_2-heavy hitters, so sampling L_2-heavy hitters are generally not helpful. However, Braverman, Chestnut, Woodruff, and Yang [19, §5] gave an *ad hoc* sketch for approximating g_{np}-moments with $O(\epsilon^{-8} \log^{14} n \log M)$ space, showing that g_{np} is indeed tractable. The Lévy-Tower constructed in Section 3.4 can estimate a class of tractable nearly periodic functions but their sketch is single-purposed, with different f requiring different sketches.

We summarize the two main drawbacks of existing universal sketches as follows.

Drawback 1: Overhead for sampling While a single L_0-sampler is quite space-efficient ($O(\log^2 n)$ bits), the class of approximable functions using polylog number of uniform samples is very limited (e.g., it fails to estimate any L_p-moment, $p > 0$). The more

[4] All space lower bounds here is implicitly capped by $O(n)$, as one may always store a whole table of the elements.

[5] In other words, $\tau(x)$ is the least significant bit position in the binary representation of $x > 0$.

powerful L_2-heavy hitter based sketch in [19] handles more function moments but it occupies significantly more space ($O(h(M)\epsilon^{-2}\log^6 n \log\log n)$ bits[6]) as CountSketch is used on every subsampling level. The natural question is

Are such auxiliary sampling-related data structures necessary?

Drawback 2: Fundamental mismatch with nearly periodic functions The polylog-space overhead for sampling is non-ideal in practice, though it can be ignored when characterizing tractability. A more fundamental limitation is that, even if L_0 or L_2 samples are given for free, the sampling based method will use *exponentially* more space when approximating certain nearly periodic functions, in comparison to the best possible sketch. Essentially, sampling based methods will need $\text{poly}(n)$ samples if the function f ever drops polynomially ($f(x) < f(1)/\text{poly}(n)$ for some $x \in [-M, M]$). Even though most "polynomially dropping" functions are proved to be not tractable [18, 19], Braverman Braverman, Chestnut, Woodruff, and Yang [19] realize some of them can be estimated (e.g., g_{np}) using polylog-space by other non-sampling based methods. It is thus intriguing to ask

Is there a universal sketch that handles tractable nearly periodic functions as well?

The discussion above suggests that one must depart from sampling based methods to remove the overhead for sampling-related subroutines, and to further improve the universality of sketches.

My Contribution: A New *Harmonic* Approach for Universal Sketching Our main contribution is a whole new sketching scheme that does not use *any* explicit samples but relies on the *harmonic structure* of the target function f. Without any auxiliary data structure for sampling, the new SymmetricPoissonTower sketch is very space-efficient, occupying only $O(\epsilon^{-2}\log^2 n)$ bits of space. It can $(1\pm\epsilon)$-approximate all the natural functions including the common L_p-moments ($f(x) = |x|^p$, $p \in [0, 2]$), logarithmic-moment ($f(x) = \log(1+|x|)$), and "soft cap"-moment ($f(x) = 1 - e^{-|x|}$). Note that L_0-sampling based methods with matching space can only estimate f-moment with $\max_{j\in[M]} f(j)/\min_{j\in[M]} f(j) = O(1)$. Thus among the examples, the L_0-sampling method [18, 39] can only estimate the "soft cap"-moment but fails to estimate L_p-moments and the logarithmic-moment.[7] The pow-

[6] h is a sub-polynomial function so that for any $0 < x < y$, $f(y)/f(x) \in [1/h(y), (y/x)^2 h(y)]$. See [19] for further details.

[7] Assuming $M = O(\text{poly}(n))$, one needs $O(\epsilon^{-2}\log M)$ L_0-samples to estimate the logarithmic-moment. Thus the total space needed is $O(\epsilon^{-2}\log^3 n)$ which is one log factor more than the space used by the harmonic approach.

erful L_2-sampling framework [19] can handle all the examples above, but it requires $\Omega(\epsilon^{-2} \log^6 n \log\log n)$ bits.

More than saving polylog factors, the new harmonic approach also *natively* tracks a large class of nearly periodic functions, including g_{np}, for which the sampling based methods provably need polynomially many samples. Thus the new approach also makes significant progress towards the ultimate goal of finding a "truly universal" sketching scheme. See Fig. 4.4.

Apart from the technical contributions, the new SymmetricPoissonTower sketch is also conceptually novel. In contrast to the previous methods which *avoid* hash collisions,[8] the new harmonic approach *embraces* hash collisions. The key new ingredient is an insight, coming from the harmonic statistics discovered in Section 4.1.2, that enables us to infer information about f-moments from a cell with an *unknown* number of indices hashed to it.

4.2.1 Insight: How to Estimate f-Moments under Hash Collisions?

Imagine hashing every element in $[n]$ to a cell, independently with probability $p = 2^{-k}$. Let $J \subset [n]$ be the set of indices hashed to the cell and let $X = \sum_{v \in J} \mathbf{x}(v)$ be stored in the cell. When k and $\lambda = |\mathrm{supp}(\mathbf{x})|$ are large, $|J|$ tends to a Poisson distribution. For simplicity, we shall *Poissonize* the cell[9] where $|J| \sim \mathrm{Poisson}(p\lambda)$ indices $\{I_j\}_{1 \le j \le |J|}$ are selected uniformly at random from $\mathrm{supp}(\mathbf{x})$ *with replacement*, so X is precisely

$$X = \sum_{j=1}^{\mathrm{Poisson}(p\lambda)} \mathbf{x}(I_j), \tag{4.2}$$

where the I_js are i.i.d. copies of $I \sim \mathrm{Uniform}(\mathrm{supp}(\mathbf{x}))$. The goal is to estimate the f-moment $f(\mathbf{x}) = \sum_{v \in [n]} f(\mathbf{x}(v))$ from the observation X. We begin with the trivial but important observation that when f is *linear*, X betrays lots of useful information about

[8] For L_0-sampling, hash collisions refer to the usual notion of multiple elements hashed to one cell. For L_2-sampling, hash collisions refer to two or more elements with significant L_2-mass hashed into one cell (there can be many insignificant elements hashed to it which serve as "background noise" in CountSketch [34]).

[9] Poissonization is a standard technique in subsampling which simulates each update with a Poisson(1) number of updates. It simplifies the analysis without affecting either update time or space. See, e.g., [37].

$f(\mathbf{x})$. In particular,

$$\begin{aligned}
\mathbb{E}f(X) &= \mathbb{E}f\left(\sum_{j=1}^{\text{Poisson}(p\lambda)} \mathbf{x}(I_j)\right) && \text{By Eq. (4.2)} \\
&= \mathbb{E}\sum_{j=1}^{\text{Poisson}(p\lambda)} f\left(\mathbf{x}(I_j)\right) && \text{From the linearity of } f \qquad (\star) \\
&= p\lambda\mathbb{E}f(\mathbf{x}(I)) && \text{Linearity of expectation} \\
&= pf(\mathbf{x}) && f(\mathbf{x}) = \lambda\mathbb{E}f(\mathbf{x}(I))
\end{aligned}$$

Thus $f(X)/p$ is an unbiased estimator for $\lambda\mathbb{E}f(\mathbf{x}(I))$.[10]

The key insight. In the example above it is not important that f is linear *per se*, but that it is a *homomorphism* (allowing the step $(\star)$), in this case from $(\mathbb{Z}, +)$ to $(\mathbb{C}, +)$. Moreover, other homomorphisms are equally useful. For example, suppose that f is instead a homomorphism from $(\mathbb{Z}, +)$ to $(\mathbb{C}, \times)$, the *multiplicative* group of complex numbers, i.e., for any $x, y \in \mathbb{Z}$, $f(x + y) = f(x)f(y)$. We can still infer information about $\mathbb{E}f(\mathbf{x}(I))$ from $f(X)$, even when X is the sum of more than one coordinate of $\mathbf{x}$, since $f(X)$ is just the product of $f(\mathbf{x}(I_j))$s. We compute $\mathbb{E}f(X)$ as follows.

$$\begin{aligned}
\mathbb{E}f(X) &= \mathbb{E}f\left(\sum_{j=1}^{\text{Poisson}(p\lambda)} \mathbf{x}(I_j)\right) && \text{By Eq. (4.2)} \\
&= \mathbb{E}\prod_{j=1}^{\text{Poisson}(p\lambda)} f\left(\mathbf{x}(I_j)\right) && \text{Since } f : (\mathbb{Z}, +) \to (\mathbb{C}, \times) \text{ is a homomorphism} \qquad (\star\star) \\
&= e^{-p\lambda}\sum_{k=0}^{\infty}\frac{(p\lambda)^k}{k!}\mathbb{E}\prod_{j=1}^{k} f(\mathbf{x}(I_j)) && \text{Definition of Poisson}(p\lambda) \\
&= e^{-p\lambda}\sum_{k=0}^{\infty}\frac{(p\lambda)^k}{k!}(\mathbb{E}f(\mathbf{x}(I)))^k && \text{Since the } \mathbf{x}(I_j)\text{s are i.i.d.} \\
&= e^{p\lambda(\mathbb{E}f(\mathbf{x}(I))-1)} && \text{Taylor expansion of } \exp(p\lambda\mathbb{E}f(\mathbf{x}(I))) \\
&= e^{pf(\mathbf{x})} && f(\mathbf{x}) = \lambda\mathbb{E}(f(\mathbf{x}(I)) - 1)
\end{aligned}$$

[10]Here the point is to demonstrate that homomorphisms f allow one to exchange the order of aggregating elements and applying f over subsampled streams, which inspires our new approach in the next paragraph. Of course, when only linear moments are considered, one does not need to subsample—they can be maintained exactly by a linear aggregation.

Note that since a multiplicative homomorphism maps $0 \in (\mathbb{Z}, +)$ to $1 \in (\mathbb{C}, \times)$, we must calibrate $f(z)$ to $f^\dagger(z) = f(z) - 1$ when computing the f-moment so that $f^\dagger(0) = 0$. Therefore we have $f(\mathbf{x}) \coloneqq \sum_{v \in [n]} (f(\mathbf{x}(v)) - 1) = \lambda \mathbb{E}(f(\mathbf{x}(I)) - 1)$ in the last step.[11] Step $(\star\star)$ allows us to extract useful information from $f(X)$ about $f(\mathbf{x})$.

Now suppose that f is some function that can be decomposed into a linear combination of homomorphisms $f_1, f_2, f_3, \ldots$ from $(\mathbb{Z}, +)$ to $(\mathbb{C}, \times)$ after calibration, i.e., $f(z) = \sum_j (f_j(z) - 1)$. We can estimate each f_j-moment separately for each j and thereby obtain an estimate of the f-moment by combining the estimates of the components. See Figure 4.3 for a visualization of the estimation process.

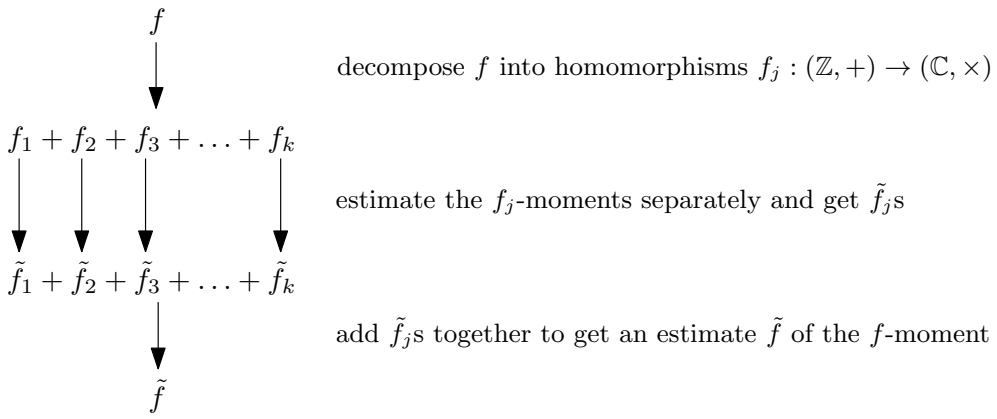


Figure 4.3: Diagram of the f-moment estimation process. Decompose f as $f_1 + \ldots + f_k$ where f_js are homomorphisms. For $j = 1, \ldots, k$, compute $\tilde{f}_j$ that estimates the f_j-moment of $\mathbf{x}$. Then $\tilde{f} = \tilde{f}_1 + \ldots + \tilde{f}_k$ is the estimator for the f-moment.

There is a famous tool to decompose functions into a linear combination of homomorphisms, namely the *Fourier transform*. Nevertheless, the Fourier transform requires f to be integrable which does not cover common functions like $f(x) = |x|^p$, $p \in (0, 2)$. A few more tricks are needed to construct the final sketch though, this decomposition-into-homomorphisms trick is the key to overcome the barriers that held back previous sampling based methods.

4.2.2 Organization

In Section 4.3, we give a technical overview with our formal results in Section 4.3.3. In Section 4.4, we introduce two additional tricks in the construction and estimation of the main

[11] In general, one may define the f-moment to be $\sum_{v \in [n]} (f(\mathbf{x}(v)) - f(0))$, which agrees with the classic definition $\sum_{v \in [n]} f(\mathbf{x}(v))$ when $f(0) = 0$.

sketch SymmetricPoissonTower. In Section 4.5, we formally define the SymmetricPoissonTower sketch with a full analysis of the estimation for individual harmonic moments. We analyze the combined estimators in Section 4.6. We assemble the proof of the main theorem in Section 4.7.

4.3 Technical Introduction & Related Work

4.3.1 (Symmetric) Harmonic Moments

We now formalize the estimation procedure. Note that the set of bounded homomorphisms from $(\mathbb{Z}, +)$ to $(\mathbb{C}, \times)$ are nothing secret but just $\{g_\gamma(z) = e^{i\gamma z} : \gamma \in [0, 2\pi)\}$.[12] The scheme in Fig. 4.3 decomposes $f(z)$ to $\int_0^{2\pi}(e^{i\gamma z} - 1)\,\mu(d\gamma)$ where μ is some signed measure. By convention [23, 19], we will focus on functions that are symmetric and non-negative. Thus for now, we will only consider functions f that can be decomposed to $\int_0^{2\pi}(1-\cos(\gamma z))\,\nu(d\gamma)$ where ν is a *positive* measure.

Definition 4.1 ((symmetric) harmonic moments)**.** For any $\gamma > 0$, define $f_\gamma(x) = 1-\cos(\gamma x)$. The *$\gamma$-harmonic moment* of a stream with frequency vector $\mathbf{x} \in \mathbb{Z}^n$ is

$$f_\gamma(\mathbf{x}) = \sum_{v\in[n]} f_\gamma(\mathbf{x}(v)) = \sum_{v\in[n]}(1 - \cos(\gamma\mathbf{x}(v))).$$

In other words, the γ-harmonic moment is the f_γ-moment.

The computation around $(\star\star)$ shows that the $e^{i\gamma z}$-moment can be estimated by poissonized subsampling. Since $\cos(\gamma z) = (e^{i\gamma z} + e^{-i\gamma z})/2$, the symmetric harmonic moments can be estimated similarly with *symmetric Poisson random variables*.

Definition 4.2 (symmetric Poisson distribution)**.** A symmetric Poisson random variable with rate λ distributes as the difference of two independent Poisson random variables with rate $\lambda/2$. This distribution is denoted SymmetricPoisson(λ).

Consider the following 1-cell linear sketch, parameterized by a *subsampling-level* $p > 0$.

$$R_p = \sum_{v\in[n]} \mathbf{x}(v)Y_v,$$

[12] Technically, $g_r \colon z \mapsto e^{rz}$ is also a multiplicative homomorphism parameterized by $r \in \mathbb{R}$. However, we will not use g_r since we only consider symmetric functions here. Nevertheless, g_r might be useful when considering incremental-only streams.

where $Y_v \sim \text{SymmetricPoisson}(p)$ are i.i.d. random variables, provided by a random oracle.[13] The following lemma symmetrizes the computation around $(\star\star)$.

Lemma 4.1. *For any $\gamma > 0$,*

$$\mathbb{E}e^{i\gamma R_p} = e^{-pf_\gamma(\mathbf{x})}.$$

Proof. By the computation around $(\star\star)$, for $Y \sim \text{SymmetricPoisson}(p)$ and $Z, Z' \sim \text{Poisson}(p/2)$, we have

$$\mathbb{E}e^{i\gamma Y} = \mathbb{E}e^{i\gamma(Z-Z')} = \mathbb{E}e^{i\gamma Z}\mathbb{E}e^{-i\gamma Z'} = e^{-\frac{p}{2}(1-e^{i\gamma})}e^{-\frac{p}{2}(1-e^{-i\gamma})} = e^{-p(1-\cos(\gamma))}. \tag{4.3}$$

We can now relate R_p to the f_γ-moment as follows.

$$\begin{aligned}
\mathbb{E}e^{i\gamma R_p} &= \mathbb{E}e^{i\gamma\sum_{v\in[n]}\mathbf{x}(v)Y_v} \\
&= \prod_{v\in[n]} \mathbb{E}e^{i\gamma\mathbf{x}(v)Y_v} && (Y_v\text{s are independent}) \\
&= \prod_{v\in[n]} e^{-t(1-\cos(\gamma\mathbf{x}(v)))} && (Y_v \sim \text{SymmetricPoisson}(t) \text{ and Eq. (4.3)}) \\
&= e^{-t\sum_{v\in[n]}(1-\cos(\gamma\mathbf{x}(v)))} = e^{-pf_\gamma(\mathbf{x})}. && (\text{Definition 4.1})
\end{aligned}$$

□

With the identity in Lemma 4.1, one may estimate the γ-harmonic moment $f_\gamma(\mathbf{x})$ by choosing a suitable level $p = \Theta(1/f_\gamma(\mathbf{x}))$. As the "angular rate" γ and the input $\mathbf{x}$ vary, the size of $f_\gamma(\mathbf{x})$ can vary and therefore we need to maintain R_p for $O(\log n)$ geometrically spaced levels p. We thus can $(1 \pm \epsilon)$-approximate the f_γ-moments for all $\gamma > 0$ with $O(\epsilon^{-2}\log n)$ cells.[14] Further details are in Section 4.5.

To show how this approach already significantly expands the universality, we prove that, surprisingly, for *almost all* f_γ-moments, sampling based methods [18, 19] need polynomial space (exponentially worse than the harmonic approach above!) because $1 - \cos(\gamma x)$ can occasionally drop to $O(1/\operatorname{poly}(x))$.

[13] Throughout the paper we work in the *random oracle* model, in which we can evaluate uniformly random hash functions $H : [n] \to [0, 1]$ and Y_v can be simulated through the randomness in $H(v)$. The hash function is *not* stored by the sketch but provided externally.

[14] Since the input $\mathbf{x}$ is integer-valued, $f_\gamma(\mathbf{x}) = f_{\gamma+2k\pi}(\mathbf{x})$ for any $k \in \mathbb{Z}$. We assume $f_\gamma(\mathbf{x}) = \Omega(1)$ here for simplicity.

Lemma 4.2. *Let $\gamma \in \mathbb{R}_+$ such that $\gamma/\pi \notin \mathbb{Q}$ and let $f_\gamma(x) = 1 - \cos(\gamma x)$. Then*

$$\forall k \in \mathbb{Z}_+, \exists w_k \in \{1, \ldots, k\}, f_\gamma(w_k) = O(1/k^2) \qquad (f_\gamma \text{ drops polynomially})$$

Remark 4.1. Recall that a function f drops polynomially if $0 < \min_{z \in [M]} f(z) \leq O(1/\operatorname{poly}(M))$ as $M \to \infty$.

Proof. We use the following classic pigeonhole argument. Let $\beta = \gamma/(2\pi)$. For any $k \geq 1$, by the pigeonhole principle, there exist distinct $a, b \in \{1, 2, 3, \ldots, k+1\}$, such that $|(a\beta - \lfloor a\beta \rfloor) - (b\beta - \lfloor b\beta \rfloor)| \leq 1/k$. Therefore,

$$\begin{aligned} f_\gamma(a-b) &= 1 - \cos((a-b)\gamma) = 1 - \cos((a\beta - b\beta)2\pi) \\ &= 1 - \cos(((a\beta - \lfloor a\beta \rfloor) - (b\beta - \lfloor b\beta \rfloor))2\pi) \\ &\leq 2\pi^2((a\beta - \lfloor a\beta \rfloor) - (b\beta - \lfloor b\beta \rfloor))^2 \leq 2\pi^2/k^2, \end{aligned}$$

where we used the fact that $1 - \cos(z) \leq z^2/2$ for any $z \in \mathbb{R}$. Define $w_k = |a-b| \in \{1, \ldots, k\}$ and we have $f_\gamma(w_k) \leq 2\pi^2/k^2$. □

In addition, the harmonic approach immediately gives an *explanation* why the function $g_{np}(x)$ in [19, §5] (defined in Eq. (4.1)) occasionally drops to $O(1/x)$ but is still tractable by some ad hoc sketch: the g_{np}-moment can be decomposed into harmonic moments.

$$g_{np}(x) = \frac{4}{3} \sum_{\gamma \in \mathbb{B}} (1 - \cos(2\pi\gamma x)) 2^{-2\tau_*(\gamma)} \qquad (\text{the nearly periodic function example})$$

The set $\mathbb{B}$ is the set of real numbers in $[0, 1)$ with finite binary representation. The function $\tau_*(\gamma) = \min\{j \in \mathbb{N} \mid 2^{-j}|\gamma\}$ returns the length of the binary representation of γ. See Section 4.9.1 for a justification of the decomposition of g_{np}. Given the decomposition, g_{np} can be automatically estimated by combining the estimates of harmonic moments and this approach is much more space-efficient than the ad hoc sketch in [19]. We will discuss why the estimates can be safely combined without blowing up the error in Section 4.3.2.

So far, it may seem that the harmonic approach is only useful in the theoretical pursuit of fully characterizing tractability. However, it turns out that the information scattered in the harmonic moments can be combined to answer meaningful queries (L_p, logarithm, etc.), making this approach an improved *practical* sketch *without any overhead for sampling*. We

now list the decompositions for common functions. See Fig. 1.3 for visualization.

$$
\begin{aligned}
|x|^p &= \int_0^\infty (1-\cos(\gamma x)) \frac{1}{(-\Gamma(-p))\cos(p\pi/2)\gamma^{1+p}}\, d\gamma && (L_p\text{-moment, } p \in (0,1)\cup(1,2)) \\
|x| &= \int_0^\infty (1-\cos(\gamma x)) \frac{2}{\pi\gamma^2}\, d\gamma && (L_1\text{-moment}) \\
\mathbb{1}[x \neq 0] &= \int_0^\pi (1-\cos(\gamma x)) \frac{1}{\pi}\, d\gamma && (L_0\text{-moment/support size, } x \in \mathbb{Z}) \\
1-e^{-r|x|} &= \int_0^\infty (1-\cos(\gamma x)) \frac{2r}{\pi(\gamma^2+r^2)}\, d\gamma && (\text{symmetric ``soft cap'' moment})
\end{aligned}
$$

As seen above, all L_p-moments ($p \in [0,2)$) are within the span of the harmonic moments. The symmetric version of Cohen's basis functions [50, 51] $G_r^*(x) = 1 - e^{-r|x|}$ are in the span of the harmonic moments, as well as their linear combinations. For example,

$$
\begin{aligned}
\log(1+|x|) &= \int_0^\infty (1-e^{-r|x|}) e^{-r} r^{-1}\, dr \\
&= \int_0^\infty \left[\int_0^\infty (1-\cos(\gamma x)) \frac{2r}{\pi(\gamma^2+r^2)}\, d\gamma \right] e^{-r} r^{-1}\, dr \\
&= \int_0^\infty (1-\cos(\gamma x)) \left[\int_0^\infty \frac{2e^{-r}}{\pi(\gamma^2+r^2)}\, dr \right] d\gamma && (\text{logarithmic-moment})
\end{aligned}
$$

The L_2-moment, on the other hand, lies on the "border" of the harmonic moments, since $|x|^2 = \lim_{\gamma\to 0} \frac{2}{\gamma^2}(1-\cos(\gamma x))$. Section 4.6.1 explains how to estimate the L_2-moment with the SymmetricPoissonTower.

4.3.2 Combination of Harmonic Moments

We now discuss the error of the combined estimator. Let $\{V_\gamma\}_{\gamma\in\mathbb{R}_+}$ be a family of estimates where V_γ is an approximation of the f_γ-moment. Let $f(x) = \int_0^\infty f_\gamma(x)\, \nu(d\gamma)$ where ν is some positive measure. We define the *combined estimator* of f as $\int_0^\infty V_\gamma\, \nu(d\gamma)$.

At first sight, it might look like the combined estimator can have a large error, since the estimates $\{V_\gamma\}_{\gamma\in\mathbb{R}_+}$ are *arbitrarily correlated*. They are produced with a *single* universal sketch! However, Cohen [50, 51] observed that as long as the *mean* and *variance* of each individual estimate V_γ have good guarantees, such guarantees can be transported to the combined estimator. In [50, 51], Cohen developed a universal sketch over *incremental streams* with a min-based sampling scheme.[15] In contrast to the turnstile model, incremental streams

[15] Min-based schemes are not applicable over turnstile streams.

have a frequency vector $\mathbf{x}$ in $\mathbb{N}^n$ and only positive updates are allowed. For any $r > 0$, let $G_r(x) = 1 - e^{-rx}$. Cohen's sketch has a collection of estimators $\{V_r^+\}_{r>0}$ such that for any $r > 0$,

$$\mathbb{E}V_r^+ = G_r(\mathbf{x}) = \sum_{v\in[n]}(1 - e^{-r\mathbf{x}(v)}) \tag{4.4}$$

$$\mathbb{V}V_r^+ = O(\epsilon^2)G_r(\mathbf{x})^2. \tag{4.5}$$

For any positive measure ν on $(0,\infty)$, let $G_\nu(x) = \int_0^\infty G_r(x)\,\nu(dr)$.[16] Cohen noted [50, 51] that

$$\begin{aligned}
\mathbb{E}\int_0^\infty V_r^+\,\nu(dr) &= \int_0^\infty \mathbb{E}V_r^+\,\nu(dr) = \sum_{v\in[n]} G_\nu(\mathbf{x}(v)) = G_\nu(\mathbf{x}) && \text{(Eq. (4.4))}\\
\mathbb{V}\int_0^\infty V_r^+\,\nu(dr) &= \int_0^\infty\int_0^\infty \mathrm{Cov}(V_r^+, V_{r'}^+)\,\nu(dr)\nu(dr')\\
&\le \int_0^\infty\int_0^\infty \sqrt{\mathbb{V}V_r^+}\sqrt{\mathbb{V}V_{r'}^+}\,\nu(dr)\nu(dr') && \text{(Cauchy-Schwartz)}\\
&= \left(\int_0^\infty \sqrt{\mathbb{V}V_r^+}\,\nu(dr)\right)^2\\
&= \left(\int_0^\infty O(\epsilon)G_r(\mathbf{x})\,\nu(dr)\right)^2 && \text{(Eq. (4.5))}\\
&= O(\epsilon^2)G_\nu(\mathbf{x})^2.
\end{aligned}$$

Therefore, by Chebyshev's inequality, $\int_0^\infty V_r^+\,\nu(dr)$ is a $(1\pm\epsilon)$-approximation of the G_ν-moment $G_\nu(\mathbf{x})$. Next, we will construct estimators whose mean and variance can be analyzed so that we can apply Cohen's argument to our harmonic framework.

4.3.3 Formal Statement of the Results

We now formally state the main theorem. We first define the function family

$$\mathcal{F}_* = \{f(x) = cx^2 + \int_0^\infty (1-\cos(\gamma x))\,\nu(d\gamma) \mid c\in\mathbb{R}_+, \nu \text{ is a positive measure}\},$$

which is the *mathematical* span of the harmonic moments as well as x^2. The simple take-away message is that the universal harmonic sketch is *almost* $\mathcal{F}_*$-universal: It can $(1\pm\epsilon)$-approximate any f-moment with $f\in\mathcal{F}_*$, conditioning on $f(\mathbf{x})\in[1/\operatorname{poly}(n), \operatorname{poly}(n)]$. Nev-

[16] For G_ν to exist, the measure ν has to satisfy that $\int_0^\infty \min(r,1)\,\nu(dr) < \infty$.

ertheless, to be fully rigorous, we need a few more technical conditions on f that guarantee $f(\mathbf{x}) \in [1/\operatorname{poly}(n), \operatorname{poly}(n)]$ for any $\mathbf{x} \in [-M, M]^n$.

For any $f \in \mathcal{F}_*$ with $f(x) = cx^2 + \int_0^\infty (1 - \cos(\gamma x))\, \nu(d\gamma)$, define $C_f = \int_0^\infty \min(\gamma^2, 1)\, \nu(d\gamma)$. Note that for any $x \in \mathbb{R}$, $f(x)$ is finite if and only if $C_f < \infty$. Recall $M \in \mathbb{Z}_+$ is the frequency bound. Define $\xi(f, M) = \min_{x \in [M]} \{f(x) \mid f(x) > 0\}$.

Theorem 4.1 (Universal Harmonic Sketch)**.** *There exists an $O(\epsilon^{-2} \log^2(n\epsilon^{-1}M))$-bit space sketch (SymmetricPoissonTower) such that for any turnstile stream with frequency vector $\mathbf{x} \in \{-M, \ldots, M\}^n$, the following statements are true.*

Combination *For any function $f \in \mathcal{F}_*$ such that $C_f/\xi(f, M) = O(\operatorname{poly}(nM\epsilon^{-1}))$, a $(1 \pm \epsilon)$-approximation of the f-moment can be returned with probability 2/3.*

Signed combination *For any functions $f, g \in \mathcal{F}_*$ such that $f(x) - g(x) = \Omega(f(x) + g(x))$ for $x \in [M]$ and $(C_f + C_g)/\xi(f - g, M) = O(\operatorname{poly}(nM\epsilon^{-1}))$, a $(1 \pm \epsilon)$-approximation of the $(f - g)$-moment can be returned with probability 2/3.*

Remark 4.2. In the usual region where $\epsilon^{-1} = O(\operatorname{poly}(n))$ and $M = O(\operatorname{poly}(n))$, the space is $O(\epsilon^{-2} \log^2 n)$. See Fig. 4.4 for a summary.

Remark 4.3. To further extend the universality, one may set $\tilde{h}(M) = (\max_{x \in [M]} \frac{f(x)+g(x)}{f(x)-g(x)})^2$. For a non-negative function $f^* = f - g$, the f^*-moment can be $(1 \pm \epsilon)$-approximated by the signed combination with $O(\tilde{h}(M) \cdot \epsilon^{-2} \log^2 n)$ bits of space.

We now discuss the parameters in the conditions of the main theorem.

Function constant C_f. For any *fixed* function $f \in \mathcal{F}_*$, C_f is a constant. For example, one may compute that for the L_p-moments where $f(x) = |x|^p$, the constants $C_{|x|^p} \le 2$ for any $p \in (0, 2)$ and for the L_0-moment, $C_{\mathbb{1}[x \neq 0]} < 1$. For the g_{np} function, $C_{g_{np}} \le \frac{2}{3}$; see Section 4.9.1. However, given that the sketch is universal, when a *family* of functions f are considered, the constraint $C_f = O(\operatorname{poly}(nM\epsilon^{-1}))$ may matter.

Drop rate $\xi(f, M)$. For any *increasing* function $f \in \mathcal{F}_*$, $\xi(f, M) = f(1)$ is a constant. The constraint $\xi(f, M) = \Omega(1/\operatorname{poly}(nM\epsilon^{-1}))$ only matters when f can become very small over the frequency range $\{1, \ldots, M\}$. Note that this condition in the main theorem is much weaker than the corresponding drop rate condition in previous techniques [23, 18, 19] which will fail whenever $\xi(f, M)$ is not polylog. In our main theorem, $\xi(f, M)$ can decrease with any polynomial rate. This allows the harmonic sketch to overcome the previous barrier to approximate g_{np} for which $\xi(g_{np}, M) = \Theta(1/M)$

and the golden harmonic function g_{gold} (Lemma 4.3, below) for which $\xi(g_{gold}, M) = \Theta(1/M^2)$.

Does the harmonic approach completely dominate the L_2-heavy hitter method [19] in terms of universality? This is a difficult statement to check as the criteria in [19] is a set of properties (slow-jumping, slow-dropping, and predictable) coming from a deep analysis of the Indyk-Woodruff subsample-and-CountSketch technique [99]. On the other hand, the new harmonic approach sets criteria on the harmonic structure of the functions. However, we do know the following.

- Common target functions like L_p-moments ($f(x) = |x|^p$, $p \in [0, 2]$), logarithmic-moment ($f(x) = \log(1 + |x|)$), and "soft cap"-moment ($f(x) = 1 - e^{-|x|}$) all have simple harmonic structures and thus can be tracked efficiently with the SymmetricPoissonTower sketch. Thus the SymmetricPoissonTower is a provably more space efficient universal sketch for the common tasks above, removing all sampling-related data structures.

- There exist functions that can be tracked with polylog space using the harmonic approach, while sampling based approaches need polynomial space. One example is the g_{np} function from [19], which has an *ad hoc* solution, but is covered by our main Theorem 4.1 using a much smaller space. For a new example, we also prove that the *golden harmonic moment* is not slow-dropping [19], but covered by Theorem 4.1. See Lemma 4.3 below (the proof is in Section 4.9.2).

In all, a provably "truly universal" sketch has not been found yet, but the harmonic approach is a significant step towards the final goal.

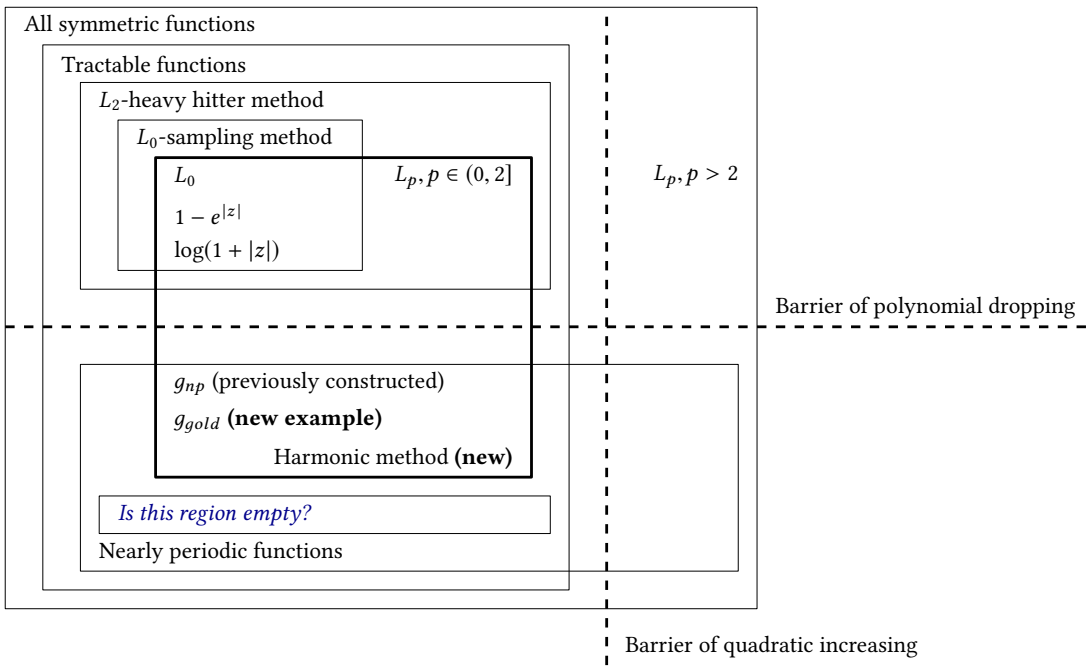
Lemma 4.3 (golden harmonic moment)**.** *Let* $g_{gold}(x) = 1 - \cos(\frac{1+\sqrt{5}}{2} \cdot 2\pi x)$ *for* $x \in \mathbb{Z}$*. Then*

- g_{gold} *is not slow-dropping since* $g_{gold}(1) > 1.7$ *and* $\xi(g_{gold}, M) = O(1/M^2)$*.*

- g_{gold} *is covered by Theorem 4.1 since* $C_{g_{gold}} = 1$ *and* $\xi(g_{gold}, M) = \Omega(1/M^2)$*.*

4.3.4 Related Work

Błasiok, Braverman, Chestnut, Krauthgamer, and Yang [14] further extend the Indyk-Woodruff [99] heavy hitter technique to estimate *symmetric norms* of the dynamic vector $\mathbf{x}$. Such heavy hitter based methods are proved to be optimal for all symmetric norms up to



sketch	space
L_0-sampling based [18]	$O(h^*(M) \cdot \epsilon^{-2} \log^2 n)$
L_2-heavy hitter based [19]	$O(h(M) \cdot \epsilon^{-2} \log^6 n \log \log n)$
ad hoc sketch for g_{np} [19]	$O(\epsilon^{-8} \log^{15} n)$
harmonic approach **(new)**	$O(\tilde{h}(M) \cdot \epsilon^{-2} \log^2 n)$

Figure 4.4: Summary of tractability and universality. Pivotal functions are shown to demonstrate the estimation power of different universal sketching schemes. See [23, 19] for L_2-heavy hitter based methods (h satisfies $f(y)/f(x) \in [1/h(y), (y/x)^2 h(y)]$ for any $0 < x < y \leq M$) and [18, 39] for L_0-sampling based methods ($h^*(n) \geq \max_{j \in [M]} f(j) / \min_{j \in [M]} f(j)$). The functions h and h^* describe the growing overhead to approximate harder and harder f-moments ("hard" with respect to the corresponding scheme). The harmonic approach has a different "boundary behavior" with overhead $\tilde{h}$ which is described in Remark 4.3. The g_{np} (defined in Eq. (4.1)) is constructed in [19] to show the existence of tractable functions with occasional polynomial dropping ($g_{np}(z) = 1/z$ if z is a power of 2). g_{gold} is newly constructed in Lemma 4.3 which occasionally drops quadratically ($g_{gold}(z) = O(1/z^2)$ for infinitely many z). It is proved in [8] that L_p is not tractable if $p > 2$.

polylog factors. However, they face the same barrier as [19] when trying to estimate generic f-moments that do not correspond to norms.

The idea of "solving" or "inverting" hash collisions is not entirely new. Especially in the incremental streaming model, Kumar, Sung, Xu, and Wang [116] have used the *Expectation Maximization* technique to iteratively estimate $\phi_1, \phi_2, \ldots$ where ϕ_j is the number of elements with value j. A more recent work by Chen, Indyk, and Woodruff [37] estimates $\phi_1, \phi_2, \ldots$ iteratively with dynamic programming. However, both [116] and [37] crucially rely on the incremental structure, so, e.g., if a "0" is observed, it must be an empty cell; if a "1" is observed, it has to be a singleton with value 1; if a "2" is observed, it is either the sum of two ones or a singleton with value two; and so on. It is not clear whether such techniques [116, 37] can be generalized to the turnstile model since if elements have both positive and negative values, then, e.g., a "0" cell can be either empty, $(1)+(-1)$, or $(2)+(-3)+(1)$, etc.—there are infinitely many combinations of integer-values to collide into a zero. From this perspective, our insight in Section 4.2.1 is really powerful and potentially provides a new simple scheme for incremental streams as well (with multiplicative homomorphism $f_r^*\colon z \mapsto e^{-rz}$ which is bounded now since the argument z is non-negative in the incremental model).

As discussed in Section 4.3.2, the idea of combining estimates of "basis moments" to estimate functions that are in the "span" has been first used by Cohen [50, 51] in the incremental setting. Nevertheless, the context is very different—Cohen uses min-based, multi-objective samplers to estimate the basis moments, whereas we consider the more generic turnstile streams (with both insertion and deletion) and our basis are the harmonic moments.

4.4 Implicit Level Selection and Smoothed Subsampling

The take-away message from Section 4.3.1 is that if one maintains the linear sketch $R_p = \sum_{v\in[n]} \mathbf{x}(v)Y_v$, where $Y_v \sim \text{SymmetricPoisson}(p)$, then for any $\gamma > 0$,

$$\mathbb{E}e^{i\gamma R_p} = e^{-pf_\gamma(\mathbf{x})}. \qquad \text{(Lemma 4.1)}$$

By selecting $p = \Theta(1/f_\gamma(\mathbf{x}))$, the γ-harmonic moment $f_\gamma(\mathbf{x})$ can be $(1 \pm \epsilon)$-approximated with $O(\epsilon^{-2})$ i.i.d. copies of R_p.

To minimize the space complexity, we will select the suitable level p *implicitly*.[17] Instead

[17]Kane, Nelson, and Woodruff faced a similar *level selection* problem in estimating distinct elements [110].

of using auxiliary data structures to select a proper level p, we maintain levels[18] R_p with $p = e^{-k/m}$, $k \in \mathbb{Z}$, and aggregate them as follows.

$$U_\gamma = \sum_{k\in\mathbb{Z}} (1 - e^{i\gamma R_{e^{-k/m}}}) e^{\frac{1}{3}k/m},$$

whose mean is[19]

$$\begin{aligned}
\mathbb{E}U_\gamma &= \sum_{k\in\mathbb{Z}} \mathbb{E}(1 - e^{i\gamma R_{e^{-k/m}}}) e^{\frac{1}{3}k/m} \\
&= \sum_{k\in\mathbb{Z}} (1 - e^{-e^{-k/m} f_\gamma(\mathbf{x})}) e^{\frac{1}{3}k/m} && \text{(Lemma 4.1)} \\
&= f_\gamma(\mathbf{x})^{1/3} \sum_{k\in\mathbb{Z}} (1 - e^{-e^{-(k - m\log f_\gamma(\mathbf{x}))/m}}) e^{\frac{1}{3}(k - m\log f_\gamma(\mathbf{x}))/m} \\
&\approx f_\gamma(\mathbf{x})^{1/3} \sum_{k\in\mathbb{Z}} (1 - e^{-e^{-k/m}}) e^{\frac{1}{3}k/m} && \text{(Shift } k \leftarrow \lfloor k - m\log f_\gamma(\mathbf{x})\rfloor)
\end{aligned}$$

Roughly speaking we have $\mathbb{E}U_\gamma \propto f_\gamma(\mathbf{x})^{1/3}$. Let $U_\gamma^{(1)}, U_\gamma^{(2)}, U_\gamma^{(3)}$ be three i.i.d. copies of U_γ. We know $U_\gamma^{(1)} U_\gamma^{(2)} U_\gamma^{(3)}$ will be a roughly unbiased estimator for $f_\gamma(\mathbf{x})$ after normalization. The reason that this sum implicitly chooses the correct levels is because the contribution to the mean decays exponentially outside the suitable levels.

- When $k \gg \log_2 f_\gamma(\mathbf{x})$, $\left(1 - e^{-2^{-k} f_\gamma(\mathbf{x})}\right) 2^{k/3} \approx 2^{-k} f_\gamma(\mathbf{x}) 2^{k/3} = f_\gamma(\mathbf{x}) 2^{-2k/3}$. Thus the contribution to the mean $\mathbb{E}U_\gamma$ is vanishing exponentially as $k \to \infty$.
- When $k \ll \log_2 f_\gamma(\mathbf{x})$, $\left(1 - e^{-2^{-k} f_\gamma(\mathbf{x})}\right) 2^{k/3} \approx 2^{k/3}$. Thus the contribution to the mean is also vanishing exponentially as $k \to -\infty$.

The construction of U_γ is similar to Pettie and Wang's [160] recent "τ-GRA" cardinality estimators for HyperLogLog and PCSA, which aggregate statistics at each level 2^{-k} with weight $2^{-\tau k}$, called *τ-aggregation*. U_γ can be considered a $(-1/3)$-aggregation in [160]'s framework.

Their approach would be to choose t from a *rough estimator* of $f_\gamma(\mathbf{x})$, which would return a large constant factor approximate in the range $[c_1 f_\gamma(\mathbf{x}), c_2 f_\gamma(\mathbf{x})]$ with probability $1 - \delta$. However, here the *variance* is complicated to bound with such explicit level selectors. A rough bound for the increase in variance for failed selections is $O(n^2)$ (note that $f_\gamma(\mathbf{x}) = O(n)$), which will require $\delta = O(1/\operatorname{poly}(n))$ to suppress. This will add an additional $\log(n)$ factor to the algorithm.

[18] While it is more common in the streaming literature [83] to maintain m i.i.d. cells at levels 2^{-k}, $k \in \mathbb{Z}$, here we sample a *single* cell at *finer* levels $e^{-k/m}$, $k \in \mathbb{Z}$. This trick is known as *smoothing* in [137], which is used for removing the oscillating component in the estimators.

[19] Later on we will establish that U_γ is absolutely convergent almost surely and the summation and expectation can be interchanged.

Remark 4.4. It may seem unnatural to first construct an estimator U_γ for $f_\gamma(\mathbf{x})^{1/3}$ and then estimate $f_\gamma(\mathbf{x})$ with three i.i.d. copies of U_γ. It would be simpler to set $U'_\gamma = \sum_{k\in\mathbb{Z}}(1 - e^{i\gamma R_{e^{-k/m}}})e^{k/m}$ or $U''_\gamma = \sum_{k\in\mathbb{Z}}(1 - e^{i\gamma R_{e^{-k/m}}})e^{\frac{1}{2}k/m}$, where by the same line of argument, one gets $\mathbb{E}U'_\gamma \propto f_\gamma(\mathbf{x})$ and $\mathbb{E}U''_\gamma \propto f_\gamma(\mathbf{x})^{1/2}$. However, we note that $\mathbb{E}U'_\gamma$ does not, in fact, exist, and $\mathbb{E}(U''_\gamma)^2$ also does not exist. Thus, for the estimator to have finite mean and variance, three copies are needed in this approach.

4.5 SymmetricPoissonTower

Recall that a SymmetricPoisson(λ) random variable distributes as the difference of two independent Poisson random variables with rate $\lambda/2$. We first list a few more properties of symmetric Poissons.

Lemma 4.4. *Let* $Y \sim$ SymmetricPoisson(λ).

- $\mathbb{E}Y = 0$, $\mathbb{E}Y^2 = \mathbb{V}Y = \lambda$, $\mathbb{E}Y^4 = \lambda + 3\lambda^2$, $\mathbb{V}Y^2 = \lambda + 2\lambda^2$.
- $\mathbb{E}e^{izY} = e^{-\lambda(1-\cos(z))}$ *for any* $z \in \mathbb{R}$.
- $\mathbb{P}(Y = 0) = e^{-\lambda/2}$.

Following the discussion in Sections 4.3.1 and 4.4, we now formally define the SymmetricPoissonTower sketch.

Definition 4.3. An *abstract SymmetricPoissonTower* is parameterized by a single integer m, which controls the variance of the estimates, and consists of an infinite vector $(X_j)_{j\in\mathbb{Z}}$ of *cells*, initialized as $0^{\mathbb{Z}}$. For any $j \in \mathbb{Z}$,

$$X_j = \sum_{v\in[n]} \mathbf{x}(v)Z_{v,j},$$

where $Z_{v,j} \sim$ SymmetricPoisson($e^{-j/m}$). All $Z_{v,j}$s are independent and assigned by the random oracle. In other words, Update(v, y) is implemented as follows.

Update(v, y)**:** For each $j \in \mathbb{Z}$, set $X_j \leftarrow X_j + Z_{v,j}y$, where $Z_{v,j}$s are independent Poisson($e^{-j/m}$) random variables.

The $[a, b)$-*truncated SymmetricPoissonTower* only stores $(X_j)_{j\in[a,b)}$.

We will first analyze this abstract (infinite) SymmetricPoissonTower and then prove that it suffices to store an $[a, b)$-truncated SymmetricPoissonTower with $b - a = O(m \log(nmM))$ cells, at a cost of $O(1/\operatorname{poly}(nmM))$ to the bias of the estimates. Let me recap the relevant notations below.

Notation	Definition	Name	Note
$\|\mathbf{x}\|_0$	$\sum_{v\in[n]} \mathbb{1}[\mathbf{x}(v) \neq 0]$	support size	$\mathbf{x} \in \mathbb{Z}^n$
$\|\mathbf{x}\|_p$	$(\sum_{v\in[n]} \|\mathbf{x}(v)\|^p)^{1/p}$	p-norm	$\mathbf{x} \in \mathbb{Z}^n$, $p \in (0, \infty)$
$\|\mathbf{x}\|_\infty$	$\max_{v\in[n]} \|\mathbf{x}(v)\|$	maximum norm	$\mathbf{x} \in \mathbb{Z}^n$
$f_\gamma(\mathbf{x})$	$\sum_{v\in[n]}(1 - \cos(\gamma \mathbf{x}(v)))$	γ-harmonic moment	$\mathbf{x} \in \mathbb{Z}^n, \gamma \in \mathbb{R}_+$
$\mathbb{V}Z$	$\mathbb{E}((Z - \mathbb{E}Z)\overline{(Z - \mathbb{E}Z)})$	variance	complex random variable Z

4.5.1 $(-1/3)$-Aggregation

As discussed in Section 4.4, we will use the following statistic that selects suitable levels implicitly.

Definition 4.4 ($(-1/3)$-aggregation)**.** For any $\gamma \in \mathbb{R}_+$, define

$$U_\gamma = \sum_{k=-\infty}^{\infty} (1 - e^{i\gamma X_k}) e^{\frac{1}{3}k/m}.$$

To invoke Fubini's theorem when computing $\mathbb{E}U_\gamma$ and $\mathbb{E}U_\gamma^2$, we first prove the following lemma.

Lemma 4.5. *For any $\gamma \in \mathbb{R}_+$ and $m \in \mathbb{Z}_+$,*

- $\sum_{k=-\infty}^{\infty} \mathbb{E}|1 - e^{i\gamma X_k}| e^{\frac{1}{3}k/m} \leq 14m\|\mathbf{x}\|_0^{1/3}$,
- $\sum_{k=-\infty}^{\infty} \mathbb{E}|1 - e^{i\gamma X_k}|^2 e^{\frac{2}{3}k/m} \leq 44m\|\mathbf{x}\|_0^{2/3}$.

Proof. Recall that $Z_{v,k} \sim \text{SymmetricPoisson}(e^{-k/m})$ and thus by Lemma 4.4, $\mathbb{P}(Z_{v,k} = 0) = e^{-e^{-k/m}/2}$. Since $X_k = \sum_{v=1}^n Z_{v,k}\mathbf{x}(v)$,

$$\mathbb{P}(X_k = 0) \geq \mathbb{P}(\forall v \in \operatorname{supp}(\mathbf{x}), Z_{v,k} = 0) = e^{-e^{-k/m}\|\mathbf{x}\|_0/2}.$$

Note that $|1 - e^{i\gamma X_k}| \leq 2$. Thus $\mathbb{E}|1 - e^{i\gamma X_k}| \leq 2\mathbb{P}(X_k \neq 0) \leq 2(1 - e^{-e^{-k/m}\|\mathbf{x}\|_0/2})$ and $\mathbb{E}|1 - e^{i\gamma X_k}|^2 \leq 4\mathbb{P}(X_k \neq 0) \leq 4(1 - e^{-e^{-k/m}\|\mathbf{x}\|_0/2})$. Now we compute

$$\sum_{k=-\infty}^{\infty} \mathbb{E}|1 - e^{i\gamma X_k}| e^{\frac{1}{3}k/m}$$

$$
\begin{aligned}
&\leq 2 \sum_{k=-\infty}^{\infty} (1 - e^{-e^{-k/m}\|\mathbf{x}\|_0/2}) e^{\frac{1}{3}k/m} \\
&= 2 \sum_{k=-\infty}^{\infty} \eta_1(k/m; a, b, c)
\end{aligned}
$$

where η_1 is defined in Lemma 4.17 and $a = 0, b = \|\mathbf{x}\|_0/2, c = 1/3$

$$
\begin{aligned}
&\leq 2m \int_{-\infty}^{\infty} \eta_1(x; a, b, c)\, dx + 2 \left| m \int_{-\infty}^{\infty} \eta_1(x; a, b, c) - \sum_{k=-\infty}^{\infty} \eta_1(k/m; a, b, c) \right| \\
&\leq 2(-\Gamma(-1/3))m(\|\mathbf{x}\|_0/2)^{1/3} + 2(3 + \Gamma(2/3))(\|\mathbf{x}\|_0/2)^{1/3} \qquad \text{(Lemma 4.17(1,6))} \\
&\leq 14m\|\mathbf{x}\|_0^{1/3}.
\end{aligned}
$$

Along the same lines, we have

$$
\begin{aligned}
&\sum_{k=-\infty}^{\infty} \mathbb{E}|1 - e^{i\gamma X_k}|^2 e^{\frac{1}{3}k/m} \\
&\leq 4 \sum_{k=-\infty}^{\infty} \eta_1(k/m; a, b, c)
\end{aligned}
$$

where $a = 0, b = \|\mathbf{x}\|_0/2, c = 2/3$

$$
\begin{aligned}
&\leq 4(-\Gamma(-2/3))m(\|\mathbf{x}\|_0/2)^{2/3} + 4(6 + \Gamma(1/3))(\|\mathbf{x}\|_0/2)^{2/3} \qquad \text{(Lemma 4.17(1,6))} \\
&\leq 44m\|\mathbf{x}\|_0^{2/3}.
\end{aligned}
$$

□

We now compute the mean and variance of U_γ. Note that we will use the Gamma function $\Gamma(x)$, which is negative over $(-1, 0)$ and positive over $(0, 1)$. See Fig. 4.5.

Lemma 4.6 (mean of U_γ)**.** *For any $\gamma \in \mathbb{R}_+$,*

$$
m^{-1}\mathbb{E}U_\gamma = (-\Gamma(-1/3) + O(m^{-1}))f_\gamma(\mathbf{x})^{1/3},
$$

as $m \to \infty$.

Proof. By Lemma 4.5, we invoke Fubini's theorem and exchange the order of expectation

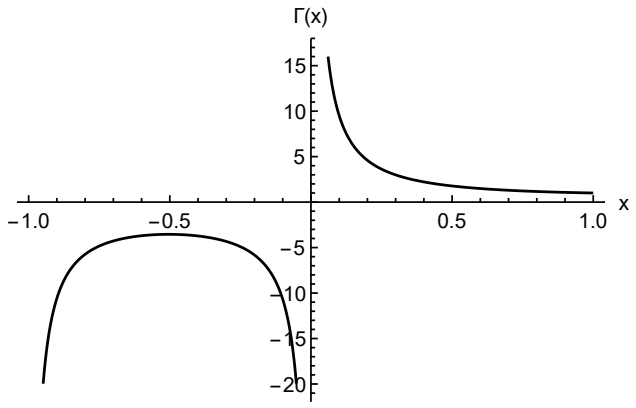


Figure 4.5: Gamma function $\Gamma(x)$ over $(-1,0)\cup(0,1)$

and sum.

$$\mathbb{E}U_\gamma = \sum_{k=-\infty}^{\infty}(1-\mathbb{E}e^{i\gamma X_k})e^{\frac{1}{3}k/m}$$

By Lemma 4.1,

$$\begin{aligned}&= \sum_{k=-\infty}^{\infty}(1-e^{-e^{-k/m}f_\gamma(\mathbf{x})})e^{\frac{1}{3}k/m}\\ &= \sum_{k=-\infty}^{\infty}\eta_1(k/m;a,b,c),\end{aligned}$$

where $a=0, b=f_\gamma(\mathbf{x}), c=1/3$. By Lemma 4.17, we know

$$\left|\mathbb{E}U_\gamma - mf_\gamma(\mathbf{x})^{1/3}(-\Gamma(-1/3))\right| \le (3+\Gamma(2/3))f_\gamma(\mathbf{x})^{1/3}.$$

□

Lemma 4.7 (variance of U_γ)**.** *For any $\gamma\in\mathbb{R}_+$,*

$$m^{-1}\mathbb{V}U_\gamma = 2^{2/3}(-\Gamma(-2/3))(1+O(m^{-1}))f_\gamma(\mathbf{x})^{2/3},$$

as $m\to\infty$.

Proof. We first check the integrability.

$$\sum_{k=-\infty}^{\infty}\sum_{j=-\infty}^{\infty}\mathbb{E}|1-e^{i\gamma X_k}||1-e^{i\gamma X_j}|e^{\frac{1}{3}(k+j)/m}$$

$$\leq \sum_{k=-\infty}^{\infty} \mathbb{E}|1-e^{i\gamma X_k}|^2 e^{\frac{2}{3}k/m} + (\sum_{k=-\infty}^{\infty} \mathbb{E}|1-e^{i\gamma X_k}|e^{\frac{1}{3}k/m})^2$$
$$\leq 240m^2\|\mathbf{x}\|_0^{2/3} < \infty. \qquad \text{(by Lemma 4.5)}$$

Then, by Fubini's theorem, we have

$$\begin{aligned}
\mathbb{V}U_\gamma = \mathbb{E}U_\gamma\overline{U_\gamma} - \mathbb{E}U_\gamma\mathbb{E}\overline{U_\gamma} &= \sum_{k=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} \mathbb{E}(1-e^{i\gamma X_k})(1-e^{-i\gamma X_j})e^{\frac{1}{3}(k+j)/m} \\
&\quad - \sum_{k=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} \mathbb{E}(1-e^{i\gamma X_k})\mathbb{E}(1-e^{-i\gamma X_j})e^{\frac{1}{3}(k+j)/m} \\
&= \sum_{k=-\infty}^{\infty} (1-|\mathbb{E}e^{i\gamma X_k}|^2)e^{\frac{2}{3}k/m} \\
&= \sum_{k=-\infty}^{\infty} (1-e^{-2e^{-k/m}[\mathrm{x};\gamma]})e^{\frac{2}{3}k/m} \\
&= \eta_1(k/m; 0, 2f_\gamma(\mathbf{x}), 2/3).
\end{aligned}$$

By Lemma 4.17, we know

$$\left|\mathbb{V}U_\gamma - m(2f_\gamma(\mathbf{x}))^{2/3}(-\Gamma(-2/3))\right| \leq (6+\Gamma(1/3))(2f_\gamma(\mathbf{x}))^{2/3}$$

□

4.5.2 Triple SymmetricPoissonTower

As in Section 4.4, the final trick is to store three independent copies of the SymmetricPoissonTower and obtain three i.i.d. copies of U_γ, denoted as $U_\gamma^{(1)}$, $U_\gamma^{(2)}$, $U_\gamma^{(3)}$. Define V_γ as

$$V_\gamma = (-\Gamma(-1/3))^{-3}m^{-3}U_\gamma^{(1)}U_\gamma^{(2)}U_\gamma^{(3)}.$$

Theorem 4.2 (triple SymmetricPoissonTower)**.** *Let $X = (X_k^{(j)})_{k\in\mathbb{Z},j=1,2,3}$ be a triple SymmetricPoissonTower. For any $\gamma \in \mathbb{R}_+$, let*

$$V_\gamma = (-\Gamma(-1/3))^{-3}m^{-3}\prod_{j=1}^{3}\left(\sum_{k=-\infty}^{\infty}(1-e^{i\gamma X_k^{(j)}})e^{\frac{1}{3}k/m}\right)$$

Then

$$\begin{aligned}\mathbb{E}V_\gamma &= (1+O(m^{-1}))f_\gamma(\mathbf{x})\\ \mathbb{V}V_\gamma &= (m^{-1}+O(m^{-2}))\cdot\frac{3\cdot 2^{2/3}(-\Gamma(-2/3))}{(-\Gamma(-1/3))^2}\cdot f_\gamma(\mathbf{x})^2.\end{aligned}$$

Proof. Recall that $(-\Gamma(-1/3))^3V_\gamma = m^{-3}U_\gamma^{(1)}U_\gamma^{(2)}U_\gamma^{(3)}$, where $U^{(j)\gamma}$s are i.i.d. copies of U_γ. We first compute the mean. By Lemma 4.6,

$$\begin{aligned}\mathbb{E}V_\gamma &= \left((-\Gamma(-1/3))^{-1}m^{-1}\mathbb{E}U_\gamma\right)^3\\ &= \left((1+O(m^{-1}))f_\gamma(\mathbf{x})^{1/3}\right)^3\\ &= (1+O(m^{-1}))f_\gamma(\mathbf{x}).\end{aligned}$$

We now compute the variance.

$$\begin{aligned}(-\Gamma(-1/3))^6\mathbb{E}V_\gamma\overline{V_\gamma} &= \left(m^{-2}\mathbb{E}U_\gamma\overline{U_\gamma}\right)^3\\ &= \left(m^{-2}\mathbb{V}U_\gamma + m^{-2}\mathbb{E}U_\gamma\mathbb{E}\overline{U_\gamma}\right)^3\\ (-\Gamma(-1/3))^6\mathbb{E}V_\gamma\mathbb{E}\overline{V_\gamma} &= \left(m^{-2}\mathbb{E}U_\gamma\mathbb{E}\overline{U_\gamma}\right)^3\\ (-\Gamma(-1/3))^6\mathbb{V}V_\gamma &= \left(m^{-2}\mathbb{V}U_\gamma + m^{-2}\mathbb{E}U_\gamma\mathbb{E}\overline{U_\gamma}\right)^3 - \left(m^{-2}\mathbb{E}U_\gamma\mathbb{E}\overline{U_\gamma}\right)^3\\ &= m^{-3}\left(m^{-1}\mathbb{V}U_\gamma\right)^3 + 3m^{-2}\left(m^{-1}\mathbb{V}U_\gamma\right)^2\left(m^{-2}\mathbb{E}U_\gamma\mathbb{E}\overline{U_\gamma}\right)\\ &\quad + 3m^{-1}\left(m^{-1}\mathbb{V}U\right)\left(m^{-2}\mathbb{E}U_\gamma\mathbb{E}\overline{U_\gamma}\right)^2.\end{aligned}$$

Note that by Lemma 4.7, $m^{-1}\mathbb{V}U_\gamma = (2^{2/3}(-\Gamma(-2/3))+O(m^{-1}))f_\gamma(\mathbf{x})^{2/3}$ and by Lemma 4.6, $|m^{-1}\mathbb{E}U_\gamma| = (-\Gamma(-1/3)+O(m^{-1}))f_\gamma(\mathbf{x})^{1/3}$. Thus the last term dominates and we have

$$\begin{aligned}&= 3m^{-1}\left(m^{-1}\mathbb{V}U_\gamma\right)\left(m^{-2}\mathbb{E}U_\gamma\mathbb{E}\overline{U_\gamma}\right)^2\left(1+O(m^{-1})\right)\\ &= 3m^{-1}\left(2^{2/3}(-\Gamma(-2/3))\right)f_\gamma(\mathbf{x})^{2/3}(-\Gamma(-1/3))^4 f_\gamma(\mathbf{x})^{4/3}\left(1+O(m^{-1})\right).\end{aligned}$$

We conclude that

$$\mathbb{V}V_\gamma = (m^{-1}+O(m^{-2}))\cdot\frac{3\cdot 2^{2/3}(-\Gamma(-2/3))}{(-\Gamma(-1/3))^2}\cdot f_\gamma(\mathbf{x})^2.$$

□

4.5.3 The Truncated Tower

We now consider a finite truncation $(X_k)_{a \le k < b}$. The discussion in Section 4.4 suggests that we only need to store an interval of cells in order to estimate the γ-harmonic moment $f_\gamma(\mathbf{x})$. Define the $[a, b)$-truncated $(-1/3)$-aggregation as

$$U_{\gamma;a,b} = \sum_{k=a}^{b-1} \left(1 - e^{i\gamma X_k}\right) e^{\frac{1}{3}k/m},$$

The statistic $U_{\gamma;a,b}$ can be obtained by maintaining only $(b-a)$ cells. We now formally show that the contribution to the first and second moments being truncated decays exponentially in both directions.

Lemma 4.8. *Let $A = U_\gamma$, $B = U_{\gamma;-\infty,a}$ and $C = U_{b,\infty}$. Define $\|\mathbf{x}\|_0 = \sum_{v\in[n]} \mathbb{1}\left[\mathbf{x}(v) \neq 0\right]$. Then*

1. $\mathbb{E}|A| \le \mathrm{poly}(\|\mathbf{x}\|_0 m)$.
2. $|B| \le e^{\frac{1}{3}a/m}\,\mathrm{poly}(\|\mathbf{x}\|_0 m)$.
3. $\mathbb{E}|C| \le e^{-\frac{2}{3}b/m}\,\mathrm{poly}(\|\mathbf{x}\|_0 m)$.
4. $\mathbb{E}|A|^2 \le \mathrm{poly}(\|\mathbf{x}\|_0 m)$.
5. $|B|^2 \le e^{\frac{2}{3}a/m}\,\mathrm{poly}(\|\mathbf{x}\|_0 m)$.
6. $\mathbb{E}|C|^2 \le e^{-\frac{4}{3}b/m}\,\mathrm{poly}(\|\mathbf{x}\|_0 m)$.
7. $\mathbb{E}|A||B| \le e^{\frac{1}{3}a/m}\,\mathrm{poly}(\|\mathbf{x}\|_0 m)$.
8. $\mathbb{E}|B||C| \le e^{\frac{1}{3}a/m - \frac{2}{3}b/m}\,\mathrm{poly}(\|\mathbf{x}\|_0 m)$.
9. $\mathbb{E}|C||A| \le e^{-\frac{2}{3}b/m}\,\mathrm{poly}(\|\mathbf{x}\|_0 m)$.

Proof.

1. Follow from Lemma 4.5 by noting that $\mathbb{E}|\sum_{k\in\mathbb{Z}}(1-e^{i\gamma X_k})e^{\frac{1}{3}k/m}| \le \sum_{k\in\mathbb{Z}}\mathbb{E}|1-e^{i\gamma X_k}|e^{\frac{1}{3}k/m}$.
2. $|B| = |\sum_{k=-\infty}^{a-1}(1-e^{i\gamma X_k})e^{\frac{1}{3}k/m}| \le 2\sum_{k=-\infty}^{a-1} e^{\frac{1}{3}k/m} = e^{\frac{1}{3}a/m}\cdot O(m)$.
3. $\mathbb{E}|C| \le \sum_{k=b}^{\infty}\mathbb{E}|1-e^{i\gamma X_k}|e^{\frac{1}{3}k/m}$. As in Lemma 4.5, $\mathbb{E}|1-e^{i\gamma X_k}| \le 2(1-e^{-e^{-k/m}\|\mathbf{x}\|_0/2}) \le e^{-k/m}\|\mathbf{x}\|_0$. Thus $\mathbb{E}|C| \le \sum_{k=b}^{\infty}\|\mathbf{x}\|_0 e^{-\frac{2}{3}k/m} = \|\mathbf{x}\|_0 e^{-\frac{2}{3}b/m}\cdot O(m)$.
4. Proved in Lemma 4.7.
5. By part 2.
6. As in Lemma 4.5, we have $\mathbb{E}|1-e^{i\gamma X_k}|^2 \le 4(1-e^{-e^{-k/m}\|\mathbf{x}\|_0/2}) \le 2e^{-k/m}\|\mathbf{x}\|_0$ and for $j \neq k$, $\mathbb{E}|1-e^{i\gamma X_k}||1-e^{i\gamma X_j}| = \mathbb{E}|1-e^{i\gamma X_k}|\mathbb{E}|1-e^{i\gamma X_j}| \le e^{-(k+j)/m}\|\mathbf{x}\|_0^2$. Thus

$$\mathbb{E}|C|^2 \le \sum_{j\ge b}\sum_{k\ge b}\mathbb{E}|1-e^{i\gamma X_j}||1-e^{i\gamma X_k}|e^{\frac{j+k}{3}/m}$$

$$
\begin{aligned}
&\leq \sum_{j\geq b}\sum_{k\geq b} e^{-(j+k)/m}\|\mathbf{x}\|_0^2 e^{\frac{j+k}{3}/m} \\
&= \|\mathbf{x}\|_0^2 (\sum_{j\geq b} e^{-\frac{2}{3}j/m})^2 \\
&= e^{-\frac{4}{3}b/m}\,\mathrm{poly}(\|\mathbf{x}\|_0 m).
\end{aligned}
$$

7. By parts 1 and 2.

8. By parts 2 and 3.

9. Follows from parts 4 and 6. Note that $\mathbb{E}|A||C| \leq \sqrt{\mathbb{E}|A|^2\mathbb{E}|C|^2}$ by Cauchy-Schwartz.

□

Theorem 4.3 (truncated harmonic estimator)**.** *For any $\gamma > 0$, define the truncated estimator*

$$
V_{\gamma;a,b} = m^{-3}(-\Gamma(-1/3))^{-3}\prod_{j=1}^{3}\left(\sum_{k=a}^{b-1}\left(1 - e^{i\gamma X_k^{(j)}}\right)e^{\frac{1}{3}k/m}\right)
$$

For any $a \leq 0 \leq b$,

$$
\begin{aligned}
|\mathbb{E}V_{\gamma;a,b} - \mathbb{E}V_\gamma| &\leq (e^{\frac{1}{3}a/m} + e^{-\frac{2}{3}b/m})\,\mathrm{poly}(\|\mathbf{x}\|_0 m) \\
|\mathbb{V}V_{\gamma;a,b} - \mathbb{V}V_\gamma| &\leq (e^{\frac{1}{3}a/m} + e^{-\frac{2}{3}b/m})\,\mathrm{poly}(\|\mathbf{x}\|_0 m).
\end{aligned}
$$

Proof. Let the $A^{(j)}$, $B^{(j)}$, and $C^{(j)}$ be i.i.d. copies of A, B, and C, defined in Lemma 4.8. Recall that the infinite SymmetricPoissonTower has estimator

$$
V_\gamma = \frac{A^{(1)}A^{(2)}A^{(3)}}{m^3(-\Gamma(-1/3))^3}
$$

and the truncated SymmetricPoissonTower has estimator

$$
V_{\gamma;a,b} = \frac{\left(A^{(1)} - B^{(1)} - C^{(1)}\right)\left(A^{(2)} - B^{(2)} - C^{(2)}\right)\left(A^{(3)} - B^{(3)} - C^{(3)}\right)}{m^3(-\Gamma(-1/3))^3}.
$$

We first bound their difference in the mean.

$$
\begin{aligned}
&|\mathbb{E}V_\gamma - \mathbb{E}V_{\gamma;a,b}| \\
&= \left|(\mathbb{E}(A - B - C))^3 - (\mathbb{E}A)^3\right| \\
&\leq 3(\mathbb{E}|A|)^2(\mathbb{E}|B| + \mathbb{E}|C|) + 3(\mathbb{E}|A|)(\mathbb{E}|B| + \mathbb{E}|C|)^2 + (\mathbb{E}|B| + \mathbb{E}|C|)^3
\end{aligned}
$$

According to Lemma 4.8, we know $\mathbb{E}|B| \le e^{\frac{1}{3}a/m}\,\mathrm{poly}(\|\mathbf{x}\|_0 m)$, $\mathbb{E}|C| \le e^{-\frac{2}{3}b/m}\,\mathrm{poly}(\|\mathbf{x}\|_0 m)$, and $\mathbb{E}|A| \le \mathrm{poly}(\|\mathbf{x}\|_0 m)$. Since we assume $a \le 0 \le b$, we have

$$\le (e^{\frac{1}{3}a/m} + e^{-\frac{2}{3}b/m})\,\mathrm{poly}(\|\mathbf{x}\|_0 m),$$

Now we bound the difference in the second moment.

$$\begin{aligned}
&\left|\mathbb{E}|V_\gamma|^2 - \mathbb{E}|V_{\gamma;a,b}|^2\right| \\
&= \left|\left(\mathbb{E}|A-B-C|^2\right)^3 - \left(\mathbb{E}|A|^2\right)^3\right| \\
&= \left|\left(\mathbb{E}(|A|^2+|B|^2+|C|^2 - A\overline{B} - A\overline{C} - B\overline{A} + B\overline{C} - C\overline{A} + C\overline{B})\right)^3 - \left(\mathbb{E}|A|^2\right)^3\right|,
\end{aligned}$$

where the dominating term $(\mathbb{E}|A|^2)^3$ is canceled and we have

$$= O((\mathbb{E}|A|^2)^2 \cdot (\mathbb{E}|A||B| + \mathbb{E}|A||C|))$$

Similarly by Lemma 4.8, we know $\mathbb{E}|A|^2 = \mathrm{poly}(\|\mathbf{x}\|_0 m)$, $\mathbb{E}|A||B| = e^{\frac{1}{3}a/m}\,\mathrm{poly}(\|\mathbf{x}\|_0 m)$, and $\mathbb{E}|A||C| = e^{-\frac{2}{3}b/m}\,\mathrm{poly}(\|\mathbf{x}\|_0 m)$. We thus have

$$\le (e^{\frac{1}{3}a/m} + e^{-\frac{2}{3}b/m})\,\mathrm{poly}(\|\mathbf{x}\|_0 m).$$

We have proved first and second moment have vanishing differences, which implies the difference in variance is also $(e^{\frac{1}{3}a/m} + e^{-\frac{2}{3}b/m})\,\mathrm{poly}(\|\mathbf{x}\|_0 m)$. □

4.6 Analysis of Combined Estimators

Let $\{V_{\gamma;a,b}\}_{\gamma>0}$ be the truncated harmonic estimators in Theorem 4.3. We just showed that the mean of $V_{\gamma;a,b}$ can be made arbitrarily close to the mean of V_γ by picking small enough a and large enough b. Choosing $a = -\Omega(m\log(mn))$ and $b = \Omega(m\log(mn))$, the difference $|\mathbb{E}V_{\gamma;a,b} - \mathbb{E}V_\gamma|$ becomes $O(1/\mathrm{poly}(mn))$. For a function $f(x) = \int_0^\infty (1-\cos(\gamma x))\,\nu(d\gamma)$, to estimate the f-moment one may try to combine the individual estimates as discussed in Section 4.3.2. However, note that

$$\begin{aligned}
\mathbb{E}\int_0^\infty V_{\gamma;a,b}\,\nu(d\gamma) &= \int_0^\infty (\mathbb{E}V_\gamma + O(1/\mathrm{poly}(mn)))\,\nu(d\gamma) && \text{(Theorem 4.3)} \\
&= \int_0^\infty ((1+O(m^{-1}))f_\gamma(\mathbf{x}) + O(1/\mathrm{poly}(mn)))\,\nu(d\gamma) && \text{(Theorem 4.2)}
\end{aligned}$$

$$= (1 + O(m^{-1}))f(\mathbf{x}) + O(1/\operatorname{poly}(mn)) \cdot \int_0^\infty \nu(d\gamma),$$

where in general $\int_0^\infty \nu(d\gamma)$ can be *infinite*. For example, to estimate the L_1-moment, we have $f(x) = |x| = \int_0^\infty (1 - \cos(\gamma x)) \frac{2}{\pi\gamma^2}\, d\gamma$. For this function f, we have $\int_0^\infty \frac{2}{\pi\gamma^2}\, d\gamma = \infty$. Recall that for $\int_0^\infty (1 - \cos(\gamma x))\, \nu(d\gamma)$ to be well-defined, one only needs $\int_0^\infty \min(\gamma^2, 1)\, \nu(d\gamma)$ to be finite. The reason that the attempt above fails is that, when γ is near 0, the density $\nu(d\gamma)$ is unbounded and any non-zero additive bias is too large for the combined estimator. We thus need to split f into two parts $f_{<\zeta}(x) = \int_0^\zeta (1 - \cos(\gamma x))\, \nu(d\gamma)$ and $f_{\geq\zeta}(x) = \int_\zeta^\infty (1 - \cos(\gamma x))\, \nu(d\gamma)$ for estimation, where ζ is to be determined later.

Before we proceed, we first define a quantity that measures the "size" of the function f.

Definition 4.5 (function constant C_f)**.** For any $f = \int_0^\infty (1 - \cos(\gamma x))\, \nu(d\gamma)$ where ν is a positive measure, define $C_f = \int_0^\infty \min(\gamma^2, 1)\, \nu(d\gamma)$. Note that for any $x \in \mathbb{R}$, $f(x)$ is finite if and only if $C_f < \infty$.

4.6.1 Estimation of the $f_{<\zeta}$-Moment

We first show that, when ζ is small enough, the $f_{<\zeta}$-moment can be approximated by the L_2-moment $\|\mathbf{x}\|_2^2$.

Lemma 4.9. *Recall that $f_{<\zeta}(x) = \int_0^\zeta (1 - \cos(\gamma x))\, \nu(d\gamma)$. The $f_{<\zeta}$-moment of $\mathbf{x} \in \mathbb{Z}^n$ can be bounded as follows.*

$$\left(1 - \frac{\zeta^2 \|\mathbf{x}\|_2^2}{12}\right) \cdot \frac{1}{2}\|\mathbf{x}\|_2^2 \int_0^\zeta \gamma^2\, \nu(d\gamma) \leq f_{<\zeta}(\mathbf{x}) \leq \frac{1}{2}\|\mathbf{x}\|_2^2 \int_0^\zeta \gamma^2\, \nu(d\gamma).$$

Proof. We first note the following approximation of $(1 - \cos(x))$. For any $x \in \mathbb{R}$,

$$x^2/2 - x^4/24 \leq 1 - \cos(x) \leq x^2/2.$$

On the one hand, we have

$$\begin{aligned}\sum_{v \in [n]} \int_0^\zeta (1 - \cos(\gamma \mathbf{x}(v)))\, \nu(d\gamma) &\leq \sum_{v \in [n]} \int_0^\zeta \gamma^2 \mathbf{x}(v)^2/2\, \nu(d\gamma) \\ &= \frac{1}{2}\|\mathbf{x}\|_2^2 \int_0^\zeta \gamma^2\, \nu(d\gamma).\end{aligned}$$

On the other hand,

$$\begin{aligned}\sum_{v\in[n]}\int_0^\zeta (1-\cos(\gamma\mathbf{x}(v)))\,\nu(d\gamma) &\geq \sum_{v\in[n]}\int_0^\zeta (\gamma^2\mathbf{x}(v)^2/2-\gamma^4\mathbf{x}(v)^4/24)\,\nu(d\gamma)\\ &= \frac{1}{2}\|\mathbf{x}\|_2^2\int_0^\zeta \gamma^2\,\nu(d\gamma)-\frac{1}{24}\|\mathbf{x}\|_4^4\int_0^\zeta \gamma^4\,\nu(d\gamma)\\ &\geq \frac{1}{2}\|\mathbf{x}\|_2^2\int_0^\zeta \gamma^2\,\nu(d\gamma)-\frac{1}{24}\|\mathbf{x}\|_4^4\zeta^2\int_0^\zeta \gamma^2\,\nu(d\gamma).\end{aligned}$$

Finally we note that $\|\mathbf{x}\|_4 \leq \|\mathbf{x}\|_2$. □

Thus by choosing $\zeta = O(1/\operatorname{poly}(m\|\mathbf{x}\|_2))$, $\frac{1}{2}\|\mathbf{x}\|_2^2\int_0^\zeta \gamma^2\,\nu(d\gamma)$ will be a $(1\pm O(m^{-1}))$-approximation of the $f_{<\zeta}$-moment. It suffices now to approximate the L_2-moment $\|\mathbf{x}\|_2^2$. The classic L_2-moment can be estimated with a separate AMS sketch [5]. Nevertheless, to demonstrate the universality of the SymmetricPoissonTower sketch, we will use its cells $X_0,\dots,X_{m-1}$ to estimate L_2, which are *not* independent from the harmonic estimators $\{V_{\gamma;a,b}\}$.

Lemma 4.10 (L_2-moment estimator)**.** *Let $(X_k)_{a\leq k<b}$ be a truncated symmetric Poisson tower with parameter m where $a\leq 0$ and $b\geq m$. Then we define the L_2-estimator,*

$$\Psi = \frac{1}{m}\sum_{k=0}^{m-1} X_k^2 e^{k/m}.$$

We have

$$\begin{aligned}\mathbb{E}\Psi &= \|\mathbf{x}\|_2^2\\ \mathbb{V}\Psi &\leq (3+e)m^{-1}\|\mathbf{x}\|_2^4.\end{aligned}$$

Proof. Recall that $X_k = \sum_{v\in[n]} Z_{v,k}\mathbf{x}(v)$ where $Z_{v,k}\sim \text{SymmetricPoisson}(e^{-k/m})$. By Lemma 4.4, we know $\mathbb{E}Z_{v,k}=0$, $\mathbb{E}Z_{v,k}^2 = e^{-k/m}$, and $\mathbb{E}Z_{v,k}^4 = e^{-k/m}+3e^{-2k/m}$. Thus similar to the AMS sketch,

$$\mathbb{E}X_k^2 = \sum_{v\in[n]}\mathbf{x}(v)^2\mathbb{E}Z_{v,k}^2 = e^{-k/m}\|\mathbf{x}\|_2^2.$$

Thus $\mathbb{E}\Psi = \|\mathbf{x}\|_2^2$. Then

$$\mathbb{E}X_k^4 = \sum_{v\in[n]}\mathbf{x}(v)^4\mathbb{E}Z_{v,k}^4 + 6\sum_{s<t}\mathbf{x}(s)^2\mathbf{x}(t)^2\mathbb{E}Z_{v,s}^2\mathbb{E}Z_{v,t}^2$$

$$\leq (e^{-k/m} + 3e^{-2k/m})\|\mathbf{x}\|_2^4$$

Now

$$\begin{aligned}
\mathbb{E}\Psi^2 &\leq \frac{1}{m^2}\sum_{k=0}^{m-1}\mathbb{E}X_k^4 e^{2k/m} + (\frac{1}{m}\sum_{k=0}^{m-1}\mathbb{E}X_k^2 e^{k/m})^2 \\
&= \frac{\|\mathbf{x}\|_2^4}{m^2}\sum_{k=0}^{m-1}(e^{k/m}+3) + \|\mathbf{x}\|_2^4 \\
&\leq \frac{\|\mathbf{x}\|_2^4}{m}(3+e) + \|\mathbf{x}\|_2^4.
\end{aligned}$$

Thus $\mathbb{V}\Psi \leq \mathbb{E}\Psi^2 - (\mathbb{E}\Psi)^2 \leq (3+e)m^{-1}\|\mathbf{x}\|_2^4$. □

4.6.2 Estimation of the $f_{\geq\zeta}$-Moment

We proceed as in Section 4.3.2. Let $\{V_{\gamma;a,b}\}_{\gamma\geq\zeta}$ be the set of harmonic estimators generated from the truncated triple SymmetricPoissonTower in Theorem 4.3. For $f_{\geq\zeta}(x) = \int_\zeta^\infty (1-\cos(\gamma x))\,\nu(d\gamma)$, define

$$\phi_{f_{\geq\zeta};a,b} = \int_\zeta^\infty V_{\gamma;a,b}\,\nu(d\gamma).$$

By Theorem 4.3, we know $\mathbb{E}V_{\gamma;a,b} = \mathbb{E}V_\gamma + O(\delta)$ where $\delta = (e^{\frac{1}{3}a/m} + e^{-\frac{2}{3}b/m})\,\mathrm{poly}(\|\mathbf{x}\|_0 m)$.

We have

$$\begin{aligned}
\mathbb{E}\phi_{f_{\geq\zeta};a,b} &= \int_\zeta^\infty \mathbb{E}V_{\gamma;a,b}\,\nu(d\gamma) \\
&= \int_\zeta^\infty ((1+O(m^{-1}))f_\gamma(\mathbf{x}) + O(\delta))\,\nu(d\gamma) \\
&= (1+O(m^{-1}))\int_\zeta^\infty f_\gamma(\mathbf{x})\,\nu(d\gamma) + O(\delta)\int_\zeta^\infty \nu(d\gamma)
\end{aligned}$$

note that since $\zeta < 1$, we have $\int_\zeta^\infty \nu(d\gamma) \leq \zeta^{-2}\int_0^\infty \min(\gamma^2,1)\,\nu(d\gamma) = \zeta^{-2}C_f$

$$= (1+O(m^{-1}))\int_\zeta^\infty f_\gamma(\mathbf{x})\,\nu(d\gamma) + O(\delta)\zeta^{-2}C_f.$$

The variance can be bounded similarly and we will leave it in the proof of Theorem 4.4.

4.6.3 Combination

We now prove the main technical theorem.

Theorem 4.4 ("frequency-domain" estimator)**.** *Let $\mathbf{x} \in \mathbb{Z}^n$ be the input frequency vector and $(X_k^{(j)})_{a\le k<b, j=1,2,3}$ be a $[a,b)$-truncated triple* SymmetricPoissonTower *with parameter m. For any function $f(x) = \sigma^2 x^2/2 + \int_0^\infty (1-\cos(\gamma x))\,\nu(d\gamma)$ where ν is a positive measure with $C_f = \int_0^\infty \min(\gamma^2, 1)\,\nu(d\gamma)$. Denote the estimator for f as*

$$\phi_{f;a,b} = \frac{1}{2}\left(\sigma^2 + \int_0^\zeta \gamma^2\,\nu(d\gamma)\right)\frac{\sum_{j=1}^3\sum_{k=0}^{m-1}(X_k^{(j)})^2 e^{k/m}}{3m} + \int_\zeta^\infty \frac{\prod_{j=1}^3\left(\sum_{k=a}^{b-1}(1-e^{i\gamma X_k^{(j)}})e^{\frac{1}{3}k/m}\right)}{m^3(-\Gamma(-1/3))^3}\,\nu(d\gamma).$$

Set $\zeta = O(1/\operatorname{poly}(\|\mathbf{x}\|_2 m))$. If $a = -\Omega(m\log(\|\mathbf{x}\|_2 m))$ and $b = \Omega(m\log(\|\mathbf{x}\|_2 m))$, then $\phi_{f;a,b}$ is a $(1\pm O(m^{-1}))$-approximate of the f-moment with negligible bias.

$$\mathbb{E}\phi_{f;a,b} = (1+O(m^{-1}))\sum_{v\in[n]} f(\mathbf{x}(v)) + O(\frac{C_f}{\operatorname{poly}(\|\mathbf{x}\|_2 m)})$$
$$\mathbb{V}\phi_{f;a,b} \le O(m^{-1})\cdot\left(\sum_{v\in[n]} f(\mathbf{x}(v))\right)^2 + O(\frac{C_f^2}{\operatorname{poly}(\|\mathbf{x}\|_2 m)}).$$

Proof. For simplicity, we write $\phi_{f;a,b}$ as $\alpha A + B$ where $\alpha = \frac{1}{2}(\sigma^2 + \int_0^\zeta \gamma^2\,\nu(d\gamma))$, $A = \frac{\sum_{j=1}^3\sum_{k=0}^{m-1}(X_k^{(j)})^2 e^{k/m}}{3m}$, and $B = \int_\zeta^\infty \frac{\prod_{j=1}^3(\sum_{k=a}^{b-1}(1-e^{i\gamma X_k^{(j)}})e^{\frac{1}{3}k/m})}{m^3(-\Gamma(-1/3))^3}\,\nu(d\gamma)$. By Lemma 4.10, we know

$$\mathbb{E}A = \|\mathbf{x}\|_2^2$$
$$\mathbb{V}A \le (1+e/3)m^{-1}\|\mathbf{x}\|_2^4.$$

With Lemma 4.9, we know

$$\mathbb{E}\alpha A = (1+O(m^{-1}))f_{<\zeta}(\mathbf{x})$$
$$\mathbb{V}(\alpha A) \le (1+e/3)m^{-1}\left(f_{<\zeta}(\mathbf{x})\right)^2(1+O(m^{-1}))$$

Since $\mathbf{x}\in\mathbb{Z}^n$, we always have $\|\mathbf{x}\|_0 \le \|\mathbf{x}\|_2^2$. By the calculation in Section 4.6.2 and the

assumptions on ζ, a, b, we have

$$\mathbb{E}B = (1 + O(m^{-1})) \sum_{v \in [n]} f_{\geq \zeta}(\mathbf{x}(v)) + C_f \cdot O(1/\operatorname{poly}(\|\mathbf{x}\|_2 m))$$

Similarly, for the variance we have

$$\begin{aligned}
\mathbb{V}B &\leq \left(\int_\zeta^\infty \sqrt{\mathbb{V}V_{\gamma;a,b}}\, \nu(\gamma) \right)^2 \\
&\leq \left(\int_\zeta^\infty \sqrt{\mathbb{V}V_\gamma + O(1/\operatorname{poly}(\|\mathbf{x}\|_2 m))}\, \nu(\gamma) \right)^2 && \text{(Theorem 4.3)} \\
&\leq \left(\int_\zeta^\infty \sqrt{(m^{-1} + O(m^{-2})) \cdot C \cdot f_\gamma(\mathbf{x})^2 + O(1/\operatorname{poly}(\|\mathbf{x}\|_2 m))}\, \nu(\gamma) \right)^2 && \text{(Theorem 4.2)}
\end{aligned}$$

note that $f_\gamma(\mathbf{x}) \leq 2\|\mathbf{x}\|_0 \leq 2\|\mathbf{x}\|_2^2$ and $\int_\zeta^\infty \nu(d\gamma) \leq \zeta^{-2} \int_\zeta^\infty \min(\gamma^2, 1)\, \nu(d\gamma) = \zeta^{-2} C_f$.

$$\begin{aligned}
&\leq \left(\int_\zeta^\infty \sqrt{(m^{-1} + O(m^{-2})) \cdot C \cdot f_\gamma(\mathbf{x})^2} + \sqrt{O(1/\operatorname{poly}(\|\mathbf{x}\|_2 m))}\, \nu(\gamma) \right)^2 \\
&\leq \left(\sqrt{(m^{-1} + O(m^{-2})) \cdot C} f_{\geq \zeta}(\mathbf{x}) + \sqrt{O(1/\operatorname{poly}(\|\mathbf{x}\|_2 m))} \zeta^{-2} C_f \right)^2
\end{aligned}$$

Note that we have $1 + e/3 < 2$ and $C = \frac{3 \cdot 2^{2/3}(-\Gamma(-2/3))}{(-\Gamma(-1/3))^2} < 2$. We conclude that

$$\begin{aligned}
\mathbb{E}(\alpha A + B) &= (1 + O(m^{-1})) f(\mathbf{x}) + C_f \cdot O(1/\operatorname{poly}(\|\mathbf{x}\|_2 m)) \\
\mathbb{V}(\alpha A + B) &\leq (\sqrt{\mathbb{V}(\alpha A)} + \sqrt{\mathbb{V}B})^2 \\
&\leq \left(\sqrt{2(m + O(m^{-2}))} f(\mathbf{x}) + O(1/\operatorname{poly}(\|\mathbf{x}\|_2 m)) \cdot C_f \right)^2
\end{aligned}$$

We know $f_{\geq \zeta}(\mathbf{x}) \leq \sum_{v \in [n]} \int_\zeta^\infty 2\, \nu(d\gamma) = O(\zeta^{-2} C_f \|\mathbf{x}\|_0)$ and $f_{<\zeta}(\mathbf{x}) \leq \frac{1}{2}\|\mathbf{x}\|_2^2 C_f$ (Lemma 4.9). Therefore $f(\mathbf{x}) = O(\zeta^{-2} C_f \|\mathbf{x}\|_0)$. We have assumed $\zeta = O(1/\operatorname{poly}(\|\mathbf{x}\|_2 m))$ and thus

$$\leq O(m^{-1}) \left(f(\mathbf{x})\right)^2 + O(1/\operatorname{poly}(\|\mathbf{x}\|_2 m)) \cdot C_f^2.$$

□

The "frequency domain" estimator above might look daunting but in practice f can usually be approximated with a few main harmonic moments. In the case where f cannot be approximated by sparse harmonics, we provide the following alternative estimator, which can be computed with at most $O(m^3 \log^3(nMm))$ evaluations of the function $f_{\geq \zeta}$.

Corollary 4.1 ("time-domain" estimator)**.** *With the setup in Theorem 4.4, define* $\psi_{f;a,b}$ *as*

$$\begin{aligned}\psi_{f;a,b} = \frac{1}{2}\left(\sigma^2 + \int_0^\zeta \gamma^2\,\nu(d\gamma)\right)\frac{\sum_{j=1}^3\sum_{k=0}^{m-1}(X_k^{(j)})^2 e^{k/m}}{3m} \\ + m^{-3}(-\Gamma(-1/3))^{-3}\sum_{j=a}^{b-1}\sum_{k=a}^{b-1}\sum_{l=a}^{b-1} e^{\frac{k+j+l}{3m}}\Big(f_{\geq\zeta}(X_j^{(1)}) + f_{\geq\zeta}(X_k^{(2)}) + f_{\geq\zeta}(X_l^{(3)}) \\ - f_{\geq\zeta}(X_j^{(1)} + X_k^{(2)}) - f_{\geq\zeta}(X_k^{(2)} + X_l^{(3)}) - f_{\geq\zeta}(X_l^{(3)} + X_j^{(1)}) + f_{\geq\zeta}(X_j^{(1)} + X_k^{(2)} + X_l^{(3)})\Big).\end{aligned}$$

Then $\psi_{f;a,b}$ *is a* $(1\pm O(m^{-1}))$*-approximate of the* f*-moment with negligible bias. In particular,*

$$\psi_{f;a,b} = \mathfrak{Re}(\phi_{f;a,b}).$$

Proof. Note that

$$\begin{aligned}&\prod_{j=1}^3\left(\sum_{k=a}^{b-1}(1-e^{i\gamma X_k^{(j)}})e^{\frac{1}{3}k/m}\right) \\ &= \sum_{j=a}^{b-1}\sum_{k=a}^{b-1}\sum_{l=a}^{b-1}(1-e^{i\gamma X_j^{(1)}})(1-e^{i\gamma X_k^{(2)}})(1-e^{i\gamma X_l^{(3)}})e^{\frac{1}{3}(k+j+l)/m}\end{aligned}$$

where for any integers A_1, A_2, A_3

$$\begin{aligned}&(1-e^{i\gamma A_1})(1-e^{i\gamma A_2})(1-e^{i\gamma A_3}) \\ &= 1 - \sum_j e^{i\gamma A_j} + \sum_{j<k} e^{i\gamma(A_1+A_2)} - e^{i\gamma(A_1+A_2+A_3)} \\ &= (1-e^{i\gamma(A_1+A_2+A_3)}) + \sum(1-e^{i\gamma A_j}) - \sum_{j<k}(1-e^{i\gamma(A_j+A_k)}).\end{aligned}$$

Note that $\mathfrak{Re}(1-e^{i\gamma k}) = 1-\cos(\gamma k)$ for any $k\in\mathbb{Z}$. Thus by change the order of the triple-summation and integration we have what we want. Finally, note that for any complex random variable W, $\mathbb{V}W = \mathbb{V}\mathfrak{Re}W + \mathbb{V}\mathfrak{Im}W$ and thus $\mathbb{V}\psi_{f;a,b} \leq \mathbb{V}\phi_{f;a,b}$. □

4.6.4 Signed Combination

As in [50, 51], the transportation of the mean and the relative variance of $\{V_\gamma\}_{\gamma>0}$ in the combined estimator $\int_0^\infty V_\gamma\,\nu(d\gamma)$ relies crucially on the fact that ν is a positive measure. In general, when ν is a signed measure, the combined mean Eq. (4.4) still holds but the combined variance Eq. (4.5) now depend on $|\nu|$.

Corollary 4.2 (signed combination)**.** *Let* $f(x) = \sigma_1^2 x^2/2 + \int_0^\infty (1 - \cos(\gamma x))\, \nu_1(d\gamma)$ *and* $g(x) = \sigma_2^2 x^2/2 + \int_0^\infty (1 - \cos(\gamma x))\, \nu_2(d\gamma)$ *where* ν_1, ν_2 *are positive measures. Let* $\phi_{f;a,b}$ *and* $\phi_{g;a,b}$ *be generated from a same truncated triple* SymmetricPoissonTower *sketch in Theorem 4.4. Then*

$$\mathbb{E}(\phi_{f;a,b} - \phi_{g;a,b}) = (1 + O(m^{-1}))(f(\mathbf{x}) - g(\mathbf{x})) + O(\frac{C_f + C_g}{\text{poly}(\|\mathbf{x}\|_2 m)})$$
$$\mathbb{V}(\phi_{f;a,b} - \phi_{g;a,b}) \le O(m^{-1}) \cdot (f(\mathbf{x}) + g(\mathbf{x}))^2 + O(\frac{C_f^2 + C_g^2}{\text{poly}(\|\mathbf{x}\|_2 m)}).$$

Proof. Both follow from Theorem 4.4. The mean results from the linearity of expectation. For the variance, note that

$$\mathbb{V}(\phi_{f;a,b} - \phi_{g;a,b}) \le (\sqrt{\mathbb{V}\phi_{f;a,b}} + \sqrt{\mathbb{V}\phi_{g;a,b}})^2.$$

□

Thus for $(\phi_{f;a,b} - \phi_{g;a,b})$ to be a multiplicative approximation of the $(f-g)$-moment, one needs $f(\mathbf{x}) + g(\mathbf{x}) = \Omega(f(\mathbf{x}) - g(\mathbf{x}))$ for any $\mathbf{x} \in \mathbb{Z}^n$.

4.7 Proof of Theorem 4.1

The main theorem (Theorem 4.1) almost directly follows from Theorem 4.4 and Corollary 4.2. We only need to analyze the space and translate the variance guarantee to $(1 \pm \epsilon)$-approximation with Chebyshev's inequality.

Lemma 4.11. *With probability* $5/6$*, all the cells in the triple* SymmetricPoissonTower *have magnitude at most* $O(\text{poly}(Mmn))$*.*

Proof. The truncated tower stores X_js from $a = -\Omega(m \log(\|\mathbf{x}\|_2 m))$ to $b = \Omega(m \log(\|\mathbf{x}\|_2 m))$. Suppose $b - a = C_1 m \log(\|\mathbf{x}\|_2 m)$ for some $C_1 > 0$. Each cell j stores an integer. Specifically $X_j \sim \sum_{v \in [n]} \mathbf{x}(v) Y_v$ where $Y_v \sim \text{SymmetricPoisson}(e^{-k/m})$. By a simple coupling argument, we have $|X_j| \le MY$ where M is the frequency upper bound and $Y \sim \text{Poisson}(ne^{-k/m})$. Therefore by Markov's inequality,

$$\mathbb{P}(|X_j| \ge C_2 m \log(\|\mathbf{x}\|_2 m) M n e^{-k/m}) \le \mathbb{P}(Y \ge C_2 m \log(\|\mathbf{x}\|_2 m) n e^{-k/m}) \le \frac{1}{C_2 m \log(\|\mathbf{x}\|_2 m)}.$$

Choosing $C_2 = 18C_1$, by the union bound, with probability $1 - \frac{C_1 m \log(\|\mathbf{x}\|_2 m)}{C_2 m \log(\|\mathbf{x}\|_2 m)} = 17/18$, that $|X_j| < C_2 m \log(\|\mathbf{x}\|_2 m) M n e^{-k/m} = O(\text{poly}(Mmn))$ for all $j \in [a, b)$, where we bound $\|\mathbf{x}\|_2$ by nM^2. With probability at least 5/6 that this event occurs for all three truncated towers. □

Lemma 4.12. *With probability 5/6, the triple* SymmetricPoissonTower *with* $m = \Omega(\epsilon^{-2})$ *outputs a correct* $(1 \pm \epsilon)$*-approximation of the* f*-moment.*

Proof. By Theorem 4.4, an estimate of the f-moment can be returned with $O(m^{-1})$ relative variance and an $O(C_f / O(\text{poly}(\|\mathbf{x}\|_2 m)))$ additive bias. Note that for any non-empty stream, the f-moment $f(\mathbf{x})$ is at least $\xi(f, M)$. Thus the additive bias can be absorbed in the multiplicative term since we have $\xi(f, M) = \Omega(C_f / \text{poly}(nMm))$ by assumption in Theorem 4.1. Therefore, the estimator is a $(1 \pm O(m^{-1/2}))$-approximation with probability 5/6 by Chebyshev's inequality with a suitable constant in $O(m^{-1/2})$. It suffices to set $m = \Omega(\epsilon^{-2})$ for a $(1 \pm \epsilon)$-approximation. □

Proof of Theorem 4.1. By a union bound of the events in Lemmas 4.11 and 4.12, with probability 2/3 the estimate is a correct $(1 \pm \epsilon)$-approximation and at the same time $O(\log(Mmn))$ bits suffice to store each cell for the three i.i.d. copies of truncated tower. There are $O(m \log(\|\mathbf{x}\|_2 m)) = O(m \log(nMm))$ cells. Thus the total space in bits is $O(m \log^2(nMm)) = O(\epsilon^{-2} \log^2(nM\epsilon^{-1}))$. The proof for the signed combination goes the same with the variance guarantee provided by Corollary 4.2. □

4.8 Summary

The set of tractable f-moments are those that can be approximated in polylog space. It is a major open question to fully characterize this class. To date, Braverman, Chestnut, Woodruff, and Yang [19] have a nearly-complete characterization of this class, but it cannot handle the class of tractable nearly periodic functions.

In this chapter, we proposed a new method for estimating f-moments based on a harmonic decomposition of f. The SymmetricPoissonTower is a new $O(\epsilon^{-2} \log^2 n)$-space data structure that can estimate all the important functions in Braverman et al.'s class [19], as well as many nearly periodic functions. We conjecture that the SymmetricPoissonTower is a universal sketch for the class of tractable functions.

Conjecture 4.1. *The* SymmetricPoissonTower *is universal for the class of tractable functions.*

There is some subtlety in this conjecture, as the tractable class probably does not have a sharp boundary. For fixed $f, g \in \mathcal{F}_*$ (Section 4.3.3), define $\tilde{h}(M) = (\max_{x\in[M]} \frac{f(x)+g(x)}{f(x)-g(x)})^2$. As $\tilde{h}(M)$ becomes large, more and more functions that can be written as $f - g$ can be included in the range with the cost of having an increasingly large $\tilde{h}(M)$ factor in space. This is analogous to the role that $h(M)$ plays in Braverman et al. [19].

4.9 Example Functions

4.9.1 Nearly Periodic g_{np}

The function g_{np} is constructed in [19] for the purpose of showing there are tractable functions outside the reach of L_2-heavy hitters framework. We decompose g_{np} into harmonic moments and compute its function constant $C_{g_{np}}$ in this section. Recall the following definitions.

- $\mathbb{B} = \{\sum_{j=1}^{k} a_j 2^{-j} \mid \forall j, a_j \in \{0, 1\}, k \in \mathbb{N}\}$ is the set of finite-precision binary numbers in $[0, 1)$;
- $\tau : \mathbb{Z}_+ \to \mathbb{Z}_+$ with $\tau(x) = \max\{j \in \mathbb{N} \mid 2^j | x\}$ which returns the position of the least significant bit of x;
- $\tau_* : \mathbb{B} \to \mathbb{Z}_+$ with $\tau_*(r) = \min\{j \in \mathbb{N} \mid 2^{-j} | r\}$ which returns the length of the binary representation of r.

Lemma 4.13. *For any $x \in \mathbb{N}$,*

$$g_{np}(x) = 2^{-\tau(x)} = \frac{4}{3} \sum_{\gamma\in\mathbb{B}} (1 - \cos(2\pi\gamma x)) 2^{-2\tau_*(\gamma)}$$

Proof. Partition $\mathbb{B}$ as $B_0, B_1, B_2, \ldots$ where $B_k = \{r \in \mathbb{B} \mid \tau_*(r) = k\}$. We have $B_0 = \{0\}$, $B_1 = \{1/2\}$, $B_2 = \{1/4, 3/4\}$, $B_3 = \{1/8, 3/8, 5/8, 7/8\}$, and so on. By symmetry, for $k \geq 2$

$$\sum_{\gamma\in B_k} (1 - \cos(2\pi\gamma x)) = \begin{cases} 2^{k-1}, & \tau(x) \leq k - 2 \\ 2^k, & \tau(x) = k - 1 \\ 0, & \tau(x) \geq k \end{cases}$$

where in the first case $\sum_{\gamma\in B_k} \cos(2\pi\gamma x) = 0$, in the second case $\cos(2\pi\gamma x) = -1$ for any $\gamma \in B_k$, and in the last case, $\cos(2\pi\gamma x) = 1$ for any $\gamma \in B_k$. Thus if $\tau(x) \geq 1$, i.e., x is even,

then

$$\begin{aligned}\sum_{\gamma\in\mathbb{B}}(1-\cos(2\pi\gamma x))2^{-2\tau_*(\gamma)} &= \sum_{k\in\mathbb{N}}\sum_{\gamma\in B_k}(1-\cos(2\pi\gamma x))2^{-2k}\\ &= 2^{\tau(x)+1}2^{-2(\tau(x)+1)} + \sum_{k=\tau(x)+2}^{\infty} 2^{k-1}\cdot 2^{-2k}\\ &= 2^{-(\tau(x)+1)} + 2^{-(\tau(x)+2)}\\ &= \frac{3}{4}2^{-\tau(x)}.\end{aligned}$$

When x is odd, then

$$\begin{aligned}\sum_{\gamma\in\mathbb{B}}(1-\cos(2\pi\gamma x))2^{-2\tau_*(\gamma)} &= \sum_{k\in\mathbb{N}}\sum_{\gamma\in B_k}(1-\cos(2\pi\gamma x))2^{-2k}\\ &= 2\cdot 2^{-2} + \sum_{k=2}^{\infty} 2^{k-1}\cdot 2^{-2k}\\ &= \frac{3}{4}.\end{aligned}$$

□

Lemma 4.14. $C_{g_{np}} \leq \frac{2}{3}$.

Proof.

$$C_{g_{np}} = \frac{4}{3}\sum_{\gamma\in\mathbb{B}}\min(4\pi^2\gamma^2,1)2^{-2\tau_*(\gamma)} \leq \frac{4}{3}\sum_{k=1}^{\infty}\sum_{\gamma\in B_k}2^{-2k} = \frac{4}{3}\sum_{k=1}^{\infty}2^{k-1}2^{-2k} = \frac{2}{3}.$$

□

4.9.2 The Golden Harmonic Moment g_{gold}

The golden harmonic moment g_{gold} is constructed for additional examples that are tractable in the harmonic sketching framework but out of the reach of prior techniques.

With the pigeonhole argument in Lemma 4.2, it is clear that $\xi(g_{gold}, M) = O(1/M^2)$. We now bound $\xi(g_{gold}, M)$ from below, which is based on the theory of continued fractions.

Lemma 4.15. *Let* $g_{gold}(x) = 1-\cos(\frac{1+\sqrt{5}}{2}\cdot 2\pi x)$ *for* $x\in\mathbb{Z}$. *Then* $\xi(g_{gold}, M) = \Omega(1/M^2)$.

Proof. The convergents of $\frac{1+\sqrt{5}}{2}$ are $(F_{n+1}/F_n)_{n\in\mathbb{N}}$ where $(F_n)_{n\in\mathbb{N}}$ is the Fibonacci sequence. It is known that for any $p, q \in \mathbb{N}$ such that $q \le F_n$, $|\frac{1+\sqrt{5}}{2} - p/q| \ge |\frac{1+\sqrt{5}}{2} - F_{n+1}/F_n|$. Thus for any $x \in \mathbb{N}$, let $\frac{1+\sqrt{5}}{2}x = w + r$ where $w \in \mathbb{N}$ and $r \in (-1/2, 1/2)$. Let $F_n \ge x$. Then

$$|r/x| = |\frac{1+\sqrt{5}}{2} - \frac{w}{x}| \ge |\frac{1+\sqrt{5}}{2} - F_{n+1}/F_n| \ge |\frac{1+\sqrt{5}}{2} - F_{n+2}/F_{n+1}|$$

Thus

$$\begin{aligned} 2|r/x| &\ge |\frac{1+\sqrt{5}}{2} - F_{n+1}/F_n| + |\frac{1+\sqrt{5}}{2} - F_{n+2}/F_{n+1}| \\ &\ge |F_{n+1}/F_n - F_{n+2}/F_{n+1}| \\ &= \frac{1}{F_n F_{n+1}} \end{aligned}$$

Note that $F_n = \Theta((\frac{1+\sqrt{5}}{2})^n)$ and thus we may choose F_n, F_{n+1} that are $O(x)$. Therefore $|r| = \Omega(1/x)$, which implies $1 - \cos(\frac{1+\sqrt{5}}{2} \cdot 2\pi x) = 1 - \cos(w2\pi + 2\pi r) = 1 - \cos(2\pi r) \le 2\pi^2 r^2 = \Omega(1/x^2)$. □

4.10 Mathematical Lemmas

In this section we estimate the rate that sums like $\frac{1}{m}\sum_{k\in\mathbb{Z}}(1 - e^{-e^{-k/m}})e^{\frac{1}{3}k/m}$ converge to its limit $\int_{-\infty}^{\infty}(1 - e^{-e^{-r}})e^{\frac{1}{3}r}\,dr$ as m increases, supporting our analysis of the smoothed (-1/3)-aggregation.

Lemma 4.16. *Let $h : \mathbb{R} \to \mathbb{C}$ be a differentiable function. If both h and h' are (Lebesgue) integrable, then*

$$\left| \int_{-\infty}^{\infty} h(s)\,ds - \frac{1}{m}\sum_{k=-\infty}^{\infty} h(k/m) \right| \le m^{-1} \int_{-\infty}^{\infty} |h'(s)|\,ds.$$

Proof. We bound the difference

$$\begin{aligned} \left| \int_{-\infty}^{\infty} h(s)\,ds - \frac{1}{m}\sum_{k=-\infty}^{\infty} h(k/m) \right| &= \left| \sum_{k=-\infty}^{\infty} \left(\int_0^{1/m} h(k/m + s)\,ds - \frac{1}{m}h(k/m) \right) \right| \\ &\le \sum_{k=-\infty}^{\infty} \left| \int_0^{1/m} h(k/m + s)\,ds - \frac{1}{m}h(k/m) \right|. \end{aligned}$$

Note that

$$\begin{aligned}\left|\int_0^{1/m} h(k/m+s)\,ds - \frac{1}{m}h(k/m)\right| &= \left|\int_0^{1/m}\int_0^{s} h'(k/m+t)\,dtds\right| \\ &\le \int_0^{1/m}\int_0^{s} |h'(k/m+t)|\,dtds \\ &\le \int_0^{1/m}\int_0^{1/m} |h'(k/m+t)|\,dtds \\ &= \frac{1}{m}\int_0^{1/m} |h'(k/m+t)|\,dt.\end{aligned}$$

Thus

$$\begin{aligned}\left|\int_{-\infty}^{\infty} h(s)\,ds - \frac{1}{m}\sum_{j=-\infty}^{\infty} h(j/m)\right| &\le \sum_{k=-\infty}^{\infty}\frac{1}{m}\int_0^{1/m} |h'(k/m+t)|\,dt \\ &= \frac{1}{m}\int_{-\infty}^{\infty} |h'(t)|\,dt.\end{aligned}$$

□

Lemma 4.17. *Let $a, b \in \mathbb{C}$ and $c \in \mathbb{R}$. Define*

$$\eta_1(x;a,b,c) = (e^{-ae^{-x}} - e^{-be^{-x}})e^{cx}, \qquad \eta_2(x;a,c) = ae^{-x}e^{-ae^{-x}}e^{cx}.$$

If $\mathfrak{Re}(a) \ge 0$, $\mathfrak{Re}(b) \ge 0$ and $c \in (0,1)$, then

1. $\int_{-\infty}^{\infty} \eta_1(x;a,b,c)\,dx = (a^c - b^c)\Gamma(-c)$, *where $\Gamma(\cdot)$ here denotes the Gamma function (Fig. 4.5);*
2. $\eta_1'(x;a,b,c) = c\eta_1(x;a,b,c) + \eta_2(x;a,c) - \eta_2(x;b,c)$;
3. $\int_{-\infty}^{\infty} |\eta_1(x;a,b,c)|\,dx \le 2\left(\frac{1}{c} + \frac{1}{1-c}\right)|b-a|^c$.
4. $\int_{-\infty}^{\infty} |\eta_2(x;a,c)|\,dx \le |a|^c\Gamma(1-c)$.
5. $\int_{-\infty}^{\infty} |\eta_1'(x;a,b,c)|\,dx \le 2\left(1 + \frac{c}{1-c}\right)|b-a|^c + (|a|^c + |b|^c)\Gamma(1-c)$.
6. $$\left|\int_{-\infty}^{\infty} \eta_1(x;a,b,c)\,dx - \frac{1}{m}\sum_{k=-\infty}^{\infty} \eta_1(k/m;a,b,c)\right|$$

$$\leq m^{-1}\left(2\left(1+\frac{c}{1-c}\right)|b-a|^c+(|a|^c+|b|^c)\Gamma(1-c)\right).$$

Proof. Part 1. Integral by parts.

$$\begin{aligned}&\int_{-\infty}^{\infty}(e^{-ae^{-x}}-e^{-be^{-x}})e^{cx}\,dx\\&=(e^{-ae^{-x}}-e^{-be^{-x}})c^{-1}e^{cx}\Big|_{-\infty}^{\infty}-\int_{-\infty}^{\infty}(ae^{-x}e^{-ae^{-x}}-be^{-x}e^{-be^{-x}})c^{-1}e^{cx}\,dx\end{aligned}$$

Since $\mathfrak{Re}(a)\geq 0$, $\mathfrak{Re}(b)\geq 0$ and $c\in(0,1)$, $(e^{-ae^{-x}}-e^{-be^{-x}})c^{-1}e^{cx}\Big|_{-\infty}^{\infty}=0$,

$$=-c^{-1}\int_{-\infty}^{\infty}ae^{-(1-c)x}e^{-ae^{-x}}\,dx+c^{-1}\int_{-\infty}^{\infty}be^{-(1-c)x}e^{-be^{-x}}\,dx.$$

Set $z=ae^{-x}$ and we have

$$\begin{aligned}c^{-1}\int_{-\infty}^{\infty}ae^{-(1-c)x}e^{-ae^{-x}}\,dx&=c^{-1}\int_0^{\infty}a^cz^{-c}e^{-z}\,dz\\&=a^c\frac{\Gamma(1-c)}{c}=a^c(-\Gamma(-c)).\end{aligned}\tag{4.6}$$

Thus

$$\int_{-\infty}^{\infty}(e^{-ae^{-x}}-e^{-be^{-x}})e^{cx}\,dx=-a^c(-\Gamma(-c))+b^c(-\Gamma(-c))=(a^c-b^c)\Gamma(-c).$$

Part 2.

$$\begin{aligned}\eta_1'(x;a,b,c)&=(e^{-ae^{-x}}-e^{-be^{-x}})ce^{cx}+(ae^{-x}e^{-ae^{-x}}-be^{-x}e^{-be^{-x}})e^{cx}\\&=c\eta_1(x;a,b,c)+\eta_2(x;a,c)-\eta_2(x;b,c).\end{aligned}$$

Part 3. Define the path $\phi(t)=(1-t)a+tb$ and $g_x(r)=e^{-re^{-x}}$. By the path integral,

$$\begin{aligned}\left|e^{-ae^{-x}}-e^{-be^{-x}}\right|&=|g_x(a)-g_x(b)|=\left|\int_0^1 g_x'(\phi(t))\phi'(t)\,dt\right|\\&=\left|\int_0^1 e^{-\phi(t)e^{-x}}(-e^{-x})(b-a)\,dt\right|\leq|b-a|e^{-x}\int_0^1|e^{-\phi(t)e^{-x}}|\,dt\end{aligned}$$

We assumed $\mathfrak{Re}(a), \mathfrak{Re}(b) \geq 0$ and thus $\mathfrak{Re}(\phi(t)) \geq 0$ for any $t \in [0,1]$, which implies $|e^{-\phi(t)e^{-x}}| < 1$. We then have

$$\left|e^{-ae^{-x}} - e^{-be^{-x}}\right| \leq |b-a|e^{-x}.$$

On the other hand, since $\mathfrak{Re}(a), \mathfrak{Re}(b) \geq 0$, we have $\left|e^{-ae^{-x}}\right|, \left|e^{-be^{-x}}\right| \leq 1$ and thus

$$\left|e^{-ae^{-x}} - e^{-be^{-x}}\right| \leq \left|e^{-ae^{-x}}\right| + \left|e^{-ae^{-x}}\right| \leq 2.$$

We conclude that for any $q \in \mathbb{R}$

$$\begin{aligned}\int_{-\infty}^{\infty} \left|(e^{-ae^{-x}} - e^{-be^{-x}})e^{cx}\right| \, dx &= \int_{-\infty}^{q} \left|e^{-ae^{-x}} - e^{-be^{-x}}\right| e^{cx} \, dx + \int_{q}^{\infty} \left|e^{-ae^{-x}} - e^{-be^{-x}}\right| e^{cx} \, dx \\ &\leq \int_{-\infty}^{q} 2e^{cx} \, dx + \int_{q}^{\infty} |b-a|e^{-x}e^{cx} \, dx \\ &= 2\frac{e^{cq}}{c} + |b-a|\frac{e^{(c-1)q}}{1-c},\end{aligned}$$

since $c \in (0,1)$. Now set $q = \log|b-a|$ and we have

$$\int_{-\infty}^{\infty} \left|(e^{-ae^{-x}} - e^{-be^{-x}})e^{cx}\right| \, dx \leq 2\left(\frac{1}{c} + \frac{1}{1-c}\right)|b-a|^c.$$

Part 4.

$$\int_{-\infty}^{\infty} \left|ae^{-x}e^{-ae^{-x}}e^{cx}\right| \, dx = |a| \int_{-\infty}^{\infty} e^{-x}e^{-\mathfrak{Re}(a)e^{-x}}e^{cx} \, dx$$

by Eq. (4.6), since $\mathfrak{Re}(a) \geq 0$

$$\begin{aligned}&= |a|\frac{1}{\mathfrak{Re}(a)}(\mathfrak{Re}(a))^c\Gamma(1-c) \\ &\leq |a|^c\Gamma(1-c).\end{aligned}$$

Part 5. Follows from 2, 3, and 4.

Part 6. Follows from 5 and Lemma 4.16.

□

CHAPTER 5

Cardinality Estimation

Cardinality estimation, aka F_0 estimation or `COUNT-DISTINCT`, is a fundamental problem in streaming and sketching with diverse applications in databases [40, 85], network monitoring [9, 29, 167, 38], nearest neighbor search [141], caching [162], and genomics [125, 75].

Definition 5.1 (cardinality/F_0/`COUNT-DISTINCT` estimation)**.** Let n be the size of the domain. A sequence $\mathcal{A} = (a_1, \ldots, a_N) \in [n]^N$ over some universe $[n]$ is revealed one element at a time. The goal is to estimate the *cardinality* $\lambda = |\{a_1, \ldots, a_N\}|$ of the set. Different error metrics are used in the standard model and in the random oracle model.

- In the standard model, an estimator $\hat{\lambda}$ has error guarantee (ϵ, δ) if $\mathbb{P}(|\hat{\lambda} - \lambda| > \epsilon\lambda) < \delta$.
- In the random oracle model, an estimator is usually required to be asymptotically unbiased (as $\lambda \to \infty$). The error is measured by the *relative* variance $\frac{1}{\lambda^2}\mathbb{V}(\hat{\lambda} \mid \lambda)$ and the *relative* standard deviation $\frac{1}{\lambda}\sqrt{\mathbb{V}(\hat{\lambda} \mid \lambda)}$, also called the *standard error*.

Sketches for cardinality estimation are evaluated along three axes: ***space complexity*** (in bits), ***estimation error***, and ***algorithmic complexity***[1]. In this chapter, I will orbit around this *three-way* tradeoff between these measures and present new theory and new algorithms for optimizing one's cardinality sketches in different contexts.

- In Section 5.2, I investigate the two-way tradeoff between space complexity and estimation error without caring about algorithmic complexity. This two-way tradeoff is *information-theoretic* in nature.
- In Section 5.7, I study a class of cardinality estimators, called *martingale estimators*, which are invented by Ting [151] and Cohen [49] simultaneously. The martingale

[1]Here the algorithmic complexity of a sketch refers to the running time and also the difficulty of implementing it.

estimators take advantage of the *history* of the sketch and achieve better estimation accuracy. This estimation scheme is both optimal in estimation error and very simple to implement, with the cost of being *non-mergeable*.

- In Section 5.8, I switch back to the mergeable setting and focus on the tradeoff between estimation error and algorithmic complexity. I define a class of statistics, called τ-*generalized remaining areas* (τ-GRA), which are simple to compute and are at the same time almost statistically efficient.

Before presenting technical results, let me first characterize the structure of mergeable cardinality sketches, so that the existing sketches can be defined in a uniform way, as in Fig. 5.1.

5.1 Prelude: The Structure of Cardinality Sketches

5.1.1 Representation of the Sketch State Space

I will work in the (M, Ω, Ψ) homomorphic sketching framework (Definition 2.3). For cardinality estimation over incremental streams, the element state space is $M = (\{0,1\}, \mathrm{OR})$. Let the sketch state space be $(\Omega, \uplus)$ and the random-morphism be Ψ. Note that since M is idempotent (check that $0 \,\mathrm{OR}\, 0 = 0$ and $1 \,\mathrm{OR}\, 1 = 1$), the idempotency is projected to $(\Omega, \uplus)$ by the random-morphism.

Lemma 5.1 (idempotency)**.** *If $M = (\{0,1\}, \mathrm{OR})$ and all sketch states in Ω can be reached with non-zero probability by some input, then the sketch state space $(\Omega, \uplus)$ is* idempotent*, i.e., for any $s \in \Omega$, $s \uplus s = s$.*

Proof. If $s = 0$, then $0 \uplus 0 = 0$ since 0 is the identity of Ω. If $s \neq 0$, then there exists an input sequence $v_1, v_2, \ldots, v_w$ such that

$$\mathbb{P}(\Psi^{(v_1)}(1) \uplus \cdots \uplus \Psi^{(v_w)}(1) = s) > 0.$$

Let us insert this sequence again and we have

$$\begin{aligned} 0 &< \mathbb{P}(\Psi^{(v_1)}(1) \uplus \cdots \uplus \Psi^{(v_w)}(1) = s) \\ &= \mathbb{P}(\Psi^{(v_1)}(1) \uplus \cdots \uplus \Psi^{(v_w)}(1) = s, \text{ and} \\ &\quad \Psi^{(v_1)}(1) \uplus \cdots \uplus \Psi^{(v_w)}(1) \uplus \Psi^{(v_1)}(1) \uplus \cdots \uplus \Psi^{(v_w)}(1) = s \uplus s) \end{aligned}$$

note that Ω is commutative and $\Psi\colon M \to \Omega$ is a morphism,

$$\begin{aligned}
&= \mathbb{P}(\Psi^{(v_1)}(1) \uplus \cdots \uplus \Psi^{(v_w)}(1) = s, \text{ and} \\
&\quad \Psi^{(v_1)}(1 \operatorname{OR} 1) \uplus \cdots \uplus \Psi^{(v_w)}(1 \operatorname{OR} 1) = s \uplus s) \qquad (\Omega \text{ is commutative and associative}) \\
&\leq \mathbb{P}(s = s \uplus s).
\end{aligned}$$

As $s = s \uplus s$ is a deterministic event, $\mathbb{P}(s = s \uplus s) > 0$ implies that $s = s \uplus s$. □

This implies that the sketch state space $(\Omega, \uplus)$ is a *commutative and idempotent monoid*. For algorithmic convenience, I will identify those monoids by covering structures.

Definition 5.2 (covering structure)**.** Let the universe be $[0, 1)$ and let $\mathcal{P}([0, 1))$ be its power set. A set system $\mathcal{C}$ is a *covering structure* if

1. it has a bottom: $\emptyset \in \mathcal{C}$;
2. it has a top: $\forall A \in \mathcal{C}, A \subset [0, 1)$ and $[0, 1) \in \mathcal{C}$;
3. it is closed under intersection: $\forall A, B \in \mathcal{C}, A \cap B \in \mathcal{C}$.

Remark 5.1. This representation is algorithmically convenient as the universe of $\mathcal{C}$ will correspond to the space of hash values returned by the random oracle. By convention we assume the random oracle returns a Uniform$(0, 1)$ random variable and thus the universe of $\mathcal{C}$ is defined to be $[0, 1)$ here. Later when we consider a sketch consisting of m i.i.d. subsketches where the random oracle returns m i.i.d. Uniform$(0, 1)$ per index, the universe will be replaced by $[0, 1)^m$.

Definition 5.3 (covers)**.** For any covering structure $\mathcal{C}$ and subset $C \subset [0, 1)$, the $\mathcal{C}$*-cover* of C is defined as $\sup|_{\mathcal{C}} C = \bigcap_{D \in \mathcal{C}, C \subset D} D$, i.e. $\sup|_{\mathcal{C}} C$ is the smallest set in $\mathcal{C}$ that is a superset of C.

Remark 5.2. This is well-defined since $\mathcal{C}$ is closed under intersection (assuming $\mathcal{C}$ is finite) and $[0, 1) \in \mathcal{C}$.

Theorem 5.1 (identification by covering structures)**.** *Let $(\Omega, \uplus)$ be any finite commutative and idempotent monoid. Then there exists a covering structure $\mathcal{C}$ such that $(\Omega, \uplus)$ is isomorphic to $(\mathcal{C}, \sup)$, where for any $A, B \in \mathcal{C}$, $\sup\{A, B\} = \sup|_{\mathcal{C}}(A \cup B)$.*

Remark 5.3. We restrict ourselves to finite Ω to avoid subtleties which necessarily need axiom of choice. Later we will consider infinite Ω that can be represented by a sequence of finite Ω_js.

Proof. A commutative and idempotent monoid is just a join-semilattice with the identity as the bottom element. Every join-semilattice can be represented as a set system that is closed under intersection. See, e.g., [32] for the structure theory of semilattices. □

In the rest of this chapter, the sketch state space of cardinality spaces will be identified by covering structures.

5.1.2 Classic Sketches

We now define the classic sketches for cardinality estimation uniformly as covering structures in Table 5.1. See Fig. 5.1 for a visualization.

Sketch	Covering Structure
PCSA [83]	$\left\{\bigcup_{k\in A}[2^{-(k+1)}, 2^{-k}) \mid A \subset \mathbb{N}\right\}$
LinearCounting [161]	$\left\{\bigcup_{k\in A}[(k-1)/m, k/m) \mid A \subset [m]\right\}$
m-AdaptiveSampling [81]	$\left\{\{r_1, \ldots, r_m\} \cup [2^{-(k+1)}, 1) \mid k \in \mathbb{N}, \forall i \in [m] : r_i \in [0,1]\right\}$
LogLog [73], HyperLogLog [82]	$\left\{[2^{-(k+1)}, 1) \mid k \in \mathbb{N}\right\}$
k-Min [90]	$\{\{r_1, \ldots, r_{k-1}\} \cup [r_k, 1) \mid \forall i \in [k] : r_i \in [0,1]\}$

Table 5.1: Classic cardinality sketches as covering structures. Note that these algorithms are quite different in the form of pesudocode but can now be uniformly represented by covering structures.

The correspondence from covering structures to actual algorithms is intuitive: add a fresh random point in $[0, 1)$ for every new distinct insertion and the current state is the smallest set $A \in \mathcal{C}$ that covers all the random points. One may check that such definitions are indeed correct simulations of the sketches in Table 5.1. For example, k-Min stores the smallest k hash values $r_1, \ldots, r_k$ and the smallest set in its covering structure that covers all the hash values is exactly $\{r_1, \ldots, r_{k-1}\} \cup [r_k, 1)$. Algorithmically, the covering structure $\mathcal{C}$ specifies the following behaviors for the induced sketch.

- The current state $C \in \mathcal{C}$ specifies the range of hash values that would *not* change the current state. By construction, if a new hash value x is added, and $x \in C$, the current state remains C. If otherwise $x \notin C$, one needs a larger cover set in $\mathcal{C}$ and thus the sketch state necessarily changes.
- Given the current state $C \in \mathcal{C}$ and a new hash value $x \notin C$, the next state C' is the smallest set in $\mathcal{C}$ that is a supset of $C \cup \{x\}$.

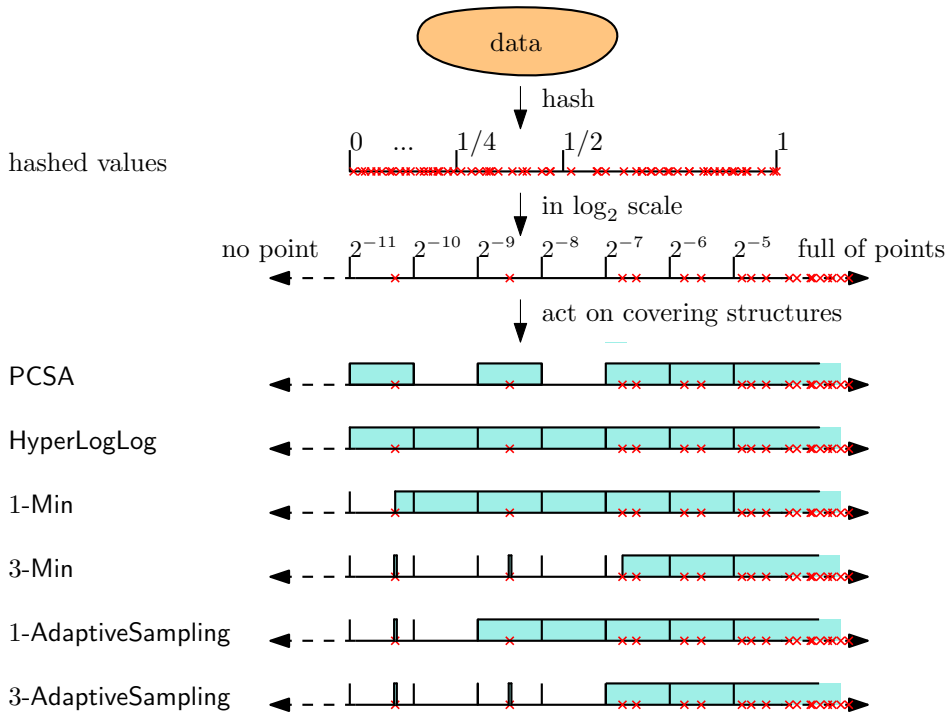


Figure 5.1: Classic cardinality sketches as covering structures

5.1.3 Poisson Activation

I will poissonize the insertions when considering algorithms in the mergeable setting. Given a sketch state space $\mathcal{C}$, we consider only one random-morphism Ψ. Since Ψ is a homorphism from M (the 1-bit monoid) to $(\mathcal{C}, \sup)$, $\Psi(0) = \emptyset$ and we only need to define $\Psi(1)$, which is a random sketch state.

Definition 5.4 (Poisson activation)**.** Given a sketch state space $(\mathcal{C}, \sup)$, the Poisson activation is a random-morphism $\Psi : \{0, 1\} \to \mathcal{C}$ such that

$$\Psi(1) \sim \sup|_{\mathcal{C}}\{X_1, \ldots, X_Y\},$$

where X_is are i.i.d. Uniform$(0, 1)$ random variables and $Y \sim$ Poisson(1).

Remark 5.4. Back in 1983, Flajolet and Martin [83] analyzed PCSA by first *poissonization*, i.e., assuming Poisson(λ) number of hash values are inserted, and then *depoissonization*, i.e., bounding the difference in estimation between the poissonized sketch and the original sketch (with λ insertions). Poissonization also has an algorithmic interpretation: hashing every element to a Poisson(1) number of points (see a recent application [37]). In our framework, Poisson activation ensures infinite divisibility, which is inherently needed for scale-invariance

(Section 3.1.2). Indeed, the current sketch state is the smallest set $A \in \mathcal{C}$ that covers all the Poisson(λ) random points in $[0, 1]$, where λ is the number of distinct elements, and for any $k \in \mathbb{Z}_+$, the current state can be written as the merged state of k i.i.d. states, each distributed as $\sup|_{\mathcal{C}}\{X_1, \ldots, X_{Y'}\}$ where $Y' \sim \text{Poisson}(\lambda/k)$.

I will assume Poisson activation in the mergeable/distributed setting. A sketch now is fully described by its covering structure $\mathcal{C}$. See Algorithm 5.1 and Fig. 5.2.

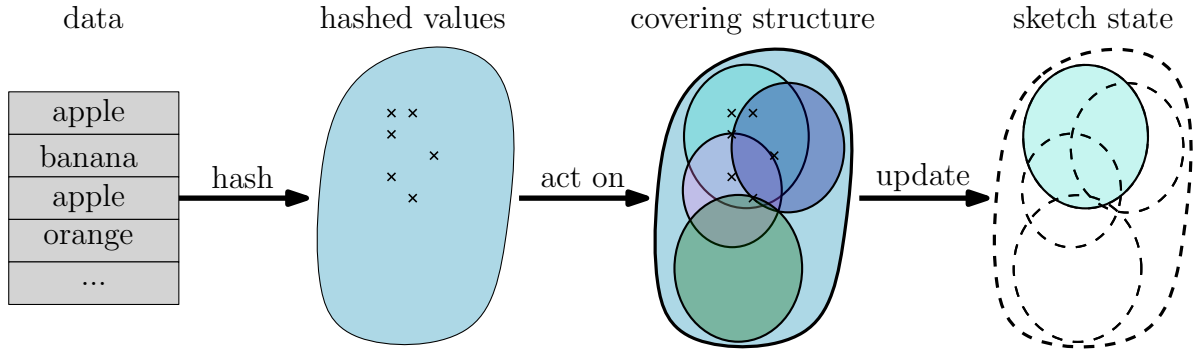


Figure 5.2: Pipeline of cardinality sketches. Every distinct element is hashed to a Poisson(1) number of random points. After λ distinct insertions, a Poisson(λ) number of random points has acted on the covering structure. The final sketch state is the smallest set that covers all the random points. The only design choice in this framework is the covering structure.

Algorithm 5.1: The cardinality sketch identified by a covering structure $\mathcal{C}$. Covering structures are defined in Definition 5.2

Sketch : current state $C \in \mathcal{C}$, initialized as $\emptyset$

Hash function: H hashes each element to a Poisson(1) number of independent, uniformly random points in $[0, 1)$

Insert (v)**:**

```
// upon inserting v ∈ [n]
// update to the minimum state that covers both the current state
   C and the inserted points H(v)
```

$C \leftarrow \sup|_{\mathcal{C}}(C \cup H(v))$

end

MergeWith (C')**:**

```
// upon merging with another sketch C' ∈ C
// merge to the minimum state that covers both C and C'
```

$C \leftarrow \sup|_{\mathcal{C}}(C \cup C')$

end

5.2 The Fisher-Shannon Number

Given the setup so far, the first natural question to ask is which covering structure induces the "best" cardinality sketch. I focus on the trade-off between the *memory space* and *relative variance*, while assuming unbounded computing power and random oracles. As we want the "best" sketch, we have to quantify *how good* is a sketch in terms of the memory-variance tradeoff. In this section, I will propose a metric of a sketch that only depends on its covering structure, based on two important information measures—Shannon entropy and Fisher information number. This metric is named as the **Fi**sher-**Sh**annon number (Fish), which characterizes the memory-variance product of the best implementation of a cardinality sketch. The rather ambiguous task of finding the "best" cardinality sketch can thus now be formally defined ato finding the covering structure with the minimum Fish-number.

5.2.1 Background

As one of the most fundamental streaming problems, cardinality estimation has been studied intensively in both the random oracle model and the standard model. Various types of cardinality sketches are invented. The homomorphic sketches we are going to consider cover a large portion of existing sketches. Nevertheless, there are sketches with weaker guarantees.

Commutative and idempotent A sketch is commutative and idempotent if it is insensitive to the order and multiplicity of the insertions. One can show that these are exactly equivalent to the homomorphic sketches and can thus be identified by covering structures. Let us call them commutative sketches for simplicity. Commutative sketches are clearly ideal for estimating cardinality in the distributed setting, because when the randomness is fixed, the final sketch state will be insensitive to the order of the insertions.

History independent A weaker condition is being *history independent*. A sketch is history independent if the *distribution* of its final state is insensitive to the order and multiplicity of the insertions (over a sequential stream). It suffices to have history independence if the inputs are sequential. A notable example are the Martingale sketches discovered by Ting [151] and Cohen [49].

Non-history independent Non-history independent sketches are mostly heuristics whose memory state distribution can depend on other factors apart from cardinality, e.g., the

insertion pattern. Such heuristics usually have poor *worst-case*[2] guarantees. E.g., a proposal of Sedgewick [147] called HyperBitBit, which is constructed as a lossy compression of LogLog, is non-history independent. It has constant relative bias and variance, no matter how many subsketches are used; see Section 5.6.1.

Table 5.2 summarizes prior work. To compare random oracle and standard model algorithms, note that an asymptotically unbiased $\tilde{O}(m)$-bit sketch with standard error $O(1/\sqrt{m})$ is morally similar to an $\tilde{O}(\epsilon^{-2})$-bit sketch with (ϵ, δ)-approximation guarantee, $\delta = O(1)$. However, the two guarantees are formally incomparable. The (ϵ, δ)-guarantee does not specifically claim anything about bias or variance, and with probability δ the error is technically not bounded.

Though the optimal (up to constants) cardinality sketches have been proved by Kane et al. [110] in 2010, one needs to be more careful when designing memory representation and the estimator if he wants to also optimize the constants.

Sketch Compression The first thing many researchers notice about classic sketches like (Hyper)LogLog and PCSA is their wastefulness in terms of space. Improving space by constant factors can have a disproportionate impact on *time*, since this allows for more sketches to be stored at lower levels of the cache-hierarchy. In low-bandwidth situations (e.g., distributed sensor networks), improving space can be an end in itself [146, 56, 132]. The idea of sketch compression goes back to the original Flajolet-Martin paper [83], who observed that the PCSA sketch matrix consists of nearly all 1s in the low-order bits, nearly all 0s in the high order bits, and a mix in between. They suggested encoding a sliding window of width 8 across the sketch matrix. By itself this idea does not work well.

In her Ph.D. thesis [72, p. 136], Durand observed that each counter in LogLog has constant entropy, and can be encoded with a prefix-free code with expected length 3.01. The state-of-the-art standard model [110, 13] algorithms use this property, and further show that a compressed representation of these counters can be updated in $O(1)$ time [12].

The practical efforts to compress (Hyper)LogLog have used fixed-length codes rather than variable length codes. Xiao, Chen, Zhou, and Luo [168] proposed a variant of HyperLogLog called HLL-Tailcut+ that codes the minimum counter and m 3-bit offsets, where $\{0, \ldots, 6\}$ retain their natural meaning but larger offsets are truncated at 7. They claimed that with a different estimation function, the variance is $1/\sqrt{m}$. This claim is incorrect; the relative bias and squared error of this estimator are constant, independent of m.[3] An implementation of

[2] which means they might work perfectly well on "usual" data streams.

[3] The two sequences in Section 5.6.1 suffice to show that the bias can be made independent of m.

Random Oracle Model

Mergeable Sketches

		Sketch Size (Bits)	Approximation Guarantee
Flajolet & Martin	(PCSA) 1983	$m \log n$	Std. err. $\approx 0.78/\sqrt{m}$
Flajolet	(AdaptiveSampling) 1990	$m \log n + \log \log n$	Std. err. $\approx 1.21/\sqrt{m}$
Durand & Flajolet	(LogLog) 2003	$m \log \log n$	Std. err. $\approx 1.3/\sqrt{m}$
Giroire	(Min) 2005	$m \log n$	Std. err. $\approx 1/\sqrt{m}$
Chassaing & Gerin	(Min) 2006	$m \log n$	Std. err. $\approx 1/\sqrt{m}$
Estan, Varghase & Fisk	(Multires.Bitmap) 2006	$m \log n$	Std. err. $O(1/\sqrt{m})$
Beyer, Haas, Reinwald Sismanis & Gemulla	2007	$m \log n$	Std. err. $\approx 1/\sqrt{m}$
Flajolet, Fusy, Gandouet & Meunier	(HyperLogLog) 2007	$m \log \log n$	Std. err. $\approx 1.04/\sqrt{m}$
Lumbroso	2010	$m \log n$	Std. err. $\approx 1/\sqrt{m}$
Lang	(Compressed FM85) 2017	$\approx \log n + 1.99m$ (in expectation)	Std. err. $\approx 1/\sqrt{m}$
new	(Fishmonger) 2020	$O(\log^2 \log n) + (1 + o(1))(H_0/I_0)m$ where $H_0/I_0 \approx 1.98016$	Std. err. $\approx 1/\sqrt{m}$

Non-mergeable Sketches

Chen, Cao, Shepp & Nguyen	(S-Bitmap) 2009	m	Std. err. $\approx \frac{\log(eU/m)/2}{\sqrt{m}}$	$(\star)$
Helmi, Lumbroso, Martínez & Viola	(Recordinality) 2012	$(1 + o(1))m \log n$	Std. err. $\tilde{O}(1)/\sqrt{m}$	$(\star)$
Cohen	(Martingale LogLog) 2014	$m \log \log n + \log n$	Std. err. $\approx 0.833/\sqrt{m}$	$(\star)$
Ting	(Martingale MinCount)	$(m + 1) \log n$	Std. err. $\approx 0.71/\sqrt{m}$	
Sedgewick	(HyperBitBit) 2016	134	? (See Section 5.6)	$(\star\star)$

Standard Model

Alon, Matias & Szegedy	1996	$O(\log n)$	$(\epsilon, 2/\epsilon)$-approx., $\epsilon \geq 2$
Gibbons & Tirthapura	2001	$O(\epsilon^{-2} \log n \log \delta^{-1})$	(ϵ, δ)-approx.
Bar-Yossef, Kumar & Sivakumar	2002	$O(\epsilon^{-3} \log n \log \delta^{-1})$	(ϵ, δ)-approx.
Bar-Yossef, Jayram, Kumar, Sivakumar & Trevisan	2002	$O\left([\epsilon^{-2} \log \log n + \log n] \log \delta^{-1}\right)$	(ϵ, δ)-approx.
Kane, Nelson & Woodruff	2015	$O([\epsilon^{-2} + \log n] \log \delta^{-1})$	(ϵ, δ)-approx.
Błasiok	2018	$O(\epsilon^{-2} \log \delta^{-1} + \log n)$	(ϵ, δ)-approx.

Lower Bounds

Trivial		$\Omega(\log \log n)$	$(O(1), O(1))$-approx. (rand. oracle)
Alon, Matias & Szegedy	1996	$\Omega(\log n)$	$(O(1), O(1))$-approx. (std. model)
Indyk & Woodruff	2003	$\Omega(\epsilon^{-2})$	$(\epsilon, O(1))$-approx. (Both)
Jayram & Woodruff	2011	$\Omega(\epsilon^{-2} \log \delta^{-1})$	(ϵ, δ)-approx. (Both)
new	2020	$(H_0/I_0)m$	Std. err. $1/\sqrt{m}$ (Linearizable)

Table 5.2: Summary of cardinality sketches. Algorithms analyzed in the RANDOM ORACLE model assume oracle access to a uniformly random hash function $h : [n] \to [0, 1]$. Algorithms in the standard model can generate uniformly random bits, but must store any hash functions explicitly. The state of a *commutative* algorithm is independent of the order elements are processed, once all randomness is fixed. All algorithms are commutative except for those marked with star(s). Algorithms marked with $(\star)$ are *history independent*, meaning before the randomness is fixed, the distribution of the final state depends only on the cardinality, not the order/identity of elements. The algorithm marked with $(\star\star)$ is neither commutative nor history independent.

HyperLogLog in Apache *DataSketches* [150] uses a 4-bit offset, where $\{0, \ldots, 14\}$ retain their normal meaning and 15 indicates that the true value is stored in a separate exception list. This is lossless compression, and therefore does not affect the estimation accuracy [82].

Scheuermann and Mauve [146] experimented with compression of PCSA and HyperLogLog sketches to their entropy bounds via arithmetic coding, and noted that, with the usual estimation functions [83, 82], Compressed-PCSA is slightly smaller than Compressed-HLL for the same standard error. Lang [117] went a step further, and considered Compressed-PCSA and Compressed-HLL sketches, but with several improved estimators including Minimum Description Length (MDL), which in this context is essentially the same as the Maximum Likelihood Estimator (MLE). Lang's numerical calculations revealed that Compressed-PCSA+MDL is *substantially* better than Compressed-LL+MDL, and that off-the-shelf compression algorithms achieve compression to within roughly 10% of the entropy bounds. A variation on Lang's scheme is included in Apache *DataSketches* under the name CPC for Compressed Probabilistic Counting [150]. By buffering stream elements and only decompressing when the buffer is full, the amortized cost per insertion can be reduced to $\tilde{O}(1)$ from $\tilde{O}(m)$, which is competitive in practice [150].

To sum up, the idea of compressing sketches has been studied since the beginning, heuristically [83, 147, 168], experimentally [146, 150], and numerically [117], but to our knowledge never analytically.

Statistical efficiency Another way to boost the performance of cardinality sketches is to use better estimators, ideally being statistically efficient. As the number of subsketches go to infinity, the Maximum Likelihood Estimator (MLE) is asymptotically efficient. MLE for the Min sketch was studied by Chassaing and Gerin [35] and Clifford and Cosma [47]. Clifford and Cosma [47] and Ertl [77] studied the computational complexity of MLE in LogLog sketches. Lang [117] experimented with MLE-type estimators for PCSA and LogLog. Cohen, Katzir, and Yehezkel [55] looked at MLE estimators for estimating the cardinality of set intersections. Nevertheless, these works have not formally defined the "asymptotic relative variance," which a suitable metric for goodness. There is a subtle barrier here, haunting every serious analysis paper about cardinality sketches—the normalized relative variance does *not* converge as the number of subsketches goes to infinity. Instead, as first noticed by Flajolet and Martin in 1983, the behavior of sketches like PCSA will *oscillate* in some small but non-vanishing magnitude. Later, I will carefully handle such subtleties.

5.2.2 Base-q Generalization of PCSA and LogLog

As binary numbers are native to computers (at least till now), both PCSA and LogLog choose base 2 for performance. Recall that, in the covering structure representation, we have

$$\mathcal{C}_{\mathsf{PCSA}} = \left\{ \bigcup_{k \in A} [2^{-(k+1)}, 2^{-k}) \;\middle|\; A \subset \mathbb{N} \right\}$$
$$\mathcal{C}_{\mathsf{LogLog}} = \left\{ [2^{-(k+1)}, 1) \;\middle|\; k \in \mathbb{N} \right\}.$$

Nevertheless, if one only cares about the meomory-variance tradeoff, it is curious to ask whether one can improve the performance by using another base. Let us now define the base-q generalizations of PCSA and LogLog.

Definition 5.5 (q-PCSA and q-LogLog)**.** Let $q \in (1, \infty)$. Denote base-q PCSA as q-PCSAand base-q LogLog as q-LogLog. They are identified by the following covering structures.

$$\mathcal{C}_{q\text{-}\mathsf{PCSA}} = \left\{ \bigcup_{k \in A} [(q-1)^{-1} q^{-(k+1)}, (q-1)^{-1} q^{-k}) \;\middle|\; A \subset \mathbb{N} \right\}$$
$$\mathcal{C}_{q\text{-}\mathsf{LogLog}} = \left\{ [q^{-(k+1)}, 1) \;\middle|\; k \in \mathbb{N} \right\}.$$

5.2.3 Overview of Results

To understand the intrinsic tradeoff between space and accuracy, we need to grapple with two of the influential notions of "information" from the 20th century: *Shannon entropy*, which controls the (expected) space of an optimal encoding, and *Fisher information*, which limits the variance of an asymptotically unbiased estimator, via the Cramér-Rao lower bound [30, 154].

To be specific, consider a sketch $S = (S(0), \ldots, S(m-1))$ composed of m i.i.d. experiments over a multiset with cardinality λ. We assume that these experiments are *useful*, in the sense that any two cardinalities λ_0, λ_1 induce distinct distributions on S. Under this condition and some mild regularity conditions, it is well known [30, 154] that the Maximum Likelihood Estimator (MLE):

$$\hat{\lambda}(S) = \arg\max_{\lambda} \Pr(S \mid \lambda)$$

is asymptotically unbiased and meets the Cramér-Rao lower bound:

$$\lim_{m \to \infty} \sqrt{m} \left(\hat{\lambda}(S) - \lambda \right) \sim \text{Normal}\left(0, \frac{1}{I_{S(0)}(\lambda)} \right).$$

Here $I_{S(0)}(\lambda)$ is the *Fisher information* number of λ associated with any one component of the vector S. This implies that as m gets large, $\hat{\lambda}(S)$ tends toward a normal distribution Normal $\left(\lambda, \frac{1}{I_S(\lambda)}\right)$ with variance $1/I_S(\lambda) = 1/(m \cdot I_{S(0)}(\lambda))$. (See Section 5.2.4.)

Suppose for the moment that $I_S(\lambda)$ is scale-invariant, in the sense that we can write it as $I_S(\lambda) = \mathcal{I}(S)/\lambda^2$, where $\mathcal{I}(S)$ does not depend on λ. We can think of $\mathcal{I}(S)$ as measuring the *value* of experiment S to estimating the parameter λ, but it also has a *cost*, namely the space required to store the outcome of S. By Shannon's source-coding theorem we cannot beat $H(S \mid \lambda)$ bits on average, which we also assume for the time being is scale-free, and can be written $\mathcal{H}(S)$, independent of λ. We measure the *efficiency* of an experiment by its Fisher-Shannon (Fish) number, defined to be the ratio of its cost to its value:

$$\mathsf{Fish}(S) = \frac{\mathcal{H}(S)}{\mathcal{I}(S)}.$$

In particular, this implies that using sketching scheme S to achieve a standard error of $\sqrt{1/b}$ (variance $1/b$) requires $\mathsf{Fish}(S) \cdot b$ bits of storage on average,[4] i.e., lower Fish-numbers are superior. The actual definition of Fish (Section 5.2.5) is slightly more complex in order to deal with sketches S that are not *strictly* scale-invariant.

Our main results are as follows.

(1) Let q-PCSA be the natural base-q analogue of PCSA, which is 2-PCSA. We prove that the Fish-number of q-PCSA does not depend on q, and is precisely:

$$\mathsf{Fish}(q\text{-PCSA}) = \frac{H_0}{I_0} \approx 1.98016.$$

where

$$H_0 = \frac{1}{\log 2} + \sum_{k=1}^{\infty} \frac{1}{k} \log_2 (1 + 1/k),$$
$$I_0 = \zeta(2) = \frac{\pi^2}{6}.$$

The constant H_0/I_0 is very close to Lang's [117] numerical calculations of 2-PCSA's entropy and mean squared error. Let q-LogLog be the natural base-q analogue of LogLog = 2-LogLog. Whereas the Fisher information for q-PCSA is expressed in terms of the *Riemann zeta* function ($\zeta(2)$), the Fisher information of q-LogLog is expressed

[4] Set m such that $b = \mathcal{I}(S) = m \cdot \mathcal{I}(S(0))$. The expected space required is $m \cdot \mathcal{H}(S(0)) = b(\mathcal{H}(S(0))/\mathcal{I}(S(0))) = b \cdot \mathsf{Fish}(S)$.

in terms of the *Hurwitz zeta* function $\zeta(2, \frac{q}{q-1}) = \sum_{k \geq 0}(k + \frac{q}{q-1})^{-2}$. We prove that q-LogLog is always worse than PCSA, but approaches the efficiency of PCSA in the limit, i.e.,

$$\forall q.\ \mathsf{Fish}(q\text{-LogLog}) > H_0/I_0 \quad \text{but} \lim_{q \to \infty} \mathsf{Fish}(q\text{-LogLog}) = H_0/I_0.$$

(2) The results of (1) should be thought of as *lower bounds* on implementing compressed representations of q-PCSA and q-LogLog. We give a new sketch called Fishmonger whose space, at all times, is $O(\log^2 \log n) + (1 + o(1))(H_0/I_0)b \approx 1.98b$ bits and whose standard error, at all times, is $1/\sqrt{b}$, with probability $1 - 1/\operatorname{poly}(b)$.[5]

(3) Is it possible to go below H_0/I_0? We define a natural class of commutative sketches called *linearizable* sketches and prove that no member of this class has Fish-number strictly smaller than H_0/I_0. Since all the popular commutative sketches are, in fact, linearizable, we take this as circumstantial evidence that Fishmonger is information-theoretically *optimal*, up to a $1 + o(1)$ factor in space/variance.

5.2.4 Preliminaries

Shannon Entropy Let X_1 be a random variable with probability density/mass function f. The *entropy* of X_1 is defined to be

$$H(X_1) = \mathbb{E}(-\log_2 f(X_1)).$$

Let (X_1, R_1) be a pair of random variables with joint probability function $f(x_1, r_1)$. When X_1 and R_1 are independent, entropy is additive: $H(X_1, R_1) = H(X_1) + H(R_1)$. We can generalize this to possibly dependent random variables by the *chain rule for entropy*. We first define the notion of *conditional entropy*. The conditional entropy of X_1 given R_1 is defined as

$$H(X_1 \mid R_1) = \mathbb{E}\left(-\log_2 f(X_1 \mid R_1)\right),$$

[5]This sketch was developed before we were aware of Lang's technical report [117]. If one combined Lang's Compressed-FM85 sketch with our analysis, it would yield a theorem to the following effect: at any particular moment in time the *expected* size of the sketch encoding is $\log n + (H_0/I_0 + \epsilon)b$ and the standard error at most $1/\sqrt{b}$, for some small constant $\epsilon > 0$ (see Section 5.2.5 concerning the periodic behavior of sketches). Fishmonger improves this by bringing the leading coefficient down to H_0/I_0 and making a "for all" guarantee: that the sketch is stored in $O(\log^2 \log n) + (1 + o(1))(H_0/I_0)b$ bits <u>*at all times*</u>, with high probability $1 - 1/\operatorname{poly}(b)$.

which is interpreted as the *average* entropy of X_1 after knowing R_1.

Theorem 5.2 (chain rule for entropy [63])**.** *Let $(X_0, X_1, \ldots, X_{m-1})$ be a tuple of random variables. Then $H(X_0, X_1, \ldots, X_{m-1}) = \sum_{i=0}^{m-1} H(X_i \mid X_0, \ldots, X_{i-1})$.*

Shannon's source coding theorem says that it is impossible to encode the outcome of a *discrete* random variable X_1 in fewer than $H(X_1)$ bits on average. On the positive side, it is possible [63] to assign code words such that the outcome $[X_1 = x]$ is communicated with less than $\lceil \log_2(1/f(x)) \rceil$ bits, e.g., using arithmetic coding [163, 127].

Fisher Information and the Cramér-Rao Lower Bound Let $F = \{f_\lambda \mid \lambda \in \mathbb{R}\}$ be a family of distributions parameterized by a single unknown parameter $\lambda \in \mathbb{R}$. (We do not assume there is a prior distribution on λ.) A *point estimator* $\hat{\lambda}(X)$ is a statistic that estimates λ from a vector $\mathbf{X} = (X_0, \ldots, X_{m-1})$ of samples drawn i.i.d. from f_λ.

The accuracy of a "reasonable" point estimator is limited by the properties of the distribution family F itself. Informally, if every $f_\lambda \in F$ is sharply concentrated and statistically far from other $f_{\lambda'}$ then f_λ is *informative*. Conversely, if f_λ is poorly concentrated and statistically close to other $f_{\lambda'}$ then f_λ is uninformative. This measure is formalized by the *Fisher information* [154, 30].

Fix $\lambda = \lambda_0$ and let $X \sim f_\lambda$ be a sample drawn from f_λ. The Fisher information number with respect to the observation X at λ_0 is defined to be:[6]

$$I_X(\lambda_0) = \mathbb{E}\left(\frac{\frac{\partial}{\partial\lambda} f_\lambda(X)}{f_\lambda(X)}\right)^2 |_{\lambda=\lambda_0} .$$

The *conditional Fisher information* of X_1 given X_0 at $\lambda = \lambda_0$ is defined as

$$I_{X_1|X_0}(\lambda_0) = \mathbb{E}\left(\frac{\frac{\partial}{\partial\lambda} f_\lambda(X_1 \mid X_0)}{f_\lambda(X_1 \mid X_0)}\right)^2 |_{\lambda=\lambda_0} .$$

Similar to Shannon's entropy, we also have a chain rule for Fisher information numbers.

Theorem 5.3 (chain rule for Fisher information [170])**.** *Let $\mathbf{X} = (X_0, X_2, \ldots, X_{m-1})$ be a tuple of random variables all depending on λ. Under mild regularity conditions, $I_{\mathbf{X}}(\lambda) = \sum_{i=0}^{m-1} I_{X_i|X_0,\ldots,X_{i-1}}(\lambda)$. Specifically if $\mathbf{X} = (X_0, \ldots, X_{m-1})$ is a set of* independent *samples from f_λ then $I_{\mathbf{X}}(\lambda) = m \cdot I_{X_0}(\lambda)$.*

[6] Since in this section the parameter is always the cardinality, the parameter λ is omitted in the notation $I_X(\lambda_0)$.

The celebrated Cramér-Rao lower bound [154, 30] states that, under mild regularity conditions (see Section 5.2.4), for any unbiased estimator $\hat{\lambda}(\mathbf{X})$ with finite variance,

$$\mathbb{V}(\hat{\lambda} \mid \lambda) \geq \frac{1}{I_{\mathbf{X}}(\lambda)}.$$

Suppose now that $\hat{\lambda}(\mathbf{X} = (X_0, \ldots, X_{m-1}))$ is, in fact, the Maximum Likelihood Estimator (MLE) from m i.i.d. observations. Under mild regularity conditions, it is asymptotically normal and efficient, i.e.,

$$\lim_{m \to \infty} \sqrt{m}(\hat{\lambda} - \lambda) \sim \text{Normal}\left(0, \frac{1}{I_{X_0}(\lambda)}\right),$$

or equivalently, $\hat{\lambda} \sim \text{Normal}\left(\lambda, \frac{1}{I_{\mathbf{X}}(\lambda)}\right)$ as $m \to \infty$. In the Cardinality Estimation problem we are concerned with *relative* variance and *relative* standard deviations (standard error). Thus, the corresponding lower bound on the relative variance is $(\lambda^2 \cdot I_{\mathbf{X}}(\lambda))^{-1}$. We define the *normalized* Fisher information number of λ with respect to the observation $\mathbf{X}$ to be $\lambda^2 \cdot I_{\mathbf{X}}(\lambda)$.

Regularity Conditions and Poissonization The asymptotic normality of MLE and the Cramér-Rao lower bound depend on various regularity conditions [170, 1, 11], e.g., that $f_\lambda(x)$ is differentiable with respect to λ and that we can swap the operators of differentiation w.r.t. λ and integration over observations x. (We only consider discrete observations here, so this is just a summation.)

A key regularity condition of Cramér-Rao is that the support of f_λ does not depend on λ, i.e., the set of possible observations is independent of λ.[7] Strictly speaking our sketches do not satisfy this property, e.g., when $\lambda = 1$ the only possible PCSA sketches have Hamming weight 1. To address this issue we *poissonize* the model, as in [73, 82]. Consider the following two processes.

Discrete counting process. Starting from time 0, an element is inserted at every time $k \in \mathbb{N}$.

Poissonized counting process. Starting from time 0, elements are inserted memorylessly with rate 1. This corresponds to a *Poisson point process* of rate 1 on $[0, \infty)$.

[7] A canonical example violating this condition (and one in which the Cramér-Rao bound can be beaten) is when θ is the parameter and the observation X is sampled uniformly from $[0, \theta]$; see [30].

For both processes, our goal would be to estimate the current time λ. In the discrete process the number of insertions is precisely $\lfloor\lambda\rfloor + 1$ whereas in the Poisson one it is $\tilde{\lambda} \sim \text{Poisson}(\lambda)$. When λ is sufficiently large, any estimator for $\tilde{\lambda}$ with standard error $c/\sqrt{m}$ also estimates λ with standard error $(1 - o(1))c/\sqrt{m}$, since $\tilde{\lambda} = \lambda \pm \tilde{O}(\sqrt{\lambda})$ with probability $1 - 1/\operatorname{poly}(\lambda)$. Since we are concerned with the asymptotic efficiency of sketches, we are indifferent between these two models.[8]

For our upper and lower bounds we will use the poissonized counting process as the mathematical model. As a consequence, for any real $\lambda > 0$ the support of the sketch state distribution is independent of λ, and f_λ will always be differentiable w.r.t. λ. Henceforth, we use the terms "time" and "cardinality" interchangeably.

5.2.5 Scale-Invariance and Fish Numbers

We are destined to measure the efficiency of observations in terms of entropy (H) and normalized information ($\lambda^2 I$), but it turns out that these quantities are slightly ill-defined, being *periodic* when we really want them to be *constant* (at least in the limit). Thus before defining the Fish number, we still need to make some more preparations. I will first switch from the covering structure view of sketches to a distributional interpretation, then I define a weak notion of *scale-invariance* for sketches. Starting from a weakly scale-invariant sketch, I describe a generic method to iron out its periodic behavior, only then are we ready to formally define the Fish number of a sketch.

Induced Distribution Family of Sketches Given a sketch scheme, Cardinality Estimation can be viewed as a point estimation problem, where the unknown parameter is the cardinality λ and f_λ is the distribution over the final state of the sketch.

Definition 5.6 (Induced Distribution Family)**.** Let $\mathcal{C}$ be covering structure of a sketch (see Section 5.1.1). The *Induced Distribution Family (IDF)* of $\mathcal{C}$ is a parameterized distribution family

$$\Xi_{\mathcal{C}} = \{\xi_{\mathcal{C},\lambda} : \mathcal{C} \to [0, 1] \mid \lambda > 0\},$$

[8] Algorithmically, the Poisson model could be simulated online as follows. When an element a arrives, use the random oracle to generate $\xi_a \sim \text{Poisson}(1)$ and then insert elements $(a, 1), \ldots, (a, \xi_a)$ into the sketch as usual.

where

$$\xi_{\mathcal{C},\lambda}(x) = \mathbb{P}(\sup|_{\mathcal{C}}(\Upsilon_\lambda) = x).$$

Here Υ_λ is a set of Poisson(λ) i.i.d. uniformly random points over the universe of $\mathcal{C}$. Recall that the current state at cardinality λ of a sketch $\mathcal{C}$ is the minimum state that covers all the points in Υ_λ (Algorithm 5.1). Define $X_{\mathcal{C},\lambda} \sim \sup|_{\mathcal{C}}(\Upsilon_\lambda)$ to be the random state at cardinality λ.

We usually consider just the basic version of each sketch, e.g., a single bit-vector for PCSA or a single counter for LogLog. When we apply the machinery laid out in Section 5.2.4 we take m *independent* copies of the basic sketch, i.e., every element is inserted into all m sketches. One could also use *stochastic averaging* [83, 82, 78], which, after poissonization, yields the same sketch with cardinality $\lambda' = m\lambda$.

Weak Scale-Invariance Consider a basic sketch $\mathcal{C}$ with IDF $\Xi_{\mathcal{C}}$, and let $\mathcal{C}^m$ denote a vector of m independent $\mathcal{C}$-sketches. From the Cramér-Rao lower bound we know the variance of an unbiased estimator is at least $\frac{1}{I_{\mathcal{C}^m}(\lambda)} = \frac{1}{m \cdot I_{\mathcal{C}}(\lambda)}$. (Here $I_{\mathcal{C}^m}(\lambda)$ is short for $I_{X_{\mathcal{C}^m,\lambda}}(\lambda)$, where $X_{\mathcal{C}^m,\lambda}$ is the observed final state of $\mathcal{C}^m$ at time λ.) The memory required to store it is at least $H(X_{\mathcal{C}^m,\lambda}) = m \cdot H(X_{\mathcal{C},\lambda})$. Thus the product of the memory and the *relative* variance is lower bounded by

$$\frac{H(X_{\mathcal{C},\lambda})}{\lambda^2 \cdot I_{\mathcal{C}}(\lambda)},$$

which only depends on the distribution family $\Xi_{\mathcal{C}}$ and the unknown parameter λ. However, ideally it would depend only on $\Xi_{\mathcal{C}}$. See Fig. 5.3 for an intuition.

Essentially every existing sketch is insensitive to the scale of λ, up to some coarse approximation. However, it is difficult to design a sketch with a countable state-space that is *strictly* scale-invariant. It turns out that a weaker form is just as good for our purposes.

Definition 5.7 (Weak Scale-Invariance)**.** Let $\mathcal{C}$ be a sketch with induced distribution family $\Xi_{\mathcal{C}}$ and $q > 1$ be a real number. We say $\mathcal{C}$ is *weakly scale-invariant with base q* if for any $\lambda > 0$, we have

$$H(X_{\mathcal{C},\lambda}) = H(X_{\mathcal{C},q\lambda}),$$
$$I_{\mathcal{C}}(\lambda) = q^2 \cdot I_{\mathcal{C}}(q\lambda).$$

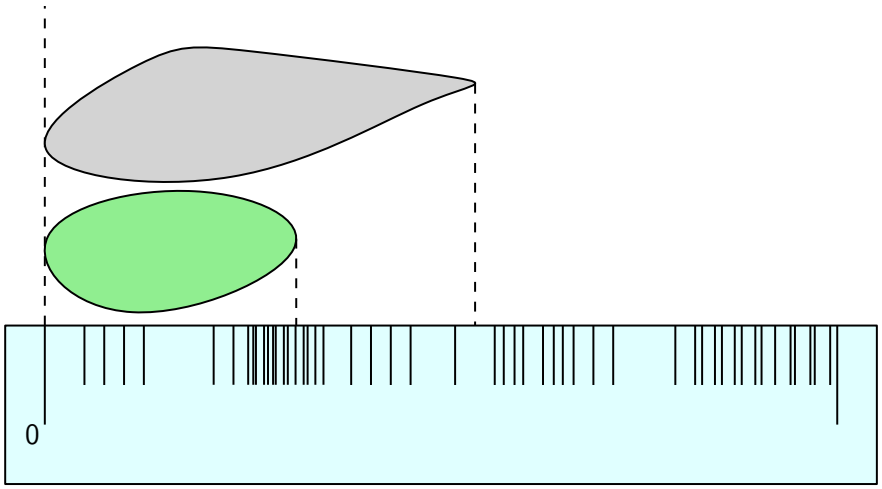


Figure 5.3: Analogy between cardinality sketches and rulers: Why scale-invariant? It is non-ideal for rulers to have non-uniform ticks. The shorter object will be measured more accurately since the ticks around its right end are denser. This non-uniform ruler is an analogy of a non-scale-invariant sketch—its estimation error will depend on the true cardinality being estimated.

Remark 5.5. For example, the original (HyperLogLog and PCSA sketches [80, 82, 73] are, after poissonization, base-2 weakly scale-invariant. See Fig. 5.4 for the ruler analogy.

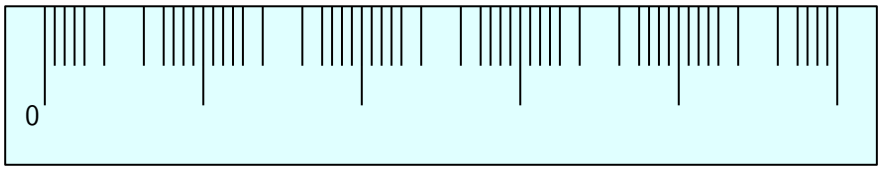


Figure 5.4: Analogy between cardinality sketches and rulers: Weak scale-invariance. The ruler above is not uniform though, its ticks are *periodic*. This periodic ruler is an analogy of existing cardinality sketches such as PCSA [83] and HyperLogLog [82], whose performance are multiplicatively periodic in the true cardinality λ. This is the reason that such sketches are difficult to analyze—the variance does not converge and one needs to bound its oscillation [83].

Observe that if a sketch $\mathcal{C}$ is weakly scale-invariant with base q, then the ratio

$$\frac{H(X_{\mathcal{C},q\lambda})}{(q\lambda)^2 \cdot I_{\mathcal{C}}(q\lambda)} = \frac{H(X_{\mathcal{C},\lambda})}{\lambda^2 \cdot I_{\mathcal{C}}(\lambda)}$$

becomes multiplicatively periodic with period q. See Fig. 5.5 for illustrations of the periodicity of the entropy (H) and normalized information ($\lambda^2 I$) of the base-q LogLog sketch.

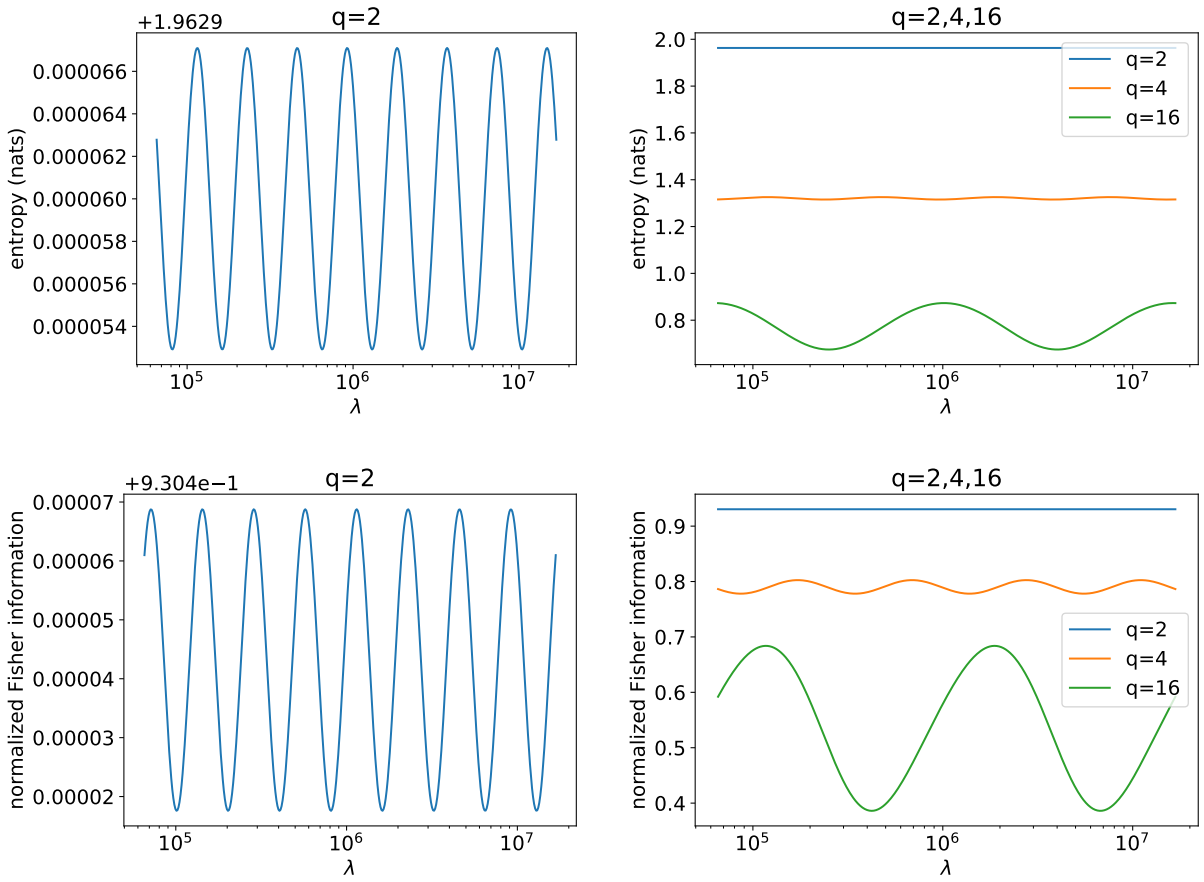


Figure 5.5: Entropy and normalized Fisher information number for q-LogLog sketches for $\lambda \in [2^{16}, 2^{24}]$. See Section 5.3.2 for the precise definitions. Left: At a sufficiently small scale, the oscillations in entropy (top) and normalized information (bottom) of 2-LogLog become visible. Right: At higher values of $q \in \{2, 4, 16\}$, the oscillations in entropy (top) and normalized information (bottom) of q-LogLog are clearly visible. From the bottom left plot one can see that the standard error coefficient lower bound $\frac{1}{\sqrt{0.93}} = 1.037$ is very close to the standard error coefficient $\beta_{128} = 1.046$ obtained by Flajolet et al. [82]. This highlights how little room for improvement there is in the HyperLogLog analysis.

Smoothing via Random Offsetting The LogLog sketch has an oscillating asymptotic relative variance but since its magnitude is very small (less than 10^{-4}), it is often ignored. However, when we consider base-q generalizations of LogLog, e.g., $q = 16$, the oscillation becomes too large to ignore; see Figs. 5.5 and 5.7. Here we give a simple mechanism to *smooth* these functions.

Rather than combine m i.i.d. copies of the basic sketch, we will combine m *randomly offsetted* copies of the sketch. Specifically, the algorithm is hard-coded with a random vector $(R_0, \ldots, R_{m-1}) \in [0,1)^m$ and for all $i \in [m]$, each element processed by the algorithm will be withheld from the ith sketch with probability $1-q^{-R_i}$. Thus, after seeing λ distinct elements, the ith sketch will have seen λq^{-R_i} distinct elements in expectation. As m goes to infinity, the memory size (entropy) and the relative variance tend to their average values.[9] Figure 5.7 illustrates the effectiveness of this smoothing operation for reasonably small values of $q = 16$ and $m = 128$. See Fig. 5.6 for the ruler analogy.

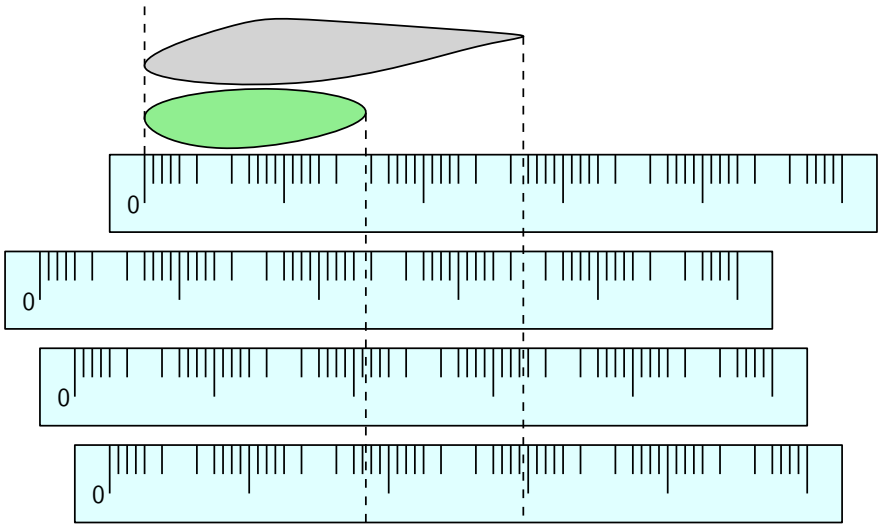


Figure 5.6: Analogy between cardinality sketches and rulers: Smoothing. Even though only periodic rulers are available, one can measure an object multiple times with *random offsets*, and the overall error will be completely insensitive to the subjects' length. This trick is an analogy of turning weakly scale-invariant sketches such as PCSA to strictly scale-invariant by random-offsetting.

Now let the covering structure $\mathcal{C}$ be a weakly scale-invariant sketch with base q and IDF $\Xi_{\mathcal{C}}$. Let $(R_1, Y_1) \in [0,1) \times \mathcal{C}$ be a pair where R_1 is uniformly random in $[0,1)$, and Y_1 is the

[9] As m goes to infinity, using the set of uniform offsets $(0, \ldots, \frac{m-1}{m})$ will also work.

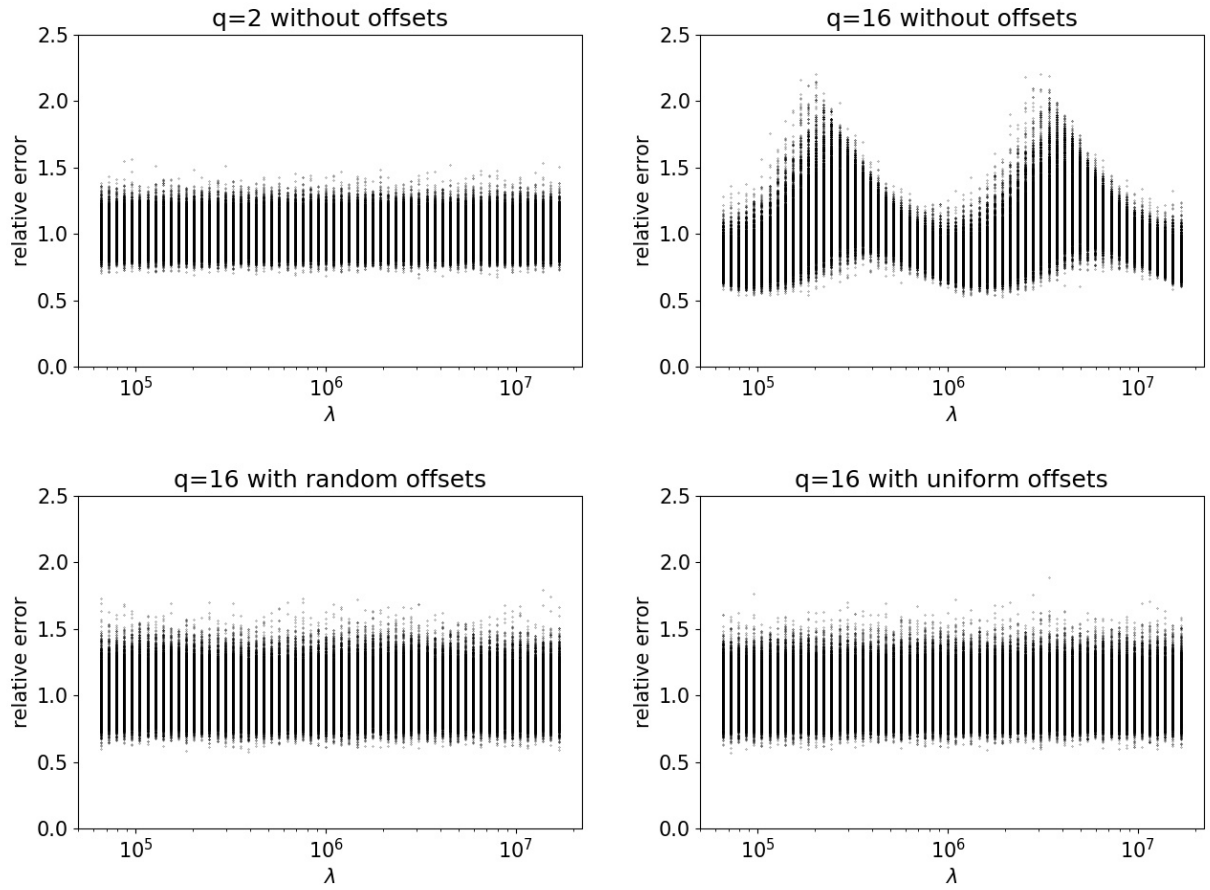


Figure 5.7: The empirical relative error $(\hat{\lambda}/\lambda)$ distribution (for $\lambda \in [2^{16}, 2^{24}]$) of q-LogLog. Four cases: (1) $q = 2$ without offsets; (2) $q = 16$ without offsets; (3) $q = 16$ with random offsets; (4) $q = 16$ with uniform offsets. All use $m = 128$ and the number of experiments is 5000 for each cardinality. We use a HyperLogLog-type estimator $\hat{\lambda}(S) = \alpha_{q,m,r} \cdot m(\sum_{k\in[m]} q^{-S(k)-r_k})^{-1}$ (without stochastic averaging), where $S(k)$ is the final state of the kth sketch and r_k is the offset for the kth sketch. The sketches without offsets have $r_k = 0$ for all $k \in [m]$. The sketches with random offsets have $r = (r_k)_{k\in[m]}$ uniformly distributed in $[0, 1)^m$. Sketches with uniform offsets use the offset vector $r = (0, 1/m, \ldots, (m-1)/m)$. The constant $\alpha_{q,m,r}$ is determined experimentally for each case.

state of the sketch after seeing λq^{-R_1} distinct insertions.[10] Then

$$\Pr(Y_1 = y_1 \mid R_1 = r_1, \lambda) = \xi_{\mathcal{C},\lambda q^{-r_1}}(y_1).$$

Thus the joint density function is

$$f_\lambda(r_1, y_1) = \xi_{\mathcal{C},\lambda q^{-r_1}}(y_1).$$

Lemma 5.2. *Fix the unknown cardinality (parameter) λ. The Fisher information of λ with respect to (R_1, Y_1) is equal to*

$$\frac{1}{\lambda^2} \int_0^1 q^{2r} I_{\mathcal{C}}(q^r)\, dr.$$

Proof. We can calculate the Fisher information of λ with respect to (R_1, Y_1) as follows.

$$I_{(R_1,Y_1)}(\lambda) = \mathbb{E}\left(\frac{\frac{d}{d\lambda} f_\lambda(R_1, Y_1)}{f_\lambda(R_1, Y_1)}\right)^2 = \int_0^1 \sum_{y_1 \in \mathcal{C}} \left(\frac{\frac{d}{d\lambda}\xi_{\mathcal{C},\lambda q^{-r_1}}(y_1)}{\xi_{\mathcal{C},\lambda q^{-r_1}}(y_1)}\right)^2 \xi_{\mathcal{C},\lambda q^{-r_1}}(y_1) dr_1. \tag{5.1}$$

Let $r = r_1$ and $w = \lambda q^{-r}$. Then we have

$$\frac{d}{d\lambda}\xi_{\mathcal{C},\lambda q^{-r}}(y_1) = \frac{dw}{d\lambda}\frac{d}{dw}\xi_{\mathcal{C},w}(y_1) = q^{-r}\frac{d}{dw}\xi_{\mathcal{C},w}(y_1).$$

Continuing, (5.1) is equal to

$$\begin{aligned} &= \int_0^1 q^{-2r} \sum_{y_1 \in \mathcal{C}} \left(\frac{\frac{d}{dw}\xi_{\mathcal{C},w}(y_1)}{\xi_{\mathcal{C},w}(y_1)}\right)^2 \xi_{\mathcal{C},w}(y_1)\, dr \\ &= \int_0^1 q^{-2r} I_{\mathcal{C}}(w)\, dr \;=\; \int_0^1 q^{-2r} I_{\mathcal{C}}(\lambda q^{-r})\, dr. \end{aligned} \tag{5.2}$$

Let $g(x) = q^{2x} I_{\mathcal{C}}(q^x)$. By the weak scale-invariance of $\mathcal{C}$, we have $g(x+1) = g(x)$ for any $x \in \mathbb{R}$. Applying the definition of g, (5.2) is equal to

$$= \frac{1}{\lambda^2}\int_0^1 g(-r + \log_q \lambda)\, dr = \frac{1}{\lambda^2}\int_0^1 g(r)\, dr = \frac{1}{\lambda^2}\int_0^1 q^{2r} I_{\mathcal{C}}(q^r)\, dr.$$

□

[10] Technically, with the offset R_1, the sketch should see $B(\lambda, q^{-R_1})$ distinct insertions, where $B(\lambda, q^{-R_1})$ is a binomial random variable with λ trials and success probability q^{-R_1}. We approximate $B(\lambda, q^{-R_1})$ by its mean λq^{-R_1} since we are only considering the *asymptotic* relative behavior as λ goes to infinity.

Lemma 5.3. *Fix the unknown cardinality (parameter) λ. The conditional entropy $H(Y_1 \mid R_1)$ is equal to*

$$\int_0^1 H(X_{\mathcal{C},q^r})\,dr.$$

Proof. By the definition of the conditional entropy, we have

$$H(Y_1 \mid R_1) = \int_0^1 H(Y_1 \mid r)\,dr = \int_0^1 H(X_{\mathcal{C},\lambda q^{-r}})\,dr.$$

Let $g(x) = H(X_{\mathcal{C},q^x})$. By the weak scale-invariance of $\mathcal{C}$, we know that $g(x) = g(x+1)$ for any $x \in \mathbb{R}$. Thus, we conclude that

$$\int_0^1 H(X_{\mathcal{C},\lambda q^{-r}})\,dr = \int_0^1 g(-r + \log_q \lambda)\,dr = \int_0^1 g(r)\,dr = \int_0^1 H(X_{\mathcal{C},q^r})\,dr.$$

□

In conclusion, with random offsetting we can transform any weakly scale-invariant sketch $\mathcal{C}$ so that the product of the memory and the relative variance is

$$\frac{\int_0^1 H(X_{\mathcal{C},q^r})\,dr}{\lambda^2 \cdot \frac{1}{\lambda^2}\int_0^1 q^{2r} I_{\mathcal{C}}(q^r)\,dr} = \frac{\int_0^1 H(X_{\mathcal{C},q^r})\,dr}{\int_0^1 q^{2r} I_{\mathcal{C}}(q^r)\,dr},$$

and hence independent of the cardinality λ.

The Fish Number of a Sketch Finally, we are ready to define the Fish number. Let $\mathcal{C}_q$ be a weakly scale-invariant sketch with base q. The *Fisher-Shannon* (Fish) *number* of $\mathcal{C}_q$ captures the maximum performance we can potentially extract out of $\mathcal{C}_q$, after applying random offsets (Section 5.2.5), optimal estimators, and compression to the entropy bound, as $m \to \infty$. In particular, any sketch composed of independent copies of $\mathcal{C}_q$ with standard error $\frac{1}{\sqrt{b}}$ must use at least $\mathsf{Fish}(\mathcal{C}_q) \cdot b$ bits. Thus, smaller Fish-numbers are better.

Definition 5.8. Let $\mathcal{C}_q$ be a weakly scale-invariant sketch with base q. The *Fish number* of $\mathcal{C}_q$ is defined to be $\mathsf{Fish}(\mathcal{C}_q) \stackrel{\text{def}}{=} \mathcal{H}(\mathcal{C}_q)/\mathcal{I}(\mathcal{C}_q)$, where

$$\mathcal{H}(\mathcal{C}_q) \stackrel{\text{def}}{=} \int_0^1 H(X_{\mathcal{C}_q,q^r})\,dr \quad \text{and} \quad \mathcal{I}(\mathcal{C}_q) \stackrel{\text{def}}{=} \int_0^1 q^{2r} I_{\mathcal{C}_q}(q^r)\,dr.$$

5.3 Fish Numbers of PCSA and LogLog

In this section, we will find the Fish numbers of generalizations of PCSA [80] and (Hyper)LogLog [73, 82]. The results are expressed in terms of two important constants, H_0 and I_0. Some of the proofs are deferred to Section 5.3.3

Definition 5.9. Let $h(x) = -x \log x - (1-x)\log(1-x)$ and $g(x) = \frac{x^2}{e^x - 1}$. We define

$$H_0 \stackrel{\text{def}}{=} \frac{1}{\log 2} \cdot \int_{-\infty}^{\infty} h\left(e^{-e^w}\right) dw \quad \text{and} \quad I_0 \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} g\left(e^w\right) dw.$$

Lemma 5.4 derives simplified expressions for H_0 and I_0.

Lemma 5.4.

$$H_0 = \frac{1}{\log 2} + \sum_{k=1}^{\infty} \frac{1}{k} \log_2\left(1 + 1/k\right), \qquad \textit{and} \qquad I_0 = \zeta(2) = \frac{\pi^2}{6},$$

where $\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}$ *is the Riemann zeta function.*

5.3.1 The Fish Numbers of q-PCSA Sketches

In the discrete counting process, a natural base-q generalization of PCSA (q-PCSA) can be represented as a bit vector $\mathbf{b} = (b_k)_{k \in \mathbb{N}}$ where $\Pr(b_i = 0) = (1 - q^{-i})^{\lambda} \approx e^{-\lambda/q_i}$ after processing a multiset with cardinality λ. The easiest way to effect this, conceptually, is to interpret $h(a)$ as a sequence $\mathbf{x} \in \{0,1\}^{\infty}$ of bits,[11] where $\Pr(x_i = 1) = q^{-i}$, then update $\mathbf{b} \leftarrow \mathbf{b} \vee \mathbf{x}$, where $\vee$ is bit-wise OR.[12] After poissonization, we have

1. The probability that the ith bit is zero is *exactly* $\Pr(b_i = 0) = e^{-\lambda/q^i}$.
2. All bits of the sketch are independent.

Since we are concerned with the *asymptotic* behavior of the sketch when $\lambda \to \infty$ we also assume that the domain of the sketch $\mathbf{b}$ is extended from $\mathbb{N}$ to $\mathbb{Z}$, e.g., together with poissonization, we have $\Pr(b_{-5} = 0) = e^{-q^5 \lambda}$. The resulting abstract sketch is weakly scale-invariant with base q, according to Definition 5.7.

[11] If we are interested in cardinalities $\ll n$, we would truncate the hash at $\log n$ bits.

[12] We can simplify this scheme with the same two levels of stochastic averaging used by Flajolet and Martin [80], namely choosing $\mathbf{x}$ to have bounded Hamming weight (weight 1 in their case), and splitting the stream into m substreams if we are maintaining m such $\mathbf{b}$-vectors.

Definition 5.10 (IDF of q-PCSA Sketches)**.** For any base $q > 1$, the induced distribution for cardinality λ is

$$\xi_{q\text{-PCSA},\lambda}(\mathbf{b}) = \prod_{k=-\infty}^{\infty} e^{-\frac{\lambda(1-b_k)}{q^k}}(1-e^{-\frac{\lambda}{q^k}})^{b_k}.$$

Theorem 5.4. *For any $q > 1$, q-PCSA is weakly scale-invariant with base q. Furthermore, we have*

$$\mathcal{H}(q\text{-PCSA}) = \frac{H_0}{\log q} \quad \textit{and} \quad \mathcal{I}(q\text{-PCSA}) = \frac{I_0}{\log q},$$

and hence

$$\mathit{Fish}(q\text{-PCSA}) = \frac{H_0}{I_0} \approx 1.98016.$$

Proof. Let λ be the unknown cardinality (the parameter) and $X_{q\text{-PCSA},\lambda} = (Z_{\lambda,k})_{k\in\mathbb{Z}} \in \{0,1\}^{\mathbb{Z}}$ be the final state of the bit-vector. For each k, $Z_{\lambda,k}$ is a Bernoulli random variable with probability mass function $f_{\lambda,k}(b_k) = e^{-\frac{\lambda(1-b_k)}{q^k}}(1-e^{-\frac{\lambda}{q^k}})^{b_k}$. Let $h(x) = -x\log x - (1-x)\log(1-x)$. Since the $\{Z_{\lambda,k}\}$ are independent, we have

$$\begin{aligned} H(X_{q\text{-PCSA},\lambda}) &= \sum_{k=-\infty}^{\infty} H(Z_{\lambda,k}) = \frac{1}{\log 2}\sum_{k=-\infty}^{\infty} h\left(e^{-\frac{\lambda}{q^k}}\right) \\ &= \frac{1}{\log 2}\sum_{k=-\infty}^{\infty} h\left(e^{-\frac{q\lambda}{q^k}}\right) = H(X_{q\text{-PCSA},q\lambda}), \end{aligned}$$

meaning q-PCSA satisfies the first criterion of weak scale-invariance. We now turn to the second criterion regarding Fisher information.

Let $g(x) = \frac{x^2e^{-2x}}{e^{-x}} + \frac{x^2e^{-2x}}{1-e^{-x}} = \frac{x^2}{e^x-1}$. Observe that the Fisher information of λ with respect to the observation $Z_{\lambda,k}$ (i.e., $I_{Z_{\lambda,k}}(\lambda)$) is equal to

$$\begin{aligned} \mathbb{E}\left(\frac{\frac{d}{d\lambda}f_{\lambda,k}(Z_{\lambda,k})}{f_{\lambda,k}(Z_{\lambda,k})}\right)^2 &= \frac{\left(\frac{d}{d\lambda}(1-e^{-\frac{\lambda}{q^k}})\right)^2}{1-e^{-\frac{\lambda}{q^k}}} + \frac{\left(\frac{d}{d\lambda}e^{-\frac{\lambda}{q^k}}\right)^2}{e^{-\frac{\lambda}{q^k}}} \\ &= \frac{\left(\frac{1}{q^k}e^{-\frac{\lambda}{q^k}}\right)^2}{1-e^{-\frac{\lambda}{q^k}}} + \frac{\left(\frac{1}{q^k}e^{-\frac{\lambda}{q^k}}\right)^2}{e^{-\frac{\lambda}{q^k}}} = \frac{1}{\lambda^2}g\left(\frac{\lambda}{q^k}\right). \end{aligned}$$

Since the $\{Z_{\lambda,k}\}$ are independent, we have

$$I_{q\text{-PCSA}}(\lambda) = \sum_{k=-\infty}^{\infty} \frac{1}{\lambda^2} g\left(\frac{\lambda}{q^k}\right) = q^2 \sum_{k=-\infty}^{\infty} \frac{1}{q^2\lambda^2} g\left(\frac{q\lambda}{q^k}\right) = q^2 I_{q\text{-PCSA}}(q\lambda).$$

We conclude that q-PCSA is weakly scale-invariant with base q. Now we compute the $\mathcal{H}(q\text{-PCSA})$ and $\mathcal{I}(q\text{-PCSA})$.

$$\begin{aligned}
\mathcal{H}(q\text{-PCSA}) &= \int_0^1 H(X_{q\text{-PCSA},q^r})\, dr = \frac{1}{\log 2} \int_0^1 \sum_{k=-\infty}^{\infty} h\left(e^{-\frac{q^r}{q^k}}\right) dr \\
&= \frac{1}{\log 2} \sum_{k=-\infty}^{\infty} \int_0^1 h\left(e^{-e^{(r-k)\log q}}\right) dr = \frac{1}{\log 2} \sum_{k=-\infty}^{\infty} \int_{-k}^{1-k} h\left(e^{-e^{r\log q}}\right) dr \\
&= \frac{1}{\log 2} \int_{-\infty}^{\infty} h\left(e^{-e^{r\log q}}\right) dr = \frac{1}{\log 2} \cdot \frac{1}{\log q} \int_{-\infty}^{\infty} h\left(e^{-e^w}\right) dw \;=\; \frac{H_0}{\log q}.
\end{aligned}$$

The final line uses the change of variable $w = r \log q$. We use similar techniques to calculate the normalized information $\mathcal{I}(q\text{-PCSA})$.

$$\begin{aligned}
\mathcal{I}(q\text{-PCSA}) &= \int_0^1 q^{2r} I_{q\text{-PCSA}}(q^r)\, dr = \int_0^1 q^{2r} \sum_{k=-\infty}^{\infty} \frac{1}{q^{2r}} g(q^{r-k})\, dr = \sum_{k=-\infty}^{\infty} \int_0^1 g(q^{r-k})\, dr \\
&= \int_{-\infty}^{\infty} g(q^r)\, dr = \frac{1}{\log q} \int_{-\infty}^{\infty} g(e^w) dw = \frac{I_0}{\log q}.
\end{aligned}$$

□

5.3.2 The Fish Numbers of q-LogLog Sketches

In a discrete counting process, the natural base-q generalization of the (Hyper)LogLog sketch (q-LogLog) works as follows. Let $Y = \min_{a\in\mathcal{C}} h(a) \in [0,1]$ be the minimum hash value seen. The q-LogLog sketch stores the integer $S = \lfloor -\log_q Y \rfloor$, so when the cardinality is λ,

$$\Pr(S = k) = \Pr(q^{-k} \le Y < q^{-k+1}) = (1-q^{-k})^\lambda - (1-q^{-k+1})^\lambda \approx e^{-\lambda/q^k} - e^{-\lambda/q^{k-1}}.$$

Once again the state space of this sketch is $\mathbb{Z}^+$ but to show weak scale-invariance it is useful to extend it to $\mathbb{Z}$. Together with poissonization, we have the following.

1. $\Pr(S = k)$ is *precisely* $e^{-\lambda/q^k} - e^{-\lambda/q^{k-1}}$.

2. The state space is $\mathbb{Z}$, e.g., together with (1) we have $\Pr(S = -1) = e^{-q\lambda} - e^{-q^2\lambda}$.

Definition 5.11 (IDF of q-LogLog sketches)**.** For any base $q > 1$, the state space of q-LogLog is $\mathcal{C}_{\mathsf{LogLog}} = \mathbb{Z}$ and the induced distribution for cardinality λ is

$$\xi_{q\text{-LogLog},\lambda}(k) = e^{-\lambda/q^k} - e^{-\lambda/q^{k-1}}.$$

In Lemma 5.6 we express the Fish number of q-LogLog in terms of two quantities $\phi(q)$ and $\rho(q)$, defined as follows.

Definition 5.12.

$$\phi(q) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} -(e^{-e^r} - e^{-e^r q}) \log_2(e^{-e^r} - e^{-e^r q})\, dr.$$
$$\rho(q) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} \frac{(-e^r e^{-e^r} + e^r q e^{-e^r q})^2}{e^{-e^r} - e^{-e^r q}}\, dr.$$

Lemma 5.5 gives simplified expressions for $\phi(q)$ and $\rho(q)$. See Section 5.3.3 for proof.

Lemma 5.5. *Let $\zeta(s,t) = \sum_{k\geq 0}(k+t)^{-s}$ be the Hurwitz zeta function. Then ϕ and ρ can be expressed as:*

$$\phi(q) = \frac{1 - 1/q}{\log 2} + \sum_{k=1}^{\infty} \frac{1}{k} \log_2 \left(\frac{k + \frac{1}{q-1} + 1}{k + \frac{1}{q-1}} \right).$$
$$\rho(q) = \zeta\left(2, \frac{q}{q-1}\right) = \sum_{k=0}^{\infty} \frac{1}{(k + \frac{q}{q-1})^2}.$$

Refer to Section 5.3.3 for proof of Lemma 5.6.

Lemma 5.6. *For any $q > 1$, q-LogLog is weakly scale-invariant with base q. Furthermore, we have*

$$\mathcal{H}(q\text{-LogLog}) = \frac{\phi(q)}{\log q} \quad \text{and} \quad \mathcal{I}(q\text{-LogLog}) = \frac{\rho(q)}{\log q}.$$

Theorem 5.5. *For any $q > 1$, the Fish number of q-LogLog is*

$$\mathsf{Fish}(q\text{-LogLog}) > \frac{H_0}{I_0}.$$

Furthermore, we have

$$\lim_{q\to\infty} \mathsf{Fish}(q\text{-LogLog}) = \frac{H_0}{I_0}.$$

Proof. We prove the second statement first. By Lemma 5.6, we have

$$\begin{aligned}
\lim_{q\to\infty} \mathsf{Fish}(q\text{-}\mathsf{LogLog}) &= \lim_{q\to\infty} \frac{\mathcal{H}(q\text{-}\mathsf{LogLog})}{\mathcal{I}(q\text{-}\mathsf{LogLog})} \\
&= \lim_{q\to\infty} \frac{\frac{1-1/q}{\log 2} + \sum_{k=1}^{\infty} \frac{1}{k} \log_2 \left(\frac{k+\frac{1}{q-1}+1}{k+\frac{1}{q-1}} \right)}{\sum_{k=1}^{\infty} \frac{1}{(k+\frac{1}{q-1})^2}} \\
&= \frac{\frac{1}{\log 2} + \sum_{k=1}^{\infty} \frac{1}{k} \log_2 \left(\frac{k+1}{k} \right)}{\sum_{k=1}^{\infty} \frac{1}{k^2}} = \frac{H_0}{I_0}.
\end{aligned}$$

The first statement follows from Lemmas 5.7 and 5.8. Refer to Section 5.3.3 for proof. □

Lemma 5.7. *Fish*(q-*LogLog*) *is strictly decreasing for* $q \geq 1.4$.

Lemma 5.8. *Fish*(q-*LogLog*) > *Fish*(2-*LogLog*) *for* $q \in (1, 1.4]$.

5.3.3 Deferred Proofs

Lemma 5.9 is applied in the proofs of Lemmas 5.4 and 5.5.

Lemma 5.9. *For any* $b > a > 0$ *we have the following identity.*

$$\int_0^{\infty} \frac{e^{-ax} - e^{-bx}}{x} dx = \log b - \log a.$$

Proof.

$$\begin{aligned}
\log b - \log a &= \int_a^b \frac{1}{t} dt \\
&= -\int_a^b \frac{e^{-xt}}{t} dt \Big|_0^{\infty} \\
&= -\int_{ax}^{bx} \frac{e^{-r}}{r} dr \Big|_0^{\infty} && \{\text{Change of variable: } r = tx\} \\
&= -\left(\int_{ax}^{\infty} \frac{e^{-r}}{r} dr - \int_{bx}^{\infty} \frac{e^{-r}}{r} dr \right) \Big|_0^{\infty}
\end{aligned}$$

Note that

$$\frac{d}{dx}\int_{ax}^{\infty}\frac{e^{-r}}{r}dr = -\frac{e^{-ax}}{ax}a = -\frac{e^{-ax}}{x}.$$

Thus we have

$$\int_0^{\infty}\frac{e^{-ax}-e^{-bx}}{x}dx = \left(-\int_{ax}^{\infty}\frac{e^{-r}}{r}dr + \int_{bx}^{\infty}\frac{e^{-r}}{r}dr\right)\Bigg|_0^{\infty} = \log b - \log a.$$

□

Proof of Lemma 5.4. Let $u = e^w$, then $\frac{du}{dw} = u$.

$$\begin{aligned}
\log 2 \cdot H_0 &= \int_0^{\infty}\frac{e^{-u}u - (1-e^{-u})\log(1-e^{-u})}{u}du \\
&= 1 + \int_0^{\infty}\frac{-(1-e^{-u})\log(1-e^{-u})}{u}du \\
&= 1 + \int_0^{\infty}\frac{(1-e^{-u})\sum_{k=1}^{\infty}\frac{e^{-ku}}{k}}{u}du && \{\text{Taylor exp.}\} \\
&= 1 + \sum_{k=1}^{\infty}\frac{1}{k}\int_0^{\infty}\frac{(e^{-ku}-e^{-(k+1)u})}{u}du.
\end{aligned}$$

Applying Lemma 5.9, we have

$$H_0 = \frac{1}{\log 2} + \sum_{k=1}^{\infty}\frac{1}{k}\log_2\left(\frac{k+1}{k}\right).$$

We now prove that $I_0 = \pi^2/6$. Letting $u = e^r$, we have $\frac{du}{dr} = u$ and can write I_0 as

$$I_0 = \int_0^{\infty}\frac{(e^r)^2}{e^{e^r}-1}dr = \int_0^{\infty}\frac{u^2}{u(e^u-1)}du = \int_0^{\infty}\frac{u}{e^u-1}du,$$

which is exactly the integral representation of the Riemann zeta function evaluated at $s = 2$. We conclude that

$$I_0 = \zeta(2) = \frac{\pi^2}{6}.$$

□

Proof of Lemma 5.5. Let $w = e^r$ and $\frac{dw}{dr} = w$. We have

$$\log 2 \cdot \phi(q)$$

$$= \int_0^\infty \frac{-(e^{-w} - e^{-qw}) \log(e^{-w} - e^{-qw})}{w} dw$$
$$= \int_0^\infty \frac{-(e^{-w} - e^{-qw}) \log(1 - e^{-(q-1)w})}{w} dw + \int_0^\infty (e^{-w} - e^{-qw}) dw.$$

Note that $\int_0^\infty (e^{-w} - e^{-qw}) dw = 1 - \frac{1}{q}$. Continuing with a Taylor expansion of the logarithm, we have

$$= \int_0^\infty \frac{(e^{-w} - e^{-qw}) \sum_{k=1}^\infty \frac{e^{-k(q-1)w}}{k}}{w} dw + 1 - \frac{1}{q}$$
$$= \sum_{k=1}^\infty \frac{1}{k} \int_0^\infty \frac{e^{-(k(q-1)+1)w} - e^{-(k(q-1)+q)w}}{w} dw + 1 - \frac{1}{q}.$$

Applying Lemma 5.9 to the integral, this is equal to

$$= 1 - \frac{1}{q} + \sum_{k=1}^\infty \frac{1}{k} \log\left(\frac{kq - k + q}{kq - k + 1}\right)$$
$$= 1 - \frac{1}{q} + \sum_{k=1}^\infty \frac{1}{k} \log\left(\frac{k + \frac{1}{q-1} + 1}{k + \frac{1}{q-1}}\right).$$

Hence $\phi(q)$ is

$$\phi(q) = \frac{1 - 1/q}{\log 2} + \sum_{k=1}^\infty \frac{1}{k} \log_2\left(\frac{k + \frac{1}{q-1} + 1}{k + \frac{1}{q-1}}\right).$$

Set $w = e^r$, then $\frac{dw}{dr} = w$. We have

$$\begin{aligned}
\rho(q) &= \int_0^\infty \frac{(-we^{-w} + qwe^{-qw})^2}{w(e^{-w} - e^{-qw})} dw \\
&= \int_0^\infty \frac{we^{-w}(1 - qe^{-(q-1)w})^2}{1 - e^{-(q-1)w}} dw \\
&= \int_0^\infty \frac{we^{-w}q^2(1 - e^{-(q-1)w} + \frac{1}{q} - 1)^2}{1 - e^{-(q-1)w}} dw \\
&= \int_0^\infty we^{-w}q^2 \left(1 - e^{-(q-1)w} + 2\left(\frac{1}{q} - 1\right) + \frac{(\frac{1}{q} - 1)^2}{1 - e^{-(q-1)w}}\right) dw \\
&= \int_0^\infty we^{-w} \left(-q^2 e^{-(q-1)w} + (-q^2 + 2q) + \frac{(q-1)^2}{1 - e^{-(q-1)w}}\right) dw \\
&= -q^2 \int_0^\infty we^{-qw} dw + (-q^2 + 2q) \int_0^\infty we^{-w} dw + (q-1)^2 \int_0^\infty \frac{we^{-w}}{1 - e^{-(q-1)w}} dw.
\end{aligned}$$

We calculate the three integrals separately. First we have

$$\int_0^\infty we^{-qw}dw = e^{-qw}\left(-\frac{w}{q} - \frac{1}{q^2}\right)\Big|_0^\infty = \frac{1}{q^2}.$$

For the second we have

$$\int_0^\infty we^{-w}dw = e^{-w}(-w-1)\Big|_0^\infty = 1.$$

For the last, let $u = (q-1)w$. Then we have

$$\int_0^\infty \frac{we^{-w}}{1-e^{-(q-1)w}}dw = \frac{1}{(q-1)^2}\int_0^\infty \frac{ue^{-\frac{u}{q-1}}}{1-e^{-u}}du,$$

where $\int_0^\infty \frac{ue^{-\frac{u}{q-1}}}{1-e^{-u}}du$ is just the integral representation of the Hurwitz zeta function $\zeta(2, \frac{1}{q-1})$. This can be written as a sum series as follows.

$$\int_0^\infty \frac{ue^{-\frac{u}{q-1}}}{1-e^{-u}}du = \sum_{k=0}^\infty \frac{1}{(k+\frac{1}{q-1})^2}.$$

Combining the three integrals, we conclude that

$$\rho(q) = -1 - q^2 + 2q + \sum_{k=0}^\infty \frac{1}{(k+\frac{1}{q-1})^2} = \sum_{k=1}^\infty \frac{1}{(k+\frac{1}{q-1})^2}.$$

□

Proof of Lemma 5.6. For a fixed λ, by the definition of Shannon entropy, we have

$$\begin{aligned} H(X_{q\text{-LogLog},\lambda}) &= \sum_{k=-\infty}^\infty -(e^{-\frac{\lambda}{q^k}} - e^{-\frac{\lambda}{q^{k-1}}})\log_2(e^{-\frac{\lambda}{q^k}} - e^{-\frac{\lambda}{q^{k-1}}}) \\ &= \sum_{k=-\infty}^\infty -(e^{-\frac{q\lambda}{q^k}} - e^{-\frac{q\lambda}{q^{k-1}}})\log_2(e^{-\frac{q\lambda}{q^k}} - e^{-\frac{q\lambda}{q^{k-1}}}) = H(X_{q\text{-LogLog},q\lambda}). \end{aligned}$$

Also, we have

$$I_{q\text{-LogLog}}(\lambda) = \sum_{k=-\infty}^\infty \frac{\left(-\frac{1}{q^k}e^{-\frac{\lambda}{q^k}} + \frac{1}{q^{k-1}}e^{-\frac{\lambda}{q^{k-1}}}\right)^2}{e^{-\frac{\lambda}{q^k}} - e^{-\frac{\lambda}{q^{k-1}}}}$$

$$= q^2 \sum_{k=-\infty}^{\infty} \frac{\left(-\frac{1}{q^k} e^{-\frac{q\lambda}{q^k}} + \frac{1}{q^{k-1}} e^{-\frac{q\lambda}{q^{k-1}}}\right)^2}{e^{-\frac{q\lambda}{q^k}} - e^{-\frac{q\lambda}{q^{k-1}}}} = q^2 \cdot I_{q\text{-}\mathsf{LogLog}}(q\lambda).$$

We conclude that q-LogLog is weakly scale-invariant with base q. We now turn to calculating $\mathcal{H}(q\text{-}\mathsf{LogLog})$ and $\mathcal{I}(q\text{-}\mathsf{LogLog})$. By Definition 5.8,

$$\begin{aligned}
\mathcal{H}(q\text{-}\mathsf{LogLog}) &= \int_0^1 H(X_{q\text{-}\mathsf{LogLog}}, q^r)\, dr \\
&= -\int_0^1 \sum_{k=-\infty}^{\infty} \left(e^{-q^{r-k}} - e^{-q^{r-k+1}}\right) \log_2 \left(e^{-q^{r-k}} - e^{-q^{r-k+1}}\right) dr \\
&= -\int_{-\infty}^{\infty} \left(e^{-q^r} - e^{-q^{r+1}}\right) \log_2 \left(e^{-q^r} - e^{-q^{r+1}}\right) dr = \frac{\phi(q)}{\log q}.
\end{aligned}$$

Again, by Definition 5.8,

$$\begin{aligned}
\mathcal{I}(q\text{-}\mathsf{LogLog}) &= \int_0^1 q^{2r} I_{q\text{-}\mathsf{LogLog}}(q^r) dr \\
&= \int_0^1 \sum_{k=-\infty}^{\infty} \frac{\left(-q^{r-k} e^{-q^{r-k}} + q^{r-k+1} e^{-q^{r-k+1}}\right)^2}{e^{-q^{r-k}} - e^{-q^{r-k+1}}} dr \\
&= \int_{-\infty}^{\infty} \frac{\left(-q^r e^{-q^r} + q^{r+1} e^{-q^{r+1}}\right)^2}{e^{-q^r} - e^{-q^{r+1}}} dr = \frac{\rho(q)}{\log q}.
\end{aligned}$$

□

Proof of Lemma 5.7 . Note that by Lemma 5.6,

$$\mathsf{Fish}(q\text{-}\mathsf{LogLog}) = \frac{\mathcal{H}(q\text{-}\mathsf{LogLog})}{\mathcal{I}(q\text{-}\mathsf{LogLog})} = \frac{\phi(q)}{\rho(q)}.$$

Then we have

$$\begin{aligned}
\log 2 \cdot \phi'(q) &= \frac{1}{q^2} + \sum_{k=1}^{\infty} \frac{1}{k} \left(\frac{1}{k + \frac{1}{q-1} + 1} - \frac{1}{k + \frac{1}{q-1}}\right) \frac{-1}{(q-1)^2} \\
&= \frac{1}{q^2} + \frac{1}{(q-1)^2} \sum_{k=1}^{\infty} \frac{1}{k(k + \frac{1}{q-1})(k + \frac{q}{q-1})} \\
&= \frac{1}{q(q-1)} \left(\frac{q-1}{q} + \sum_{k=1}^{\infty} \frac{1}{k(k + \frac{1}{q-1})(\frac{q-1}{q} k + 1)}\right)
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{q(q-1)}\left(\sum_{k=1}^{\infty}\left(\frac{1}{k+\frac{1}{q-1}}-\frac{1}{k+\frac{1}{q-1}+1}\right)+\sum_{k=1}^{\infty}\frac{1}{k(k+\frac{1}{q-1})(\frac{q-1}{q}k+1)}\right)\\
&= \frac{1}{q(q-1)}\left(\sum_{k=1}^{\infty}\frac{1}{(k+\frac{1}{q-1})(k+\frac{q}{q-1})}+\sum_{k=1}^{\infty}\frac{1}{k(k+\frac{1}{q-1})(\frac{q-1}{q}k+1)}\right)\\
&= \frac{1}{q(q-1)}\left(\sum_{k=1}^{\infty}\frac{\frac{q-1}{q}k+1}{k(k+\frac{1}{q-1})(\frac{q-1}{q}k+1)}\right)\\
&= \frac{1}{q(q-1)}\left(\sum_{k=1}^{\infty}\frac{1}{k(k+\frac{1}{q-1})}\right),
\end{aligned}
$$

and

$$
\rho'(q) = \frac{2}{(q-1)^2}\sum_{k=1}^{\infty}\frac{1}{(k+\frac{1}{q-1})^3}.
$$

Define α and β as follows.

$$
\begin{aligned}
\alpha(q) &= \sum_{k=1}^{\infty}\frac{1}{k(k+\frac{1}{q-1})}\\
\beta(q) &= \sum_{k=1}^{\infty}\frac{1}{(k+\frac{1}{q-1})^3},
\end{aligned}
$$

We then have

$$
\begin{aligned}
\log 2\cdot\frac{d}{dq}\frac{\mathcal{H}(q\text{-}\mathsf{LogLog})}{\mathcal{I}(q\text{-}\mathsf{LogLog})} &= \log 2\cdot\frac{d}{dq}\frac{\phi(q)}{\rho(q)}\\
&= \log 2\cdot\frac{\phi'(q)\rho(q)-\rho'(q)\phi(q)}{\rho(q)^2}\\
&= \frac{\frac{q-1}{q}\alpha(q)\rho(q)-2\beta(q)\log 2\cdot\phi(q)}{(q-1)^2\rho(q)^2}.
\end{aligned}
$$

We define $g(a,b) = \frac{b-1}{b}\alpha(b)\rho(b) - 2\beta(a)\log 2\cdot\phi(a)$ and thus $\frac{d}{dq}\frac{\mathcal{H}(q\text{-}\mathsf{LogLog})}{\mathcal{I}(q\text{-}\mathsf{LogLog})} < 0$ if and only if $g(q,q) < 0$.

Note that we have $\phi'(q) > 0$ and $\rho'(q) > 0$ for all $q > 1$ and thus both $\rho(q)$ and $\phi(q)$ are monotonically increasing for $q > 1$. Note that $\frac{1}{q-1}$ is monotonically decreasing for $q > 1$, thus both $\alpha(q)$ and $\beta(q)$ are also monotonically increasing.

Let $a < b$ where $a \in (1,\infty)$ and $b \in (1,\infty]$. Since α, ρ, β and ϕ are all monotonically

increasing, if $g(a,b) < 0$, then for any $q \in [a,b)$,

$$\begin{aligned} g(q,q) &= \frac{q-1}{q}\alpha(q)\rho(q) - 2\beta(q)\log 2 \cdot \phi(q) \\ &< \frac{b-1}{b}\alpha(b)\rho(b) - 2\beta(a)\log 2 \cdot \phi(a) \\ &= g(a,b) \\ &< 0. \end{aligned}$$

Thus, to prove that $\mathsf{Fish}(q\text{-}\mathsf{LogLog})$ is strictly decreasing for $q \geq 1.4$, it is sufficient to find a sequence $1.4 = q_0 < q_1, \ldots < q_n = \infty$ such that for all $k \in [n]$, $g(q_{k-1}, q_k) < 0$. The following table shows the existence of such a sequence and thus completes the proof.

k	q_k	$g(q_{k-1}, q_k)$
0	1.4	
1	1.49	-0.00228439
2	1.62	-0.00186328
3	1.81	-0.00522805
4	2.12	-0.00747658
5	2.72	-0.0038581
6	4.25	-0.00602114
7	6	-0.669626
8	∞	-0.216103

□

Proof of Lemma 5.8. We first calculate that $\log 2 \cdot \mathsf{Fish}(2\text{-}\mathsf{LogLog}) \approx 2.1097$. We then prove that for any $q \in (1, 1.4]$, $\log 2 \cdot \mathsf{Fish}(q\text{-}\mathsf{LogLog}) > 2.11$. We use the inequality $\log x > 1 - \frac{1}{x}$ for $x > 0$, and find that

$$\log\left(\frac{k + \frac{1}{q-1} + 1}{k + \frac{1}{q-1}}\right) > 1 - \frac{k + \frac{1}{q-1}}{k + \frac{1}{q-1} + 1} = \frac{1}{k + \frac{1}{q-1} + 1}.$$

Thus we have

$$\log 2 \cdot \phi(q) > 1 - \frac{1}{q} + \sum_{k=1}^{\infty} \frac{1}{k(k + \frac{q}{q-1})}.$$

We also have

$$\begin{aligned}\rho(q) &< \sum_{k=1}^{\infty}\left(\frac{1}{\frac{1}{q-1}+k-1}-\frac{1}{\frac{1}{q-1}+k}\right)\\ &= \frac{1}{\frac{1}{q-1}+1-1}\\ &= q-1.\end{aligned}$$

Combining the two, we have

$$\begin{aligned}\log 2\cdot \mathsf{Fish}(q\text{-}\mathsf{LogLog}) = \frac{\log 2\cdot\phi(q)}{\rho(q)} &> \frac{1}{q}+\sum_{k=1}^{\infty}\frac{1}{k((q-1)k+q)}\\ &\geq \frac{1}{1.4}+\sum_{k=1}^{\infty}\frac{1}{k(0.4k+1.4)}\\ &\approx 2.11863 > \log 2\cdot\mathsf{Fish}(2\text{-}\mathsf{LogLog}).\end{aligned}$$

We conclude that for any $q\in(1,1.4]$, $\mathsf{Fish}(q\text{-}\mathsf{LogLog}) > \mathsf{Fish}(2\text{-}\mathsf{LogLog})$. □

5.4 A Sharp Lower Bound on Linearizable Sketches

In Section 5.4.1 we introduce the *Dartboard* model, which is essentially the same as Ting's *area-cutting process* [151], with some minor differences.[13] This model is more general then the commutative sketches we characterized before,[14] and is intuitive to work with.

In Section 5.4.2 we define the class of *linearizable* sketches, and in Section 5.4.3 we prove that no Linearizable sketch has Fish-number strictly smaller than H_0/I_0.

5.4.1 The Dartboard Model

Define the *dartboard* to be a unit square $[0,1]^2$, partitioned into a set $\mathcal{M}$ of *cells* of various sizes. A *state space* is a set $\mathcal{S}\subseteq 2^{\mathcal{M}}$. Each state $\sigma\in\mathcal{S}$ partitions the cells into *occupied* cells (σ) and *free* cells ($\mathcal{M}\backslash\sigma$). We process a stream of elements from some multiset. When a *new*

[13] Ting's definition does not fix the state-space *a priori*, and in its full generality allows for non-deterministic sketching algorithms.

[14] Commutative sketches, identified by covering structures, will be insensitive both to duplicates and the insertion order. The dartboard model, on the other hand, is insensitive to duplicates but possibly sensitive to the insertion order.

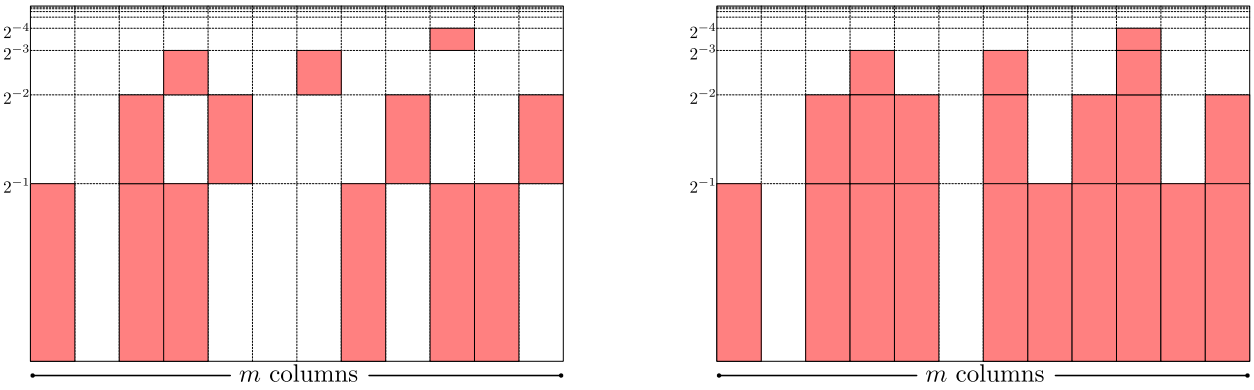


Figure 5.8: The cell partition used by q-PCSA and q-LogLog. Left: A possible state of PCSA. Occupied (red) cells are precisely those containing darts. Right: The corresponding state of LogLog. Occupied (red) cells contain a dart, or lie below a cell in the same column that contains a dart.

element arrives we throw a *dart* at the dartboard and update the state.[15] The probability that a cell $c_i \in \mathcal{M}$ is hit is p_i: the *size* of the cell. A *dartboard sketch* is defined by a transition function satisfying some simple rules.

(R1) Every cell containing a dart is occupied; occupied cells may contain no darts.

(R2) If a dart hits an occupied cell, the state does not change. Rule (R1) implies that if a dart hits a free cell, the state *must* change.

(R3) Once occupied, a cell never becomes free.

Observation 5.1. *Every commutative sketch is a dartboard sketch.*

The state of a commutative sketch is completely characterized by the set of hash-values that cause no state transition. (In particular, the state cannot depend on the order in which elements are processed.) Such a sketch is mapped to the dartboard model by realizing "dart throwing" using the random oracle, say $h : [n] \to [n]$. The dartboard is partitioned into $[n]$ equally-sized cells, where occupied cells are precisely those that cause no change to the state. Rules (R1)–(R3) then follow from the fact that the sketch transition function is commutative and idempotent. However, it is usually possible to partition the dartboard more coarsely than at the level of individual hash-values. For example, base-q PCSA and (Hyper)LogLog (without offsetting) use the same cell partition depicted in Fig. 5.8.

[15]This model can be extended to allow for insertions triggering multiple darts, or a variable number of darts. The dart throwing is effected by the random oracle, so if the same element arrives later, its dart will hit the same cell, and not register a state change, by Rule (R2), below.

After Poissonizing the dartboard, at time (cardinality) λ, Poisson(λ) darts are randomly scattered in the unit square $[0,1]^2$. By properties of the Poisson distribution, the number of darts inside each cell are independent variables. We use the words "time" and "cardinality" interchangeably.

5.4.2 Linearizable Sketches

Informally, a sketch in the dartboard model is called *linearizable* if there is a fixed permutation of cells $(c_0, c_1, \ldots, c_{|\mathcal{M}|-1})$ such that if $\sigma \in \mathcal{S}$ is the state, whether $c_i \in \sigma$ is a function of $\sigma \cap \{c_0, \ldots, c_{i-1}\}$ and whether c_i has been hit by a dart.

More formally, let Z_i be the indicator for whether c_i has been hit by a dart and Y_i be the indicator for whether c_i is occupied. A sketch is linearizable if there is a monotone function $\phi : \{0,1\}^* \to \{0,1\}$ such that

$$Y_i = Z_i \vee \phi(\mathbf{Y}_{i-1}), \quad \text{where } \mathbf{Y}_{i-1} = (Y_0, \ldots, Y_{i-1}).$$

In other words, if $\phi(\mathbf{Y}_{i-1}) = 1$ then cell c_i is "forced" to be occupied, regardless of Z_i. Such a sketch adheres to Rules (R1)–(R3), where (R3) follows from the monotonicity of ϕ. Note that Y_i is a function of $(\mathbf{Y}_{i-1}, Z_i)$, and by induction, also a function of $(Z_0, \ldots, Z_i)$. This implies that state transitions can be computed online (as darts are thrown) and that the transition function is commutative and idempotent.

Observation 5.2. *All linearizable sketches are commutative (and hence mergeable).*

Thus we have

all sketches $\supseteq$ dartboard sketches $\supseteq$ commutative sketches $\supseteq$ linearizable sketches

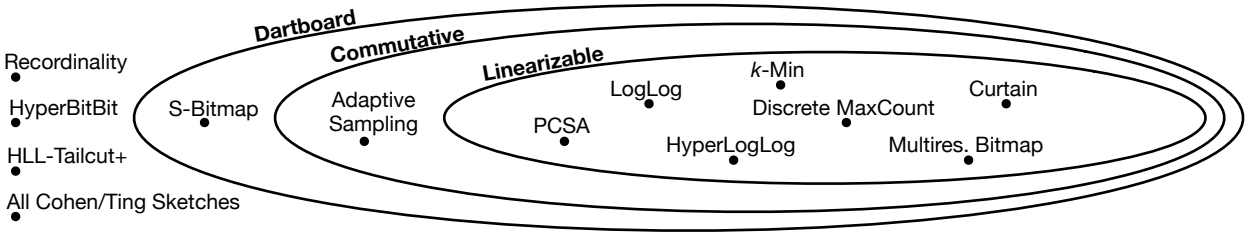


Figure 5.9: A classification of sketching algorithms for cardinality estimation

All of these containments are *strict*, but most popular commutative sketches are linearizable. See Fig. 5.9. For example, PCSA-type sketches [80, 79] are defined by the equality

$Y_i = Z_i$, and hence are linearizable w.r.t. any permutation of cells and constant $\phi(\cdot) = 0$. For LogLog, put the cells in non-decreasing order by size. The function $\phi(\mathbf{Y}_{i-1}) = 1$ iff any cell above c_i in its column is occupied. For the k-Min sketch (aka Bottom-k or Min), the cells are in 1-1 correspondence with hash values, and listed in increasing order of hash value. Then $\phi(\mathbf{Y}_{i-1}) = 1$ iff $\mathbf{Y}_{i-1}$ has Hamming weight at least k, i.e., we only remember the k smallest cells hit by darts. One can also confirm that other sketches are linearizable, such as Multires. Bitmap [79], Discrete MaxCount [151], and Curtain [140].

Strictly speaking AdaptiveSampling [81, 89] is not linearizable. Similar to k-Min, it remembers the smallest k' hash values for varying $k' \leq k$, but k' cannot be determined in a linearizable fashion. One can also invent non-linearizable variations of other sketches. For example, we could add a rule to PCSA that if, among all cells of the same size, at least 70% are occupied, then 100% of them must be occupied.

We are only aware of one sketch that fits in the dartboard model that is not commutative, namely the S-Bitmap [36]; see Section 5.6.

The sketches that fall outside the dartboard model are of two types. The first are non-commutative sketches like Recordinality or those derived by the Cohen/Ting [49, 151] transformation. These consist of a commutative (dartboard) sketch and a cardinality estimate $\hat{\lambda}$, where $\hat{\lambda}$ depends on the *order* in which the darts were thrown; see Section 5.6. The other type are heuristic sketches that violate Rule (R3) (occupied cells stay occupied), like HyperBit-Bit [147] and HLL-Tailcut+ [168]. Rule (R3) is critical if the sketch is to be (asymptotically) unbiased; see Section 5.6.1.

5.4.3 The Lower Bound

When phrased in terms of the dartboard model, our analysis of the Fish-number of PCSA (Section 5.3) took the following approach. We fixed a moment in *time* λ and *aggregated* the Shannon entropy and normalized Fisher information over all *cells* on the dartboard.

Our lower bound on linearizable sketches begins from the opposite point of view. We fix a particular cell $c_i \in \mathcal{M}$ of size p_i and consider how it might contribute to the Shannon entropy and normalized Fisher information at *various times*. The $\dot{H}, \dot{I}$ functions defined in Lemma 5.10 are useful for describing these contributions.

Lemma 5.10. *Let Z be the indicator variable for whether a particular cell of size p has been hit by a dart. At time λ, $\Pr(Z = 0) = e^{-p\lambda}$ and*

$$H(Z) = \dot{H}(p\lambda) \quad \text{and} \quad \lambda^2 \cdot I_Z(\lambda) = \dot{I}(p\lambda),$$

where

$$\dot{H}(t) \stackrel{\text{def}}{=} \frac{1}{\log 2}\left(te^{-t} - (1-e^{-t})\log(1-e^{-t})\right),$$
$$\dot{I}(t) \stackrel{\text{def}}{=} \frac{t^2}{e^t - 1}.$$

In other words, the number of darts in this cell is a Poisson(t) random variable, $t = p\lambda$, and both entropy and normalized information can be expressed in terms of t via functions $\dot{H}, \dot{I}$.

Proof. $\Pr(Z = 0) = e^{-p\lambda}$ follows from the definition of the process. Then we have, by the definition of entropy and Fisher information,

$$\begin{aligned} H(Z) &= -e^{-p\lambda}\log_2 e^{-p\lambda} - (1-e^{-p\lambda})\log_2(1-e^{-p\lambda}) \\ &= p\lambda e^{-p\lambda}/\log 2 - (1-e^{-p\lambda})\log_2(1-e^{-p\lambda}) = \dot{H}(p\lambda), \\ \lambda^2 \cdot I_Z(\lambda) &= \lambda^2\left(\frac{e^{-2p\lambda}p^2}{e^{-p\lambda}} + \frac{e^{-2p\lambda}p^2}{1-e^{-p\lambda}}\right) = \frac{e^{-p\lambda}(p\lambda)^2}{1-e^{-p\lambda}} = \dot{I}(p\lambda). \end{aligned}$$

□

Still fixing $c_i \in \mathcal{M}$ with size p_i, let us now aggregate its *potential* contributions to entropy/information *over all time*. We say potential contribution because in a linearizable sketch, it is possible for cell c_i to be "killed"; at the moment $\phi(\mathbf{Y}_{i-1})$ switches from 0 to 1, Z_i is no longer relevant. We measure time on a log-scale, so $\lambda = e^x$. Unsurprisingly, the potential contributions of c_i do not depend on p_i:

Lemma 5.11.

$$\int_{-\infty}^{\infty} \dot{H}(e^x)dx = H_0 \quad \text{and} \quad \int_{-\infty}^{\infty} \dot{I}(e^x)dx = I_0.$$

Proof. Follows from Definition 5.9 and Lemma 5.4, since $\frac{1}{\log 2} \cdot h(e^{-e^w}) = \dot{H}(e^w)$ and $g(e^w) = \dot{I}(e^w)$. □

In other words, if we let cell c_i "live" forever (fix $\phi(\mathbf{Y}_{i-1}) = 0$ for all time) it would contribute H_0 to the aggregate entropy and I_0 to the aggregate normalized Fisher information. In reality c_i may die at some particular time, which leads to a natural *optimization* question. When is the most advantageous time λ to kill c_i, as a function of its density $t_i = p_i\lambda$?

Fig. 5.10 plots $\dot{H}(t)$, $\dot{I}(t)$ and—most importantly—the ratio $\dot{H}(t)/\dot{I}(t)$. It *appears* as if $\dot{H}(t)/\dot{I}(t)$ is monotonically decreasing in t and this is, in fact, the case, as established in Lemma 5.12. See Section 5.4.4.

Lemma 5.12. *$\dot{H}(t)/\dot{I}(t)$ is decreasing in t on $(0, \infty)$.*

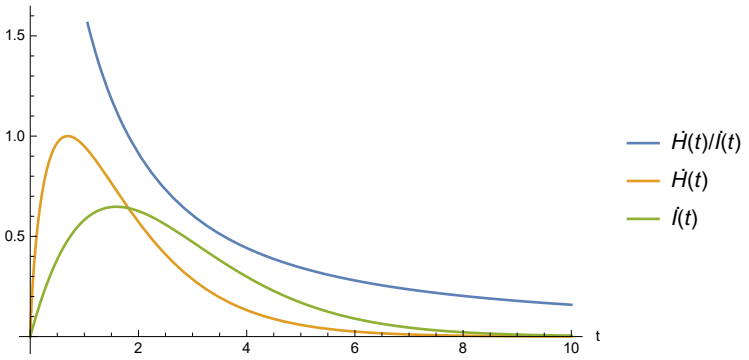


Figure 5.10: $\dot{H}(t), \dot{I}(t)$ and $\dot{H}(t)/\dot{I}(t)$

Lemma 5.12 is the critical observation. Although the *cost* $\dot{H}(t)$ and *value* $\dot{I}(t)$ fluctuate with t, the cost-per-unit-value *only improves with time*. In other words, the optimum moment to "kill" any cell c_i should be *never*, and any linearizable sketch that routinely kills cells prematurely should, on average, perform strictly worse than PCSA—the ultimate pacifist sketch.

The rest of the proof formalizes this intuition. One difficulty is that H_0/I_0 is not a *hard* lower bound at any particular moment in time. For example, if we just want to perform well when the cardinality λ is in, say, $[10^6, 2 \cdot 10^6]$, then we can easily beat H_0/I_0 by a constant factor.[16] However, if we want to perform well over a sufficiently long time interval $[a, b]$, then, at best, the worst case efficiency over that interval tends to H_0/I_0 in the limit.

Define $Z_{i,\lambda}, Y_{i,\lambda}$ to be the variables Z_i, Y_i at time λ. Let $\mathbf{Y} = \mathbf{Y}_{|\mathcal{M}|-1} = (Y_0, \ldots, Y_{|\mathcal{M}|-1})$ be the vector of indicators encoding the state of the sketch and $\mathbf{Y}_{[\lambda]} = (Y_{0,\lambda}, \ldots, Y_{|\mathcal{M}|-1,\lambda})$ refer to $\mathbf{Y}$ at time λ.

Proposition 5.1. *For any linearizable sketch and any $c_i \in \mathcal{M}$, $\Pr(\phi(\mathbf{Y}_{i-1,\lambda}) = 0)$ is non-increasing with λ.*

Proof. Follows from Rule (R3) and the monotonicity of ϕ. □

[16] Clifford and Cosma [47] calculated the optimal Fisher information for Bernoulli observables when λ was known to lie in a small range.

The proof depends on *linearizability* mainly through Lemma 5.13, which uses the chain rule to bound aggregate entropy/information in terms of a weighted sum of cell entropy/information. The weights here correspond to the probability that the cell is still alive, which, by Proposition 5.1, is non-increasing over time.

Lemma 5.13. *For any linearizable sketch and any $\lambda > 0$, we have*

$$H(\mathbf{Y}_{[\lambda]}) = \sum_{i=0}^{|\mathcal{M}|-1} \dot{H}(p_i\lambda)\Pr(\phi(\mathbf{Y}_{i-1,\lambda}) = 0),$$
$$\lambda^2 \cdot I_{\mathbf{Y}}(\lambda) = \sum_{i=0}^{|\mathcal{M}|-1} \dot{I}(p_i\lambda)\Pr(\phi(\mathbf{Y}_{i-1,\lambda}) = 0).$$

Proof. By the chain rule of entropy, we have

$$\begin{aligned} H(\mathbf{Y}_{[\lambda]}) &= \sum_{i=0}^{|\mathcal{M}|-1} H(Y_{i,\lambda} \mid \mathbf{Y}_{i-1,\lambda}) = \sum_{i=0}^{|\mathcal{M}|-1} H(Z_{i,\lambda})\Pr(\phi(\mathbf{Y}_{i-1,\lambda}) = 0) \\ &= \sum_{i=1}^{|\mathcal{M}|-1} \dot{H}(p_i\lambda)\Pr(\phi(\mathbf{Y}_{i-1,\lambda}) = 0), \end{aligned}$$

where the last equality follows from Lemma 5.10. Similarly, by the chain rule of Fisher information number, we have

$$\begin{aligned} \lambda^2 \cdot I_{\mathbf{Y}}(\lambda) &= \sum_{i=0}^{|\mathcal{M}|-1} \lambda^2 \cdot I_{Y_i|\mathbf{Y}_{i-1}}(\lambda) = \sum_{i=0}^{|\mathcal{M}|-1} \lambda^2 \cdot I_{Z_i}(\lambda)\Pr(\phi(\mathbf{Y}_{i-1}) = 0) \\ &= \sum_{i=0}^{|\mathcal{M}|-1} \dot{I}(p_i\lambda)\Pr(\phi(\mathbf{Y}_{i-1}) = 0), \end{aligned}$$

where the last equality follows from Lemma 5.10. □

Definition 5.13 introduces some useful notation for talking about the aggregate contributions of *some* cells to *some* period of time (on a log-scale) $W = [a, b]$, i.e., all $\lambda \in [e^a, e^b]$.

Definition 5.13. Fix a linearizable sketch. Let $C \subset \mathcal{M}$ be a collection of cells and $W \subset \mathbb{R}$ be an interval of the reals. Define:

$$\mathbf{H}(C \to W) = \int_W \sum_{c_i \in C} \dot{H}(p_i e^x)\Pr(\phi(\mathbf{Y}_{i-1,e^x}) = 0)dx,$$

$$\mathbf{I}(C \to W) = \int_W \sum_{c_i \in C} \dot{I}(p_i e^x) \Pr(\phi(\mathbf{Y}_{i-1,e^x}) = 0) dx.$$

A linearizable sketching *scheme* is really an algorithm that takes a few parameters, such as a desired space bound and a maximum allowable cardinality, and produces a partition $\mathcal{M}$ of the dartboard, a function ϕ (implicitly defining the state space $\mathcal{S}$), and a cardinality estimator $\hat{\lambda} : \mathcal{S} \to \mathbb{R}$. Since we are concerned with asymptotic performance we can assume $\hat{\lambda}$ is MLE, so the sketch is captured by just $\mathcal{M}, \phi$.

In Theorem 5.6 we assume that such a linearizable sketching scheme has produced $\mathcal{M}, \phi$ such that the *entropy* (i.e., space, in expectation) is *at most* $\tilde{H}$ at all times, and that the normalized information is *at least* $\tilde{I}$ for all times $\lambda \in [e^a, e^b]$. It is proved that $\tilde{H}/\tilde{I} \geq (1 - o_d(1))H_0/I_0$, where $d = b - a$ and $o_d(1) \to 0$ as $d \to \infty$. The take-away message (proved in Corollary 5.1) is that all scale-invariant linearizable sketches have Fish-number at least H_0/I_0.

Theorem 5.6. *Fix reals $a < b$ with $d = b - a > 1$. Let $\tilde{H}, \tilde{I} > 0$. If a linearizable sketch satisfies that*

- *For all $\lambda > 0$, $H(\mathbf{Y}_{[\lambda]}) \leq \tilde{H}$,*
- *For all $\lambda \in [e^a, e^b]$, $\lambda^2 \cdot I_{\mathbf{Y}}(\lambda) \geq \tilde{I}$,*

then

$$\frac{\tilde{H}}{\tilde{I}} \;\geq\; \frac{H_0}{I_0} \frac{1 - \max(8d^{-1/4}, 5e^{-d/2})}{1 + \frac{(344+4\sqrt{d})}{d} \frac{H_0}{I_0} (1 - \max(8d^{-1/4}, 5e^{-d/2}))} = (1 - o_d(1)) \frac{H_0}{I_0}.$$

The expression for this $1 - o_d(1)$ factor arises from the following two technical lemmas, proved in Section 5.4.4.

Lemma 5.14. *For any $d > 0$ and $t \geq \frac{1}{2} \log d$,*

$$\frac{\int_{-\infty}^{-t} \dot{H}(e^x) dx}{\int_{-\infty}^{-t+d} \dot{H}(e^x) dx} \leq \max(8d^{-1/4}, 5e^{-d/2}).$$

Lemma 5.15. *Let $d = b - a > 1$, $\Delta = \frac{1}{2} \log d$ and $\mathcal{M}^* = \{c_i \in \mathcal{M} \mid p_i < e^{-a-\Delta}\}$. Assume that for all $\lambda > 0$, $H(\mathbf{Y}_{[\lambda]}) \leq \tilde{H}$ (the first condition of Theorem 5.6). Then we have*

$$\mathbf{I}(\mathcal{M} \setminus \mathcal{M}^* \to [a, b]) \leq (344 + 4e^{\Delta}) \tilde{H}.$$

Proof of Theorem 5.6. First, since for all $\lambda \in [e^a, e^b]$, we have both $H(\mathbf{Y}_{[\lambda]}) \le \tilde{H}$ and $\lambda^2 \cdot I_{\mathbf{Y}}(\lambda) \ge \tilde{I}$, we know, by Lemma 5.13,

$$\frac{\mathbf{H}(\mathcal{M} \to [a,b])}{\mathbf{I}(\mathcal{M} \to [a,b])} = \frac{\int_a^b H(\mathbf{Y}_{[e^x]})dx}{\int_a^b e^{2x} I_{\mathbf{Y}}(e^x)dx} \le \frac{\tilde{H}d}{\tilde{I}d} = \frac{\tilde{H}}{\tilde{I}}. \tag{5.3}$$

Thus it is sufficient to bound $\frac{\mathbf{H}(\mathcal{M}\to[a,b])}{\mathbf{I}(\mathcal{M}\to[a,b])}$. Define $\Delta = \frac{1}{2}\log d$ and $\mathcal{M}^* = \{c_i \in \mathcal{M} \mid p_i < e^{-a-\Delta}\}$. We then have

$$\frac{\mathbf{H}(\mathcal{M} \to [a,b])}{\mathbf{I}(\mathcal{M} \to [a,b])} \ge \frac{\mathbf{H}(\mathcal{M}^* \to [a,b])}{\mathbf{I}(\mathcal{M} \to [a,b])} = \frac{\mathbf{H}(\mathcal{M}^* \to [a,b])}{\mathbf{I}(\mathcal{M}^* \to [a,b])} \cdot \frac{\mathbf{I}(\mathcal{M}^* \to [a,b])}{\mathbf{I}(\mathcal{M} \to [a,b])}. \tag{5.4}$$

We shall bound $\frac{\mathbf{H}(\mathcal{M}^*\to[a,b])}{\mathbf{I}(\mathcal{M}^*\to[a,b])}$ and $\frac{\mathbf{I}(\mathcal{M}^*\to[a,b])}{\mathbf{I}(\mathcal{M}\to[a,b])}$ separately.

First, for any cell $c_i \in \mathcal{M}^*$, let $f(t) = \dot{H}(p_i e^t)$, $g(t) = \dot{I}(p_i e^t)$ and $h(t) = \Pr(\phi(\mathbf{Y}_{i-1,e^t}) = 0)$. By Lemma 5.12 and Proposition 5.1, we know that $f(t)/g(t)$ and $h(t)$ are non-increasing in t. By Lemma 5.11, we know both $f(t)$ and $g(t)$ have finite integral over $(-\infty, \infty)$. It is also easy to see that $f(t) > 0$, $g(t) > 0$ and $h(t) \in [0,1]$ for all $t \in \mathbb{R}$. By the first part of Lemma 5.16 we conclude that

$$\frac{\int_a^b \dot{H}(p_i e^t)\Pr(\phi(\mathbf{Y}_{i-1,e^t}) = 0)\,dt}{\int_a^b \dot{I}(p_i e^t)\Pr(\phi(\mathbf{Y}_{i-1,e^t}) = 0)\,dt} \ge \frac{\int_a^b \dot{H}(p_i e^t)\,dt}{\int_a^b \dot{I}(p_i e^t)dt}.$$

In addition, we have

$$\begin{aligned}\frac{\int_a^b \dot{H}(p_i e^t)\,dt}{\int_a^b \dot{I}(p_i e^t)dt} &\ge \frac{\int_a^b \dot{H}(p_i e^t)\,dt}{\int_{-\infty}^b \dot{I}(p_i e^t)dt} = \frac{\int_{-\infty}^b \dot{H}(p_i e^t)\,dt}{\int_{-\infty}^b \dot{I}(p_i e^t)dt} \cdot \frac{\int_a^b \dot{H}(p_i e^t)\,dt}{\int_{-\infty}^b \dot{H}(p_i e^t)dt} \\ &\ge \frac{\int_{-\infty}^\infty \dot{H}(p_i e^t)\,dt}{\int_{-\infty}^\infty \dot{I}(p_i e^t)dt} \cdot \frac{\int_a^b \dot{H}(p_i e^t)\,dt}{\int_{-\infty}^b \dot{H}(p_i e^t)dt},\end{aligned}$$

where the last inequality follows from the second part of Lemma 5.16. By Lemma 5.11 we know that $\frac{\int_{-\infty}^\infty \dot{H}(p_i e^t)\,dt}{\int_{-\infty}^\infty \dot{I}(p_i e^t)dt} = H_0/I_0$. By applying Lemma 5.14, we have

$$\frac{\int_a^b \dot{H}(p_i e^t)\,dt}{\int_{-\infty}^b \dot{H}(p_i e^t)dt} = 1 - \frac{\int_{-\infty}^a \dot{H}(p_i e^t)\,dt}{\int_{-\infty}^b \dot{H}(p_i e^t)dt} = 1 - \frac{\int_{-\infty}^{a+\log p_i} \dot{H}(e^t)\,dt}{\int_{-\infty}^{a+\log p_i + d} \dot{H}(e^t)dt} \ge 1 - \max(8d^{-1/4}, 5e^{-d/2}).$$

Note here that since cell $c_i \in \mathcal{M}^*$, $a + \log p_i \le a + (-a - \Delta) = -\frac{1}{2}\log d$, as required by

Lemma 5.14. Therefore, for any $c_i \in \mathcal{M}^*$, we have

$$\frac{\int_a^b \dot{H}(p_i e^t)\Pr(\phi(\mathbf{Y}_{i-1,e^t})=0)\,dt}{\int_a^b \dot{I}(p_i e^t)\Pr(\phi(\mathbf{Y}_{i-1,e^t})=0)\,dt} \geq \frac{H_0}{I_0}(1-\max(8d^{-1/4},5e^{-d/2})).$$

Summing over all cells in $\mathcal{M}^*$, this also implies that

$$\frac{\mathbf{H}(\mathcal{M}^* \to [a,b])}{\mathbf{I}(\mathcal{M}^* \to [a,b])} = \frac{\sum_{c_i \in \mathcal{M}^*}\int_a^b \dot{H}(p_i e^t)\Pr(\phi(\mathbf{Y}_{i-1,e^t})=0)\,dt}{\sum_{c_i \in \mathcal{M}^*}\int_a^b \dot{I}(p_i e^t)\Pr(\phi(\mathbf{Y}_{i-1,e^t})=0)\,dt} \geq \frac{H_0}{I_0}(1-\max(8d^{-1/4},5e^{-d/2})). \tag{5.5}$$

Secondly, by Lemma 5.15, we have

$$\frac{\mathbf{I}(\mathcal{M}^* \to [a,b])}{\mathbf{I}(\mathcal{M} \to [a,b])} = 1 - \frac{\mathbf{I}(\mathcal{M} \setminus \mathcal{M}^* \to [a,b])}{\mathbf{I}(\mathcal{M} \to [a,b])} \geq 1 - \frac{(344+4e^{\Delta})\tilde{H}}{\mathbf{I}(\mathcal{M} \to [a,b])}.$$

Since for all $\lambda \in [e^a, e^b]$, $\lambda^2 \cdot I_{\mathbf{Y}}(\lambda) \geq \tilde{I}$, we have, by Lemma 5.13

$$\mathbf{I}(\mathcal{M} \to [a,b]) = \int_a^b e^{2t} I_{\mathbf{Y}}(e^t)\,dt \geq \tilde{I}d.$$

Thus we have

$$\frac{\mathbf{I}(\mathcal{M}^* \to [a,b])}{\mathbf{I}(\mathcal{M} \to [a,b])} \geq 1 - \frac{(344+4e^{\Delta})\tilde{H}\log 2}{\tilde{I}d}. \tag{5.6}$$

By combining inequalities (5.3), (5.4), (5.5), and (5.6), we have

$$\frac{\tilde{H}}{\tilde{I}} \geq \frac{H_0}{I_0}\left(1-\max(8d^{-1/4},5e^{-d/2})\right)\left(1-\frac{(344+4\sqrt{d})\tilde{H}}{d\tilde{I}}\right), \tag{5.7}$$

and by rearranging inequality (5.7), we finally conclude that

$$\frac{\tilde{H}}{\tilde{I}} \geq \frac{H_0}{I_0}\frac{1-\max(8d^{-1/4},5e^{-d/2})}{1+\frac{(344+4\sqrt{d})}{d}\frac{H_0}{I_0}(1-\max(8d^{-1/4},5e^{-d/2}))} = (1-o_d(1))\frac{H_0}{I_0}.$$

□

Corollary 5.1. *Let A_q be any linearizable, weakly scale-invariant sketch with base q. Then* $\mathsf{Fish}(A_q) \geq H_0/I_0$.

Proof. Fix $m > 0$. Let A_q^m be a vector of m independent, offsetted A_q sketches with respect

to the uniform offset vector $(0, 1/m, 2/m, \ldots, (m-1)/m)$. First note that, since A_q is linearizable, A_q^m is also linearizable since we can simply concatenate the linear orders on the cells of each independent subsketch.

Let $\tilde{H}_m = \sup\{H(X_{A_q^m,q^r}) \mid r \in [0, 1/m]\}$ and $\tilde{I}_m = \inf\{q^{2r} I_{A_q^m}(q^r) \mid r \in [0, 1/m]\}$. Since A_q^m is weakly scale-invariant sketch with base $q^{1/m}$, for any $\lambda > 0$ we have

$$H(X_{A_q^m,\lambda}) \leq \tilde{H}_m \quad \text{and} \quad \lambda^2 \cdot I_{A_q^m}(\lambda) \geq \tilde{I}_m.$$

Therefore we can apply Theorem 5.6 to A_q^m with arbitrary large $d = b - a$. This implies that $\tilde{H}_m/\tilde{I}_m \geq H_0/I_0$. On the other hand, note that as m becomes large, the sketch is smoothed, i.e., $\tilde{H}_m/\tilde{I}_m$ converges to $\mathsf{Fish}(A_q)$ as $m \to \infty$. We conclude that $\mathsf{Fish}(A_q) \geq H_0/I_0$. □

5.4.4 Deferred Proofs

Lemma 5.16. *Let f, g and h be real functions. If*

- *$f(t) \geq 0$, $g(t) > 0$ and $h(t) \in [0, 1]$ for all $t \in \mathbb{R}$,*
- *$f(t)/g(t)$ and $h(t)$ are (weakly) decreasing in t, and*
- *$\int_{-\infty}^{\infty} f(t)\,dt < \infty$ and $\int_{-\infty}^{\infty} g(t)\,dt < \infty$,*

then for any $-\infty \leq a < b \leq \infty$ such that $\int_a^b g(t)h(t)\,dt > 0$, we have

$$\frac{\int_a^b f(t)h(t)\,dt}{\int_a^b g(t)h(t)\,dt} \geq \frac{\int_a^b f(t)\,dt}{\int_a^b g(t)\,dt}. \tag{5.8}$$

In particular, we have

$$\frac{\int_{-\infty}^b f(t)\,dt}{\int_{-\infty}^b g(t)\,dt} \geq \frac{\int_{-\infty}^{\infty} f(t)\,dt}{\int_{-\infty}^{\infty} g(t)\,dt} \tag{5.9}$$

Proof. To show (5.8), it is sufficient to prove the following difference is non-negative.

$$\begin{aligned}
&2\int_a^b f(t)h(t)\,dt \int_a^b g(t)\,dt - 2\int_a^b g(t)h(t)\,dt \int_a^b f(t)\,dt \\
&= \int_a^b \int_a^b f(x)h(x)g(y) + f(y)h(y)g(x)dxdy - \int_a^b \int_a^b g(x)h(x)f(y) + g(y)h(y)f(x)dxdy \\
&= \int_a^b \int_a^b (f(x)g(y) - g(x)f(y))(h(x) - h(y))dxdy.
\end{aligned}$$

Note that $(f(x)g(y)-g(x)f(y))(h(x)-h(y)) = \frac{1}{g(x)g(y)}(f(x)/g(x)-f(y)/g(y))(h(x)-h(y)) \geq 0$, since both $f(t)/g(t)$ and $h(t)$ are decreasing. Thus the difference is non-negative.

Inequality (5.9) follows from 5.8 by setting $a = -\infty$, $b = \infty$ and $h(t) = \mathbb{1}(t \leq b)$. □

Proof of Lemma 5.12. Note that

$$\frac{\dot{H}(t)\log 2}{\dot{I}(t)} = \frac{1-e^{-t}}{t} + \frac{(1-e^{-t})^2}{t^2} \cdot (-e^t \log(1-e^{-t})).$$

Since $-\log(1-e^{-t})$ is decreasing on $(0,\infty)$, it suffices to prove that $\frac{1-e^{-t}}{t}$ and $-e^t\log(1-e^{-t})$ are decreasing on $(0,\infty)$. Let $f(t) = \frac{1-e^{-t}}{t}$ and $g(t) = -e^t\log(1-e^{-t})$. By taking the derivative of $f(t)$, we have

$$f'(t) = \frac{e^{-t}t - (1-e^{-t})}{t^2} = -\frac{e^{-t}}{t^2}(e^t - 1 - t) \leq 0$$

which implies $f(t)$ is decreasing. By taking the derivative of $g(t)$, we have

$$g'(t) = -e^t\left(\log(1-e^{-t}) + \frac{e^{-t}}{1-e^{-t}}\right),$$

where we want to show that $g'(t) \leq 0$ for all $t > 0$. Note that for any $x > 0$, we have $x \geq \log(1+x)$. Set $x = \frac{1}{e^t-1}$ and we have

$$\frac{1}{e^t-1} \geq \log\left(1+\frac{1}{e^t-1}\right) \iff \frac{e^{-t}}{1-e^{-t}} - \log\left(\frac{e^t}{e^t-1}\right) \geq 0 \iff \log\left(1-e^{-t}\right) + \frac{e^{-t}}{1-e^{-t}} \geq 0,$$

which implies, indeed, $g'(t) \leq 0$ for all $t > 0$. This completes the proof. □

Proof of Lemma 5.14 Lemma 5.14 is a property of the function $\dot{H}(\cdot)$. To prove that, we first need the following lemmas. Define $\dot{H}_e(\cdot)$ to be $\dot{H}(\cdot)\log 2$, i.e. the entropy measured in natural base.

Lemma 5.17. *For all $t > 0$,*

$$te^{-t} \leq \dot{H}_e(t) \leq 2\sqrt{t}.$$

Proof. The lower bound follows directly from the definition:

$$\dot{H}_e(t) = te^{-t} - (1-e^{-t})\log(1-e^{-t}) \geq te^{-t}.$$

For the upper bound, first note that since $\dot{H}_e(t)$ is the entropy (measured in "*nats*") of a Bernoulli random variable, we have $\dot{H}_e(t) \le \log(2)$. Thus we only need to prove $\dot{H}_e(t) \le 2\sqrt{t}$ for $t \in (0, \log(2)^2/4]$. Note that $te^{-t} \le t \le \sqrt{t}$ for $t \in (0, 1]$. It then suffices to show that $-(1-e^{-t})\log(1-e^{-t}) \le \sqrt{t}$ for $t \in (0, \log^2(2)/4]$. Observe the following.

- $-x \log x$ is increasing in $x \in (0, 1/e]$, since $(-x \log x)' = -\log x + 1$. Note that $1/e > 0.36 > 0.12 > 1 - e^{-\log^2(2)/4}$.
- $1 - e^{-t} \le t$ for all $t \in \mathbb{R}$.
- $-t \log t < \sqrt{t}$ for $t \in (0, 1]$. Let $f(t) = t^{-1/2} + \log t$. Then we have $f'(t) = -t^{-3/2}/2 + 1/t = t^{-3/2}(-1/2 + \sqrt{t})$. Then only zero of $f'(t)$ is at $t = 1/4$, which is the minimum point. Note that $f(1/4) = 2 - \log 4 > 0$. Therefore $t^{-1/2} + \log t > 0$ for $t \in (0, 1]$, which implies $-t \log t < \sqrt{t}$.

Then we have, for $t \in (0, \log^2(2)/4]$,

$$-(1-e^{-t})\log(1-e^{-t}) \le -t \log t \le \sqrt{t},$$

where the first inequality results from the first two observations and the second inequality follows from the last observation. □

Lemma 5.18. *For any $p > 1$, $\dot{H}_e(t/p)/\dot{H}_e(t)$ is increasing in $t \in (0, \log(2))$.*

Proof. Fix $p > 1$, let $f(t) = \dot{H}_e(t/p)/\dot{H}_e(t)$. First note that

$$\begin{aligned}
\dot{H}_e(t) &= te^{-t} - (1-e^{-t})\log(1-e^{-t}) = t - (1-e^{-t})\log(e^t - 1), \\
\dot{H}_e'(t) &= e^{-t} - te^{-t} - e^{-t}\log(1-e^{-t}) - e^{-t} = -e^{-t}\log(e^t - 1).
\end{aligned}$$

We have

$$f'(t) = \frac{\dot{H}_e'(t/p)\dot{H}_e(t)/p - \dot{H}_e(t/p)\dot{H}_e'(t)}{\dot{H}_e(t)^2}.$$

Define

$$\begin{aligned}
g(t) &= \dot{H}_e'(t/p)\dot{H}_e(t)/p - \dot{H}_e(t/p)\dot{H}_e'(t) \\
&= -e^{-t/p}\log\left(e^{t/p} - 1\right)\left(t - (1-e^{-t})\log\left(e^t - 1\right)\right)/p \\
&\qquad + e^{-t}\log\left(e^t - 1\right)\left(t/p - \left(1 - e^{-t/p}\right)\log\left(e^{t/p} - 1\right)\right)
\end{aligned}$$

$$= \frac{1}{pe^{t/p+t}}\Bigg[-\log\left(e^{t/p}-1\right)\left(te^t-\left(e^t-1\right)\log\left(e^t-1\right)\right) \\ + p\log\left(e^t-1\right)\left(\frac{t}{p}e^{t/p}-\left(e^{t/p}-1\right)\log\left(e^{t/p}-1\right)\right)\Bigg].$$

We want to show that $g(t) \geq 0$ for $t \in (0, \log(2))$. Define

$$h(t) = \frac{e^t}{\log(e^t-1)} - \frac{e^t-1}{t},$$

and then we can write

$$g(t) = \frac{t\log(e^{t/p}-1)\log(e^t-1)}{pe^{t/p+t}}(-h(t)+h(t/p)).$$

Since $t \in (0, \log(2))$ and $p > 1$, we know $\frac{t\log(e^{t/p}-1)\log(e^t-1)}{pe^{t/p+t}} > 0$. Therefore, it suffices to show $h(t)$ is decreasing in $(0, \log(2))$. Note that for $t \in (0, \log(2))$, we have $\log(e^t-1) < 0$ and $|\log(e^t-1)|$ is decreasing while e^t is increasing. Thus $\frac{e^t}{\log(e^t-1)}$ is decreasing on $(0, \log(2))$. On the other hand,

$$\frac{d}{dt}\frac{e^t-1}{t} = \frac{e^t t - e^t + 1}{t^2}.$$

Let $w(t) = e^t t - e^t + 1$. We have $w'(t) = e^t + e^t t - e^t > 0$. Thus we have $w(t) \geq w(0) = 0$. Therefore, $\frac{e^t-1}{t}$ is increasing in t. Thus, indeed, $h(t)$ is decreasing.

We conclude that $\dot{H}_e(t/p)/\dot{H}_e(t)$ is increasing in $t \in (0, \log(2))$. □

Now we can prove Lemma 5.14.

Proof of Lemma 5.14. Note that by the upper bound in Lemma 5.17, we have

$$\int_{-\infty}^{-t} \dot{H}_e(e^x)dx \leq \int_{-\infty}^{-t} 2e^{x/2}dx = 4e^{-t/2} \leq 4d^{-1/4},$$

where the last inequality follows from the assumption $t \geq \frac{1}{2}\log d$. If $-t + d > \log\log(2)$, then, by the lower bound in Lemma 5.17,

$$\int_{-\infty}^{-t+d} \dot{H}_e(e^x)dx \geq \int_{-\infty}^{\log\log(2)} \dot{H}_e(e^x)dx \geq \int_{-\infty}^{\log\log(2)} e^x e^{-e^x}dx = \frac{1}{2},$$

where we use the fact that $\int e^x e^{-e^x} dx = -e^{-e^x}$. In this case we have

$$\frac{\int_{-\infty}^{-t} \dot{H}_e(e^x)dx}{\int_{-\infty}^{-t+d} \dot{H}_e(e^x)dx} \leq 8d^{-1/4}.$$

If $-t+d \leq \log\log(2)$, then for any $x \leq -t+d$, $e^x \leq \log(2)$. Thus by Lemma 5.18, for any $x \leq -t+d$, $\dot{H}_e(e^{x-d})/\dot{H}_e(e^x) \leq \dot{H}_e(\log(2)/e^d)/\dot{H}_e(\log(2))$. Then we have

$$\int_{-\infty}^{-t+d} \dot{H}_e(e^x)dx = \int_{-\infty}^{-t+d} \dot{H}_e(e^{x-d}) \frac{\dot{H}_e(e^x)}{\dot{H}_e(e^x/e^d)} dx \geq \frac{\dot{H}_e(\log(2))}{\dot{H}_e(\ln 2/e^d)} \int_{-\infty}^{-t} \dot{H}_e(e^x)dx,$$

which implies, using the bounds of Lemma 5.17, that

$$\frac{\int_{-\infty}^{-t} \dot{H}_e(e^x)dx}{\int_{-\infty}^{-t+d} \dot{H}_e(e^x)dx} \leq \frac{\dot{H}_e(\log(2)/e^d)}{\dot{H}_e(\log(2))} \leq \frac{2\sqrt{\log(2)/e^d}}{\log(2)e^{-\log(2)}} < 5e^{-d/2}.$$

We conclude that, in both cases,

$$\frac{\int_{-\infty}^{-t} \dot{H}(e^x)dx}{\int_{-\infty}^{-t+d} \dot{H}(e^x)dx} = \frac{\int_{-\infty}^{-t} \dot{H}_e(e^x)dx}{\int_{-\infty}^{-t+d} \dot{H}_e(e^x)dx} \leq \max(8d^{-1/4}, 5e^{-d/2}).$$

□

Proof of Lemma 5.15 Lemma 5.15 is a property of the function $\dot{I}(\cdot)$. To prove that, we first need the following lemmas.

Lemma 5.19. *For all $t > 0$,*

$$\dot{I}(t) \leq 4e^{-t/2}.$$

Proof. Recall that $\dot{I}(t) = \frac{t^2}{e^t - 1}$. Therefore we have, for $t > 0$,

$$\dot{I}(t) \leq 4e^{-t/2} \iff 4e^t - t^2 e^{t/2} - 4 \geq 0.$$

Let $f(t) = 4e^t - t^2 e^{t/2} - 4$. Then we have

$$\begin{aligned} f'(t) &= 4e^t - e^{t/2}(t^2/2 + 2t) = e^{t/2}(4e^{t/2} - 2t - t^2/2) \\ &\geq e^{t/2}(4 + 2t + t^2/2 - 2t - t^2/2) \geq 0, \end{aligned}$$

since $e^x \geq 1 + x + x^2/2$ for $x \geq 0$, which implies $e^{t/2} \geq 1 + t/2 + t^2/8$. Thus $f(t)$ is increasing.

In addition, $f(0) = 0$. Thus we know $f(t) \geq 0$ for all $t \geq 0$ and therefore $\dot{I}(t) \leq 4e^{-t/2}$ holds. □

Lemma 5.20.

$$\int_0^\infty \sum_{k\geq 0} \sup\{\dot{I}(e^{x+w}) \mid w \in [k, k+1)\} dx \leq 119.$$

Proof. Using the bound derived in Lemma 5.19, we have

$$\begin{aligned}
&\int_0^\infty \sum_{k\geq 0} \sup\{\dot{I}(e^{x+w}) \mid w \in [k, k+1)\} dx \\
&\leq \int_0^\infty \sum_{k\geq 0} \sup\{4\exp(-e^{x+w}/2) \mid w \in [k, k+1)\} dx \\
&= 4\int_0^\infty \sum_{k\geq 0} \exp(-e^{x+k}/2) dx \\
&\leq 4\int_0^\infty \int_0^\infty \exp(-e^{x+y-1}/2) dy dx \\
&\leq 4\int_0^\infty \int_0^\infty \exp(-(x+y)/(2e)) dy dx \\
&= 4\left(\int_0^\infty e^{-x/(2e)} dx\right)^2 \\
&= 4 \cdot (2e)^2 < 119.
\end{aligned}$$

□

Lemma 5.21. *For any interval A, define $\underline{h}_e(A) \stackrel{\text{def}}{=} \inf\{\dot{H}_e(e^x) \mid x \in A\}$. Then for any $a < b$, $\underline{h}_e([a, b]) = \min(\dot{H}_e(e^a), \dot{H}_e(e^b))$.*

Proof. Note that, from the proof of Lemma 5.18, we know $\dot{H}'_e(t) = -e^{-t}\log(e^t - 1)$, whose only zero is $\log 2$. From this we know when $t < \log 2$, $\dot{H}_e(t)$ increases and when $t > \log 2$, it decreases. Thus given an interval $[e^a, e^b]$, the minimum is always obtained at one of the end points. □

Proof of Lemma 5.15. For any $k \geq 0$, define

$$\begin{aligned}
B_k &\stackrel{\text{def}}{=} \{c_i \in \mathcal{C} \mid p_i \in [e^{-a+k}, e^{-a+k+1})\} \\
B^* &\stackrel{\text{def}}{=} \{c_i \in \mathcal{C} \mid p_i \in (e^{-a-\Delta}, e^{-a})\}.
\end{aligned}$$

We see $\{B_k\}_{k\geq 0}$ together with B^* form a partition of $\mathcal{C}\setminus\mathcal{C}^*$. Define $w(B_k)$ as

$$w(B_k)\stackrel{\text{def}}{=}\sum_{c_i\in B_k}\Pr(\phi(\mathbf{Y}_{i-1,e^{a-k}})=0).$$

Fix some $k\geq 0$. By the entropy assumption, when $\lambda=e^{a-k}$, we have

$$\tilde{H}\geq H(\mathbf{Y}_{[\lambda]})\geq\sum_{c_i\in B_k}\dot{H}_e(p_ie^{a-k})\Pr(\phi(\mathbf{Y}_{i-1,e^{a-k}})=0)/\log(2)\geq\underline{h}_e([0,1])w(B_k)/\log(2), \tag{5.10}$$

since for all cells $c_i\in B_k$, $p_ie^{a-k}\in[e^0,e^1]$ by definition. This implies $w(B_k)\leq\frac{\tilde{H}\log(2)}{\underline{h}_e([0,1])}$ for any $k\geq 0$. By Proposition 5.1, we have, for any $k\geq 0$,

$$\begin{aligned}
\mathbf{I}(B_k\to[a,b]) &= \int_a^b\sum_{c_j\in B_k}\dot{I}(p_je^x)\Pr(\phi(\mathbf{Y}_{j-1,e^x})=0)dx\\
&\leq\int_a^b\sum_{c_j\in B_k}\dot{I}(p_je^x)\Pr(\phi(\mathbf{Y}_{j-1,e^{a-k}})=0)dx\\
&\leq\int_a^b\sup\left\{\dot{I}(p_je^x)\;\middle|\;c_j\in B_k\right\}w(B_k)dx\\
&\leq\int_a^b\sup\left\{\dot{I}(e^{-a+x+w})\;\middle|\;w\in[k,k+1)\right\}w(B_k)dx,
\end{aligned}$$

since for any cell $c_j\in B_k$, $p_j\in[e^{-a+k},e^{-a+k+1})$ by definition. Then we have

$$\begin{aligned}
\mathbf{I}\left(\bigcup_{k\geq 0}B_k\to[a,b]\right) &= \sum_{k=0}^{\infty}\mathbf{I}(B_k\to[a,b])\\
&\leq\sum_{k=0}^{\infty}\int_a^b\sup\left\{\dot{I}(e^{-a+x+w})\;\middle|\;w\in[k,k+1)\right\}w(B_k)dx\\
&=\int_0^{\infty}\sum_{k=0}^{\infty}\sup\left\{\dot{I}(e^{-x+w})\;\middle|\;w\in[k,k+1)\right\}w(B_k)dx.
\end{aligned}$$

By Lemma 5.20 and Inequality 5.10, we have

$$\mathbf{I}(\cup_{k\geq 0}B_k\to[a,b])\leq\frac{\tilde{H}\log(2)}{\underline{h}_e([0,1])}119.$$

By Lemma 5.21, we know $\underline{h}_e([0,1]) > \min(0.65, 0.24) = 0.24$. We then have

$$\mathbf{I}(\cup_{k\geq 0} B_k \to [a,b]) \leq 344\tilde{H}.$$

On the other hand, when $\lambda = e^a$, we have, by the assumption that the entropy never exceeds $\tilde{H}$,

$$\begin{aligned}\tilde{H} \geq H(\mathbf{Y}_{[\lambda]}) &\geq \sum_{c_i \in B^*} \dot{H}_e(p_i e^a) \Pr(\phi(\mathbf{Y}_{i-1,e^a}) = 0)/\log(2) \\ &\geq \underline{h}_e([-\Delta, 0]) \sum_{c_i \in B^*} \Pr(\phi(\mathbf{Y}_{i-1,e^a}) = 0)/\log(2),\end{aligned}$$

since for any cell $c_i \in B^*$, $p_i e^a \in [-\Delta, 0]$ by definition. Then we know that

$$\Pr(\phi(\mathbf{Y}_{i-1,e^a}) = 0) \leq \frac{\tilde{H}\log(2)}{\underline{h}_e([-\Delta,0])}.$$

We can calculate that, again, by Proposition 5.1,

$$\begin{aligned}\mathbf{I}(B^* \to [a,b]) &= \int_a^b \sum_{c_i \in B^*} \dot{I}(p_i e^x) \Pr(\phi(\mathbf{Y}_{i-1,e^x}) = 0) dx \\ &\leq \int_a^b \sum_{c_i \in B^*} \dot{I}(p_i e^x) \Pr(\phi(\mathbf{Y}_{i-1,e^a}) = 0) dx \\ &\leq \int_{-\infty}^{\infty} \sum_{c_i \in B^*} \dot{I}(p_i e^x) \Pr(\phi(\mathbf{Y}_{i-1,e^a}) = 0) dx.\end{aligned}$$

Note that by Lemma 5.11, for any $p_i > 0$, $\int_{-\infty}^{\infty} \dot{I}(p_i e^x) dx = \int_{-\infty}^{\infty} \dot{I}(e^x) dx = I_0$. Continuing,

$$\begin{aligned}&= I_0 \cdot \sum_{c_i \in B^*} \Pr(\phi(\mathbf{Y}_{i-1,e^a}) = 0) dx \\ &\leq I_0 \frac{\tilde{H}\log(2)}{\underline{h}_e([-\Delta,0])}.\end{aligned}$$

By Lemmas 5.17 and 5.21, $\underline{h}_e([-\Delta,0]) = \min(\dot{H}_e(e^{-\Delta}), \dot{H}_e(e^0)) \geq \min(e^{-\Delta}e^{-e^{-\Delta}}, 0.65) \geq e^{-\Delta-1}$ since $\Delta > 0$. Recall that by Lemma 5.4, $I_0 = \pi^2/6$. Thus, we have $\mathbf{I}(B^* \to [a,b]) \leq \tilde{H} 4 e^{\Delta}$. Finally, we conclude that

$$\mathbf{I}(\mathcal{C} \setminus \mathcal{C}^* \to [a,b]) = \mathbf{I}(\cup_{k\geq 0} B_k \to [a,b]) + \mathbf{I}(B^* \to [a,b]) \leq (344 + 4e^{\Delta})\tilde{H}.$$

□

5.5 Fishmonger

The results of Section 5.3 can properly be thought of as *lower bounds* on the performance of q-PCSA and q-LogLog, so it is natural to ask whether there are matching *upper bounds*, at least in principle. Specifically, we can always compress a sketch so that its *expected* size is equal to its entropy. Scheuermann and Mauve [146] and Lang [117] showed that this is effective experimentally, and Lang [117] numerically calculated the entropy of 2-PCSA and 2-LogLog. However, it is most desirable to store the sketch in a *fixed* memory footprint, i.e., to guarantee a certain worst case size bound *at all times*.

In this section we describe a sketch Fishmonger that can be stored in $O(\log^2 \log n) + (1 + o(1))mH_0$ bits and achieves a standard error of $(1 + o(1))\sqrt{1/(mI_0)}$. Through a change of variable, this sketch requires $O(\log^2 \log n) + (1+o(1))(H_0/I_0)b \approx 1.98b$ bits and has standard error $1/\sqrt{b}$. Fishmonger is based on a smoothed, compressed e-PCSA sketch, with a different estimation function, and a fixed level of redundancy. It is characterized by the following features.

- The *abstract* state-space of the sketch is $\{0,1\}^{m\times\log n}$. Due to compression the *true* state-space of the sketch is in correspondence with a *subset* of $\{0,1\}^{m\times\log n}$. (Whenever these two states need to be distinguished, denote $\check{S}$ to be the abstract state and S the true state. At time zero we have $S_0 = \check{S}_0 = 0$.)

- When $a \in [n]$ is processed we interpret $h(a)$ as a random matrix $Z_a \in \{0,1\}^{m\times\log n}$ where $\Pr(Z_a(i,j) = 1) = e^{-j-i/m}$, then set $S \leftarrow S \vee Z_a$ (component-wise OR). In other words, the rows $S(0), \ldots, S(m-1)$ are independent e-PCSA-type sketches effecting a uniform offset vector (see Section 5.2.5)

$$(0, 1/m, 2/m, \ldots, (m-1)/m).$$

- The cardinality is estimated using the Maximum Likelihood Estimator. Define $l(S \mid \lambda)$ to be the $\log_2$-likelihood of seeing S after sketching a set of cardinality λ.

$$l(S \mid \lambda) = \log_2 \left(\Pr_{Z_1,\ldots,Z_\lambda} (Z_1 \vee \cdots \vee Z_\lambda = S) \right).$$

  The estimator is then defined to be

$$\hat{\lambda}(S) = \arg\max_{\lambda} l(S \mid \lambda).$$

The MLE can be computed in $O(m \operatorname{poly}(\log n))$ time via binary search. Clifford and Cosma [47] and Ertl [77], discuss MLE algorithms with improved convergence for LogLog-type sketches but the ideas carry over to PCSA as well.

- The sketch stores the estimate $\hat{\lambda}(S)$ explicitly, then allocates $(1+o(1))m \cdot \mathcal{H}(e\text{-PCSA}) + B \leq (1+o(1))mH_0 + B$ bits for storing S. If $-l\left(S \mid \hat{\lambda}(S)\right) \leq (1+o(1))m \cdot H_0 + B$ then S is successfully stored. If not, then the last update to S cannot be recorded ($\check{S} \neq S$) and the state of the sketch reverts to its state before processing the last element.

The crux of the analysis is to show that when $B = O(\log^2 \log n) + o(m)$, it is *always* possible to store $\check{S}$ in compressed form, with high probability $1 - 1/\operatorname{poly}(m)$.

Theorem 5.7. *The Fishmonger algorithm processes a sequence $\mathcal{A} \in [n]^*$ and maintains a sketch S using*

$$O(\log^2 \log n) + (1 + o(1))mH_0 \approx O(\log^2 \log n) + 3.25724m \text{ bits}$$

that ideally represents an abstract $m \log n$-bit sketch $\check{S}$. With probability $1 - 1/\operatorname{poly}(m)$, $S = \check{S}$ at all times, and $\hat{\lambda}(S)$ is an asymptotically unbiased estimate of the cardinality λ of $\mathcal{C}$ with standard error

$$\sqrt{\frac{(1 + o(1))}{mI_0}} \approx \frac{0.77969}{\sqrt{m}}.$$

The remainder of this section constitutes a proof of Theorem 5.7. We make use of Bernstein's inequality.

Theorem 5.8 (See [71])**.** *Let $X_0, \ldots, X_{m-1}$ be independent random variables such that $X_i - \mathbb{E}(X_i) \leq M$ for all i. Let $X = \sum_i X_i$ and $V = \sum_i \mathbb{V}(X_i)$. Then*

$$\Pr(X > \mathbb{E}(X) + B) < \exp\left(-B^2 / (2V + 2MB/3)\right).$$

Observe that the number of times the abstract state $\check{S}$ can change is $m' = m \log n$. Since the sketch is idempotent, we can conflate "time" with cardinality, and let $S_\lambda, \check{S}_\lambda$ be the states after seeing λ distinct elements. We will first prove that at any particular time λ, the probability that $\check{S}_\lambda$ cannot be stored in the specified number of bits is low, namely $1/\operatorname{poly}(m')$. We then argue that this implies that $\forall \lambda . \check{S}_\lambda = S_\lambda$ holds with probability $1 - 1/\operatorname{poly}(m')$, i.e., the actual state is identical to the abstract state *at all times*.

Fix any time λ. By the independence of the rows $\{\check{S}(i)\}_{i\in[m]}$ of $\check{S}$ we have

$$\begin{aligned} H(\check{S} \mid \lambda) &= \sum_{i\in[m]} H(\check{S}(i) \mid \lambda) \\ &= \sum_{i\in[m]} \mathbb{E}(-l(\check{S}(i) \mid \lambda)) \\ &= (1+o(1))m \cdot \mathcal{H}(e\text{-}\mathsf{PCSA}) = (1+o(1))mH_0, \end{aligned}$$

where the last line follows from Theorem 5.4 and the fact that in the limit ($m \to \infty$), the offset vector is uniformly dense in $[0,1)$. By definition of the MLE $\hat{\lambda}(\check{S})$, we have for every state $\check{S}$,

$$-l\left(\check{S} \mid \hat{\lambda}(\check{S})\right) \le -l(\check{S} \mid \lambda).$$

In particular,

$$\Pr\Big(-l(\check{S} \mid \hat{\lambda}(\check{S})) > H(\check{S} \mid \lambda) + B\Big) \le \Pr\Big(-l(\check{S} \mid \lambda) > H(\check{S} \mid \lambda) + B\Big).$$

Thus, it suffices to analyze the distribution of of the upper tail of $-l(\check{S} \mid \lambda)$.

Define $X_{i,j}$ to be the log-likelihood $-l(\check{S}(i,j) \mid \lambda)$. Note that $\check{S}(i,j)$ is Bernoulli with $p_{i,j} = \Pr(\check{S}(i,j)=0) = (1-e^{-(j+i/m)})^{\lambda} \approx e^{-\lambda e^{-(j+i/m)}}$. In particular, if $j > \log\lambda$, $p_{i,j} = 1-\Theta(\lambda e^{-j})$ and if $j < \log\lambda$ then $p_{i,j} = e^{-\Theta(\lambda e^{-j})}$. Due to the independence of the $(X_{i,j})$, the total variance V is therefore

$$\begin{aligned} V &= \mathbb{V}(-l(\check{S} \mid \lambda)) \\ &= \sum_{i\in[m],j\in[\log n]} \mathbb{V}(X_{i,j}) \\ &\le \sum_{i\in[m],j\in[\log n]} \left(p_{i,j}\log_2^2 p_{i,j} + (1-p_{i,j})\log_2^2(1-p_{i,j})\right) \\ &\le Cm, \end{aligned}$$

for some sufficiently large constant C.

Define $\mathscr{I} \subset [m]\times[\log n]$ to be the set of all indices (i,j) such that

$$\log\lambda - \log(c\log m') \le j + i/m \le \log\lambda + c\log m'$$

for some constant c that controls the error probabilities. If $(i,j) \in \mathscr{I}$ with $j + i/m \ge \log\lambda$

then $\Pr(\check{S}(i,j)=1) = 1-p_{i,j} = \Theta(e^{-(j+i/m)+\log\lambda}) = \Omega((m')^{-c})$. If $(i,j) \in \mathscr{I}$ with $j+i/m \leq \log\lambda$ then $\Pr(\check{S}(i,j)=0) = \Theta(e^{\lambda e^{-(j+i/m)}}) = \Omega((m')^{-c})$. Thus, the cells within $\mathscr{I}$ satisfy a worst case deviation of

$$-l(\check{S}(i,j) \mid \lambda) - E(X_{i,j}) \leq c\log_2 m' + O(1) \stackrel{\text{def}}{=} M.$$

Redefine $X_{i,j}$ so that this deviation of M is satisfied outside $\mathscr{I}$ as well.

$$\begin{aligned} X_{i,j} &= \min\left\{-l\left(\check{S}(i,j) \;\middle|\; \lambda\right),\ M\right\}, \\ X &= \sum_{i\in[m],j\in[\log n]} X_{i,j}, \end{aligned}$$

We choose

$$B = \sqrt{2Cm\log\epsilon^{-1}} + (2/3)M\log\epsilon^{-1},$$

and apply Theorem 5.8.

$$\begin{aligned} \Pr\left(X > H(\check{S} \mid \lambda) + B\right) &\leq \exp\left(-B^2/(2V + (2/3)MB)\right) \\ &\leq \exp\left(-\frac{B^2}{2Cm + (2/3)MB}\right) \\ &< \epsilon. \end{aligned}$$

Outside of $\mathscr{I}$, the most probable outcomes (i.e., those minimizing negated log-likelihood) are to have $\check{S}(i,j) = 1$ whenever $j + i/m$ is too small to be in $\mathscr{I}$ and $\check{S}(i,j) = 0$ whenever $j + i/m$ is too large to be in $\mathscr{I}$. When this occurs, X is identical to $-l(\check{S} \mid \lambda)$. By a union bound, this fails to occur with probability at most $m' \cdot (m')^{-c} = (m')^{-c+1}$. Thus, with probability at least $1 - \epsilon - (m')^{-c+1}$ we achieve the successful outcome

$$-l(\check{S} \mid \lambda) = X \ \leq\ H(\check{S} \mid \lambda) + B \ \leq\ (1 + o(1))mH_0 + B.$$

We set $\epsilon = (m')^{-c+1}$ and hence

$$B = O\left(\sqrt{m\log m'} + \log^2(m')\right) = O\left(\sqrt{m\log m} + (\log\log n)^2\right).$$

At first glance, setting ϵ so high seems insufficient to the task of proving that w.h.p., $\forall\lambda.\, \check{S}_\lambda = S_\lambda$. Ordinarily we would take a union bound over all $\lambda \in [1,n]$, necessitating an $\epsilon \ll n^{-1}$. The key observation is that S changes at most m' times, so it suffices to take a

union bound over a set Λ of *checkpoint* times that witness all states of the sketch.

Define $\epsilon_0 = \sqrt{\epsilon}$ and $\Lambda = \{\lambda_1, \lambda_2, \ldots\}$ to be the set of all times (i.e., cardinalities) of the form

$$\lambda_k = \lfloor (1+\epsilon_0)^k \rfloor \leq n.$$

By a union bound, we fail to have success at all checkpoint times in Λ with probability at most

$$|\Lambda| \cdot 2\epsilon < (\log_{1+\epsilon_0} n) \cdot 2\epsilon = O(\epsilon \cdot \epsilon_0^{-1} \log n) = O(\epsilon_0 \log n).$$

We now need to argue that all states of the data structure can be witnessed, w.h.p., by only checking it at times in Λ, i.e., in any interval $(\lambda_k, \lambda_{k+1})$, the state changes at most once.

Observe the the probability that the next element causes a change to the sketch never increases, since bits in $\check{S}$ or S only get flipped from 0 to 1. Define P_k to be the probability, at time λ_k, that the next element causes a change to the sketch. Observe that P_k is itself a random variable: it is the probability that the next Z_a contains a 1 in some location that is 0 in $\check{S}$. It is straightforward to show that when the true cardinality is λ_k, $\mathbb{E}(P_k) = \Theta(m/\lambda_k)$, and via Chernoff-Hoeffding bounds [71], that $\Pr(P_k > c'm'/\lambda_k) = \exp(-m')$ for a sufficiently large constant c'. Thus we proceed under the assumption that $P_k = O(m'/\lambda_k)$ for all k.

If checkpoints Λ do *not* witness all states of the sketch, then there must have been an index k such that the sketch changed state *twice* in the interval $(\lambda_k, \lambda_{k+1})$. For fixed k, the probability that this occurs is, by a union bound,

$$\binom{\lambda_{k+1} - \lambda_k}{2} P_k^2 < (\epsilon_0 \lambda_k)^2 (c'm'/\lambda_k)^2 = O(\epsilon (m')^2)$$

Taking another union bound over all k shows that Λ fails to witnesses all sketch states with probability

$$\begin{aligned} O(|\Lambda| \epsilon (m')^2) &= O(\epsilon_0^{-1} \log n \epsilon (m')^2) = O(\epsilon_0 (m')^3) \\ &= O((m')^{-(c-1)/2+3}) \end{aligned}$$

Setting c sufficiently large, we conclude that

$$\Pr(\forall \lambda.\, \check{S}_\lambda = S_\lambda) \geq 1 - 1/\operatorname{poly}(m').$$

Whenever $\check{S} = S$, Theorem 5.4 implies the standard error of $\hat{\lambda}$ is

$$\sqrt{\frac{1+o(1)}{m \cdot \mathcal{I}(e\text{-PCSA})}} = \sqrt{\frac{(1+o(1))}{mI_0}} \leq \frac{(1+o(1))0.77969}{\sqrt{m}}.$$

The space used by the sketch (in bits) is

$$\begin{aligned}&\log n + O(\log^2 \log n) + (1+o(1))m \cdot \mathcal{H}(e\text{-PCSA}) \\ &= (1+o(1))\left(\log n + mH_0\right).\end{aligned}$$

Here the $\log n$ term accounts for the cost of explicitly storing the estimate $\hat{\lambda}(S)$. This can be further reduced to $O(\log \log n + \log m)$ bits by storing instead a floating point approximation

$$\tilde{\lambda} \in [\hat{\lambda}, (1+1/m')\hat{\lambda}].$$

By using $\tilde{\lambda}$ in lieu of $\hat{\lambda}$ we degrade the efficiency of the arithmetic encoding. The efficiency loss is $-l(\check{S} \mid \tilde{\lambda}) + l(\check{S} \mid \hat{\lambda})$. Fix an entry (i,j). Define $\hat{p} = e^{-\hat{\lambda}e^{-(j+i/m)}}$ to be the probability that $\check{S}(i,j) = 0$, assuming cardinality $\hat{\lambda}$, and define $\tilde{p} = e^{-\tilde{\lambda}e^{-(j+i/m)}}$ analogously for $\tilde{\lambda}$. The loss in encoding efficiency for location (i,j) is the KL-divergence between the two distributions, i.e.,

$$\begin{aligned}D_{\mathrm{KL}}(\hat{p} \,\|\, \tilde{p}) &= \hat{p}\log_2\left(\frac{\hat{p}}{\tilde{p}}\right) + (1-\hat{p})\log_2\left(\frac{1-\hat{p}}{1-\tilde{p}}\right) \\ &\leq \hat{p}\log_2\left(\frac{\hat{p}}{\tilde{p}}\right) \qquad \tilde{\lambda} \geq \hat{\lambda}, \text{ hence } \tilde{p} \leq \hat{p} \\ &= \hat{p}\frac{1}{\log 2}(\tilde{\lambda} - \hat{\lambda})e^{-(j+i/m)} \\ &\leq \hat{p}\frac{1}{\log 2}\hat{\lambda}e^{-(j+i/m)}/m' \\ &= \hat{p}\log_2(\hat{p}^{-1})/m' < H(\hat{p})/m' < 1/m'.\end{aligned}$$

In other words, over all m' entries in $\check{S}$, the total loss in encoding efficiency due to using $\tilde{\lambda}$ is less than 1 bit.

Remark 5.6. In the proof of Theorem 5.7 we treated the unlikely event that $\check{S} \neq S$ as a *failure*, but in practice nothing bad happens. As these errors occur with probability $1/\operatorname{poly}(m \log n)$ they have a negligible effect on the standard error.

The proof could be simplified considerably if we do not care about the dependence on n. For example, we could set $\epsilon = 1/\operatorname{poly}(n)$ and apply a standard union bound rather than look

at the "checkpoints" Λ. We could have also applied a recent tail bound of Zhao [172] for the log-likelihood of a set of independent Bernoulli random variables. These two simplifications would lead to a redundancy of

$$B = O(\sqrt{m' \log \epsilon^{-1}}) = O(\sqrt{m} \log n).$$

Remark 5.7. The Fishmonger sketch S is commutative *in the abstract*, in the sense that $\check{S}$ is commutative and $\forall\lambda.\check{S}_\lambda = S_\lambda$ holds with high probability. This means that it is mergeable, and can be used in a distributed environment to sketch substreams $\mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(z)}$ separately and then combine them to yield a sketch of $\mathcal{C}^{(1)} \cup \cdots \cup \mathcal{C}^{(z)}$. Strictly speaking Fishmonger is not commutative since among all permutations of $(a_1, \ldots, a_N)$, some negligible fraction will induce occasional $S \neq \check{S}$ errors. The important point is that these bad permutations depend on h and cannot be constructed by an adversary unaware of h.

5.6 Interlude: Non-Commutative Sketching

The idea of non-commutative sketching was discovered by several independent groups, and within a few years of each other, by Chen, Cao, Shepp, and Nguyen [36] in 2009, Helmi, Lumbroso, Martinez, and Viola [94] in 2012, and Cohen [49] in 2014. Moreover, Cohen [49] and Ting [151] (also 2014) discovered a simple way to transform *any* commutative sketch into a better history-independent sketch of essentially the same space complexity. Ting [151] gave a set of generic tools for analyzing the standard error of such sketches.

The S-Bitmap [36] consists of a length-m bit-vector S, initially zero, and is parameterized by a sequence of thresholds $p_0 \geq \cdots \geq p_{m-1}$. The hash function $h : [n] \to [m] \times (0, 1)$ is interpreted as producing an index ι and a real value ρ. When processing the next element a_i with $h(a_i) = (\iota, \rho)$ we set $S(\iota) \leftarrow 1$ iff $\rho \leq p_{\text{HammingWeight}(S)}$. I.e., once h is fixed, the effect that a_i has on the structure *depends on when it appears* in the input sequence. Nonetheless, before h is fixed the distribution of S at any one time clearly depends solely on the cardinality of the input. Thus, it is history independent but non-commutative. They proved that if $\{p_0, \ldots, p_{m-1}\}$ are chosen to accommodate cardinalities $\lambda \in [n]$, the standard error of the estimator is about $(\log(eU/m)/2)/\sqrt{m}$. This error is worse than HyperLogLog for large cardinalities ($n \to \infty$) but is often better when n is not too large and m is not too small.

Helmi et al. [94] began from a simple Min sketch S that stores the minimum m hash values seen so far. Rather than estimate λ based on the *values* of these hashes in S, it does so based on the *number of times* S changes. Their *Recordinality* sketch stores S, which is

commutative, and a counter tallying changes to S, which is not. The combination is history independent. Their analysis depends solely on the property that the first occurrences of the distinct values in $(h(a_1), \ldots, h(a_N))$ induce a random permutation. The standard error is shown to be $\tilde{O}(1/\sqrt{m})$.

Cohen [49] and Ting [151] gave a mechanical way to make *any* commutative sketch S into a history-independent sketch with improved accuracy. This will be the topic in Section 5.7.

One of the virtues of commutative or history independent sketches is that there is no notion of *worst case input*; *all* inputs are equally bad. Sedgewick [147] (unpublished) proposed a sketch called HyperBitBit that consists of 134 bits and empirically gets less than 10% error on several data sets. A careful inspection of the algorithm shows that it is neither commutative nor history independent.

5.6.1 The Error Distribution of HyperBitBit

In this section we describe the HyperBitBit sketch [147],[17] and give an example of two inputs sequences with the same cardinality for which HyperBitBit behaves very differently. It has relative error usually exceeding 20%.

The *purpose* of this section is to illustrate the dangers of designing sketches that are ostensibly in the *Dartboard* model from Section 5.4.1, but violate Rule (R3), that cells, once occupied, remain occupied.

One way to view the (Hyper)LogLog sketch is as representing an infinite table $A[j,k]$ of bits, initially zero, where $j \in [m]$ and $k \in \mathbb{Z}^+$. If a_i is to be processed and $h(a_i) = (j,k)$ (which holds with probability $m^{-1}2^{-k}$), we set all the bits in row j up to column k to be 1.

$$A[j,0], \ldots, A[j,k] \leftarrow 1.$$

If the true cardinality is λ we expect the first $\log_2 \lambda - O(1)$ columns to be nearly all 1s, the next $O(1)$ columns to contain a healthy mixture of 0s and 1s (and hence be the most informative for estimating λ), and the remaining columns to contain nearly all 0s. The idea of Sedgewick's heuristic HyperBitBit sketch is to effectively compress (Hyper)LogLog by only maintaining two columns of A where a constant fraction of the entries are 1.

The sketch is composed of $S = (L, S_0, S_1)$, where L is a $\log\log n$-bit index, S_0, S_1 are

[17] For demonstration, the HyperBitBit sketch we consider here is the version that Sedgewick described during a talk in 2019. Very recently, Janson, Lumbroso, and Sedgewick proposed a new version of HyperBitBit in [100] with improved choices of parameters.

two 64-bit vectors (words), and S_0 satisfies the invariant that HammingWeight$(S_0) \leq 31$. When a_i is to be processed we compute $h(a_i) = (j, k)$. If $k \geq L$ we set $S_0(j) \leftarrow 1$ and if $k \geq L + 1$ we set $S_1(j) \leftarrow 1$. At this point the invariant on S_0 could be violated. If HammingWeight$(S_0) = 32$, we set $L \leftarrow L + 1$, $S_0 \leftarrow S_1$, and $S_1 \leftarrow 0$ (the all-zero vector). Cardinality is estimated as

$$\hat{\lambda}(S) \propto 2^{L+\text{HammingWeight}(S_0)/32}.$$

We argue that this sketch always has high error in the worst case, and that the problem cannot be fixed, for example, by making $|S_0|, |S_1| = m \gg 64$, or in adjusting the threshold "32," or changing how $\hat{\lambda}(S)$ is computed. Consider the following two sequences:

$$\mathcal{A}_{\text{lo}} = (1, 2, 3, 4, 5, 6, \ldots, \lambda),$$
$$\mathcal{A}_{\text{hi}} = (1, 2, 1, 2, 3, 1, 2, 3, 4, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 6, \ldots, 1, 2, \ldots, \lambda).$$

They each have the same cardinality λ but induce very different distributions in their HyperBitBit sketches. Consider the state of the (S_0, S_1) vectors immediately *after* incrementing L. The Hamming weight of S_0 is typically between 16 and 32. For the sake of argument suppose it is about 20. The Hamming weight of S_1 is zero. In $\mathcal{A}_{\text{lo}}$ it *remains* zero for a while, but in $\mathcal{A}_{\text{hi}}$ all the items seen before are reprocessed immediately, and in expectation, as least half of the 20 items that put 1s in S_0 trigger the setting of 1s in S_1. After the *next* increment of L, the expected Hamming weight of S_0 under $\mathcal{A}_{\text{lo}}$ and $\mathcal{A}_{\text{hi}}$ differ by a constant close to 10, which distorts the estimation by about a $2^{10/32}$ factor.

This description is merely meant to highlight *how* HyperBitBit fails to be commutative or history independent. Fig. 5.11 illustrates that the distribution of $\hat{\lambda}(S)$ is dramatically different after seeing $\mathcal{A}_{\text{lo}}$ and $\mathcal{A}_{\text{hi}}$ when the cardinality is $\lambda = 400,000$. The error of Sedgewick's estimator is usually over 20% and often even higher.

It may be that these large errors can be mitigated with a different estimation function. For example, redefining

$$\hat{\lambda}(S) \propto 2^{L+(\text{HammingWeight}(S_0)-16)/16}$$

may help.[18] However, we argue that no estimator based on the statistic $(L, \text{HammingWeight}(S_0))$ can be very accurate. Fig. 5.12 shows the results of the following experiment. We fix a particular *state* $(12, 31) = (L, \text{HammingWeight}(S_0))$ and see how long

[18]This estimator should have better concentration around the mean for any given input sequence, but it has other strange properties, e.g., that the estimate over time is not guaranteed to be nondecreasing.

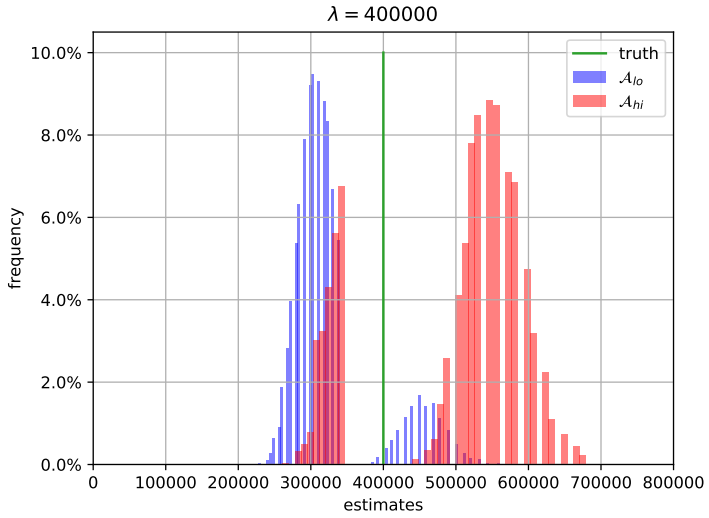


Figure 5.11: HyperBitBit Experiments with cardinality $\lambda = 400000$. We run $10,000$ experiments for each sequence type and use the original HyperBitBit estimator $2^{L+5.4+\text{HammingWeight}(S_0)/32}$ [147]. It turns out that 72.86% of the estimates from $\mathcal{A}_{\text{hi}}$ are at least 20% higher than than the true cardinality and 67.12% of the estimates from $\mathcal{A}_{\text{lo}}$ are at least 20% lower than than the true cardinality.

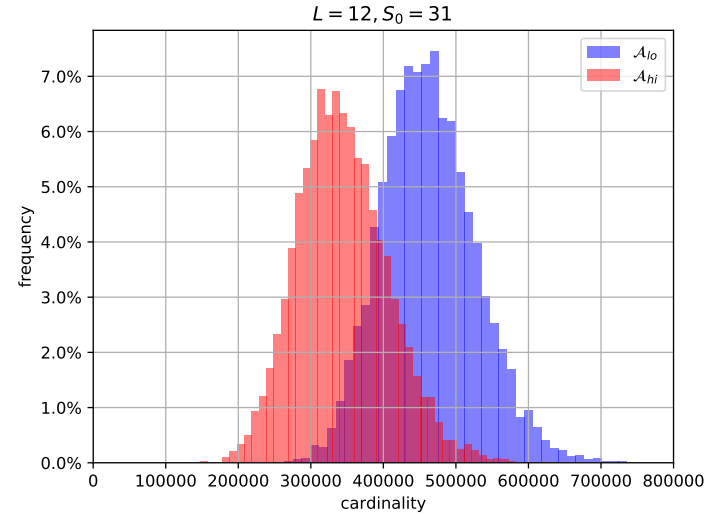


Figure 5.12: HyperBitBit Experiments with terminating condition $(L, \text{HammingWeight}(S_0)) = (12, 31)$. We run $10,000$ experiments for each sequence type and record the cardinality when the terminating condition is reached. On average, $343,928$ distinct insertions from the sequence $\mathcal{A}_{\text{hi}}$ suffice to reach the state $(L, \text{HammingWeight}(S_0)) = (12, 31)$, while $462,514$ distinct insertions from $\mathcal{A}_{\text{lo}}$ are needed to reach $(12, 31)$.

a prefix of $\mathcal{A}_{\text{lo}}$ and $\mathcal{A}_{\text{hi}}$ we need to process until the HyperBitBit sketch agrees with state $(12, 31)$. We plot the cardinality of these prefixes in two different colors. Any estimator based on this statistic that is well-calibrated for one sequence type will incur huge errors on the other sequence type.

5.7 Beyond Fisher-Shannon: the Sequential Setting

In Section 5.3, we analyzed the MVPs of *mergeable* sketches under the assumption that the sketch was compressed to its entropy bound. Fishmonger (an entropy compressed variant of PCSA with a different estimator function) was shown to have MVP $= H_0/I_0 \approx 1.98$. Furthermore, H_0/I_0 was shown to be the minimum MVP among *linearizable* sketches, a subset of *mergeable* sketches that includes all the popular sketches (HyperLogLog, PCSA, Min, etc.).

Our aim in *this* section is to build a useful framework for designing and analyzing *non-mergeable* sketching schemes, and, following Section 5.4, to develop a theory of space-variance optimality in the non-mergeable world. We work in the the random oracle model.

Martingale transform Cohen [49] and Ting [151] independently described how to turn any sketch into a non-mergeable sketch using what we call the Martingale transform. Let S_i be the state of the original sketch after seeing $(a_1, \ldots, a_i)$ and $P_{i+1} = \Pr(S_{i+1} \neq S_i \mid S_i, a_{i+1} \notin \{a_1, \ldots, a_i\})$ be the probability that it changes state upon seeing a *new* element a_{i+1}.[19] The state of the Martingale sketch is $(S_i, \hat{\lambda}_i)$. Upon processing a_{i+1} it becomes $(S_{i+1}, \hat{\lambda}_{i+1})$, where

$$\hat{\lambda}_{i+1} = \hat{\lambda}_i + P_{i+1}^{-1} \cdot \mathbb{1}\left[S_{i+1} \neq S_i\right].$$

Here $\mathbb{1}[\mathcal{E}]$ is the indicator variable for the event $\mathcal{E}$. We assume the original sketch is insensitive to duplicates, so

$$\mathbb{E}(\hat{\lambda}_{i+1}) = \begin{cases} \hat{\lambda}_i & \text{when } a_{i+1} \in \{a_1, \ldots, a_i\} \text{ (and hence } S_{i+1} = S_i) \\ \hat{\lambda}_i + 1 & \text{when } a_{i+1} \notin \{a_1, \ldots, a_i\}. \end{cases}$$

Thus, with $\hat{\lambda}_0 = \lambda_0 = 0$, $\hat{\lambda}_i$ is an unbiased estimator of the true cardinality $\lambda_i = |\{a_1, \ldots, a_i\}|$ and $(\hat{\lambda}_i - \lambda_i)_i$ is a *martingale*. The Martingale-transformed sketch requires the same space, plus just $\log n$ bits to store the estimate $\hat{\lambda}$.

[19] These probabilities are over the choice of $h(a_{i+1})$, which, in the the random oracle model, is independent of all other hash values.

Cohen and Ting [49, 151] both proved that Martingale MinCount has standard error $1/(2\sqrt{m})$ and MVP $= (\log n)/2$. They gave different estimates for the standard error of Martingale LogLog. Ting's estimate is quite accurate, and tends to $\sqrt{\log 2/m}$ as $m \to \infty$, giving it an MVP $= \log 2 \log\log n \approx 0.69 \log\log n$.

Our results are as follows.

- Although the *Martingale transform* itself is simple, analyzing the variance of these sketches is not. For example, Cohen [49] and Ting [151] estimated the standard error of Martingale LogLog to be about $\approx \sqrt{3/(4m)} \approx 0.866/\sqrt{m}$ and about $\approx 1/(2\alpha_m m)$, respectively, where the latter tends to $\sqrt{\log 2/m} \approx 0.8326/\sqrt{m}$ as $m \to \infty$.[20] We give a general method for determining the limiting relative variance of Martingale sketches that is strongly influenced by Ting's perspective.

- What is the most efficient (smallest MVP) non-mergeable sketch for cardinality estimation? The best Martingale sketches perform better than the *ad hoc* non-mergeable S-Bitmap and Recordinality, but perhaps there is a completely different, better way to systematically build non-mergeable sketches. We prove that up to some natural assumptions[21] the best non-mergeable sketch is a Martingale X sketch, for some X. Furthermore, we prove that Martingale Fishmonger, having MVP of $H_0/2 \approx 1.63$, is optimal among all Martingale X sketches, where X is *linearizable*. This provides some circumstantial evidence that Martingale Fishmonger is optimal, and that if we want, say, 1% standard error, we need to use a $H_0/2 \cdot (0.01)^{-2}$-bit sketch, ≈ 2 kilobytes.

- Martingale Fishmonger has an attractive MVP, but it is slow and cumbersome to implement. We propose a new mergeable sketch called Curtain that is "naturally" space efficient and easy to update in $O(1)$ memory accesses, and prove that Martingale Curtain has a limiting MVP ≈ 2.31.

See Table 5.3 for a summary.

Organization In Section 5.7.1 we introduce the Curtain sketch, which is a linearizable (hence mergeable) sketch in the dartboard model. In Section 5.7.2 we prove some general theorems on the bias and asymptotic relative variance of Martingale-type sketches, and in Section 5.7.3 we apply this framework to bound the limiting MVP of Martingale PCSA, Martingale Fishmonger, and Martingale Curtain.

[20] Here $\alpha_m = \left(m \int_0^\infty \left(\log_2\left(\frac{2+u}{1+u}\right)\right)^m du\right)^{-1}$ is the coefficient of Flajolet et al.'s HyperLogLog estimator.

[21] (the sketch is insensitive to duplicates, and the estimator is unbiased)

Mergeable Sketch		Limiting MVP	Notes
PCSA	[83]	$.6 \log n \approx 38.9$	For $n = 2^{64}$
LogLog	[73]	$1.69 \log \log n \approx 10.11$	For $n = 2^{64}$
MinCount	[90, 35, 122]	$\log n = 64$	For $n = 2^{64}$
HyperLogLog	[82]	$1.08 \log \log n \approx 6.48$	For $n = 2^{64}$
Fishmonger	Section 5.5	$H_0/I_0 \approx 1.98$	
Non-Mergeable Sketch			
S-Bitmap	[36]	$O(\log^2(n/m))$	
Recordinality	[94]	$O(\log(\lambda/m) \log n)$	
Martingale PCSA	**new**	$0.35 \log n \approx 22.4$	For $n = 2^{64}$
Martingale LogLog	[49, 151]	$0.69 \log \log n \approx 4.16$	For $n = 2^{64}$
Martingale MinCount	[49, 151]	$0.5 \log n = 32$	For $n = 2^{64}$
Martingale Fishmonger	**new**	$H_0/2 \approx 1.63$	$H_0 = (\log 2)^{-1} + \sum_{k \geq 1} \frac{\log_2(1+1/k)}{k}$
Martingale Curtain	**new**	≈ 2.31	Theorem 5.9 with $(q, a, h) = (2.91, 2, 1)$
Non-Mergeable Lower Bound			
Martingale X	**new**	$\geq H_0/2$	X is a *linearizable* sketch

Table 5.3: Memory-variance products of mergeable and non-mergeable sketches. A selection of results on composable sketches (top) and non-composable Martingale sketches (bottom) in terms of their limiting memory-variance product (MVP). Logarithms are base 2.

In Section 5.7.4 we prove some results on the optimality of the Martingale transform itself, and that Martingale Fishmonger has the lowest variance among those based on linearizable sketches.

Section 5.7.5 presents some experimental findings that demonstrate that the conclusions drawn from the asymptotic analysis of Martingale sketches are extremely accurate in the pre-asymptotic regime as well, and that Martingale Curtain has lower variance than Martingale LogLog.

5.7.1 The Curtain Sketch

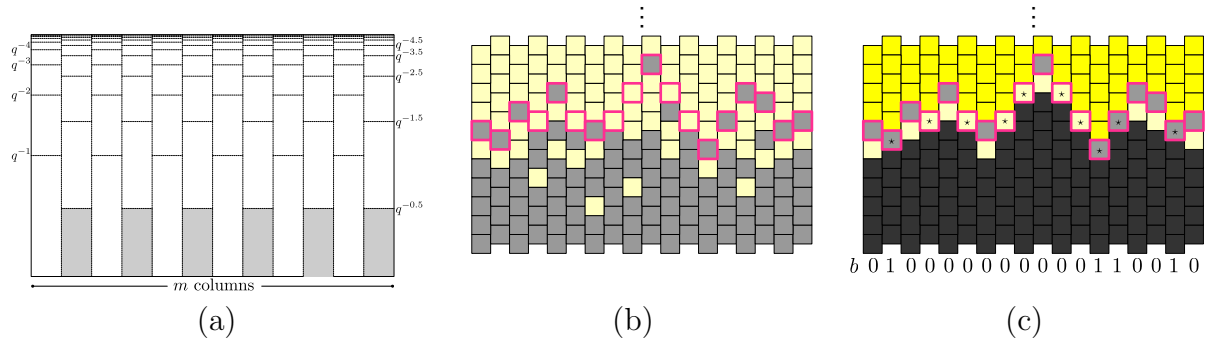
Design Philosophy. Our goal is to strike a nice balance between the simplicity and time-efficiency of (Hyper)LogLog, and the superior information-theoretic efficiency of PCSA, which can only be fully realized under extreme (and time-inefficient) compression to its entropy bound [137, 117]. Informally, if we are dedicating at least 1 bit to encode the status of a cell, the *best* cells to encode have mass $\Theta(\lambda^{-1})$ and we should design a sketch that maximizes the number of such cells encoded.

We assume the dartboard is partitioned into m columns; define $\text{Cell}(j, i)$ to be the cell in column i covering the vertical interval $[q^{-(j+1)}, q^{-j})$. In a PCSA sketch, the occupied cells

are precisely those with at least one dart. In LogLog, the occupied cells in each column are contiguous, extending to the highest cell containing a dart. See Fig. 5.8.

Consider the vector $v = (g_0, g_1, \ldots, g_{m-1})$ where $\text{Cell}(g_i, i)$ is the highest occupied cell in LogLog/PCSA. The *curtain* of v w.r.t. allowable offsets $\mathscr{O}$ is a vector $v_{\text{curt}} = (\hat{g}_0, \hat{g}_1, \ldots, \hat{g}_{m-1})$ such that (i) $\forall i \in [1, m-1].\, \hat{g}_i - \hat{g}_{i-1} \in \mathscr{O}$, and (ii) v_{curt} is the minimal such vector dominating v, i.e., $\forall i.\, \hat{g}_i \geq g_i$. Although we have described v_{curt} as a function of v, it is clearly possible to maintain v_{curt} as darts are thrown, without knowing v.

We have an interest in $|\mathscr{O}|$ being a power of 2 so that curtain vectors may be encoded efficiently, as a series of offsets. On the other hand, it is most efficient if $\mathscr{O}$ is symmetric around zero. For these reasons, we use a base-q "sawtooth" cell partition of the dartboard; see Fig. 5.13. Henceforth $\text{Cell}(j, i)$ is defined as usual, except j is an integer when i is even and a half-integer when i is odd. Then the allowable offsets are $\mathscr{O}_a = \{-(a - 1/2), -(a - 3/2), \ldots, -1/2, 1/2, \ldots, a - 3/2, a - 1/2\}$, for some a that is a power of 2.



(a) (b) (c)

Figure 5.13: Visualization of Curtain. (a) The base-q "sawtooth" cell partition. (b) and (c) depict a Curtain sketch w.r.t. $\mathscr{O} = \{-3/2, -1/2, 1/2, 3/2\}$ and $h = 1$. (b) Gray cells contain at least one dart; light yellow cells contain none. The curtain $v_{\text{curt}} = (\hat{g}_i)$ is highlighted with a pink boundary. (c) Columns that are in *tension* have a $\star$ in their curtain cell. All *dark* gray cells are occupied and all *dark* yellow cells are free according to Rule 3. All other cells are occupied/free (light gray, light yellow) according to Rules 1 and 2.

Let $\text{Cell}(g_i, i)$ is the highest cell containing a dart in column i in the *sawtooth* cell partition and $v_{\text{curt}} = (\hat{g}_i)$ be the curtain vector of $v = (g_i)$ w.r.t. offsets $\mathscr{O} = \mathscr{O}_a$. We say column i is *in tension* if $(\cdots, \hat{g}_{i-1}, \hat{g}_i - 1, \hat{g}_{i+1}, \cdots)$ is not a valid curtain, i.e., if $\hat{g}_i - \hat{g}_{i-1} = \min(\mathscr{O})$ or $\hat{g}_{i+1} - \hat{g}_i = \max(\mathscr{O})$. In particular, if column i is *not* in tension, then $\text{Cell}(\hat{g}_i, i)$ must contain at least one dart, for if it contained no darts the curtain would be dropped to $\hat{g}_i - 1$ at column i. However, if column i *is* in tension, then $\text{Cell}(\hat{g}_i, i)$ might not contain a dart.

The Curtain sketch encodes $v_{\text{curt}} = (\hat{g}_i)$ w.r.t. the base-q sawtooth cell partition and offsets $\mathscr{O}_a$, and a bit-array $b = \{0, 1\}^{h \times m}$. This sketch designates each cell *occupied* or *free* as follows.

Rule 1. If column i is not in tension then $\text{Cell}(\hat{g}_i, i)$ is occupied, and $b(\cdot, i)$ encodes the status of the h cells below the curtain, i.e., $\text{Cell}(\hat{g}_i - (j+1), i)$ is occupied iff $b(j, i) = 1$, $j \in \{0, \ldots, h-1\}$.

Rule 2. If column i is in tension, then $\text{Cell}(\hat{g}_i - j, i)$ is occupied iff $b(j, i) = 1$, $j \in \{0, \ldots, h-1\}$.

Rule 3. Every cell above the curtain is free ($\text{Cell}(\hat{g}_i + j, i)$, when $j \geq 1$) and all remaining cells are occupied.

Fig. 5.13 gives an example of a Curtain sketch, with $\mathscr{O} = \{-3/2, -1/2, 1/2, 3/2\}$ and $h = 1$. (The base q of the cell partition is unspecified in this example.)

Theorem 5.9. *Consider the Martingale Curtain sketch with parameters q, a, h (base q, $\mathscr{O}_a = \{-(a-1/2), \ldots, a-1/2\}$, and $b \in \{0,1\}^{h \times m}$), and let $\hat{\lambda}$ be its estimate of the true cardinality λ.*

1. *$\hat{\lambda}$ is an unbiased estimate of λ.*
2. *The relative variance of $\hat{\lambda}$ is:*

$$\frac{1}{\lambda^2}\mathbb{V}(\hat{\lambda} \mid \lambda) = \frac{(1 + o_{\lambda/m}(1) + o_m(1))q \log q}{2m(q-1)} \left(\frac{q-1}{q} + \frac{2}{q^h(q^{a-1/2} - 1)} + \frac{1}{q^{h+1}} \right),$$

As a result, the limiting MVP of Martingale Curtain is

$$\mathit{MVP} = (\log_2(2a) + h) \times \frac{q \log q}{2(q-1)} \left(\frac{q-1}{q} + \frac{2}{q^h(q^{a-1/2} - 1)} + \frac{1}{q^{h+1}} \right).$$

Proof. Follows from Theorems 5.12 and 5.15. □

Here $o_{\lambda/m}(1)$ and $o_m(1)$ are terms that go to zero as m and λ/m get large. Recall that for practical reasons we want to parameterize Theorem 5.9 with a a power of 2 and h an integer, but it is realistic to set $q > 1$ to be any real. Given these constraints, the optimal setting is $q = 2.91$, $a = 2$, and $h = 1$, exactly as in the example in Fig. 5.13. This uses $\log\log n + 3(m-1)$ bits to store the sketch proper, $\log n$ bits[22] to store $\hat{\lambda}$, and achieves a limiting MVP ≈ 2.31. In other words, to achieve a standard error $1/\sqrt{b}$, we need about $2.31b$ bits.

[22]It is fine to store an approximation $\tilde{\lambda}$ of $\hat{\lambda}$ with $O(\log m)$ bits of precision.

Implementation Considerations. We encode a curtain $(\hat{g}_0, \hat{g}_1, \ldots, \hat{g}_{m-1})$ as $\hat{g}_0$ and an offset vector $(o_1, o_2, \ldots, o_{m-1})$, $o_i = \hat{g}_i - \hat{g}_{i-1}$, where $\hat{g}_0$ takes $\log_2 \log_q n \leq 6$ bits and o_i takes $\log_2 |\mathscr{O}| = \log_2(2a)$ bits. Clearly, to evaluate $\hat{g}_i$ we need to compute the prefix sum $\hat{g}_0 + \sum_{i' \leq i} o_{i'}$.

Lemma 5.22. *Let $(x_0, \ldots, x_{\ell-1})$ be a vector of t-bit unsigned integers packed into $\lceil t\ell/w \rceil$ words, where each word has $w = \Omega(\log(t\ell))$ bits. The prefix sum $\sum_{j \in [0,i]} x_j$ can be evaluated in $O(t\ell/w + \log w)$ time.*

Proof. W.l.o.g. we can assume $i = \ell - 1$, so the task is to sum the entire list. In $O(\lceil (t\ell)/w \rceil)$ time we can halve the number of summands, by masking out the odd and even summands and adding these vectors together. After halving twice in this way, we have a vector of $\ell/4$ $(t+2)$-bit integers, each allocated $4t$ bits. At this point we can halve the number of words by adding the $(2i+1)$th word to the $2i$th word. Thus, if $T_w(\ell, t)$ is the time needed to solve this problem, $T_w(\ell, t) = T_w(\ell/8, 4t) + O(\lceil (t\ell)/w \rceil)$, which is $O((t\ell)/w + \log w)$. □

In our context $t = \log_2(2a) = 2$, so even if m is a medium-size constant, say at most 256 or 512, we only have to do prefix sums over 8 or 16 consecutive 64-bit words. If m is much larger then it would be prudent to partition the dartboard into m/c independent curtains, each with $c = 256$ or 512 columns. This keeps the update time independent of m and increases the space overhead negligibly.

The SecondCurtain Sketch The SecondCurtain sketch is a variable-length sketch that lies between Curtain and Fishmonger, both in terms of conceptual complexity and MVP. It is based on the observation that the positions of the *second* highest cell hit by a dart in each column exhibits considerably less variation than the highest.

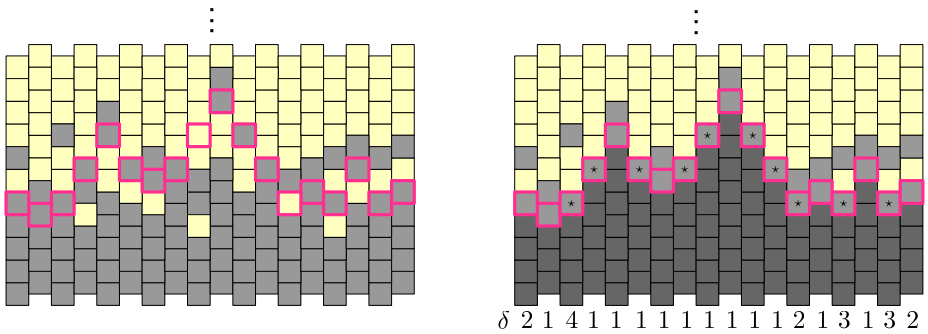
Once again, consider a sawtooth cell partition. Initially all cells of the form $\mathrm{Cell}(j, \cdot)$, $j \in \{0, 1/2, 1, 3/2\}$ are regarded as containing darts. Define $v = (g_i)$ to be such that $\mathrm{Cell}(g_i, i)$ is the *second* highest cell in column i containing a dart, and let $v_{\mathrm{curt}} = (\hat{g}_i)$ be the curtain of v w.r.t. $\mathscr{O}_a = \{-(a - 1/2), \ldots, a - 1/2\}$.

Observe that if column i is *not* in tension, then it must be the case that some $\mathrm{Cell}(\hat{g}_i + j, i)$ with $j \geq 1$ contains at least one dart, for otherwise the curtain height $\hat{g}_i$ at column i should be lower. On the other hand, if column i *is* in tension, then it may be that for all $j \geq 1$, $\mathrm{Cell}(\hat{g}_i + j, i)$ contains no dart. The SecondCurtain sketch consists of $v_{\mathrm{curt}} = (\hat{g}_i)$, and a variable-length encoding of $\delta = (\delta_0, \ldots, \delta_{m-1})$, where each δ_i encodes a positive integer in *unary*, i.e., 1=1, 2=01, 3=001, and so on. The interpretation of the sketch is as follows.

Rule 1. If column i is *not* in tension then $\mathrm{Cell}(j,i)$ is occupied iff $j \le \hat{g}_i$ or $j = \hat{g}_i + \delta_i$.

Rule 2. If column i is in tension, then $\mathrm{Cell}(j,i)$ is occupied iff $j \le \hat{g}_i$ or $j = \hat{g}_i + (\delta_i - 1)$. I.e., $\delta_i = 1$ if all darts in column i are in cells at or below $\mathrm{Cell}(\hat{g}_i, i)$.

Fig. 5.14 gives an example of a SecondCurtain sketch.



(a) (b)

Figure 5.14: Visualization of SecondCurtain. (a) Gray cells contain at least one dart; yellow cells contain none. The second curtain $v_{\mathrm{curt}} = (\hat{g}_i)$ is indicated by a pink boundary. (b) Columns that are in tension are marked with a $\star$. The δ-vector is indicated below.

Theorem 5.10. *Consider the Martingale SecondCurtain sketch with parameters q, a, and let $\hat{\lambda}$ be its estimate of the true cardinality λ.*

1. *$\hat{\lambda}$ is an unbiased estimator of λ.*
2. *Define $\varphi(x) = \sum_{i=0}^{\infty} \left(e^{\frac{x}{q^i}\frac{q-1}{q}} - 1 \right)$. The relative variance $\frac{1}{\lambda^2}\mathbb{V}(\hat{\lambda} \mid \lambda)$ is*

$$\frac{(1 + o_{\lambda/m}(1) + o_m(1))}{2m} \left(\int_{-\infty}^{\infty} e^{-\frac{1}{q^t}\frac{q^{a-1/2}+1}{q^{a-1/2}-1}} \left(1 + \varphi\left(\frac{1}{q^{t+1}}\right)\right) \prod_{j=1}^{\infty} \left(1 + \varphi\left(\frac{1}{q^{t+j(a-1/2)}}\right)\right)^2 \frac{1}{q^t}\frac{q-1}{q}\, dt \right)^{-1}.$$

The memory (in bits) used by the sketch is a random variable, having expectation

$$(1 + o_{\lambda/m}(1) + o_m(1))m \left(\log_2(2a) + 1 + \int_{-\infty}^{\infty} e^{-\frac{\lambda}{q^t}\frac{2}{q^{a-1/2}-1}} \prod_{j=1}^{\infty} \left(1 + \varphi\left(\frac{\lambda}{q^{t+j(a-1/2)}}\right)\right)^2 \left(e^{-\frac{\lambda}{q^t}} \varphi\left(\frac{\lambda}{q^t}\right) - \left(e^{-\frac{\lambda}{q^{t+1}}} - e^{-\frac{\lambda}{q^t}}\right) \varphi\left(\frac{\lambda}{q^{t+1}}\right) \right) dt \right).$$

Proof. Follows from Theorem 5.12, 5.16 and 5.17. □

The best parameterization of Theorem 5.10 uses $q = 2.2$ and $a = 2$, with a limiting MVP ≈ 2.06.

5.7.2 Foundations of the Martingale Transform

In this subsection we present a simple framework for analyzing the limiting variance of Martingale sketches, which is strongly influenced by Ting's [151] work. Theorem 5.11 gives simple unbiased estimators for the cardinality and the variance of the the cardinality estimator. The upshot of Theorem 5.11 is that to analyze the variance of the estimator, we only need to bound $\mathbb{E}(P_k^{-1})$, where P_k is the probability the kth distinct element changes the sketch. Theorem 5.12 further shows that for sketches composed of m subsketches (like Curtain, HyperLogLog, and PCSA), the limiting variance tends to $\frac{1}{2\kappa m}$, where κ is a constant that depends on the sketch scheme. Section 5.7.3 analyzes the constant κ for each of PCSA, LogLog, and Curtain. Using results of [137] on the entropy of PCSA we can calculate the limiting MVP of PCSA, LogLog, Curtain, and Fishmonger.

Martingale Estimators and Retrospective Variance Consider an arbitrary sketch with state space $\mathcal{S}$. We assume the sketch state does not change upon seeing duplicated elements, hence it suffices to consider streams of *distinct* elements. We model the evolution of the sketch as a Markov chain $(S_k)_{k\geq 0} \in \mathcal{S}^*$, where S_k is the state after seeing k *distinct* elements. Define $P_k = \Pr(S_k \neq S_{k-1} \mid S_{k-1})$ to be the *state changing probability*, which depends only on S_{k-1}. In the dartboard terminology P_k is the total size of all unoccupied cells in S_{k-1}.

Definition 5.14. Let $\mathbb{1}[\mathcal{E}]$ be the indicator variable for event $\mathcal{E}$. For any $\lambda \geq 0$, define:

$$E_\lambda = \sum_{k=1}^{\lambda} \mathbb{1}[S_k \neq S_{k-1}] \cdot \frac{1}{P_k}, \qquad \text{the martingale estimator,}$$

$$\text{and } V_\lambda = \sum_{k=1}^{\lambda} \mathbb{1}[S_k \neq S_{k-1}] \cdot \frac{1-P_k}{P_k^2}, \qquad \text{the "retrospective" variance.}$$

Note that $E_0 = V_0 = 0$.

The Martingale transform of this sketch stores $\hat{\lambda} = E_\lambda$ in one machine word and returns it as a cardinality estimate. It can also store V_λ in one machine word as well. Theorem 5.11

shows that the retrospective variance V_λ is a good running estimate of the empirical squared error $(E_\lambda - \lambda)^2$.

Theorem 5.11. *The martingale estimator E_λ is an unbiased estimator of λ and the retrospective variance V_λ is an unbiased estimator of $\mathbb{V}(E_\lambda)$. Specifically, we have,*

$$\mathbb{E}(E_\lambda) = \lambda, \text{ and } \mathbb{V}(E_\lambda) = \mathbb{E}(V_\lambda) = \sum_{k=1}^{\lambda} \mathbb{E}\left(\frac{1}{P_k}\right) - \lambda.$$

Remark 5.8. Theorem 5.11 contradicts Ting's claim [151], that V_λ is unbiased *only at "jump" times*, i.e., those λ for which $S_\lambda \neq S_{\lambda-1}$, and therefore inadequate to estimate the variance. In order to correct for this, Ting introduced a Bayesian method for estimating the time that has passed since the last jump time. The reason for thinking that jump times are different is actually quite natural. Suppose we record the list of <u>*distinct*</u> states $s_0, \ldots, s_k$ encountered while inserting λ elements, λ being unknown, and let p_i be the probability of changing from s_i to some other state. The amount of time spent in state s_i is a geometric random variable with mean p_i^{-1} and variance $(1-p_i)/p_i^2$. Furthermore, these waiting times are independent. Thus, $\sum_{i\in[0,k)} p_i^{-1}$ and $\sum_{i\in[0,k)}(1-p_i^{-1})/p_i^2$ are unbiased estimates of the cardinality λ' and squared error *upon entering state* s_k. These exactly correspond to E_λ and V_λ, but they *should* be biased since they do not take into account the $\lambda - \lambda'$ elements that had no effect on s_k. As Theorem 5.11 shows, this is a mathematical optical illusion. The history is a random variable, and although the last $\lambda - \lambda'$ elements did not change the state, *they could have*, which would have altered the observed history $s_0, \ldots, s_k$ and hence the estimates E_λ and V_λ.

Proof of Theorem 5.11. Note that P_k is a function of S_{k-1}. By the linearity of expectation and the law of total expectation, we have

$$\begin{aligned}\mathbb{E}(E_k) &= \mathbb{E}(\mathbb{E}(E_k \mid S_{k-1})) = \mathbb{E}\left(\mathbb{E}(E_{k-1} \mid S_{k-1}) + \mathbb{E}\left(\mathbb{1}\left[S_k \neq S_{k-1}\right] \cdot \frac{1}{P_k} \;\middle|\; S_{k-1}\right)\right) \\ &= \mathbb{E}(E_{k-1}) + 1 = \mathbb{E}(E_{k-2}) + 2 = \ldots = \mathbb{E}(E_0) + k = k.\end{aligned}$$

and

$$\begin{aligned}\mathbb{E}(V_k) &= \mathbb{E}(\mathbb{E}(V_k \mid S_{k-1})) = \mathbb{E}\left(\mathbb{E}(V_{k-1} \mid S_{k-1}) + \mathbb{E}\left(\mathbb{1}\left[S_k \neq S_{k-1}\right] \cdot \frac{1-P_k}{P_k^2} \;\middle|\; S_{k-1}\right)\right) \\ &= \mathbb{E}(V_{k-1}) + \mathbb{E}\left(\frac{1-P_k}{P_k}\right) = \mathbb{E}(V_{k-2}) + \mathbb{E}\left(\frac{1-P_k}{P_k}\right) + \mathbb{E}\left(\frac{1-P_{k-1}}{P_{k-1}}\right) = \ldots \\ &= \mathbb{E}(V_0) + \sum_{i=1}^{k} \mathbb{E}\left(\frac{1-P_i}{P_i}\right) = \sum_{i=1}^{k} \mathbb{E}\left(\frac{1}{P_i}\right) - k.\end{aligned}$$

For the variance, we have

$$\mathbb{V}(E_\lambda) = \mathbb{E}(E_\lambda^2) - (\mathbb{E}(E_\lambda))^2 = \mathbb{E}(E_\lambda^2) - \lambda^2.$$

Note that

$$\begin{aligned}
\mathbb{E}(E_k^2 \mid S_{k-1}) &= \mathbb{E}\left(\left(E_{k-1} + \mathbb{1}\left[S_k \neq S_{k-1}\right] \cdot \frac{1}{P_k}\right)^2 \,\middle|\, S_{k-1}\right) \\
&= E_{k-1}^2 + 2\frac{E_{k-1}}{P_k} \cdot \mathbb{E}\left(\mathbb{1}\left[S_k \neq S_{k-1}\right] \mid S_{k-1}\right) + \frac{1}{P_k^2} \cdot \mathbb{E}\left(\mathbb{1}\left[S_k \neq S_{k-1}\right]^2 \mid S_{k-1}\right) \\
&= E_{k-1}^2 + 2E_{k-1} + \frac{1}{P_k}.
\end{aligned}$$

Then by the law of total expectation and the linearity of expectation, we have

$$\begin{aligned}
\mathbb{E}\left(E_k^2\right) &= \mathbb{E}\left(\mathbb{E}\left(E_k^2 \mid S_{k-1}\right)\right) = \mathbb{E}\left(E_{k-1}^2 + 2E_{k-1} + \frac{1}{P_k}\right) \\
&= \mathbb{E}\left(E_{k-1}^2\right) + 2(k-1) + \mathbb{E}\left(\frac{1}{P_k}\right).
\end{aligned}$$

From this recurrence relation, we have

$$\begin{aligned}
\mathbb{E}\left(E_\lambda^2\right) &= \mathbb{E}\left(E_0^2\right) + 2\sum_{k=1}^{\lambda}(k-1) + \sum_{k=1}^{\lambda}\mathbb{E}\left(\frac{1}{P_k}\right) \\
&= \sum_{k=1}^{\lambda}\mathbb{E}\left(\frac{1}{P_k}\right) + \lambda(\lambda-1).
\end{aligned}$$

We conclude that

$$\mathbb{V}(E_\lambda) = \sum_{k=1}^{\lambda}\mathbb{E}\left(\frac{1}{P_k}\right) + \lambda(\lambda-1) - \lambda^2 = \sum_{k=1}^{\lambda}\mathbb{E}\left(\frac{1}{P_k}\right) - \lambda = \mathbb{E}(V_\lambda).$$

□

Asymptotic Relative Variance

The ARV Factor We consider classes of sketches composed of m *subsketches*, which controls the size and variance. In LogLog, PCSA, and Curtain these subsketches are the m columns. When considering a sketch with m subsketches, instead of using λ as the total

number of insertions, we always use λ to denote the number of insertions *per subsketch* and therefore the total number of insertions is λm. We care about the *asymptotic relative variance* (ARV) as m and λ both go to infinity (defined below). A reasonable sketch should have relative variance $O(1/m)$. Informally, the ARV factor is just the leading constant of this expression.

Definition 5.15 (ARV factor)**.** Consider a class of sketches whose size is parameterized by m. For any $k \geq 0$, define $P_{m,k}$ to be the probability the sketch changes state upon the kth insertion and $E_{m,k}$ the martingale estimator. The *ARV factor* of this class of sketches is defined as

$$\lim_{\lambda \to \infty} \lim_{m \to \infty} m \cdot \frac{\mathbb{V}(E_{m,\lambda m})}{(\lambda m)^2}. \tag{5.11}$$

Scale-Invariance and the Constant κ Few sketches have *strictly* well-defined ARV factors. In Martingale LogLog, for example, the quantity $\left(\lim_{m\to\infty} m\frac{\mathbb{V}(E_{m,\lambda m})}{(\lambda m)^2}\right)$ is not constant, but periodic in $\log_2 \lambda$; it does not converge as $\lambda \to \infty$. We explain how to fix this issue using *smoothing* in Section 5.7.2. *Scale-invariant* sketches must have well-defined ARV factors.

Definition 5.16 (scale-invariance and constant κ)**.** A combined sketch is *scale-invariant* if

1. For any λ, there exists a constant κ_λ such that $\lambda \cdot P_{m,\lambda m}$ converges to κ_λ almost surely as $m \to \infty$.
2. The limit of κ_λ as $\lambda \to \infty$ exists, and $\kappa \stackrel{\text{def}}{=} \lim_{\lambda\to\infty} \kappa_\lambda$.

The constant of a sketch A is denoted as κ_A, where the subscript A is often dropped when the context is clear.

The next theorem proves that under mild regularity conditions, all scale-invariant sketches have well defined ARV factors and there is a direct relation between the ARV factor and the constant κ.

Theorem 5.12 (ARV factor of a scale-invariant sketch)**.** *Consider a sketching scheme satisfying the following properties.*

1. *It is scale-invariant with constant κ.*
2. *For any $\lambda > 0$, the limit operator and the expectation operator of $\{\frac{1}{P_{m,\lambda m}}\}_m$ can be interchanged.*

Then the ARV factor of the sketch exists and equals $\frac{1}{2\kappa}$.

Proof. First note that, by the assumptions, we have that

$$\lim_{m\to\infty} \mathbb{E}\left(\frac{1}{P_{m,\lambda m}}\right) = \mathbb{E}\left(\lim_{m\to\infty} \frac{1}{P_{m,\lambda m}}\right) = \mathbb{E}\left(\frac{\lambda}{\kappa_\lambda}\right) = \frac{\lambda}{\kappa_\lambda}.$$

Also note that since $P_{m,k}$ are non-increasing as k increases, by simple coupling argument, we see that for any $k \le k'$, $\mathbb{E}\left(1/P_{m,k}\right) \le \mathbb{E}\left(1/P_{m,k'}\right)$ and $\frac{\kappa_k}{k} \ge \frac{\kappa_{k'}}{k'}$.

Fix $\lambda > 0$, we have, by Theorem 5.11,

$$\begin{aligned}\lim_{m\to\infty} \frac{1}{\lambda^2 m}\mathbb{V}(E_{m,\lambda m}) &= \lim_{m\to\infty}\left(\frac{1}{\lambda^2 m}\sum_{k=1}^{\lambda m}\mathbb{E}\left(\frac{1}{P_{m,k}}\right) - \frac{1}{\lambda}\right)\\ &= \lim_{m\to\infty}\frac{1}{\lambda^2 m}\sum_{i=0}^{\lambda-1}\sum_{j=1}^{m}\mathbb{E}\left(\frac{1}{P_{m,im+j}}\right) - \frac{1}{\lambda} \qquad (5.12)\end{aligned}$$

Since for any $j \in [1, m]$, $\mathbb{E}\left(\frac{1}{P_{m,im+j}}\right) \le \mathbb{E}\left(\frac{1}{P_{m,(i+1)m}}\right)$, we have

$$\begin{aligned}\lim_{m\to\infty} \frac{1}{\lambda^2 m}\mathbb{V}(E_{m,\lambda m}) &\le \lim_{m\to\infty}\frac{1}{\lambda^2 m}\sum_{i=0}^{\lambda-1}\sum_{j=1}^{m}\mathbb{E}\left(\frac{1}{P_{m,(i+1)m}}\right) - \frac{1}{\lambda}\\ &= \frac{1}{\lambda^2}\sum_{i=0}^{\lambda-1}\lim_{m\to\infty}\mathbb{E}\left(\frac{1}{P_{m,(i+1)m}}\right) - \frac{1}{\lambda}\\ &= \frac{1}{\lambda^2}\sum_{i=0}^{\lambda-1}\frac{i+1}{\kappa_{i+1}} - \frac{1}{\lambda},\end{aligned}$$

Denote the ARV factor as v. Fix $W > 0$. Note that for any $i \in [0, \lambda/W - 1]$, $\frac{k\lambda/W+i+1}{\kappa_{k\lambda/W+i+1}} \le \frac{(k+1)\lambda/W}{\kappa_{(k+1)\lambda/W}}$.

$$\begin{aligned}v &\le \lim_{\lambda\to\infty}\left(\frac{1}{\lambda^2}\sum_{i=0}^{\lambda-1}\frac{i+1}{\kappa_{i+1}} - \frac{1}{\lambda}\right) = \lim_{\lambda\to\infty}\left(\frac{1}{\lambda^2}\sum_{k=0}^{W-1}\sum_{i=0}^{\lambda/W-1}\frac{k\lambda/W+i+1}{\kappa_{k\lambda/W+i+1}}\right)\\ &\le \lim_{\lambda\to\infty}\left(\frac{1}{\lambda^2}\sum_{k=0}^{W-1}\sum_{i=0}^{\lambda/W-1}\frac{(k+1)\lambda/W}{\kappa_{(k+1)\lambda/W}}\right) = \frac{1}{W^2}\sum_{k=0}^{W-1}\lim_{\lambda\to\infty}\frac{k+1}{\kappa_{(k+1)\lambda/W}}\end{aligned}$$

note that $\lim_{\lambda\to\infty} \kappa_{(k+1)\lambda/W} = \kappa$ by the definition of scale-invariance,

$$= \frac{1}{W^2} \sum_{k=0}^{W-1} \frac{k+1}{\kappa} = \frac{1}{2\kappa} \frac{W(W+1)}{W^2}. \tag{5.13}$$

On the other hand, we can bound it from below similarly. We will only outline the key steps since it is almost identical to the previous one. Note that for any $j \in [1, m]$, $\mathbb{E}\left(\frac{1}{P_{m,im+j}}\right) \geq \mathbb{E}\left(\frac{1}{P_{m,im}}\right)$. Using this inequality in (5.12), we have

$$\lim_{m\to\infty} \frac{1}{\lambda^2 m} \mathbb{V}(E_{m,\lambda m}) \geq \lim_{m\to\infty} \frac{1}{\lambda^2 m} \sum_{i=0}^{\lambda-1} \sum_{j=1}^{m} \mathbb{E}\left(\frac{1}{P_{m,im}}\right) - \frac{1}{\lambda} = \frac{1}{\lambda^2} \sum_{i=0}^{\lambda-1} \frac{i}{\kappa_i} - \frac{1}{\lambda}.$$

Similarly, we have

$$\begin{aligned} v &\geq \lim_{\lambda\to\infty} \left(\frac{1}{\lambda^2} \sum_{k=0}^{W-1} \sum_{i=0}^{\lambda/(W)-1} \frac{k\lambda/W + i}{\kappa_{k\lambda/W+i}} \right) \geq \lim_{\lambda\to\infty} \left(\frac{1}{\lambda^2} \sum_{k=0}^{W-1} \sum_{i=0}^{\lambda/W-1} \frac{k\lambda/W}{\kappa_{k\lambda/W}} \right) \\ &= \frac{1}{W^2} \sum_{k=0}^{W-1} \frac{k}{\kappa} = \frac{1}{2\kappa} \frac{W(W-1)}{W^2}. \end{aligned} \tag{5.14}$$

Thus by combining (5.13) and (5.14), we have

$$\frac{1}{2\kappa} \frac{W(W-1)}{W^2} \leq v \leq \frac{1}{2\kappa} \frac{W(W+1)}{W^2}.$$

Since the choice of W is arbitrary, we conclude that the ARV factor v is well-defined and $v = \frac{1}{2\kappa}$. □

The constant κ together with Theorem 5.12 is useful in that it gives a simple and systematic way to evaluate the asymptotic performance of a well behaved (scale-invariant) sketch scheme.

MinCount [90, 35, 122] is an example of a scale-invariant sketch. The function $h(a) = (i, v) \in [m] \times [0, 1]$ is interpreted as a pair containing a bucket index and a real hash value. A (k, m)-MinCount sketch stores the smallest k hash values in each bucket.

Theorem 5.13. *(k, m)-MinCount is scale-invariant and $\kappa_{(k,m)\text{-MinCount}} = k$.*

Proof. When a total of λm elements are inserted to the combined sketch, each subsketch receives $(1 + o(1))\lambda$ elements as $\lambda \to \infty$. Since we only care the asymptotic behavior, we

assume for simplicity that each subsketch receives exactly λ elements.

Let $P_\lambda^{(i)}$ be the probability that the sketch of the ith bucket changes after the λth element is thrown into the ith bucket. Then by definition, we have

$$P_{m,\lambda m} = \frac{\sum_{i=1}^{m} P_\lambda^{(i)}}{m}.$$

Since all the subsketches are i.i.d., by the law of large numbers, $\lambda \cdot P_{m,\lambda} \to \lambda \cdot \mathbb{E}\left(P_\lambda^{(1)}\right)$ almost surely as $m \to \infty$.

Let X be the kth smallest hash value among λ uniformly random numbers in $[0, 1]$, which distributes identically with $P_\lambda^{(1)}$. By standard order statistics, X is a Beta random variable $\text{Beta}(k, \lambda - 1 + k)$ which has mean $\frac{k}{\lambda+1}$. Thus $\kappa_\lambda = \lambda \cdot \mathbb{E}(X) = \frac{k\lambda}{\lambda+1}$. We conclude that

$$\kappa = \lim_{\lambda\to\infty} \kappa_\lambda = \lim_{\lambda\to\infty} \frac{k\lambda}{\lambda + 1} = k.$$

□

Applying Theorem 5.12 to (k, m)-MinCount, we see its ARV is $\frac{1}{2km}$,[23] matching Cohen [49] and Ting [151]. Technically its MVP is unbounded since hash values were real numbers, but any realistic implementation would store them to $\log n$ bits of precision, for a total of $km \log n$ bits. Hence we regard its MVP to be $\frac{1}{2} \cdot \log_2 n$.

Smoothing Discrete Sketches Sketches that partition the dartboard in some exponential fashion with base q (like LogLog, PCSA, and Curtain) have the property that their estimates and variance are periodic in $\log_q \lambda$. In Section 5.7.2, I presented a simple method to *smooth* these sketches and make them truly scale-invariant as $m \to \infty$.

We assume that the dartboard is partitioned into m columns. The base-q *smoothing* operation uses an *offset vector* $\vec{r} = (r_0, \ldots, r_{m-1})$. We scale down all the cells in column i by the factor q^{-r_i}, then add a dummy cell spanning $[q^{-r_i}, 1)$ which is always occupied. (Phrased algorithmically, if a dart is destined for column i, we filter it out with probability $1-q^{-r_i}$ and insert it into the sketch with probability q^{-r_i}.) When analyzing variants of (Hyper)LogLog and PCSA, we use the uniform offset vector $(0, 1/m, 2/m, \ldots, (m-1)/m)$. The Curtain sketch can be viewed as having a built-in offset vector of $(0, 1/2, 0, 1/2, 0, 1/2, \ldots)$ which effects the

[23] For simplicity, we assume the second condition of Theorem 4 holds for all the sketches analyzed in this section.

"sawtooth" cell partition. To smooth it, we use the offset vector[24]

$$(0,\ 1/2,\ 1/m,\ 1/2+1/m,\ 2/m,\ 1/2+2/m,\ \ldots,\ 1/2-1/m,\ 1-1/m).$$

As $m \to \infty$, $\vec{r}$ becomes uniformly dense in $[0,1]$.

The smoothing technique makes the empirical estimation more scale-invariant (see [137, Figs. 1& 2]) but also makes the sketch theoretically scale-invariant according to Definition 5.16. Thus, in the analysis, we will always assume the sketches are smoothed. However, in practice it is probably not necessary to do smoothing if $q < 3$.

In the next subsection, we will prove that *smoothed* q-LogLog, q-PCSA, and Curtain are all scale-invariant.

5.7.3 Analysis of Dartboard Based Sketches

Consider a dartboard cell that covers the vertical interval $[q^{-(t+1)}, q^{-t})$. We define the *height* of the cell to be t. In a smoothed cell partition, no two cells have the same height and all heights are of the form $t = j/m$, for some integer j. Thus, we may refer to it unambiguously as *cell* t. Note that cell t is an $m^{-1} \times \frac{1}{q^t}\frac{q-1}{q}$ rectangle.

Poissonized Dartboard Since we care about the asymptotic case where $\lambda \to \infty$, we model the process of "throwing darts" by a Poisson point process on the dart board (similar to the "poissonization" in the analysis of HyperLogLog [82]). Specifically, after throwing λm darts (events) to the dartboard, we assume the number of darts in cell t is a Poisson random variable with mean $\lambda\frac{1}{q^t}\frac{q-1}{q}$ and the number of darts in different cells are independent. For the poissonized dartboard, the range of height of cells naturally extend to the whole set of real numbers, instead of just having cells with positive height.

For any $t \in \mathbb{R}$, let $Y_{t,\lambda}$ be the indicator whether cell t contains at least one dart. Note that the probability that a Poisson random variable with mean λ' is zero is $e^{-\lambda'}$. Thus we have,

$$\Pr(Y_{t,\lambda} = 0) = e^{-\frac{\lambda}{q^t}\frac{q-1}{q}}.$$

Here, we note some simple identities for integrals that we will use frequently in the

[24] In [137], the smoothing was implemented via *random* offsetting, instead of the *uniform* offsetting. In Curtain we need to use uniform offsetting so that the offset values of columns are similar to their neighbors.

analysis.

Lemma 5.23. *For any $q > 1$, we have*

$$\int \frac{1}{q^t} e^{-\frac{1}{q^t}} dt = \frac{1}{\log q} e^{-\frac{1}{q^t}} + C.$$

Furthermore, let c_0, c_1 be any positive numbers, we have

$$\int_{-\infty}^{\infty} \frac{c_0}{q^t} e^{-\frac{c_1}{q^t}} dt = \frac{c_0}{c_1} \frac{1}{\log q}.$$

Proof. Use standard calculus. □

The Constant κ Let $Z_{t,\lambda}$ be the indicator of whether the cell t is *free*. Unlike $Y_{t,\lambda}$, $Z_{t,\lambda}$ depends on which sketching algorithm we are analyzing. Since the state changing probability is equal to the sum of the area of free cells, we have

$$P_{m,\lambda m} = \sum_{j=0}^{\infty} \frac{1}{m} \left(\frac{1}{q^{j/m}} - \frac{1}{q^{j/m+1}} \right) Z_{j/m,\lambda}. \tag{5.15}$$

If $P_{m,\lambda m}$ converges to κ_λ/λ almost surely as $m \to \infty$, then $\mathbb{E}(P_{m,\lambda m})$ also converges to κ_λ/λ as $m \to \infty$. Thus we have, from (5.15),

$$\begin{aligned} \kappa_\lambda/\lambda &= \lim_{m\to\infty} \mathbb{E}(P_{m,\lambda m}) = \lim_{m\to\infty} \sum_{j=0}^{\infty} \frac{1}{m} \left(\frac{1}{q^{j/m}} - \frac{1}{q^{j/m+1}} \right) \mathbb{E}(Z_{j/m,\lambda}) \\ &= \int_0^{\infty} \left(\frac{1}{q^t} - \frac{1}{q^{t+1}} \right) \mathbb{E}(Z_{t,\lambda}) dt \approx \int_{-\infty}^{\infty} \left(\frac{1}{q^t} - \frac{1}{q^{t+1}} \right) \mathbb{E}(Z_{t,\lambda}) dt, \end{aligned} \tag{5.16}$$

where we can extend the integration range to negative infinity without affecting the limit of κ_λ as $\lambda \to \infty$.[25] We conclude that

$$\kappa = \lim_{\lambda\to\infty} \kappa_\lambda = \lim_{\lambda\to\infty} \lambda \int_{-\infty}^{\infty} \left(\frac{1}{q^t} - \frac{1}{q^{t+1}} \right) \mathbb{E}(Z_{t,\lambda}) dt. \tag{5.17}$$

The formula (5.17) is novel in the sense that, in order to evaluate κ, we now only need to

[25] Note that for any t, λ, we all have $\mathbb{E}(Z_{t,\lambda}) \le \mathbb{E}(1 - Y_{t,\lambda})$ (free cell has no dart). Therefore, by extending the integration (5.16) to the whole real line, the increment is bounded by $\int_{-\infty}^{0} \left(\frac{1}{q^t} - \frac{1}{q^{t+1}} \right) \mathbb{E}(1 - Y_{t,\lambda}) dt = \int_{-\infty}^{0} \left(\frac{1}{q^t} - \frac{1}{q^{t+1}} \right) e^{-\lambda(1/q^t - 1/q^{t+1})} dt = \frac{1}{\lambda} \int_{-\infty}^{-\log_q(\lambda(q-1)/q)} 1/q^t e^{-1/q^t} dt = \frac{1}{\lambda} \frac{e^{-\lambda\frac{q-1}{q}}}{\log q}$ where $\frac{e^{-\lambda\frac{q-1}{q}}}{\log q} \to 0$ as $\lambda \to \infty$. Thus it will not affect the value of $\lim_{\lambda\to\infty} \kappa_\lambda$.

understand the probability that $Z_{t,\lambda}$ is 1 for fixed t and λ.[26]

Analysis of Smoothed q-PCSA and q-LogLog The sketches q-PCSA and q-LogLog are the natural smoothed base-q generalizations of PCSA [83] and LogLog [73].

Theorem 5.14. *q-PCSA and q-LogLog are scale-invariant. In particular, we have,*

$$\kappa_{q\text{-PCSA}} = \frac{1}{\log q}, \text{ and } \kappa_{q\text{-LogLog}} = \frac{1}{\log q}\frac{q-1}{q}.$$

Proof. For q-LogLog, cell t is free iff both itself and all the cells above it in its column contain no darts. Thus we have

$$\mathbb{E}(Z_{t,\lambda}) = \prod_{i=0}^{\infty} \Pr(Y_{t+i,\lambda} = 0) = \prod_{i=0}^{\infty} e^{-\frac{\lambda}{q^{t+i}}\frac{q-1}{q}} = e^{-\frac{\lambda}{q^t}}.$$

Insert it to formula (5.17) and we get

$$\kappa_{q\text{-LogLog}} = \lim_{\lambda\to\infty} \lambda \int_{-\infty}^{\infty} \left(\frac{1}{q^t} - \frac{1}{q^{t+1}}\right) e^{-\frac{\lambda}{q^t}} dt = \frac{1}{\log q}\frac{q-1}{q}.$$

For q-PCSA, cell t is free iff it has no dart. Thus $Z_{t,\lambda} = 1 - Y_{t,\lambda}$ and by formula (5.17) we have

$$\kappa_{q\text{-PCSA}} = \lim_{\lambda\to\infty} \lambda \int_{-\infty}^{\infty} \left(\frac{1}{q^t} - \frac{1}{q^{t+1}}\right) e^{-\frac{\lambda}{q^t}\frac{q-1}{q}} dt = \frac{1}{\log q}.$$

□

The Fishmonger sketch is based on a smoothed, entropy compressed version of base-e PCSA. The memory footprint of Fishmonger approaches its entropy as $m \to \infty$, which was calculated to be mH_0 (Section 5.3). From Theorem 5.14, we know $\kappa_{e\text{-PCSA}} = 1$.

Corollary 5.2. *Fishmonger has limiting MVP $H_0/2 \approx 1.63$.*

Proof. By Theorem 5.12, limiting MVP equals $mH_0 \cdot \frac{1}{2m} = \frac{H_0}{2}$. □

Asymptotic Local View For any t and λ, since we want to evaluate $Z_{t,\lambda}$, whose value may depend on its "neighbors" on the dartboard, we need to understand the configurations

[26] Technically, to apply formula (5.17) one needs to first prove that the state changing probability $P_{m,\lambda m}$ converges almost surely to some constant κ_λ/λ for any λ, which is a mild regularity condition for any reasonable sketch.

of the cells near cell t. Since we consider the case where m goes to infinity, we may ignore the effect of smoothing to the cells in the immediate vicinity of cell t.

After taking these asymptotic approximations, we can index the cells near cell t as follows.

Definition 5.17 (neighbors of cell t)**.** Fix a cell t. Let $i \in \mathbb{Z}$ and $c \in \mathbb{R}$. The (i, c)-neighbor of cell t is a cell whose column index differs by i (negative i means to the left, positive to the right) and has height $t + c$, it covers the vertical interval $[q^{-(t+c+1)}, q^{-(t+c)})$. In the sawtooth partition, c is an integer when i is even and a half-integer when i is odd. (Note that we are locally ignoring the effect of smoothing.)

Once cell t is fixed, define $W(i, c)$ to be the indicator for whether the (i, c)-neighbor of cell t has at least one dart in it. Thus, for fixed t, λ, we have

$$\Pr(W(i,c) = 0) = \Pr(Y_{t+c,\lambda} = 0) = e^{-\frac{\lambda}{q^{t+c}}\frac{q-1}{q}}.$$

In the asymptotic local view, we lose the property that a cell can be uniquely identified by its height, hence the need to refer to nearby cells by their position *relative* to cell t.

Analysis of Curtain We first briefly state some properties of curtain. For any $a \geq 1$, recall that $\mathscr{O}_a = \{-(a-1/2), -(a-3/2), \ldots, -1/2, 1/2, \ldots, a-3/2, a-1/2\}$. It is easy to see that for any vector $v = (g_0, g_1, \ldots, g_{m-1})$, $v_{\text{curt}} = (\hat{g}_i)$ can be expressed as

$$\hat{g}_i = \max_{j \in [0,m-1]} \{g_j - |i-j|(a-1/2)\}.$$

For each i, we define the *tension point* τ_i to be the lowest allowable value of $\hat{g}_i$, given the context of its neighboring columns.

$$\tau_i = \max_{j \in [0,m-1]\setminus\{i\}} \{g_j - |i-j|(a-1/2)\},$$

and thus we have $\hat{g}_i = \max(g_i, \tau_i)$. We see that the column i is *in tension* iff $g_i \leq \tau_i$, that is, $\hat{g}_i = \tau_i$.

Theorem 5.15. *Curtain is scale-invariant with*

$$\kappa_{\textit{Curtain}} = \frac{1}{\log q}\frac{q-1}{q}\frac{1}{\frac{q-1}{q} + \frac{2}{q^h(q^{a-1/2}-1)} + \frac{1}{q^{h+1}}}.$$

Proof. Fix cell t and λ. Define $W_1(k)$ to be the height of the highest cell containing darts

in the column k away from t's column. I.e., define $\iota = \mathbb{1}[k \text{ is odd}]/2$ to be $1/2$ if k is odd and zero if k is even, and $W_1(k) \stackrel{\text{def}}{=} \max\{t+i+\iota \mid i \in \mathbb{Z} \text{ and } W(k, i+\iota) = 1\}$.

We have for any $i \in \mathbb{Z}$,

$$\Pr(W_1(k) \le t+i+\iota) = \prod_{j=1}^{\infty} \Pr(W(k, i+j+\iota) = 0) = e^{-\frac{\lambda}{q^{t+1+i+\iota}}}.$$

Let T_1 be the tension point of the column of cell t, which equals $\max_{j \in \mathbb{Z}\setminus\{0\}} \{W_1(j) - |j|(a-1/2)\}$. We have for any $i \in \mathbb{Z}$,

$$\begin{aligned}
\Pr(T_1 \le i+t) &= \Pr\left(\max_{j \in \mathbb{Z}\setminus\{0\}} \{W_1(j) - |j|(a-1/2)\} \le i+t\right) \\
&= \prod_{j \in \mathbb{Z}\setminus\{0\}} \Pr(W_1(j) - |j|(a-1/2) \le i+t) \\
&= \left(\prod_{j=1}^{\infty} e^{-\lambda \frac{1}{q^{t+i+1+j(a-1/2)}}}\right)^2 = e^{-\lambda \frac{2}{q^{t+i+1}} \frac{1}{q^{a-1/2}-1}}.
\end{aligned}$$

From the rules of Curtain, we know that a cell is free iff it contains no dart, it is at most $h-1$ below its column's tension point, and at most h below the highest cell in its column containing darts. Thus,

$$Z_{t,\lambda} = \mathbb{1}[Y_{t,\lambda} = 0] \cdot \mathbb{1}[t \ge T_1 - (h-1)] \cdot \mathbb{1}[t \ge W_1(0) - h],$$

Note that T_1 is independent from $Y_{t,\lambda}$ and $W_1(0)$. In addition, $Y_{t,\lambda}$ is also independent from $\mathbb{1}[t \ge W_1(0) - h]$, since the latter only depends on $Y_{t',\lambda}$ with $t' \ge h+t+1$. Thus, we have

$$\begin{aligned}
\mathbb{E}(Z_{t,\lambda}) &= \Pr(Y_{t,\lambda} = 0) \cdot \Pr(T_1 \le t+h-1) \cdot \Pr(W_1(0) \le t+h) \\
&= e^{-\frac{\lambda}{q^t}\frac{q-1}{q}} e^{-\lambda \frac{2}{q^{t+h}} \frac{1}{q^{a-1/2}-1}} e^{-\frac{\lambda}{q^{t+h+1}}} \\
&= \exp\left(-\frac{\lambda}{q^t}\left(\frac{q-1}{q} + \frac{2}{q^h(q^{a-1/2}-1)} + \frac{1}{q^{h+1}}\right)\right).
\end{aligned}$$

Thus by formula (5.17), we have

$$\begin{aligned}
\kappa_{\mathsf{Curtain}} &= \lim_{\lambda \to \infty} \lambda \int_{-\infty}^{\infty} \left(\frac{1}{q^t} - \frac{1}{q^{t+1}}\right) \exp\left(-\frac{\lambda}{q^t}\left(\frac{q-1}{q} + \frac{2}{q^h(q^{a-1/2}-1)} + \frac{1}{q^{h+1}}\right)\right) dt \\
&= \frac{1}{\log q} \frac{q-1}{q} \frac{1}{\frac{q-1}{q} + \frac{2}{q^h(q^{a-1/2}-1)} + \frac{1}{q^{h+1}}}.
\end{aligned}$$

□

Analysis of SecondCurtain

Theorem 5.16. *SecondCurtain is scale-invariant with*

$$\kappa_{\mathsf{SecondCurtain}} = \int_{-\infty}^{\infty} e^{-\frac{1}{q^t}\frac{q^{a-1/2}+1}{q^{a-1/2}-1}} \left(1+\varphi\left(\frac{1}{q^{t+1}}\right)\right) \prod_{j=1}^{\infty} \left(1+\varphi\left(\frac{1}{q^{t+j(a-1/2)}}\right)\right)^2 \frac{1}{q^t}\frac{q-1}{q}dt,$$

where $\varphi(x) = \sum_{i=0}^{\infty}\left(e^{\frac{x}{q^i}\frac{q-1}{q}}-1\right)$.

Proof. Fix a cell t. Let $W_2(k)$ be the height of the *second* highest cell containing a dart in the kth column relative to column of cell t. Let $\iota = \mathbb{1}\left[k \text{ is odd}\right]/2$. We have for any $i \in \mathbb{Z}$,

$$\begin{aligned}
&\Pr(W_2(k) \le t+i+\iota) \\
&= \prod_{j\ge 1}\Pr(Y_{t+i+\iota+j,\lambda}=0) + \sum_{j\ge 1}\Pr(Y_{t+i+\iota+j,\lambda}=1)\prod_{w\ge 1, w\ne j}\Pr(Y_{t+i+\iota+w,\lambda}=0) \\
&= \left(\prod_{j\ge 1}\Pr(Y_{t+i+\iota+j,\lambda}=0)\right)\left(1+\sum_{j\ge 1}\frac{\Pr(Y_{t+i+\iota+j,\lambda}=1)}{\Pr(Y_{t+i+\iota+j,\lambda}=0)}\right) \\
&= e^{-\frac{\lambda}{q^{t+i+\iota+1}}}\left(1+\sum_{j\ge 1}\frac{1-e^{-\frac{\lambda}{q^{t+i+\iota+j}}\frac{q-1}{q}}}{e^{-\frac{\lambda}{q^{t+i+\iota+j}}\frac{q-1}{q}}}\right) \\
&= e^{-\frac{\lambda}{q^{t+i+\iota+j+1}}}\left(1+\sum_{j\ge 1}\left(e^{\frac{\lambda}{q^{t+i+\iota+j}}\frac{q-1}{q}}-1\right)\right) \\
&= e^{-\frac{\lambda}{q^{t+i+\iota+j+1}}}\left(1+\varphi\left(\frac{\lambda}{q^{t+i+\iota+j+1}}\right)\right).
\end{aligned}$$

Now we can express the cumulative expression for the tension point T_2 of SecondCurtain:

$$\begin{aligned}
\Pr(T_2 \le i+t) &= \Pr\left(\max_{j\in\mathbb{Z}\setminus\{0\}}\{W_2(j)-|j|(a-1/2)\} \le i+t\right) \\
&= \prod_{j\in\mathbb{Z}\setminus\{0\}}\Pr(W_2(j)-|j|(a-1/2)\le i+t) \\
&= \left(\prod_{j=1}^{\infty} e^{-\frac{\lambda}{q^{t+i+1+j(a-1/2)}}}\left(1+\varphi\left(\frac{\lambda}{q^{t+i+1+j(a-1/2)}}\right)\right)\right)^2 \\
&= e^{-\frac{\lambda}{q^{t+i+1}}\frac{2}{q^{a-1/2}-1}}\prod_{j=1}^{\infty}\left(1+\varphi\left(\frac{\lambda}{q^{t+i+1+j(a-1/2)}}\right)\right)^2 \qquad (5.18)
\end{aligned}$$

From the rules of SecondCurtain, we know

$$Z_{t,\lambda} = \mathbb{1}\left[Y_{t,\lambda=0}\right] \cdot \mathbb{1}\left[t > T_2\right] \cdot \mathbb{1}\left[t > W_2(0)\right].$$

Note that T_2 are independent from $Y_{t,\lambda}$ and W_2. We have

$$\begin{aligned}\mathbb{E}(Z_{t,\lambda}) &= \Pr(Y_{t,\lambda} = 0, t > W_2(0)) \cdot \Pr(t > T_2) \\ &= (\Pr(t > W_1(0)) + \Pr(W_1(0) > t > W_2(0))) \cdot \Pr(T_2 \leq t-1). \end{aligned} \tag{5.19}$$

Note that

$$\begin{aligned}\Pr(W_1(0) > t > W_2(0)) &= \Pr(Y_{t,\lambda} = 0) \cdot \sum_{j\geq 1} \Pr(Y_{t+j,\lambda} = 1) \prod_{k\geq 1, k\neq j} \Pr(Y_{t+k,\lambda} = 0) \\ &= e^{-\frac{\lambda}{q^t}} \varphi\left(\frac{\lambda}{q^{t+1}}\right), \end{aligned} \tag{5.20}$$

and from the proof of Theorem 5.15 we know

$$\Pr(t > W_1(0)) = \Pr(W_1(0) \leq t-1) = e^{-\frac{\lambda}{t}}. \tag{5.21}$$

Apply the expressions (5.18), (5.20) and (5.21) to (5.19) and we have

$$\begin{aligned}\mathbb{E}(Z_{t,\lambda}) &= e^{-\frac{\lambda}{q^t}\frac{2}{q^{a-1/2}-1}} \prod_{j=1}^{\infty} \left(1+\varphi\left(\frac{\lambda}{q^{t+j(a-1/2)}}\right)\right)^2 \left(e^{-\frac{\lambda}{q^t}} + e^{-\frac{\lambda}{q^t}} \varphi\left(\frac{\lambda}{q^{t+1}}\right)\right) \\ &= e^{-\frac{\lambda}{q^t}\frac{q^{a-1/2}+1}{q^{a-1/2}-1}} \left(1+\varphi\left(\frac{\lambda}{q^{t+1}}\right)\right) \prod_{j=1}^{\infty} \left(1+\varphi\left(\frac{\lambda}{q^{t+j(a-1/2)}}\right)\right)^2. \end{aligned}$$

Then we have, from formula (5.17),

$$\begin{aligned}\kappa_{\mathsf{SecondCurtain}} &= \lim_{\lambda\to\infty} \lambda \int_{-\infty}^{\infty} e^{-\frac{\lambda}{q^t}\frac{q^{a-1/2}+1}{q^{a-1/2}-1}} \left(1+\varphi\left(\frac{\lambda}{q^{t+1}}\right)\right) \prod_{j=1}^{\infty} \left(1+\varphi\left(\frac{\lambda}{q^{t+j(a-1/2)}}\right)\right)^2 \frac{1}{q^t}\frac{q-1}{q} dt \\ &= \int_{-\infty}^{\infty} e^{-\frac{1}{q^t}\frac{q^{a-1/2}+1}{q^{a-1/2}-1}} \left(1+\varphi\left(\frac{1}{q^{t+1}}\right)\right) \prod_{j=1}^{\infty} \left(1+\varphi\left(\frac{1}{q^{t+j(a-1/2)}}\right)\right)^2 \frac{1}{q^t}\frac{q-1}{q} dt. \end{aligned}$$

□

Average Number of Bits Per column Each column uses $\log_2(2a)$ bits to store its offset in the second curtain. Note that SecondCurtain uses the unary encoding to store the

distance between the highest cell and the curtain. Next, we analyze the average number of bits SecondCurtain uses per column for the unary encoding.

Let $r \in [0,1)$ be the offset. We call the column with cells of height $\{i + r \mid i \in \mathbb{Z}\}$ as *column* r. Note that $W_1(0)$, $W_2(0)$ and T_2 can be defined for column r by using any cell in that column as the reference cell. When in tension, it takes $\max(W_1(0) - T_2 + 1, 1)$ bits to store the distance in the δ vector. When not in tension, it takes $(W_1(0) - W_2(0))$ bits. Let K_r be the number of bits for the unary coding for a column r. Thus we have

$$K_r = \mathbb{1}\left[W_2(0) \le T_2\right] \max(W_1(0) - T_2 + 1, 1) + \mathbb{1}\left[W_2(0) > T_2\right] (W_1(0) - W_2(0)).$$

Theorem 5.17. *The average unary encoding length is independent from λ. Specifically,*

$$\int_0^1 \mathbb{E}(K_r)dr = 1 + \int_{-\infty}^{\infty} e^{-\frac{\lambda}{q^t}\frac{2}{q^{a-1/2}-1}} \prod_{j=1}^{\infty} \left(1 + \varphi\left(\frac{\lambda}{q^{t+j(a-1/2)}}\right)\right)^2 \left(e^{-\frac{\lambda}{q^t}}\varphi\left(\frac{\lambda}{q^t}\right) - \left(e^{-\frac{\lambda}{q^{t+1}}} - e^{-\frac{\lambda}{q^t}}\right)\varphi\left(\frac{\lambda}{q^{t+1}}\right)\right) dt.$$

Proof. For simplicity, we write $W_1(0)$ and $W_2(0)$ as W_1 and W_2. We also write T_2 as T. Note that T is independent from W_1 and W_2.

First we note that

$$\max(W_1 - T + 1, 1) = 1 + \mathbb{1}\left[W_1 \ge T\right](W_1 - T).$$

Then we have

$$\begin{aligned} K_r &= \left(\mathbb{1}\left[W_1 \ge T\right](W_1 - T) + 1\right)\mathbb{1}\left[W_2 \le T\right] + (W_1 - W_2)\mathbb{1}\left[W_2 > T\right] \\ &= \mathbb{1}\left[W_2 \le T\right] + \sum_{i=-\infty}^{\infty} \left(\mathbb{1}\left[W_2 \le T\right]\mathbb{1}\left[W_1 \ge r + i > T\right] + \mathbb{1}\left[W_2 > T\right]\mathbb{1}\left[W_1 \ge r + i > W_2\right]\right) \\ &= \mathbb{1}\left[W_2 \le T\right] + \sum_{i=-\infty}^{\infty} \mathbb{1}\left[r + i > T\right]\mathbb{1}\left[W_1 \ge r + i > W_2\right]. \end{aligned} \tag{5.22}$$

Note that we have

$$\Pr(W_1 \ge r + i > W_2) = \sum_{j \ge 0} \Pr(Y_{i+r+j,\lambda} = 1) \prod_{k \ge 0, k \ne j} \Pr(Y_{i+r+k,\lambda} = 0) = e^{-\frac{\lambda}{q^{i+r}}}\varphi\left(\frac{\lambda}{q^{i+r}}\right), \tag{5.23}$$

$$\Pr(W_2 \le T) = \sum_{i=-\infty}^{\infty} \Pr(T \ge r+i)\Pr(W_2 = r+i) = 1 - \sum_{i=-\infty}^{\infty} \Pr(T < r+i)\Pr(W_2 = r+i), \tag{5.24}$$

and

$$\begin{aligned}\Pr(W_2 = r+i) &= \Pr(Y_{r+i,\lambda} = 1) \sum_{j\ge 1} \Pr(Y_{r+i+j,\lambda} = 1) \prod_{k\ge 1, k\ne j} \Pr(Y_{r+i+k,\lambda} = 0) \\ &= \left(1 - e^{-\frac{\lambda}{q^{r+i}}\frac{q-1}{q}}\right) e^{-\frac{\lambda}{q^{r+i+1}}} \varphi\left(\frac{\lambda}{q^{r+i+1}}\right) \\ &= \left(e^{-\frac{\lambda}{q^{r+i+1}}} - e^{-\frac{\lambda}{q^r}}\right) \varphi\left(\frac{\lambda}{q^{r+i+1}}\right). \end{aligned} \tag{5.25}$$

Apply the expressions (5.23), (5.24) and (5.25) to (5.22) and we have,

$$\begin{aligned}\mathbb{E}(K_r) &= \mathbb{E}\left(\mathbb{1}\left[W_2 \le T\right] + \sum_{i=-\infty}^{\infty} \mathbb{1}\left[r+i > T\right] \mathbb{1}\left[W_1 \ge r+i > W_2\right]\right) \\ &= \Pr(W_2 \le T) + \sum_{i=-\infty}^{\infty} \Pr(i > T)\Pr(W_1 \ge i > W_2) \\ &= 1 + \sum_{i=-\infty}^{\infty} \Pr(r+i > T)\left(\Pr(W_1 \ge r+i > W_2) - \Pr(W_2 = r+i)\right) \\ &= 1 + \sum_{i=-\infty}^{\infty} e^{-\frac{\lambda}{q^{i+r}}\frac{2}{q^{a-1/2}-1}} \prod_{j=1}^{\infty}\left(1 + \varphi\left(\frac{\lambda}{q^{i+r+j(a-1/2)}}\right)\right)^2 \\ &\qquad \left(e^{-\frac{\lambda}{q^{i+r}}}\varphi\left(\frac{\lambda}{q^{i+r}}\right) - \left(e^{-\frac{\lambda}{q^{i+r+1}}} - e^{-\frac{\lambda}{q^{i+r}}}\right)\varphi\left(\frac{\lambda}{q^{i+r+1}}\right)\right). \end{aligned}$$

Finally, due to smoothing, we have (set $t = i + r$)

$$\begin{aligned}\int_0^1 \mathbb{E}(K_r)dr = 1 + \int_{-\infty}^{\infty} e^{-\frac{\lambda}{q^t}\frac{2}{q^{a-1/2}-1}} \prod_{j=1}^{\infty}\left(1 + \varphi\left(\frac{\lambda}{q^{t+j(a-1/2)}}\right)\right)^2 \\ \left(e^{-\frac{\lambda}{q^t}}\varphi\left(\frac{\lambda}{q^t}\right) - \left(e^{-\frac{\lambda}{q^{t+1}}} - e^{-\frac{\lambda}{q^t}}\right)\varphi\left(\frac{\lambda}{q^{t+1}}\right)\right) dt. \end{aligned}$$

□

By numerical evaluations, the optimal parametrization such that a is a power of 2 is about $(q, a) = (2.2, 2)$. Taking these parameters, we have $\kappa_{\mathsf{SecondCurtain}} \approx 0.937$ and the average number of bits per sketch 3.86. Applying Theorem 5.12, we get its MVP $\frac{3.86}{2\cdot 0.937} \approx 2.06$.

5.7.4 Optimality of Martingale Fishmonger

Martingale sketches have several attractive properties, e.g., being strictly unbiased and insensitive to duplicate elements in the data stream. In Section 5.7.4 we argue that any sketch that satisfies these natural assumptions can be systematically transform into a Martingale X sketch with equal or lesser variance, where X is a dartboard sketch. In other words, the Martingale *transform* is optimal.

In Section 5.7.4 we prove that within the class of *linearizable* dartboard sketches, Martingale Fishmonger is optimal. The class of linearizable sketches is broad and includes state-of-the-art sketches, which lends strong *circumstantial* evidence that the memory-variance product of Martingale Fishmonger cannot be improved.

Optimality of the Martingale Transform Consider a non-mergeable sketch processing a stream $\mathcal{A} = (a_1, a_2, \ldots)$. Let S_i be its state after seeing $(a_1, \ldots, a_i)$, $\lambda_i = |\{a_1, \ldots, a_i\}|$, and $\hat{\lambda}(S_i)$ be the estimate of cardinality λ_i when in state S_i. We make the following natural assumptions.

Randomness. The random oracle h is the only source of randomness. In particular, S_i is a function of $(h(a_1), h(a_2), \ldots, h(a_i))$.

Duplicates. If $a_i \in \{a_1, \ldots, a_{i-1}\}$, $S_i = S_{i-1}$, i.e., duplicates do not trigger state transitions.

Unbiasedness. Suppose one examines the data structure at time i and sees $S_i = \mathbf{s}_i$ and then examines it at time j. Then $\hat{\lambda}(S_j) - \hat{\lambda}(\mathbf{s}_i)$ is an unbiased estimate of $\lambda_j - \lambda_i$.

Definition 5.18. The *state history* at time i, denoted $W_i = (S_0, S_{j_1}, S_{j_2}, \ldots, S_{j_\ell} = S_i)$, lists all the *distinct* states encountered when processing $(a_1, \ldots, a_i)$. Note that $\ell, \{j_k\}_{1 \le k \le \ell}$, and W_i are all random variables. When we want to fix a particular state-history w_i we write $W_i = w_i$.

The *duplicates* and *randomness* assumptions imply that the distribution of S_i and W_i depends only on λ_i. Thus, we henceforth assume for simplicity that there are no duplicates and that $\lambda_i = i$.

Suppose that the algorithm could magically make its cardinality estimates based not just on S_i, but the entire state history W_i. Let $\mathcal{H}$ be the (countably infinite) set of all histories and $\mathbf{H} \in [0,1]^{\mathcal{H} \times \mathcal{H}}$ be the stochastic matrix governing transition between histories.[27] Let

[27] I.e., if $w' = (S_0, \ldots, S, S')$ and w is the prefix of w' missing S', then $\mathbf{H}(w, w')$ is the probability that the next (distinct) element would cause the sketch to transition from S to S'.

$e_w \in [0,1]^{\mathcal{H}}$ be the probability distribution that puts unit probability on history w.[28] Let $\hat{\lambda} \in \mathbb{R}^{\mathcal{H}}$ be the vector of cardinality estimates at each history-state. From the *Randomness*, *Duplicates*, and *Unbiasedness* assumptions, it follows that if we observe that $W_i = w_i$, then

$$e_{w_i}^\top \mathbf{H}^{j-i}\, \hat{\lambda} = j - i.$$

is the expectation of $\hat{\lambda}(W_j) - \hat{\lambda}(w_i)$. Here $e_{w_i}^\top \mathbf{H}^{j-i}$ is the distribution of the history-state W_j conditioned on $W_i = w_i$. In the special case that $j = i+1$, we have

$$\mathbb{E}(\hat{\lambda}(W_{i+1}) \mid W_i = w_i) = \hat{\lambda}(w_i) + \mathbb{E}(\hat{\lambda}(W_{i+1}) - \hat{\lambda}(w_i) \mid W_{i+1} \neq w_i) \cdot (1 - \mathbf{H}(w_i, w_i))$$

and due to the *Unbiased* assumption this must be

$$= \hat{\lambda}(w_i) + 1.$$

Hence, for any w_i,

$$\mathbb{E}(\hat{\lambda}(W_{i+1}) - \hat{\lambda}(w_i) \mid W_{i+1} \neq w_i) = (1 - \mathbf{H}(w_i, w_i))^{-1}. \tag{5.26}$$

Phrased differently, the *Unbiased* assumption implies that (X_i) is a martingale, where $X_i = \hat{\lambda}(W_i) - i$. Define $Z_i = X_i - X_{i-1}$. Because (X_i) is a martingale the covariances of the (Z_i) are all zero. We have

$$\begin{aligned}
\mathbb{V}(X_i) &= \sum_{j=1}^{i} \mathbb{V}(Z_j) \\
&= \sum_{j=1}^{i} \big(\mathbb{E}(\mathbb{V}(Z_j \mid W_{j-1})) + \mathbb{V}(\mathbb{E}(Z_j \mid W_{j-1}))\big)
\end{aligned}$$

Observe that $\mathbb{E}(Z_j \mid W_{j-1}) = 0$, so this is

$$\begin{aligned}
&= \sum_{j=1}^{i} \mathbb{E}(\mathbb{V}(Z_j \mid W_{j-1})) \\
&= \sum_{j=1}^{i} \sum_{w_{j-1}} \Pr(W_{j-1} = w_{j-1}) \cdot \mathbb{E}\left(\left(\hat{\lambda}(W_j) - \hat{\lambda}(w_{j-1}) - 1\right)^2\right)
\end{aligned} \tag{5.27}$$

[28] I.e., $e_w(w) = 1$ and $e_w(w') = 0$ for all $w' \neq w$.

Note that the expression inside the expectation in (5.27) is constant when $W_j = w_{j-1}$, which holds with probability $c_0 = \mathbf{H}(w_{j-1}, w_{j-1})$. Let $c_1, c_2, \ldots$ be other constants that depend only on w_{j-1}. Continuing, this is equal to

$$= \sum_{j=1}^{i} \sum_{w_{j-1}} \left(\mathbb{E}\left(\left(\hat{\lambda}(W_j) - \hat{\lambda}(w_{j-1}) - 1 \right)^2 \,\middle|\, W_j \neq w_{j-1} \right) \cdot c_1 + c_2 \right) \tag{5.28}$$

By (5.26), $\mathbb{E}\left(\hat{\lambda}(W_j) - \hat{\lambda}(w_{j-1}) - 1\right) = (1 - \mathbf{H}(w_{j-1}, w_{j-1}))^{-1} - 1$, which also depends only on w_{j-1}, hence (5.28) is equal to

$$= \sum_{j=1}^{i} \sum_{w_{j-1}} \left(\mathbb{V}\left(\hat{\lambda}(W_j) - \hat{\lambda}(w_{j-1}) \,\middle|\, W_j \neq w_{j-1} \right) \cdot c_3 + c_4 \right). \tag{5.29}$$

At this point we can ask which estimate vector $\hat{\lambda}$ minimizes (5.29). The variances in (5.29) are non-negative, and it is possible to make them all zero, subject to (5.26), by setting

$$\hat{\lambda}(w_j) = \hat{\lambda}(w_{j-1}) + (1 - \mathbf{H}(w_{j-1}, w_{j-1}))^{-1} \tag{5.30}$$

for every w_j such that $\mathbf{H}(w_{j-1}, w_j) > 0$. Note that the transitions in $\mathbf{H}$ that occur with non-zero probability, excluding self-loops, form a directed arborescence (out-tree) rooted at the initial history (S_0). Thus, all the constraints of the form (5.30) can be satisfied simultaneously.

To recapitulate, as a consequence of the *Randomness*, *Duplicates*, and *Unbiased* assumptions, the Martingale estimator has minimum variance. Define the *h-state* of a sketch state S, denoted S^{h}, to be the set of hash values, that, if encountered, would cause no state transition. Then we can write S_i as $(\hat{\lambda}_i, S_i^{\mathrm{h}}, S_i^{\mathrm{l}})$, where $\hat{\lambda}_i$ is the Martingale estimate (which depends on the history), and S_i^{l} is any leftover state information not implied by S_i^{h} and $\hat{\lambda}_i$. We have shown that $\hat{\lambda}_i$ is the only information from the history useful for making cardinality estimates. Thus, the data structure is free to change S_i^{l} to any value consistent with S_i^{h} at will, and therefore S_i^{l} should not be stored at all. In other words, we can simply store the state S_i as $(\hat{\lambda}_i, S_i^{\mathrm{h}})$ and impute any S_i^{l} which is most advantageous.[29] Note that since (S_i^{h}) is a dartboard sketch,[30] $(\hat{\lambda}_i, S_i^{\mathrm{h}})$ is derived by a Martingale transform and is not worse than

[29] In particular, if a_{i+1} is such that it would cause $(\hat{\lambda}_i, S_i^{\mathrm{h}}, S_i^{\mathrm{l}})$ to become $(\hat{\lambda}_{i+1}, S_{i+1}^{\mathrm{h}}, S_{i+1}^{\mathrm{l}})$ and cause $(\hat{\lambda}_i, S_i^{\mathrm{h}}, S_i^{\mathrm{h}\prime})$ to become $(\hat{\lambda}_{i+1}, S_{i+1}^{\mathrm{h}\prime}, S_{i+1}^{\mathrm{l}\prime\prime})$, then we are free to choose our next state to be $(\hat{\lambda}_{i+1}, S_{i+1}^{\mathrm{h}})$ or $(\hat{\lambda}_{i+1}, S_{i+1}^{\mathrm{h}\prime})$, whichever is more advantageous. As variances improve when $|S^{\mathrm{h}}|$ is small, we would choose the one minimizing $|S_{i+1}^{\mathrm{h}}|, |S_{i+1}^{\mathrm{h}\prime}|$.

[30] (occupied cells = hash values that cause no transition)

the original sketch (S_i).

Remark 5.9. We should note that under some circumstances it is possible to achieve smaller variance by violating the *duplicates* and *unbiasedness* assumptions. For example, suppose the sketch state after seeing i elements were $(\hat{\lambda}_i, S_i, i)$. If the stream is duplicate-heavy, "i" carries no useful information, but if nearly all elements are distinct, i is also a good cardinality estimate. Since $\lambda_i \leq i$, the cardinality estimate $\min\{\hat{\lambda}_i, i\}$ is never worse than $\hat{\lambda}_i$ alone, but when $\lambda_i \approx i$, it is biased and has a constant factor lower variance.

Optimality of Martingale Fishmonger Given an abstract linearizable sketching scheme X, its space is minimized by compressing it to its entropy. On the other hand, by Theorem 5.12 the variance of Martingale X is controlled by the normalized expected probability of changing state: $2\lambda \cdot \mathbb{E}(P_\lambda)$. Theorem 5.18 lower bounds the ratio of these two quantities for any sketch that behaves well over a sufficiently large interval of cardinalities $\lambda \in [e^a, e^b]$. The proof technique is very similar to [137], as is the take-away message (that X=Fishmonger is optimal up to some assumptions). However, the two proofs are mathematically distinct as [137] focuses on Fisher information while Theorem 5.18 focuses on the *probability of state change*.

Theorem 5.18. *Fix reals $a < b$ with $d = b - a > 1$. Let $\bar{H}, \bar{R} > 0$. For any linearizable sketch, let $H(\lambda)$ be the entropy of its state and P_λ be the probability of state change*[31] *at cardinality λ satisfies that*

1. *for all $\lambda > 0$, $H(\lambda) \leq \bar{H}$, and*
2. *for all $\lambda \in [e^a, e^b]$, $2\lambda\mathbb{E}(P_\lambda) \geq \bar{R}$,*

then

$$\frac{\bar{H}}{\bar{R}} \geq \frac{H_0}{2} \frac{1 - \max(8d^{-1/4}, 5e^{-d/2})}{1 + \frac{(344+4\sqrt{d})}{d}\frac{H_0}{I_0}(1 - \max(8d^{-1/4}, 5e^{-d/2}))} = \frac{H_0}{2}(1 - o_d(1)).$$

Proof. By the assumptions of the theorem, we have $\frac{\int_a^b H(e^x)dx}{2\int_a^b e^x \cdot \mathbb{E}(P_{e^x})dx} \leq \frac{\bar{H}}{\bar{R}}$. Thus it suffices to prove that

$$\frac{\int_a^b H(e^x)dx}{2\int_a^b e^x \cdot \mathbb{E}(P_{e^x})dx} \geq \frac{H_0}{2}(1 - o_d(1)).$$

[31] The probability of state change P_λ is itself a random variable.

Now we will write the expressions in terms of the cells.

Let $\mathscr{C}$ be the of cells. By linearizability, we can write cells as $c_0, c_1, \ldots, c_{|\mathscr{C}|-1}$, where c_i has area p_i. Let Z_i be the indicator whether c_i is hit by a dart and Y_i be the indicator whether c_i is occupied. Let $F_i = (Y_0, \ldots, Y_i)$. Since it is linearizable, there is some monotone function $\phi : \{0,1\}^* \to \{0,1\}$ such that $Y_i = Z_i \vee \phi(F_{i-1})$. Assume poissonization[32], by Lemma 13 in [137], we can write

$$H(\lambda) = \sum_{i=0}^{|\mathscr{C}|-1} H_B(e^{-p_i\lambda}) \Pr(\phi(F_{i-1,\lambda}) = 0).$$

Then by the linearity of expectation, we have

$$\lambda \cdot \mathbb{E}(P_\lambda) = \lambda \sum_{i=0}^{|\mathscr{C}|-1} p_i \Pr(Z_i = 0) \Pr(\phi(F_{i-1}, \lambda) = 0) = \sum_{i=0}^{|\mathscr{C}|-1} p_i \lambda e^{-p_i\lambda} \Pr(\phi(F_{i-1,\lambda}) = 0).$$

For clear presentation, we introduce the following definitions.

Definition 5.19. Fix a linearizable sketch. Let $C \subset \mathscr{C}$ be a collection of cells and $W \subset \mathbb{R}$ be an interval of the reals. Define:

$$\mathbf{H}(C \to W) = \int_W \sum_{c_i \in C} \dot{H}(p_i e^x) \Pr(\phi(\mathbf{Y}_{i-1,e^x}) = 0) dx,$$
$$\mathbf{R}(C \to W) = \int_W \sum_{c_i \in C} \dot{R}(p_i e^x) \Pr(\phi(\mathbf{Y}_{i-1,e^x}) = 0) dx,$$

where

$$\dot{H}(t) = H_B(e^{-t}) = \frac{1}{\log 2}(te^{-t} - (1-e^{-t})\log(1-e^{-t})),$$
$$\dot{R}(t) = 2te^{-t}.$$

Now we can write $\int_a^b H(e^x)dx$ as $\mathbf{H}(\mathscr{C} \to [a,b])$ and $2\int_a^b e^x \cdot \mathbb{E}(P_{e^x})dx$ as $\mathbf{R}(\mathscr{C} \to [a,b])$.

Note that $\int_{-\infty}^{\infty} \dot{R}(e^x)dx = 2$ and it is proved in [137] that $\int_{-\infty}^{\infty} \dot{H}(e^x)dx = H_0$.[33] Thus we want to prove $\frac{\mathbf{H}(\mathscr{C}\to[a,b])}{\mathbf{R}(\mathscr{C}\to[a,b])} \geq \frac{\int_{-\infty}^{\infty} \dot{H}(e^x)dx}{\int_{-\infty}^{\infty} \dot{R}(e^x)dx}(1 - o_d(1))$. A similar statement is proved in Theorem 5 in [137] where it is showed that $\frac{\mathbf{H}(\mathscr{C}\to[a,b])}{\mathbf{I}(\mathscr{C}\to[a,b])} \geq \frac{\int_{-\infty}^{\infty} \dot{H}(e^x)dx}{\int_{-\infty}^{\infty} \dot{I}(e^x)dx}(1 - o_d(1))$ and $\mathbf{I}$ and $\dot{I}$ are defined

[32] A cell of size p will have probability $e^{-p\lambda}$ to be without a dart at cardinality λ.
[33] $H_0 = (\log 2)^{-1} + \sum_{k=1}^{\infty} k^{-1} \log_2(1 + 1/k)$.

as follows.

$$\mathbf{I}(C \to W) = \int_W \sum_{c_i \in C} \dot{I}(p_i e^x) \Pr(\phi(\mathbf{Y}_{i-1,e^x}) = 0) dx,$$

where $\dot{I}(t) = \frac{t^2}{e^t - 1}$. Note that the only difference between this theorem and Theorem 5 in [137] is between $\dot{R}$ and $\dot{I}$. However, one can verify that $\dot{R}(t)$ satisfies all the properties (see the lemma below) it is used for $\dot{I}(t)$ in the proof of Theorem 5 in [137]. Thus the similar lower bound is obtained here. □

Lemma 5.24. *The following statements are true.*

1. $\frac{\dot{H}(t)}{\dot{R}(t)}$ *is decreasing in* t *on* $(0, \infty)$. *This corresponds to Lemma 12 in [137].*
2. $\dot{R}(t) \leq 4e^{-t/2}$ *for all* $t > 0$. *This corresponds to Lemma 20 in [137].*

Proof. 1. $\frac{\dot{H}(t)}{\dot{R}(t)} = \frac{1}{2 \log 2}(1 - \frac{(1-e^{-t}) \log(1-e^{-t})}{te^{-t}})$. Note that $\frac{(1-e^{-t}) \log(1-e^{-t})}{-te^{-t}} = \frac{1-e^{-t}}{t} \cdot \frac{-\log(1-e^{-t})}{e^{-t}}$. Let $g(t) = \frac{1-e^{-t}}{t}$ and $h(x) = -\frac{\log(1-x)}{x}$ where $t > 0$ and $x \in (0, 1)$. It suffices to prove that $g(t)$ is decreasing and $g(x)$ is increasing. Indeed, $g'(t) = \frac{e^{-t}t - 1 + e^{-t}}{t^2} < \frac{e^{-t}e^t - 1}{t^2} = 0$ since $t + 1 < e^t$ for $t > 0$; $h'(x) = \frac{\frac{x}{1-x} + \log(1-x)}{x^2} > 0$ for $x \in (0, 1)$.

2. It suffices to prove $t/2 \leq e^{t/2}$, which is true.

□

Corollary 5.3. *The MVP of any linearizable and scale-invariant sketch is at least* $\frac{H_0}{2}$.

Proof. Let S be a scale-invariant combined sketch with constant κ. First recall Definition 5.16 that for any λ, we have $\lambda \cdot P_{m,\lambda m}$ converges to κ_λ almost surely as $m \to \infty$ where $P_{m,\lambda m}$ is the updating probability after λm insertions to a combined sketch consisting of m base-sketches[34]. By the dominated convergence theorem, we have $\lambda \cdot \mathbb{E}(P_{m,\lambda m})$ converges to κ_λ. Furthermore, recall that κ is the limit of κ_λ as $\lambda \to \infty$. Therefore, for any $\epsilon_1 > 0$, there exist sufficiently large m_1 and λ_1, such that for any $\lambda > \lambda_1$, $\lambda \cdot \mathbb{E}(P_{m_1,\lambda m_1}) > \kappa - \epsilon_1$.

Now by Theorem 5.12 and the definition of ARV factor (Definition 5.15), for any $\epsilon_2 > 0$, there exist sufficiently large $m_2 > m_1$ and $\lambda_2 > \lambda_1$, such that for any $\lambda > \lambda_2$, $m_2 \cdot \frac{\mathbb{V}(E_{m_2,\lambda m_2})}{(\lambda m_2)^2} > \frac{1}{2\kappa} - \epsilon_2 > \frac{1}{2\lambda \mathbb{E}(P_{m_2,\lambda m_2}) + 2\epsilon_1} - \epsilon_2$ where $E_{m_2,\lambda m_2}$ is the Martingale estimator.

Then, fixing m_2, view the combined sketch as *a single sketch*[35] and we have that for any $\lambda > \lambda_2 m_2$, $\frac{\mathbb{V}(E_\lambda)}{\lambda^2} > \frac{1}{2\lambda \mathbb{E}(P_\lambda) + 2\epsilon_1 m_2} - \frac{\epsilon_2}{m_2}$. Suppose the sketch uses $\bar{H}$ bits of memory and for

[34] The combined sketch is assumed to be smoothed.

[35] Thus $E_{m_2,\lambda m_2}$ should be written as $E_{\lambda m_2}$ and $P_{m_2,\lambda m_2}$ should be $P_{\lambda m_2}$.

sufficiently large λ, the relative variance is bounded by δ. Thus we have

$$\delta \geq \frac{\mathbb{V}(E_\lambda)}{\lambda^2} > \frac{1}{2\lambda\mathbb{E}(P_\lambda) + 2\epsilon_1 m_2} - \frac{\epsilon_2}{m_2},$$

which says

$$2\lambda\mathbb{E}(P_\lambda) \geq \frac{1}{\delta + \frac{\epsilon_2}{m_2}} - 2\epsilon_1 m_2.$$

Invoking Theorem 5.18 where a and b can be chosen arbitrarily far away, as long as $b > a > \log \lambda_2$, we have

$$\frac{\bar{H}}{\frac{1}{\delta + \frac{\epsilon_2}{m_2}} - 2\epsilon_1 m_2} \geq \frac{H_0}{2}.$$

Finally note that ϵ_1, ϵ_2 can be made arbitrarily small and m_2 can be made arbitrarily large. We conclude that the MVP $\bar{H} \cdot \delta \geq \frac{H_0}{2}$. □

5.7.5 Experimental Validation

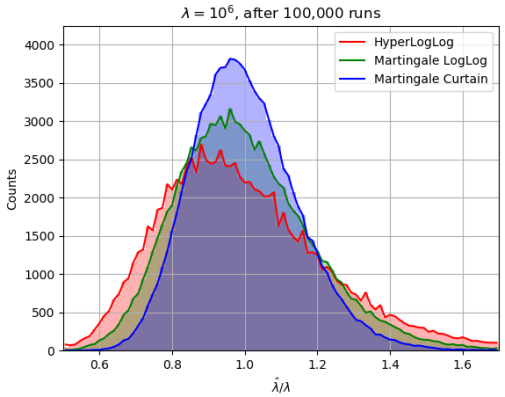
Throughout the paper we have maintained a possibly unhealthy devotion to asymptotic analysis, taking $m \to \infty$ whenever it was convenient. In practice m will be a constant, and possibly a smallish constant. How do the sketches perform in the pre-asymptotic region?

In turns out that the theoretical analysis predicts the performance of Martingale sketches pretty well, even when m is small. In the experiment of Fig. 5.15a, we fixed the sketch size at a tiny 128 bits. Therefore HyperLogLog uses $m_1 = \lfloor 128/6 \rfloor = 21$ counters. The Martingale LogLog and Martingale Curtain sketches encode the martingale estimator with a floating point *approximation* of $\hat{\lambda}$ in 14 bits, with a 6-bit exponent and 8-bit mantissa. Thus, Martingale LogLog uses $m_2 = (128 - 14)/6 = 19$ counters, and Martingale Curtain uses $m_3 = 37$.[36]

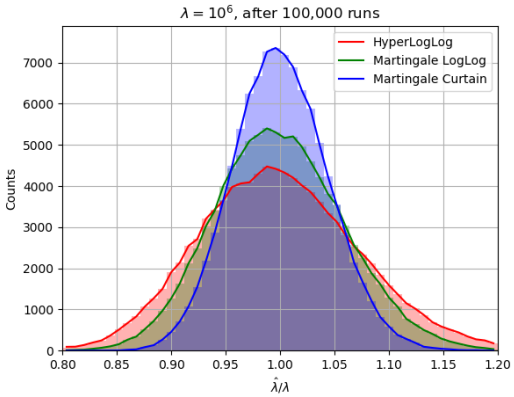
For larger sketch sizes, the distribution of $\hat{\lambda}/\lambda$ is more symmetric, and closer to the predicted performance. Fig. 5.15b gives the empirical distribution of $\hat{\lambda}/\lambda$ over 100,000 runs when $\lambda = 10^6$ and the sketch size is fixed at 1,200 bits. Here Martingale Curtain uses $m = 400$, and both Martingale LogLog and HyperLogLog use $m = 200$.

The experimental and predicted relative variances and standard errors are given in Table 5.4.

[36]It uses the optimal parameterization $(q, a, h) = (2.91, 2, 1)$ of Theorem 5.9.



(a) The sketch size is fixed at 128 bits.



(b) The sketch size is fixed at 1200 bits.

Figure 5.15: Experiments for Martingale estimators: Comparison of error distributions.

Sketch	Using 128 bits				Using 1200 bits			
	Experiment		Prediction		Experiment		Prediction	
	Var	StdErr	Var	StdErr	Var	StdErr	Var	StdErr
HyperLogLog	0.0573	23.94%	0.0549	23.44%	0.00541	7.36%	0.00539	7.35%
Martingale LogLog	0.0348	18.65%	0.0365	19.10%	0.00350	5.91%	0.00347	5.89%
Martingale Curtain	0.0211	14.54%	0.0208	14.43%	0.00189	4.35%	0.00193	4.39%

Table 5.4: Experiments of Martingale sketches. The relative variance is $\frac{1}{\lambda^2}\mathbb{V}(\hat{\lambda} \mid \lambda)$ and standard error is $\frac{1}{\lambda}\sqrt{\mathbb{V}(\hat{\lambda} \mid \lambda)}$. The predictions for Martingale LogLog and Martingale Curtain use Theorems 5.12, 5.14, and 5.15. The predictions for HyperLogLog are from Flajolet et al. [82, p. 139].

5.7.6 Summary of the Martingale X Sketches

The Martingale transform is attractive due to its simplicity and low variance, but it results in *non*-mergeable sketches. We proved that under natural assumptions[37], it generates optimal estimators automatically, allowing one to design structurally more complicated sketches, without having to worry about designing or analyzing *ad hoc* estimators. We proposed the Curtain sketch, in which each subsketch only needs a constant number of bits of memory, for *arbitrarily large* cardinality n.[38]

The analytic framework of Theorems 5.11 and 5.12 simplifies Cohen [49] and Ting [151], and gives a user-friendly formula for the asymptotic relative variance (ARV) of the Martingale estimator, as a function of the sketch's constant κ. We applied this framework to Martingale Curtain as well as the Martingale version of the classic sketches (MinCount, HLL and PCSA).

Assuming perfect compression, one gets the *memory-variance product* (MVP) of an sketch by multiplying its entropy and ARV. It is proved that for linearizable sketches, Fishmonger is optimal for mergeable sketches [137] (limiting MVP $= H_0/I_2 \approx 1.98$). In this section we proved that in the sequential (non-mergeable) setting, if we restrict our attention to linearizable sketches, that Martingale Fishmonger is optimal, with limiting MVP $= H_0/2 \approx 1.63$ (Section 5.7.4). We conjecture that these two lower bounds hold for general, possibly *non*-linearizable sketches.

5.8 The τ-GRA Statistics

In this section we switch back to the mergeable setting and analyze a class of statistics called *τ-generalized remaining area* (τ-GRA). Let me first discuss about previous estimation methods.

Sketches and Estimators. In 1983 Flajolet and Martin [83] developed the first non-trivial sketch called Probabilistic Counting with Stochastic Averaging (PCSA). A PCSA sketch S_{PCSA} consists of an array of m bit vectors or *subsketches*. The random oracle produces a pair $(h, g)(x)$, where $h(x) \in [m]$ is a uniformly random subsketch index and $g(x) \in \mathbb{Z}^+$ is equal to k with probability 2^{-k}. The bit $S_{\mathsf{PCSA}}(j, k)$ is 1 if there exists an x_i in the stream with $h(x_i) = j$ and $g(x_i) = k$, and 0 otherwise. Define $z(j) = \min\{k : S_{\mathsf{PCSA}}(j, k) = 0)$ to be the position of the least significant zero in the jth subsketch. Each $z(j)$ is individually a decent

[37] (insensitivity to duplicates, and unbiasedness)

[38] Note that an $O(\log \log n)$-bit offset register is needed for the whole sketch.

estimate of $\log(\lambda/m)$. Flajolet and Martin [83] analyzed the "first zero" estimator for PCSA, namely

$$\hat{\lambda}_{\mathrm{FM}}(S_{\mathsf{PCSA}}) \propto m \cdot 2^{\frac{1}{m}\sum_{j=1}^{m} z(j)}$$

and proved it has relative variance about $0.6/m$ and hence standard error about $0.78/\sqrt{m}$. It suffices to keep $\log n$ bits per subsketch, so PCSA requires $m \log n$ bits. Although the "first zero" has better concentration than the "last one," the latter is much cheaper to store. In 2003 Durand and Flajolet [73] implemented this idea in the LogLog sketch S_{LogLog}, which requires only $m \log\log n$ bits.

$$S_{\mathsf{LogLog}}(j) = \max\{k \,:\, S_{\mathsf{PCSA}}(j,k) = 1\}.$$

Durand and Flajolet proved that the estimator

$$\hat{\lambda}_{\mathrm{DF}}(S_{\mathsf{LogLog}}) \propto m \cdot 2^{\frac{1}{m}\sum_{j=1}^{m} S_{\mathsf{LogLog}}(j)}$$

has relative variance about C_{DF}/m and standard error about $\sqrt{C_{\mathrm{DF}}/m} \approx 1.3/\sqrt{m}$, where $C_{\mathrm{DF}} = \frac{2\pi^2+\log^2 2}{12} < 1.69$.[39] This estimator can be regarded as taking the *geometric mean* of individual estimates $2^{S_{\mathrm{DF}}(1)}, \ldots, 2^{S_{\mathrm{DF}}(m)}$. In 2007, Flajolet, Fusy, Gandouet, and Meunier [82] proposed a better estimator for LogLog based on the *harmonic* mean:

$$\hat{\lambda}_{\mathrm{FFGM}}(S_{\mathsf{LogLog}}) \propto m^2 \cdot \left(\sum_{j=1}^{m} 2^{-S_{\mathsf{LogLog}}(j)}\right)^{-1}$$

and called the resulting sketch HyperLogLog. It has relative variance roughly C_{FFGM}/m and standard error $\sqrt{C_{\mathrm{FFGM}}/m} \approx 1.04/\sqrt{m}$, where $C_{\mathrm{FFGM}} = 3\log 2 - 1 \approx 1.07944$. (The constants C_{DF} and C_{FFGM} are, in fact, limiting constants as $m \to \infty$.)

Dartboards and Remaining Area The intuition of "remaining" area comes from Ting's *area cutting process* [151], a very *visual* way to think about cardinality sketches. We will use a variant of it, the *Dartboard* model, introduced in Section 5.4.1.[40] The elements of this model are as follows:

Dartboard and Darts. The *dartboard* is a unit square $[0,1]^2$. When an element (dart) $x \in [n]$ arrives, it is *thrown* at a point $h(x) \in [0,1]^2$ in the dartboard determined by

[39] All logarithms are natural unless specified otherwise.

[40] The two are essentially identical, except that Ting's model does not have an explicit state space, and allows for non-deterministic state transitions.

the random oracle h.

Cells and States. The dartboard is partitioned into a countable set $\mathcal{B}$ of *cells*. Every cell may be *occupied* or *free*. The *state* of the sketch is defined by the set $\sigma \subseteq \mathcal{B}$ of occupied cells. The *state space* is some subset of $2^{\mathcal{B}}$.

Occupation Rules. If a dart is thrown at an occupied cell, the state does not change. If a dart is thrown at a free cell c, and the current state is σ, the new state is $f(\sigma, c) \supseteq \sigma \cup \{c\}$ in the state space.

Observe that the state transition function $f(\sigma, c)$ may force a cell to become *occupied* even though it contains no dart, which occurs in (Hyper)LogLog, for example.

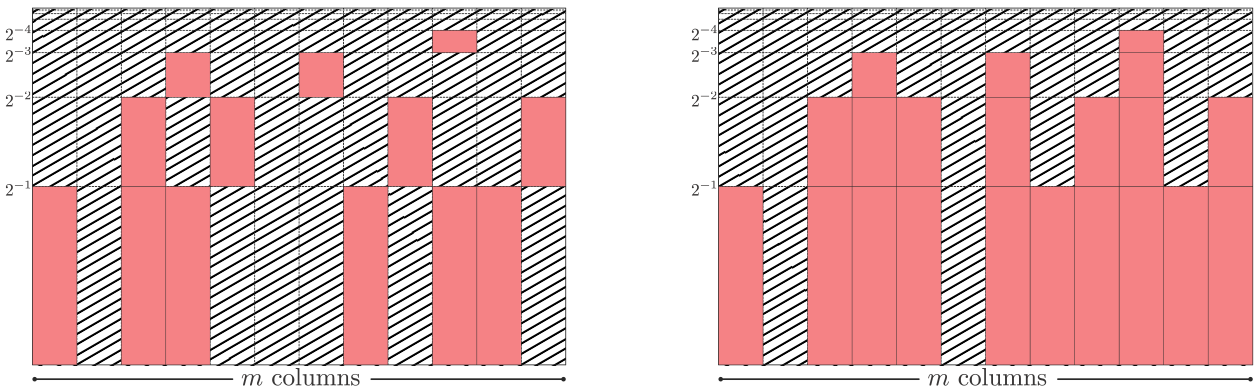


Figure 5.16: Visualization of remaining areas. The shaded area is the remaining area, whose size is the probability of changing state. Left: PCSA. Right: HyperLogLog.

A useful summary statistic of state σ is its *remaining area*

$$\text{RemainingArea}(\sigma) = \sum_{c \in \mathcal{B} \setminus \sigma} |c|,$$

where $|c|$ is the size of cell c. In other words, the remaining area is the total size of all free cells, or equivalently, the probability that the sketch changes upon seeing the next *distinct* element. See Fig. 5.16. Remaining area plays a key role in the (non-mergeable) Martingale sketches in Section 5.7. It also gives us a less fancy way to describe the HyperLogLog estimator without mentioning harmonic means:

$$\hat{\lambda}_{\text{FFGM}}(S_{\mathsf{LogLog}}) \propto m \left(\text{RemainingArea}(S_{\mathsf{LogLog}})\right)^{-1}.$$

Estimating the cardinality proportional to the reciprocal of the remaining area is reasonable for *any* sketch. This is the optimal estimator for k-Min-type sketches [35, 122], and as

we will see, superior to Flajolet and Martin's original $\hat{\lambda}_{\text{FM}}$ estimator for PCSA.

Generalized Remaining Area. Rather than have each cell $c \notin \sigma$ contribute $|c|$ to the remaining area of state σ, we could let it contribute $|c|^\tau$ instead for some fixed exponent $\tau > 0$. The resulting summary statistic is called *τ-generalized remaining area*, or τ-GRA.

$$\tau\text{-GRA}(\sigma) = \sum_{c \in \mathcal{B} \setminus \sigma} |c|^\tau.$$

Note that 0-GRA counts the number of free cells, which we regard as equivalent to counting the number of occupied cells, as is done explicitly by $\hat{\lambda}_{\text{Lang}}$.[41] It is also possible to analyze τ-GRA when $\tau < 0$ by summing over occupied cells rather than free cells. However, in this section we focus only on $\tau \geq 0$.

Summary of Contributions A conceptual contribution is the introduction of the τ-GRA summary statistic. The main technical contribution is a relatively simple analysis of the limiting efficiency of estimators for PCSA and (Hyper)LogLog based on the τ-GRA statistic. Our analysis has several benefits.

A Unified View. We show that HyperLogLog is based on 1-GRA and that, properly interpreted, LogLog is based on 0-GRA. Moreover, Lang's "coupon collector" estimator $\hat{\lambda}_{\text{Lang}}$ for PCSA is based on 0-GRA. Our analysis confirms Lang's back-of-the-envelope calculations that $\hat{\lambda}_{\text{Lang}}$ has limiting relative variance $(\log^2 2)/m$.

Simplicity. We use two techniques to dramatically simplify the analysis of τ-GRA-based estimators. The first, which has been used before [83, 82, 137, 140], is to consider a "poissonized" dartboard model, which allows us to avoid issues with small cardinalities and infinitesimal negative correlations between cells. The second is a smoothing operation similar to the one introduced in [137]. The combined effect of poissonization and smoothing is to make the sketch truly scale-invariant at every cardinality, without any periodic behavior.

Efficiency. A *statistically* optimal estimator for PCSA or LogLog meets the Cramér-Rao lower bound, which depends on the Fisher information of the given sketch (Section 5.2). It is known [30, 154] that the maximum likelihood estimator $\hat{\lambda}_{\text{MLE}}$ meets the Cramér-Rao lower bound asymptotically, as $m \to \infty$, but MLE is not particularly simple to

[41] In our stylized model there are an infinite number of cells, but in practice there are a finite number, so counting free cells is equivalent to counting occupied cells.

update as the sketch changes. The limiting relative variance of HyperLogLog's $\hat{\lambda}_{\text{FFGM}}$ is $(3\log 2 - 1)/m \approx 1.07944/m$, plus a tiny periodic function. Section 5.3 shows that the Cramér-Rao lower bound for (Hyper)LogLog is $\frac{\log 2}{\pi^2/6-1}/m \approx 1.07475/m$, which does not leave much room for improvement! In contrast, there is a wider gap between the limiting variance of PCSA's $\hat{\lambda}_{\text{DF}}$, namely $0.6/m$, or Lang's improvement $\hat{\lambda}_{\text{Lang}}$, namely $(\log^2 2)/m \approx 0.48/m$, and the Cramér-Rao lower bound [137, Theorem 3] of $\frac{\pi^2}{6\log 2}/m \approx 0.42138/m$. By choosing the optimal τs, our τ-GRA-based estimators achieve relative variance $1.0750/m$ for the LogLog sketch and $0.435532/m$ for the PCSA sketch, in both cases *nearly closing the gap* between the best known explicit estimators and the Cramér-Rao lower bound. The improvement to HyperLogLog is probably not worth implementing, but the improvement to PCSA can lead to an immediate improvement to practical implementations of PCSA, e.g., the CPC (Compressed Probabilistic Counting) sketch included in Apache *DataSketches* [150].

Fig. 1.5 illustrate the efficiency of τ-GRA-based estimators relative to other estimators, and Table 5.5 summarizes the same information symbolically.

Sketch & Estimator		Limiting Relative Variance	Citation
PCSA	Flajolet & Martin 1983		[83]
First Zero ($\hat{\lambda}_{\text{FM}}$)		$\approx 0.6/m + \theta(\lambda)$	[83]
Coupon Collector ($\hat{\lambda}_{\text{Lang}}$)		$\approx (\log^2 2)/m + \theta(\lambda) \approx 0.48/m$	[117]
Smoothed 0-GRA		$(\log^2 2)/m$	**Theorem 5.22**
Smoothed 1-GRA		$\frac{3\log 2}{4}/m \approx 0.51986/m$	**Theorem 5.22**
Smoothed τ-GRA	$\tau = 0.343557$	$\approx 0.435532/m$	**Theorem 5.22**
MLE / Cramér-Rao Lower Bound		$\frac{\pi^2}{6\log 2}/m \approx 0.42138/m$	Lemma 5.6
LogLog	Durand and Flajolet 2003		[73]
Geometric Mean ($\hat{\lambda}_{\text{DF}}$)		$\frac{2\pi^2+\log^2 2}{12}/m + \theta(\lambda) \approx 1.69/m$	[73]
Harmonic Mean ($\hat{\lambda}_{\text{FFGM}}$)		$(3\log 2 - 1)/m + \theta(\lambda) \approx 1.07944/m$	[82]
Smoothed 0-GRA		$\frac{2\pi^2+\log^2 2}{12}/m \approx 1.69/m$	**Theorem 5.21**
Smoothed 1-GRA		$(3\log 2 - 1)/m \approx 1.07944/m$	**Theorem 5.21**
Smoothed τ-GRA	$\tau = 0.889897$	$\approx 1.07507/m$	**Theorem 5.21**
MLE / Cramér-Rao Lower Bound		$\frac{\log 2}{\pi^2/6-1}/m \approx 1.07475/m$	Theorem 5.4

Table 5.5: Summary of estimators for PCSA and LogLog. Relative variance as $m, \lambda \to \infty$. All $\theta(\lambda)$ functions are multiplicatively periodic with period 2, which have a small magnitude independent of m. The "smoothing" mechanism (Section 5.8.1) eliminates periodic behavior.

Organization In Section 5.8.1 we define *scale-invariance*, and introduce *poissonization* and *smoothing* mechanisms to achieve scale-invariance. Section 5.8.2 introduces cardinality

estimates based on the τ-GRA statistic, and Sections 5.8.3 and 5.8.4 analyze the behavior of these estimators on the LogLog and PCSA sketches, respectively. We conclude with some remarks in Section 5.7.6.

5.8.1 Poissonization and Smoothing

Here we define the version of scale-invariance we need for the analysis in this section. Suppose we have an estimator E_λ at cardinality λ. Ideally, for a statistic to be a *measurement* of cardinality the relative error should distribute identically for any cardinality, i.e., E_λ should be *scale-invariant*.

Definition 5.20 (scale-invariance)**.** Let E_λ be an estimator of λ. We say E_λ is *scale-invariant* if for any $\lambda > 0$,

$$\frac{E_\lambda}{\lambda} \sim E_1.$$

Note that scale-invariance implies unbiasedness and constant relative variance that is independent of λ. Since $\mathbb{E}E_\lambda = \lambda \mathbb{E}E_1$, if $\mathbb{E}E_1 \neq 1$ then we can replace E_λ with $\frac{E_\lambda}{\mathbb{E}E_1}$ to make it an unbiased estimate of λ. Moreover, $\mathbb{V}(E_\lambda) = \lambda^2 \mathbb{V}(E_1)$, where $\mathbb{V}(E_1)$ is some fixed constant independent of λ.

Much of the simplicity and elegance of our analysis relies on beginning from *this* definition of strict scale-invariance. Unfortunately, in the real world the PCSA and HyperLogLog sketches are only *approximately* scale-invariant, stemming from two causes discussed in the introduction.

- Fixed-size sketches have two "edge effects," when $\lambda = \tilde{O}(m)$ is small and when $\lambda = \tilde{\Omega}(n)$ is approaching the size of the universe. The latter problem cropped up when $n = 2^{32}$ was small [82] but is generally not an issue when $n \geq 2^{64}$. See [82, 95, 77] for improved estimation methods for small cardinalities.

- Sketches that store continuous random variables (like k-Min [48, 90, 35, 122]) exhibit no periodic behavior but sketches that discretize their data are multiplicatively periodic in the base of the discretization, which is 2 in the case of standard PCSA and (Hyper)LogLog. Therefore, the relative variance of sketches like PCSA and HyperLogLog are *not* actually fixed constants independent of λ but periodic functions of λ, whose magnitude is small but *independent* of m, the size of the sketch.

As before, we consider a *smoothed*, *poissonized*, and *infinite* dartboard model to make the task of variance analysis simpler.

Definition 5.21 (Smoothed, poissonized, Infinite model)**.** The dartboard model and cell partition of PCSA and (Hyper)LogLog are changed as follows.

Smoothing. The sketch consists of m subsketches; these correspond to the columns in Fig. 5.8. Pick a vector $\mathbf{R} = (R_1, \ldots, R_m)$ of *offsets*. Cell j in column i now covers the vertical interval $(2^{-j-R_i}, 2^{-(j-1)-R_i}]$. We will pick $\mathbf{R}$ in two ways as is convenient. In Section 5.8.3 we choose each $R_i \in [0, 1)$ uniformly at random, independent of other offsets. In Section 5.8.4 we choose $\mathbf{R} = (0, 1/m, 2/m, \ldots, (m-1)/m)$ to be the uniformly spaced offset vector.

Infinite Dartboard. Rather than index cells by $\mathbb{Z}^+$, index them by $\mathbb{Z}$, i.e., the dartboard has unit width and infinite height. For example, cell 0 covers the vertical interval $(2^{-R_i}, 2^{1-R_i}]$, cell -5 covers the vertical interval $(2^{5-R_i}, 2^{6-R_i}]$, etc.

Poissonization. In the usual dartboard, the probability that a cell c remains free at cardinality λ is $(1 - |c|)^\lambda \to e^{-|c|\lambda}$ and the correlation between cells vanishes as $\lambda \to \infty$. For simplicity, these asymptotic properties can be achieved even for small λ with *poissonization*. Informally speaking, with poissonization, for each insertion, instead of throwing *one* dart at the board, darts *appear* on the board memorylessly with density 1. Formally speaking, for every new insertion, a Poisson point process on the infinite board with density 1 is added to the board, where each point in the process corresponds to a dart. Thus, after λ insertions, the darts on the board form a Poisson point process with density λ. By construction, for any λ—even $\lambda = 1$—the cells are independent and a cell c will remain free with probability precisely $e^{-|c|\lambda}$.

Smoothing eliminates periodic behavior, and the combination of poissonization and the Infinite Dartboard makes the distribution of the sketch scale invariant for all λ. Our justification for these changes is that they make the analysis simpler, and it does not really matter whether they are implemented in practice once λ is not too small. For example, w.h.p., there is no way to detect whether we are in a unit or infinite dartboard once $\lambda = \Omega(m \log m)$ as all cells indexed by $\mathbb{Z} - \mathbb{Z}^+$ will be occupied. Moreover, as $\lambda \to \infty$ the distribution of the true dartboard converges toward the poissonized dartboard. Smoothing eliminates the tiny periodic behavior of the estimator, but these effects are too small to worry about unless the magnitude of this periodic function is close to the desired variance, in which case smoothing *should* be implemented in practice.

From this point on, the smoothed, poissonized, and infinite dartboard model is assumed.

5.8.2 Estimation by Generalized Remaining Area

Cardinality estimation can be viewed as a *point estimation* problem where the number of subsketches is the number of independent samples/observations. Classically, one can produce i.i.d. estimates $\left(E_\lambda^{(i)}\right)_{i\in[1,m]}$ of λ with each subsketch and then use the sample mean as the combined estimator. A more general framework is to produce estimates $\left(E_{\lambda;f}^{(i)}\right)_{i\in[1,m]}$ of $f(\lambda)$ for some monotonic function f, then take the sample mean $\frac{1}{m}\sum_{i=1}^m E_{\lambda;f}^{(i)}$, which is concentrated around $f(\lambda)$. Thus we can recover an estimator of λ by applying f^{-1} to the sample mean. This process is summarized as follows.

$$\boxed{\begin{array}{c}E_{\lambda;f}^{(1)}, E_{\lambda;f}^{(2)}, \ldots, E_{\lambda;f}^{(m)}\\ (m \text{ independent}\\ \text{estimators of } f(\lambda))\end{array}} \xrightarrow{\text{sample mean}} \boxed{\begin{array}{c}\frac{1}{m}\sum_{i=1}^m E_{\lambda;f}^{(i)}\\ (\text{a concentrated}\\ \text{estimator of } f(\lambda))\end{array}} \xrightarrow{f^{-1}} \boxed{\begin{array}{c}f^{-1}\left(\frac{1}{m}\sum_{i=1}^m E_{\lambda;f}^{(i)}\right)\\ (\text{a concentrated}\\ \text{estimator of } \lambda)\end{array}}$$

An important example is its application to the *remaining area*. The remaining area (of one subsketch) offers a natural estimate for λ^{-1}. One can get a concentrated estimation for λ^{-1} using the sample mean of remaining areas of the subsketches and then take the reciprocal to get a concentrated estimator for λ. This is exactly what the HyperLogLog estimator $\hat{\lambda}_{\mathrm{FFGM}}$ does.

The remaining area estimates λ^{-1} and in general, the τ-generalized remaining area estimates $\lambda^{-\tau}$. Let $A_{\lambda;\tau}$ be the τ-generalized remaining area of one subsketch and $A_{\lambda;\tau}^{(1)}, A_{\lambda;\tau}^{(2)}, \ldots, A_{\lambda;\tau}^{(m)}$ be m i.i.d. copies. Thus by the same process, we get a generic estimator $\hat{\lambda}_{\tau;m}$ based on τ-GRA.

$$\hat{\lambda}_{\tau;m} \propto \left(\frac{1}{m}\sum_{i=1}^m A_{\lambda;\tau}^{(i)}\right)^{-\tau^{-1}}.$$

For any sketch, it turns out that the induced estimator $\hat{\lambda}_{\tau;m}$ is scale-invariant if the τ-GRA statistic itself is *τ-scale-invariant*.

Definition 5.22 (τ-scale-invariance)**.** Let $A_{\lambda;\tau}$ be the τ-generalized remaining area of a sketch. We say $A_{\lambda;\tau}$ is τ-scale-invariant if $A_{\lambda;\tau} \sim \lambda^{-\tau} A_{1;\tau}$ for any $\lambda > 0$.

Theorem 5.19. *If $A_{\lambda;\tau}$ is τ-scale-invariant, then*

$$\hat{\lambda}_{\tau;m}^* = \left(\frac{1}{m}\sum_{i=1}^m A_{\lambda;\tau}^{(i)}\right)^{-\tau^{-1}}.$$

is a scale-invariant estimator for λ.

Proof. By default, $\hat{\lambda}^*_{\tau;m}$ is the estimator at cardinality λ. When needed, we use $\hat{\lambda}^*_{\tau;m}[\lambda']$ to indicate that it is being evaluated on a sketch with cardinality λ'. By the τ-scale-invariance of $A_{\lambda;\tau}$, we have $A_{\lambda;\tau} \sim \lambda^{-\tau} A_{1;\tau}$. Thus

$$\hat{\lambda}^*_{\tau;m}[\lambda] \sim \left(\frac{1}{m}\sum_{i=1}^{m} \lambda^{-\tau} A^{(i)}_{1;\tau}\right)^{-\tau^{-1}} = \left(\frac{1}{m}\lambda^{-\tau}\sum_{i=1}^{m} A^{(i)}_{1;\tau}\right)^{-\tau^{-1}} = \lambda \cdot \hat{\lambda}^*_{\tau;m}[1].$$

□

Since we care about the asymptotic region, we prove the following useful theorem that expresses the asymptotic mean and variance of $\hat{\lambda}^*_{\tau;m}$ by the mean and variance of $A_{1;\tau}$ as $m \to \infty$. Note that although $\hat{\lambda}^*_{\tau;m}$ is scale-invariant, it is not yet normalized to be unbiased; the estimator $\hat{\lambda}_{\tau;m}$ will be the unbiased version of $\hat{\lambda}^*_{\tau;m}$. The asymptotic relative variance after normalization is also given in the lemma.

Theorem 5.20. *If $A_{1;\tau}$ is τ-scale-invariant with finite variance, we have for any $\lambda > 0$,*

1. $\lim_{m\to\infty} \mathbb{E}\hat{\lambda}^*_{\tau;m} = \lambda(\mathbb{E}A_{1;\tau})^{-\tau^{-1}}$.
2. $\lim_{m\to\infty} m\lambda^{-2}\mathbb{V}(\hat{\lambda}^*_{\tau;m}) = \tau^{-2}(\mathbb{E}A_{1;\tau})^{-2\tau^{-1}-2}\mathbb{V}(A_{1;\tau})$.
3. *For any $\lambda > 0$, the normalized estimator $\hat{\lambda}_{\tau;m} = (\mathbb{E}A_{1;\tau})^{\tau^{-1}}\hat{\lambda}^*_{\tau;m}$ is asymptotically unbiased and has limit relative variance*

$$\lim_{m\to\infty} m\lambda^{-2}\mathbb{V}(\hat{\lambda}_{\tau;m}) = \tau^{-2}\,(\mathbb{E}A_{1;\tau})^{-2}\,\mathbb{V}(A_{1;\tau}).$$

Proof. By scale-invariance, it suffices to consider the case $\lambda = 1$. Let $X = A_{1;\tau}$ and $Y_m = \frac{1}{m}\sum_{i=1}^{m} A^{(i)}_{1;\tau}$ be the mean of m copies of X. Define $f(x) = x^{-\tau^{-1}}$. Then $\hat{\lambda}^*_{\tau;m} = f(Y_m)$. Since we consider the case as $m \to \infty$, by the central limit theorem, Y_m is asymptotically normal around $\mathbb{E}X$. With high probability we have $Y_m \in (\mathbb{E}X - \frac{\log m}{\sqrt{m}}, \mathbb{E}X + \frac{\log m}{\sqrt{m}})$. Consider the first order approximation in this small neighborhood.

$$\begin{aligned} f(x) &= f(\mathbb{E}X) + (x - \mathbb{E}X)f'(\mathbb{E}X) + O((x-\mathbb{E}X)^2) \\ &= (\mathbb{E}X)^{-\tau^{-1}} - (x-\mathbb{E}X)\tau^{-1}(\mathbb{E}X)^{-\tau^{-1}-1} + O((x-\mathbb{E}X)^2). \end{aligned} \tag{5.31}$$

Note that $\mathbb{E}Y_m = \mathbb{E}X$ and $\mathbb{V}(Y_m) = \frac{1}{m}\mathbb{V}(X) = O(\frac{1}{m})$. Then we have

$$\mathbb{E}f(Y_m) = (\mathbb{E}X)^{-\tau^{-1}} - (\mathbb{E}Y_m - \mathbb{E}X)\tau^{-1}(\mathbb{E}X)^{-\tau^{-1}-1} + O(\mathbb{V}(Y_m))$$

$$= (\mathbb{E}X)^{-\tau^{-1}} + O(\frac{1}{m}). \tag{5.32}$$

Turning now to the variance,

$$\begin{aligned}
\mathbb{V}(f(Y_m)) &= \mathbb{E}\left(f(Y_m) - \mathbb{E}f(Y_m)\right)^2 \\
&= \mathbb{E}\left((Y_m - \mathbb{E}X)\tau^{-1}(\mathbb{E}X)^{-\tau^{-1}-1} + O(\frac{1}{m})\right)^2 && \text{(5.31) and (5.32)} \\
&= \mathbb{V}(Y_m)\tau^{-2}(\mathbb{E}X)^{-2\tau^{-1}-2} + O(\frac{1}{m^2}),
\end{aligned}$$

where we note that $\mathbb{E}(Y_m - \mathbb{E}X)^2 = \mathbb{V}(Y_m)$ and $\mathbb{E}(Y_m - \mathbb{E}X) = 0$. As $\mathbb{V}(Y_m) = \frac{1}{m}\mathbb{V}(X)$, this implies that the normalized variance is

$$m\mathbb{V}(f(Y_m)) = \mathbb{V}(X)\tau^{-2}(\mathbb{E}X)^{-2\tau^{-1}-2} + O(\frac{1}{m}). \tag{5.33}$$

We can now obtain a strictly unbiased estimator $(\mathbb{E}\hat{\lambda}^*_{\tau;m}[1])^{-1}\hat{\lambda}^*_{\tau;m}$, where $\hat{\lambda}^*_{\tau;m}[1]$ is the output of the estimator at cardinality (density) 1. We do not know precisely what $\mathbb{E}\hat{\lambda}^*_{\tau;m}[1]$ is, but $\lim_{m\to\infty}\mathbb{E}\hat{\lambda}^*_{\tau;m}[1] = (\mathbb{E}A_{1;\tau})^{-\tau^{-1}}$, so $\hat{\lambda}_{\tau;m} = (\mathbb{E}A_{1;\tau})^{\tau^{-1}}\hat{\lambda}^*_{\tau;m}$ is asymptotically unbiased, establishing Part (1). Part (2) follows from Part (1) and Eqn (5.33). Finally, observe that $\mathbb{V}(\hat{\lambda}_{\tau;m}) = (\mathbb{E}A_{1;\tau})^{2\tau^{-1}}\mathbb{V}(\hat{\lambda}^*_{\tau;m})$. Since $\lim_{m\to\infty} m\lambda^{-2}\mathbb{V}(\hat{\lambda}^*_{\tau;m}) = \tau^{-2}\left(\mathbb{E}A_{1;\tau}\right)^{-2\tau^{-1}-2}\mathbb{V}(A_{1;\tau})$, we have

$$\lim_{m\to\infty} m\lambda^{-2}\mathbb{V}(\hat{\lambda}_{\tau;m}) = \tau^{-2}\left(\mathbb{E}A_{1;\tau}\right)^{-2}\mathbb{V}(A_{1;\tau}),$$

proving Part (3). □

Theorems 5.19 and 5.20 give us a simple recipe for calculating the limiting relative variance of τ-GRA-based estimators. In Sections 5.8.3 and 5.8.4 we follow the following three-step process:

1. Calculate the mean $\mu = \mathbb{E}A_{1;\tau}$ and the variance $\sigma^2 = \mathbb{V}(A_{1;\tau})$ of the τ-generalized remaining area at density 1.

2. By Theorem 5.19, the induced estimator $\hat{\lambda}^*_{\tau;m} = \left(\frac{1}{m}\sum_{i=1}^m A^{(i)}_{\lambda;\tau}\right)^{-\tau^{-1}}$ is a scale-invariant estimator for λ, but possibly biased.

3. After normalization, we get the estimator $\hat{\lambda}_{\tau;m} = \mu^{\tau^{-1}}\hat{\lambda}^*_{\tau;m}$ which is asymptotically unbiased. By Theorem 5.20, its relative variance is asymptotically $\tau^{-2}\mu^{-2}\sigma^2/m$.

5.8.3 Generalized Remaining Area for the LogLog Sketch

Consider a LogLog sketch consisting of m subsketches (columns in Fig. 5.8), and let us focus on one subsketch with offset R. At cardinality λ, let X_λ be the index (an integer) of the highest occupied cell in this subsketch. Recall that the τ-generalized remaining area for this subsketch is summed up cell-by-cell:

$$\sum_{i=X_\lambda+1}^{\infty} (2^{-i-R})^\tau = \frac{1}{2^\tau - 1} 2^{-\tau(R+X_\lambda)} \propto 2^{-\tau(R+X_\lambda)}.$$

Because the cell sizes decay geometrically, this is linearly equivalent to taking the remaining area of the whole subsketch, $2^{-(R+X_\lambda)}$, to the τth power. For simplicity we calculate τ-GRA in this way, summing over subsketches rather than cells, thereby avoiding the leading constant $1/(2^\tau - 1)$. Thus, $A_{\lambda;\tau} = 2^{-\tau(R+X_\lambda)}$ is the contribution of this subsketch to the τ-GRA.

Now we analyze X_λ. Fix an offset $r \in [0, 1)$. For any $x > 0$, the event that $X_\lambda \le x$ is the event that the cells at or above $\lfloor x \rfloor + 1$ are all free. The sum of the heights of those cells is equal to $2^{-(\lfloor x \rfloor + r)}$ and they all have width $1/m$. Thus at cardinality λ, the number of darts in those cells is Poisson$(m^{-1}\lambda 2^{-\lfloor x \rfloor + r})$. We have

$$\mathbb{P}(X_\lambda \le x | R = r) = e^{-m^{-1}\lambda 2^{-(\lfloor x \rfloor + r)}}.$$

Having the floor function appear in this expression is annoying. Lemma 5.25 shows that by picking $R \in [0, 1)$ at random, we obtain a floor-free expression for the distribution of $A_{\lambda;\tau} = 2^{-\tau(R+X_\lambda)}$.

Lemma 5.25. *For any $x > 0$,*

$$\mathbb{P}(A_{\lambda;\tau} \ge x) = \int_0^1 e^{-x^{\tau^{-1}} 2^r m^{-1}\lambda}\, dr.$$

Proof. Let $x > 0$ and $r \in [0, 1)$. Then we have

$$\mathbb{P}(A_{\lambda;\tau} \ge x | R = r) = \mathbb{P}(X_\lambda \le -\log_2(x^{\tau^{-1}}) - r | R = r) = \exp(-m^{-1}\lambda 2^{-(\lfloor -\log_2(x^{\tau^{-1}}) - r \rfloor + r)}).$$

Set $y = \lfloor -\log_2(x^{\tau^{-1}}) - r \rfloor + r$ and $k = \lfloor -\log_2(x^{\tau^{-1}}) \rfloor$. Then for $r \in [0, -\log_2(x^{\tau^{-1}}) - k]$, we have $y = k + r$. For $r \in (-\log_2(x^{\tau^{-1}}) - k, 1)$, we have $y = k - 1 + r$. Therefore y iterates

$(-\log_2(x^{\tau^{-1}}) - 1, -\log_2(x^{\tau^{-1}})]$ as r iterates $[0,1)$. Therefore, we have

$$\begin{aligned}\mathbb{P}(A_{\lambda;\tau} \geq x) &= \int_0^1 \mathbb{P}(A_{\lambda;\tau} \geq x | R = r)\, dr = \int_0^1 \exp(-m^{-1}\lambda 2^{-(\lfloor -\log_2(x^{\tau^{-1}}) - r \rfloor + r)})\, dr \\ &= \int_0^1 \exp(-m^{-1}\lambda 2^{\log_2(x^{\tau^{-1}}) + r})\, dr = \int_0^1 e^{-x^{\tau^{-1}} 2^r m^{-1}\lambda}\, dr.\end{aligned}$$

□

We now prove that $A_{\lambda;\tau}$ is τ-scale-invariant.

Lemma 5.26. *For any $\tau > 0$, $A_{\lambda;\tau}$ is a τ-scale-invariant estimator for $\lambda^{-\tau}$.*

Proof. We need to prove that, for any $\tau, \lambda > 0$, $A_{\lambda;\tau} \sim \lambda^{-\tau} A_{1;\tau}$. Now note that, for any $x > 0$, we have

$$\begin{aligned}\mathbb{P}(A_{\lambda;\tau} \geq x) &= \int_0^1 e^{-x^{\tau^{-1}} 2^r m^{-1}\lambda}\, dr = \int_0^1 e^{-(\lambda x^{\tau^{-1}}) 2^r \cdot m^{-1}}\, dr \\ &= \mathbb{P}(A_{1;\tau} \geq \lambda^{\tau} x) = \mathbb{P}(\lambda^{-\tau} A_{1;\tau} \geq x).\end{aligned}$$

□

Recall that Γ is the continuous extension of the factorial function, with $\Gamma(n+1) = n!$ when $n \in \mathbb{N}$. Its integral form is $\Gamma(z) = \int_0^\infty u^{z-1} e^{-u} du$.

Proposition 5.2. *For any $\tau > 0$,*

$$m^{-\tau}\, \mathbb{E} A_{1;\tau} = \Gamma(\tau) \frac{1 - 2^{-\tau}}{\log 2}, \quad \textit{and} \quad m^{-2\tau}\, \mathbb{V}(A_{1;\tau}) = \Gamma(2\tau) \frac{1 - 2^{-2\tau}}{\log 2} - \left(\Gamma(\tau) \frac{1 - 2^{-\tau}}{\log 2} \right)^2.$$

Proof. We first prove some useful identities. Let $a, b > 0$. Then

$$\eta(a,b) \stackrel{\text{def}}{=} \int_0^\infty e^{-ax^b}\, dx = \int_0^\infty e^{-t} a^{-1} b^{-1} (t/a)^{-\frac{b-1}{b}} dt = a^{-b^{-1}} b^{-1} \Gamma(b^{-1}),$$

where we set $t = ax^b$. Let $q, c > 0$.

$$\begin{aligned}\xi(q,b,c) &\stackrel{\text{def}}{=} \int_0^\infty \int_0^1 e^{-q^r x^b c}\, dr dx = \int_0^1 \eta(cq^r, b)\, dr = c^{-b^{-1}} b^{-1} \Gamma(b^{-1}) \int_0^1 q^{-rb^{-1}}\, dr \\ &= c^{-b^{-1}} b^{-1} \Gamma(b^{-1}) \frac{1 - q^{-b^{-1}}}{b^{-1} \log q} = c^{-b^{-1}} \Gamma(b^{-1}) \frac{1 - q^{-b^{-1}}}{\log q}.\end{aligned}$$

The first and second moments can now be calculated as follows.

$$\begin{aligned}\mathbb{E}A_{1;\tau} &= \int_0^\infty \mathbb{P}(A_{1;\tau} \geq x)\, dx = \int_0^\infty \int_0^1 e^{-2^r x^{\tau^{-1}} m^{-1}}\, dr dx \\ &= \xi(2, \tau^{-1}, m^{-1}) = m^\tau \Gamma(\tau) \frac{1 - 2^{-\tau}}{\log 2}\end{aligned}$$

Turning to the second moment,

$$\begin{aligned}\mathbb{E}A_{1;\tau}^2 &= \int_0^\infty \mathbb{P}(A_{1;\tau}^2 \geq x)\, dx = \int_0^\infty \mathbb{P}(A_{1;\tau} \geq x^{1/2})\, dx \\ &= \int_0^\infty \int_0^1 e^{-2^r x^{\tau^{-1}/2} m^{-1}}\, dr dx = \xi\left(2, \frac{\tau^{-1}}{2}, m^{-1}\right) \\ &= m^{2\tau} \Gamma(2\tau) \frac{1 - 2^{-2\tau}}{\log 2}.\end{aligned}$$

We obtain the following closed form expression of the variance.

$$\mathbb{V}(A_{1;\tau}) = \mathbb{E}A_{1;\tau}^2 - (\mathbb{E}A_{1;\tau})^2 = m^{2\tau}\Gamma(2\tau)\frac{1 - 2^{-2\tau}}{\log 2} - m^{2\tau}\left(\Gamma(\tau)\frac{1 - 2^{-\tau}}{\log 2}\right)^2.$$

□

Theorem 5.21 (τ-GRA for the LogLog sketch)**.** *Let the offset vector $(R_i) \in [0, 1)^m$ be selected uniformly at random. Let $X_\lambda^{(i)}$ be the integer index of the highest one in the ith subsketch after λ insertions. Then for any $\tau > 0$,*

$$\hat{\lambda}_{\tau;m} = m\left(\Gamma(\tau)\frac{1 - 2^{-\tau}}{\log 2}\right)^{\tau^{-1}} \left(\frac{1}{m}\sum_{i=1}^m 2^{-\tau(R_i + X_\lambda^{(i)})}\right)^{-\tau^{-1}}$$

is a scale-invariant estimator for λ that is asymptotically unbiased. The asymptotic normalized relative variance is

$$\lim_{m\to\infty} m\lambda^{-2}\mathbb{V}(\hat{\lambda}_{\tau;m}) = \tau^{-2}\left(\frac{\Gamma(2\tau)\log 2}{\Gamma(\tau)^2} \cdot \frac{1 + 2^{-\tau}}{1 - 2^{-\tau}} - 1\right).$$

Proof. This follows directly from Theorem 5.20 and Proposition 5.2. □

Remark 5.10. The celebrated estimator $\hat{\lambda}_{\text{FFGM}}$ of HyperLogLog corresponds to $\tau = 1$. Inserting $\tau = 1$ to the variance formula, we have $\Gamma(2) = \Gamma(1) = 1$ and the leading constant of the variance is $3\log 2 - 1 \approx 1.07944$. The bias term at $\tau = 1$ is $\frac{1}{2\log 2}$, which match the constants from Flajolet et al. [82] as $m \to \infty$.

Remark 5.11. Note that for any $x_1, x_2, \ldots, x_m > 0$, $\lim_{\tau\to 0} \left(\frac{1}{m}\sum_{i=1}^m x_1^{-\tau}\right)^{-\tau^{-1}} = \left(\prod_{i=1}^m x_i\right)^{m^{-1}}$, i.e., the τ-mean converges towards the geometric mean as $\tau \to 0$. In other words, Durand and Flajolet's estimator $\hat{\lambda}_{\mathrm{DF}}$ for LogLog corresponds to 0-GRA. We have the normalized relative variance[42]

$$\lim_{\tau\to 0} \tau^{-2}\left(\frac{\Gamma(2\tau)\log 2}{\Gamma(\tau)^2}\cdot\frac{1+2^{-\tau}}{1-2^{-\tau}} - 1\right) = \frac{2\pi^2+\log^2 2}{12} \approx 1.68497,$$

which matches the limiting constant calculated by Durand and Flajolet [73, 72].

See Figure 1.5 as a visualization of the relative variance of the τ-GRA estimators for the LogLog sketch. By numerical optimization, the minimal variance 1.07507 is obtained at $\tau^* = 0.889897$. This comes quite close to the Cramér-Rao lower bound for LogLog sketches, which computes to be $\frac{\log 2}{\pi^2/6-1} \approx 1.07475$ in Lemma 5.6.

5.8.4 Generalized Remaining Area for the PCSA Sketch

Consider a PCSA sketch with m subsketches. Due to poissonization, the sketch consists of a set of independent indicator variables corresponding to whether each cell has been hit by at least one dart. To simplify notation, let $X(t)$ be a "fresh" binary random variable such that

$$X(t) = \begin{cases} 0, & \text{with probability } e^{-t} \\ 1, & \text{with probability } 1-e^{-t}. \end{cases}$$

Consider one subsketch with offset R. Cell i has height $2^{-(i+R)}$ and width $1/m$. At cardinality λ the number of points in the cell is Poisson$(m^{-1}\lambda 2^{-(i+R)})$. Thus the bit vector representing this subsketch distributes identically with $(X(m^{-1}\lambda 2^{-(i+R)}))_{i\in\mathbb{Z}}$. Let $\mathbb{1}[\mathcal{E}]$ denote the indicator variable for event $\mathcal{E}$. The τ-generalized remaining area for the PCSA sketch is then defined as follows.

$$A_{\lambda;\tau} = \sum_{i\in\mathbb{Z}} \mathbb{1}\left[X(m^{-1}\lambda 2^{-(i+R)}) = 0\right] 2^{-(i+R)\tau}.$$

Lemma 5.27. *For any $\tau > 0$, $A_{\lambda;\tau}$ is a τ-scale-invariant estimator for $\lambda^{-\tau}$.*

[42] This calculation is done in the algebraic system *Mathematica*.

Proof. We need to prove that for any $\lambda > 0$, $A_{\lambda;\tau} \sim \lambda^{-\tau} A_{1;\tau}$. Note that

$$\begin{aligned} A_{\lambda;\tau} &= \sum_{i\in\mathbb{Z}} \mathbb{1}\left[X(m^{-1}\lambda 2^{-(i+R)}) = 0\right] 2^{-\tau(i+R)} \\ &= \lambda^{-\tau} \sum_{i\in\mathbb{Z}} \mathbb{1}\left[X(m^{-1}2^{-(i+R-\log_2\lambda)}) = 0\right] 2^{-\tau(i+R-\log_2\lambda)} \end{aligned}$$

Note that because R is uniform over $[0, 1)$ and we are summing over $\mathbb{Z}$, this sum is invariant under shifts, e.g., by $\log_2 \lambda$. Continuing,

$$A_{\lambda;\tau} \sim \lambda^{-\tau} \sum_{i\in\mathbb{Z}} \mathbb{1}\left[X(m^{-1}2^{-(i+R)}) = 0\right] 2^{-\tau(i+R)} = \lambda^{-\tau} A_{1;\tau}.$$

□

In contrast to our smoothing of (Hyper)LogLog, it actually *does* matter that we use the uniform offset vector $\mathbf{R} = (0, 1/m, \ldots, (m-1)/m)$ rather than random offsets. Random offsets would introduce subtle correlations between cells in the same column, and increase the variance by some tiny constant. Uniform offsets have the property that there is a cell of size $2^{-i/m}$ for every $i \in \mathbb{Z}$, so the conceptual organization of cells into columns is no longer relevant.

Proposition 5.3. *Let $A^{(1)}_{1;\tau}, A^{(2)}_{1;\tau}, \ldots, A^{(m)}_{1;\tau}$ be the τ-GRA of m subsketches with uniform offsetting. For $\tau > 0$,*

$$\lim_{m\to\infty} m^{-1-\tau} \sum_{i=1}^{m} \mathbb{E}(A^{(i)}_{1;\tau}) = \frac{\Gamma(\tau)}{\log 2}, \quad \textit{and} \quad \lim_{m\to\infty} m^{-1-2\tau} \sum_{i=1}^{m} \mathbb{V}(A^{(i)}_{1;\tau}) = \frac{(1-2^{-2\tau})\Gamma(2\tau)}{\log 2}.$$

Proof. First note the following identity. Assume $q > 1, \tau > 0, \lambda > 0$. Then

$$\psi(\lambda, q, \tau) = \int_{-\infty}^{\infty} e^{-q^{-x}\lambda} q^{-\tau x}\, dx = \int_0^{\infty} e^{-t} (t/\lambda)^{\tau} (t \log q)^{-1}\, dt = \frac{1}{\log q} \lambda^{-\tau} \Gamma(\tau).$$

Here $t = q^{-x}$. After uniform offsetting, a PCSA sketch with m subsketches have cells of size $2^{-i/m}$ for all $i \in \mathbb{Z}$. Thus we we have

$$\lim_{m\to\infty} m^{-1-\tau} \sum_{i=1}^{m} \mathbb{E}\left(A^{(i)}_{1;\tau}\right) = \lim_{m\to\infty} m^{-1} \sum_{i\in\mathbb{Z}} \mathbb{E}\left(\mathbb{1}\left[X(m^{-1}2^{-i/m}) = 0\right] (m^{-1}2^{-(i/m)})^{\tau}\right)$$

Setting $h(t) = \mathbb{E}\left(\mathbb{1}\left[X(2^{-t}) = 0\right](2^{-t})^{\tau}\right) = e^{-2^{-t}}2^{-\tau t}$, the sum becomes $m^{-1}\sum_{i\in\mathbb{Z}} h(i/m + \log_2 m)$. Since we are summing over $\mathbb{Z}$, the shift $\log_2 m$ in the argument affects the sum vanishingly as $m \to \infty$. Thus $\lim_{m\to\infty} m^{-1}\sum_{i\in\mathbb{Z}} h(i/m+\log_2 m) = \lim_{m\to\infty} m^{-1}\sum_{i\in\mathbb{Z}} h(i/m) = \int_{-\infty}^{\infty} h(t)\,dt$. Thus,

$$\lim_{m\to\infty} m^{-1-\tau}\sum_{i=1}^{m}\mathbb{E}\left(A_{1;\tau}^{(i)}\right) = \int_{-\infty}^{\infty} e^{-2^{-t}}2^{-\tau t}\,dt = \psi(1,2,\tau) = \frac{\Gamma(\tau)}{\log 2}.$$

Note that by poissonization, cells are independent and thus all co-variances are zero.

$$\lim_{m\to\infty} m^{-1-2\tau}\sum_{i=1}^{m}\mathbb{V}(A_{1;\tau}^{(i)}) = \lim_{m\to\infty} m^{-1}\sum_{i\in\mathbb{Z}}\mathbb{V}\left(\mathbb{1}\left[X(m^{-1}2^{-i/m}) = 0\right](m^{-1}2^{-i/m})^{\tau}\right)$$

by the same limiting argument laid out above, this is equal to

$$\begin{aligned}
&= \int_{-\infty}^{\infty}\mathbb{V}(\mathbb{1}\left[X(2^{-t}) = 0\right]2^{-\tau t})\,dt \\
&= \int_{-\infty}^{\infty} e^{-2^{-t}}2^{-2\tau t} - e^{-2\cdot 2^{-t}}2^{-2\tau t}\,dt \\
&= \psi(1,2,2\tau) - \psi(2,2,2\tau) = \frac{\Gamma(2\tau)}{\log 2}(1-2^{-2\tau}).
\end{aligned}$$

□

Theorem 5.22 (τ-GRA for the PCSA sketch)**.** *Let $A_{\lambda;\tau}^{(i)}$ be the τ-generalized remaining area of the ith subsketch with uniform offsetting, and $A = \sum_{i=1}^{m} A_{\lambda;\tau}^{(i)}$ be the τ-GRA. Then for any $\tau > 0$,*

$$\hat{\lambda}_{\tau;m} = m\left(\frac{\Gamma(\tau)}{\log 2}\right)^{\tau^{-1}}\left(\frac{A}{m}\right)^{-\tau^{-1}}$$

is a scale-invariant estimator for λ that is asymptotically unbiased. The asymptotic normalized relative variance is

$$\lim_{m\to\infty} m\lambda^{-2}\mathbb{V}(\hat{\lambda}_{\tau;m}) = \frac{(1-2^{-2\tau})\Gamma(2\tau)\log 2}{\tau^2\Gamma(\tau)^2}.$$

Proof. This follows directly from Theorem 5.20 and Proposition 5.3. □

Remark 5.12. The remaining area estimator (1-GRA) has normalized relative variance $\frac{(1-2^{-2})\Gamma(2)}{1^2\Gamma(1)^2}\log 2 = \frac{3}{4}\log 2 \approx 0.51986$, which is better than Flajolet and Martin's original "first zero" estimator $\hat{\lambda}_{\mathrm{FM}}$.

Remark 5.13. As τ goes to 0, $\hat{\lambda}_{\tau;m}$ is essentially counting the number of free cells (0s in the sketch), which corresponds to Lang's [117] "coupon collector" estimator $\hat{\lambda}_{\text{Lang}}$ that counts occupied cells (1s in the sketch). The limiting variance of this estimator is[43]

$$\lim_{\tau \to 0} \frac{(1 - 2^{-2\tau})\Gamma(2\tau) \log 2}{\tau^2 \Gamma(\tau)^2} = \log^2 2 \approx 0.480453,$$

which confirms Lang's [117] back-of-the-envelope calculation that it should be $\log^2 2$.

See Figure 1.5 for a visualization of the relative variance of the τ-GRA estimators for the PCSA sketch. By numerical optimization, the minimal variance 0.435532 is obtained at $\tau^* = 0.343557$. This comes very close to the Cramér-Rao lower bound of $\frac{\pi^2}{6 \log 2} \approx 0.42138$ for PCSA sketches, as computed in Theorem 5.4.

5.8.5 Summary of τ-GRA based Estimators

We introduced a class of estimators based on the concept of *τ-generalized remaining area*, which has significant explanatory power, as it subsumes many existing estimators [73, 82, 117]. This concept is quite powerful, and allows us to *almost* completely close the gap between the best explicit (non-MLE) estimators for HyperLogLog [82] and PCSA [117] and their respective Cramér-Rao lower bounds (Section 5.3). See Fig. 1.5 and Table 5.5.

One distinction of our proofs is that they are very precise, but assume only basic probability and calculus. They avoid the daunting complexity of a Flajolet-style analysis [83, 81, 73, 82]. The key ingredients in our approach are (i) a deliberate simplification of the probabilistic model (see Definitions 5.20, 5.21, and 5.22), and (ii) restricting the analysis to *limiting* relative variance rather than try to understand the variance at *every* value of m.[44]

[43] This calculation is done in the algebraic system *Mathematica*.

[44] In practice m is typically between 2^8 and 2^{14}, which is already in the asymptotic regime. A Flajolet-style analysis—which obtains bias correction constants and a precise variance when m is *fixed*—is most helpful when m is a small constant.

CHAPTER 6

Generalized Cardinality Estimation

I have intensively studied the problem of cardinality estimation in Chapter 5, but only over *incremental streams*. The problem of cardinality estimation is naturally defined in any M-turnstile model. See Table 2.1. One important special case is the problem over ($\mathbb{Z}$-)turnstile streams, which models real-world data streams that have both insertions and deletions.

Definition 6.1 (M-turnstile cardinality estimation)**.** Recall that in M-turnstile streams both elements' states and the updates reside in M (Section 2.1). Let $\mathbf{x} \in M^n$ be the state vector over a M-turnstile stream. The goal is to estimate the cardinality $\|\mathbf{x}\|_0 = \sum_{v=1}^{n} \mathbb{1}\left[\mathbf{x}(v) \neq 0_M\right]$.

In this chapter, I will show how to construct M-turnstile cardinality sketches and estimators for an arbitrary finite commutative monoid M, by applying the theory and techniques I have developed in Chapters 3 to 5.

6.1 Preliminaries: Commutative Monoids

Let us review the structural theory of commutative monoids. I use Clifford and Preston's text [46] for reference.

6.1.1 Decomposition of Commutative Monoids

As we are only considering cardinality estimation problem in this chapter, it suffices to characterize the structure of the underlying monoid M as follows.

Theorem 6.1 (Decomposition of commutative monoids)**.** *Let $(M, +)$ be a commutative monoid. Define $G = \{x \in M \mid \exists y \in M, x + y = 0\}$ and $H = M \setminus G$. Then*

1. $(G, +|_G)$ *is a group.*

2. $(H, +|_H)$ *is a semigroup.*

3. *For any* $g \in G$ *and* $h \in H$, $g + h \in H$.

Proof. Follow from the maximal decomposition of M into archimedean semigroups. See [46, Theorem 4.13]. Given the maximal decomposition, G is the semigroup that contains 0 and H is the union of other semigroups. □

6.1.2 Pontryagin Duality for LCA Groups

Let us import some duality theory from Folland's text [84]. We start with the fundamental notion of *character*.

Definition 6.2 (character)**.** Let $(G, +)$ be an LCA (locally compact and abelian) group. A character γ is a continuous[1] function from G to $\mathbb{C}$ such that for any $x, y \in G$, $\gamma(x + y) = \gamma(x)\gamma(y)$ and $|\gamma(x)| = 1$.

Theorem 6.2 (Pontryagin duality, [84])**.** *Let* Γ *be the set of characters of* G. Γ *is an LCA group with point-wise multiplication and the topology of local uniform convergence on compact sets. The character group of* G*'s character group* Γ *is isomorphic to* G.

To emphasize this symmetry, we use the convention of using "+" as the operator in the group Γ and writing $\gamma(x)$ as (x, γ) where $x \in G$ and $\gamma \in \Gamma$. Thus for $\gamma, \gamma' \in \Gamma$ and $x \in G$, $(x, \gamma + \gamma') = (\gamma + \gamma')(x) = \gamma(x)\gamma'(x) = (x, \gamma)(x, \gamma')$. Some common (G, Γ) pairs are listed as below.

- The character group of $\mathbb{Z}_p$ is $\mathbb{Z}_p$, where for $x \in \mathbb{Z}_p$ and $\gamma \in \mathbb{Z}_p$, $(x, \gamma) = e^{2\pi x\gamma i/p}$.[2]
- The character group of $\mathbb{Z}$ is $\mathbb{T}$, where for $x \in \mathbb{Z}$ and $\gamma \in \mathbb{T}$, $(x, \gamma) = e^{2\pi x\gamma i}$.
- The character group of $\mathbb{R}$ is $\mathbb{R}$, where for $x \in \mathbb{R}$ and $\gamma \in \mathbb{R}$, $(x, \gamma) = e^{2\pi x\gamma i}$.
- Let G_1 and G_2 be two LCA groups. The product group $G_1 \times G_2$ is the set $\{(x, y) : x \in G_1, y \in G_2\}$ with entry-wise addition. Note that $G_1 \times G_2$ is also an LCA group. The character group of $G_1 \times G_2$ is $\Gamma_1 \times \Gamma_2$ where for $x_1 \in G_1, x_2 \in G_2, \gamma_1 \in \Gamma_1, \gamma_2 \in \Gamma_2$, $((x_1, x_2), (\gamma_1, \gamma_2)) = (x_1, \gamma_1)(x_2, \gamma_2)$.

[1] When the group G is finite, any function $f : G \to \mathbb{C}$ is continuous w.r.t. the discrete topology.
[2] x, γ are canonically embedded as integers in the exponent.

We list a few properties of characters that are used frequently later.

Proposition 6.1. *Let $x, y \in G$ and $\gamma, \gamma' \in \Gamma$. The following statements are true.*

- $(x + y, \gamma) = (x, \gamma)(y, \gamma)$.
- $(x, \gamma + \gamma') = (x, \gamma)(x, \gamma')$.
- $|(x, \gamma)| = 1$.
- $(0, \gamma) = (x, 0) = 1$.
- $(-x, \gamma) = (x, -\gamma) = \overline{(x, \gamma)}$, *the complex conjugate of* (x, γ).

The next important notion is the *Haar measure*.

Theorem 6.3 (Haar measure of LCA groups)**.** *For any LCA group G, there exists a unique (non-trivial) measure μ_G (up to a multiplicative scalar) of G such that for any $x \in G$ and measurable set $E \subset G$,*

$$\mu_G(x + E) = \mu_G(E), \text{ where } x + E \text{ is the translation of } E \text{ by } x.$$

For discrete groups, the Haar measure is just the counting measure. For $\mathbb{R}$ or $\mathbb{R}^k$, the Haar measure is just the usual Lebesgue measure. From now on, we will integrate over LCA groups with the Haar measure by default.

6.2 `COUNT-DISTINCT` over Abelian Groups

Let G be an abelian group. We assume the dual group Γ is compact and the Haar measure is normalized to be a *probabilistic measure*, i.e.,

$$\int_\Gamma 1 \, \mu_\Gamma(d\gamma) = 1.$$

One may approach the G-turnstile cardinality estimation problem from the Lévy-Khintchine perspective used in Chapter 3 or from the harmonic decomposition perspective used in Chapter 4. For simplicity, I will take the decomposition approach, which saves us from defining Lévy processes over generic spaces.

The target function f for the G-turnstile cardinality estimation problem is

$$f \colon G \to \mathbb{R}_+, \quad f(x) = \mathbb{1}\left[x \neq 0_G\right],$$

which can be decomposed as

$$f(x) = \int_\Gamma (1 - (x, \gamma))\, \mu_\Gamma(d\gamma), \qquad \text{(harmonic decomposition)}$$

because

$$\int_\Gamma 1\, \mu_\Gamma(d\gamma) = 1, \text{ and } \int_\Gamma (x, \gamma)\, \mu_\Gamma(d\gamma) = \mathbb{1}[x = 0]. \qquad (\mu_\Gamma \text{ is normalized})$$

Based on the decomposition above, for a G-turnstile stream $\mathbf{x} \in G^n$, let us define the γ-moment to be

$$\gamma(\mathbf{x}) = \sum_{v \in [n]} (1 - (\mathbf{x}(v), \gamma)) \qquad (\gamma\text{-moment})$$

One can thus decompose the cardinality as follows.

$$\|\mathbf{x}\|_0 = \int_\Gamma \gamma(\mathbf{x})\, \mu_\Gamma(d\gamma). \qquad \text{(decomposition of cardinality)}$$

Therefore, we will apply the approach in Chapter 4 to form an estimator for the cardinality by combining the estimates of γ-moments. To estimate the γ-moment, we can apply the SymmetricPoissonTower sketch over the group G. Given the form of decomposition in this section, let us use the usual Poisson projection instead of symmetric Poisson.

Definition 6.3 (G-PoissonTower)**.** Let G be a abelian group. A G-PoissonTower is parameterized by a single integer m, which controls the variance of the estimates, and consists of an infinite vector $(X_j)_{j \in \mathbb{Z}}$ of G-valued *cells*, initialized as $0^{\mathbb{Z}}$. For any $j \in \mathbb{Z}$ and input stream $\mathbf{x} \in G^n$,

$$X_j = \sum_{v \in [n]} \underbrace{\mathbf{x}(v) + \cdots + \mathbf{x}(v)}_{Z_{v,j}}, \qquad \text{(summation is in } (G, +))$$

where $Z_{v,j} \sim \text{Poisson}(e^{-j/m})$. All $Z_{v,j}$s are independent and assigned by the random oracle. Let $\lambda = \|\mathbf{x}\|_0$ be the number of non-zero elements (i.e., non-identity in G) in the stream as usual.

Note that by definition $(\cdot, \gamma)$ is a homomorphism from $(G, +)$ to $(\mathbb{C}, \times)$. Thus by the same computation in Section 4.2.1 around $(\star\star)$, we have

$$\mathbb{E}(X_j, \gamma) = e^{-e^{-j/m} \sum_{v \in [n]} (1 - (\mathbf{x}(v), \gamma))} = e^{-e^{-j/m} \gamma(\mathbf{x})}. \qquad (6.1)$$

Modify the proof in Chapter 4 and we get the following theorem.

Theorem 6.4 (G-turnstile cardinality estimation)**.** *Let G be a compact and abelian group. Let $X = (X_k^{(j)})_{k\in\mathbb{Z},j=1,2,3}$ be a triple G-PoissonTower. The G-turnstile cardinality estimator is defined as*

$$\hat{\lambda}(X) = \int_\Gamma \frac{\prod_{j=1}^3 \left(\sum_{k=-\infty}^{\infty}(1-(X_j,\gamma))e^{\frac{1}{3}k/m}\right)}{m^3|\Gamma(-1/3)|^3}\, d\mu_\gamma(\gamma).$$

Then

$$\mathbb{E}\hat{\lambda}(X) = \lambda, \qquad \textit{and} \qquad \mathbb{V}\hat{\lambda}(x) \le \lambda^2 \cdot O(m^{-1}).$$

Proof. Follows from Theorem 4.4 and Eq. (6.1). The truncation analysis is omitted. □

As in Chapter 4, G-PoissonTower can be truncated to store only $O(\log n)$ layers in the cost of a polynomially small bias. Each cell now is a G-valued register which can be stored using $O(\log|G|)$ bits. Therefore, the constructed sketch can $(1\pm\epsilon)$-approximate the cardinality of a G-turnstile stream with $O(\epsilon^{-2}\log(n)\log(|G|))$ bits of space.

When the group G is large or even infinite such as $\mathbb{Z}$, there are two workarounds.

- One may homomorphically map G to a suitable finite group G' and count as if the stream is G'-turnstile. Let this map be ϕ. The approach above will give an accurate estimate of

  $$\|\mathbf{x}\|_{0,G\to G'} \stackrel{\text{def}}{=} \sum_{v\in[n]} \mathbb{1}\left[\phi(\mathbf{x}(v)) \neq 0_{G'}\right],$$

  which is expected to approximate the actual $\|\mathbf{x}\|_0$. This is always possible with additional assumptions on the stream, such as bounding the stream length and/or bounding the "magnitude" of elements' states. A standard example is Kane et al.'s $\mathbb{Z}$-turnstile cardinality estimator, which projects $\mathbb{Z}$ to $\mathbb{Z}_p$ where p is a random prime.

- Apply the Lévy-Khintchine approach in Chapter 3 instead. Roughly speaking, with the Lévy-Khintchine approach, one maintains (X_j, R_j) where R_j is a uniformly random element in G, and then applies the Lévy-Tower's estimator. Note that now (X_j, R_j) resides in the complex unit circle and can be represented by a floating number between $[0, 2\pi)$. To guarantee the stored number will not drift too much, one needs to choose a proper precision based on the total number of updates in the stream.

6.3 COUNT-DISTINCT over Commutative Monoids

Let $(M, +)$ be a commutative monoid. As discussed in Section 2.1, the M-turnstile cardinality estimation problem is the *most* generic cardinality estimation that can be naturally defined over distributed streams. By Theorem 6.1, M can be decomposed to a group G and a semigroup H such that

- $G \cap H = \emptyset$;
- $0_M \in G$;
- $G + H = H$.

Therefore, the target function f for the M-turnstile cardinality estimation problem can be decomposed as follows.

$$\begin{aligned} f \colon M \to \mathbb{R}_+, f(x) &= \mathbb{1}\left[x \neq 0_M\right] \\ &= \mathbb{1}\left[x \in H\right] + \mathbb{1}\left[x \in G \wedge x \neq 0_M\right] \end{aligned}$$

The cardinality can be decomposed accordingly,

$$\begin{aligned} \lambda &= \sum_{v \in [n]} \left(\mathbb{1}\left[\mathbf{x}(v) \in H\right] + \mathbb{1}\left[\mathbf{x}(v) \in G \wedge x \neq 0_M\right]\right) \\ &= \underbrace{\sum_{v \in [n]} \mathbb{1}\left[\mathbf{x}(v) \in H\right]}_{\lambda_H} + \underbrace{\sum_{v \in [n]} \mathbb{1}\left[\mathbf{x}(v) \in G \wedge x \neq 0_M\right]}_{\lambda_G} \\ &= \lambda_H + \lambda_G. \end{aligned}$$

Observe that we can project M to the monoid $G \cup \{\infty\}$ without affecting the cardinality.

Definition 6.4 (killing a group)**.** Given any group $(G, +)$, we define its *killed monoid* $(G \cup \{\infty\}, +)$ where for any $g \in G$, $g + \infty = \infty + g = \infty + \infty = \infty$.

Lemma 6.1 (collapse M for cardinality estimation)**.** *Let M be a commutative monoid and G be the group in the decomposition (Theorem 6.1). Then*

$$\phi \colon M \to G \cup \{\infty\}, \quad \phi(x) = \begin{cases} x, & x \in G \\ \infty, & x \in M \setminus G \end{cases}$$

is a homomorphism from M to the killed monoid $G \cup \{\infty\}$ with a trivial kernel ($\ker(M) = \{0\}$).

Remark 6.1. Since the map has a trivial kernel, the M-stream $\mathbf{x}$ and the *collapsed* M-stream $(\phi(\mathbf{x}(v)))_{v\in[n]}$ have equal cardinality.

We now modify the G-PoissonTower in Definition 6.3 slightly to handle collapsed M-streams.

Definition 6.5 (collapsed M-PoissonTower)**.** Let M be a commutative monoid. An *collapsed M-PoissonTower* is parameterized by a single integer m, which controls the variance of the estimates, and consists of an infinite vector $(X_j)_{j\in\mathbb{Z}}$ of $G\cup\{\infty\}$-valued *cells*, initialized as $0^{\mathbb{Z}}$. For any $j\in\mathbb{Z}$ and input stream $\mathbf{x}\in M^n$,

$$X_j = \sum_{v\in[n]} \phi(\mathbf{x}(v)Z_{v,j}),$$

where $Z_{v,j}\sim \text{Poisson}(e^{-j/m})$, and $\phi(\mathbf{x}(v)Z_{v,j})$ is defined as

$$\phi(\mathbf{x}(v)Z_{v,j}) = \begin{cases} 0, & Z_{v,j}=0 \\ \infty, & Z_{v,j}>0 \text{ and } \mathbf{x}(v)\in M\setminus G \\ \underbrace{\mathbf{x}(v)+\cdots+\mathbf{x}(v)}_{Z_{v,j}}, & \text{otherwise} \end{cases}$$

All $Z_{v,j}$s are independent and assigned by the random oracle. X_j resides in the killed monoid $G\cup\{\infty\}$ and is a homomorphic image of the input.

In the worst case when $\lambda_H = 0$, the space taken by a collapsed M-PoissonTower is equivalent to the space taken by a corresponding G-PoissonTower—the killed monoid has only one more element than the group G, the killed state ∞. When $\lambda_H \gg \lambda_G$, the collapsed M-PoissonTower will essentially become a PCSA sketch and can be compressed to $O(\epsilon^{-2})$ bits of space (becoming ∞ is irrevocable and thus the ∞ prefix can be stored by a offset counter). See Fig. 6.1

Let us reuse the analysis in Section 6.2. By the construction of the collapsed tower, we only need to consider the case $M = G\cup\{\infty\}$. Let γ be any character of G, we define

$$(\infty,\gamma) \stackrel{\text{def}}{=} 0.$$

Clearly $(\cdot,\gamma)$ is patched to be a homomorphism from the killed monoid to $(\mathbb{C},\times)$ as one may check that for any $g\in G$

$$(\infty+g,\gamma) = (\infty,\gamma) = 0$$

$$
\begin{array}{cccccccccccccccccccc}
\infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & \infty & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & \infty & \infty & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & \infty & 0 & 0 & 0 & 0 & \infty & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & \infty & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & 0 & \infty & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

(a) PCSA ($\{0, \infty\}$-PoissonTower)

$$
\begin{array}{cccccccccccccccccccc}
G & G & G & G & G & G & G & G & G & G & 0 & G & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
G & G & G & G & G & G & G & G & 0 & G & G & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
G & G & G & G & G & G & G & 0 & G & 0 & 0 & 0 & 0 & G & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
G & G & G & G & G & G & G & G & G & G & G & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
G & G & G & G & G & G & G & G & G & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
G & G & G & G & G & G & G & G & G & G & 0 & G & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
G & G & G & G & G & G & G & G & G & 0 & 0 & G & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

(b) G-PoissonTower

$$
\begin{array}{cccccccccccccccccccc}
\infty & \infty & \infty & \infty & G & G & \infty & G & G & G & 0 & G & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\infty & \infty & \infty & \infty & \infty & \infty & G & G & 0 & G & G & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\infty & \infty & \infty & \infty & G & G & G & 0 & \infty & 0 & 0 & 0 & 0 & G & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\infty & \infty & \infty & \infty & \infty & \infty & \infty & G & G & G & G & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\infty & \infty & \infty & G & G & G & G & G & G & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\infty & \infty & \infty & \infty & \infty & \infty & G & G & G & \infty & 0 & G & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\infty & \infty & \infty & \infty & \infty & G & G & G & G & 0 & 0 & G & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

(c) Collapsed M-PoissonTower

Figure 6.1: PCSA, G-PoissonTower, and collapsed M-PoissonTower. Cells are arranged in an array. The subsampling rate decreases from left to right. PCSA handles insertion-only streams. G-PoissonTower handles invertible streams where updates can be inverted later. The collapsed M-PoissonTower sketch generalizes both—one element may "linger" in the group G and "commit" to ∞ at any time.

$$(\infty, \gamma)(g, \gamma) = 0 \cdot (g, \gamma) = 0$$
$$(\infty + \infty, \gamma) = (\infty, \gamma) = 0$$
$$(\infty, \gamma)(\infty, \gamma) = 0 \cdot 0 = 0.$$

Thus all the analysis in Section 6.2 will work for the killed monoid as well. We are only left to check that, the decomposition does sum back to the target cardinality. Indeed, we have exactly

$$\begin{aligned}\int_\Gamma (1 - (x, \gamma))\, \mu_\Gamma(d\gamma) &= \begin{cases} 0, & x = 0 \\ 1, & x \in G \wedge x \neq 0 \\ 1, & x \in M \setminus H \end{cases} \\ &= \mathbb{1}\left[x \neq 0_M\right].\end{aligned}$$

We summarize the M-turnstile cardinality sketch we have just developed.

Theorem 6.5 (M-turnstile cardinality estimation)**.** *Let M be a commutative monoid with the decomposition (G, H) defined in Theorem 6.1. Assume G is a compact and abelian group. Let $X = (X_k^{(j)})_{k\in\mathbb{Z}, j=1,2,3}$ be a triple collapsed M-PoissonTower. The M-turnstile cardinality estimator is defined as*

$$\hat{\lambda}(X) = \int_\Gamma \frac{\prod_{j=1}^3 \left(\sum_{k=-\infty}^{\infty} (1 - (X_j, \gamma)) e^{\frac{1}{3}k/m}\right)}{m^3 |\Gamma(-1/3)|^3}\, d\mu_\gamma(\gamma),$$

where $(\infty, \gamma) = 0$ for any $\gamma \in \Gamma$. Then

$$\mathbb{E}\hat{\lambda}(X) = \lambda, \quad \textit{and} \quad \mathbb{V}\hat{\lambda}(x) \leq \lambda^2 \cdot O(m^{-1}).$$

Proof. Follows from Theorem 6.4. □

6.4 Applications

Why care about abstract cardinality estimation at all? I will present two applications to show how such theory can be useful in practice.

Union estimation Suppose now we have two turnstile streams with a common domain. Let us assume all frequencies are less than p and use the $\mathbb{Z}_p$-turnstile model instead. The

target is to estimate the size of the *union* of these two streams, i.e., the number of elements that are non-zero in at least one of the streams. Note that this quantity is not the cardinality of the *merged stream*. For example, if an element has value $+1$ in the first stream, and value -1 in the second stream, then after merging the two streams, the element will not be in the support anymore. However, one needs to count this element for the union query defined above. To my knowledge, there have been no systematic methods of doing this apart from sampling. Nevertheless, one can observe that, though it is not the cardinality of the merged stream, but it *is* the cardinality of the *product stream*. The product stream is a $\mathbb{Z}_p \times \mathbb{Z}_p$-stream where an element with value $x \in \mathbb{Z}_p \times \mathbb{Z}_p$ means it has value x_1 in the first stream and value x_2 in the second stream. $\mathbb{Z}_p \times \mathbb{Z}_p$ is of course a compact and abelian group. Therefore Theorem 6.4 implies the following.

Theorem 6.6 (union estimator)**.** *Let $G = \mathbb{Z}_p$. Let $[n]$ be the universe of elements and $\mathbf{x}(1), \mathbf{x}(2) \in \mathbb{Z}_p^n$ be two streams where $\mathbf{x}(1)(v), \mathbf{x}(2)(v)$ are two attributes of the element v. Let $(X_{k;1}^{(j)})_{k\in\mathbb{Z},j=1,2,3}$ and $(X_{k;2}^{(j)})_{k\in\mathbb{Z},j=1,2,3}$ be the triple G-PoissonTower with parameter m for $\mathbf{x}(1)$ and $\mathbf{x}(2)$ respectively. Define the union estimator $\psi_\cup$ as*

$$\psi_\cup\left(\left(X_{k;1}^{(j)}, X_{k;2}^{(j)}\right)_{k\in\mathbb{Z},j=1,2,3}\right) = \frac{1}{p^2}\sum_{\gamma_1=0}^{p-1}\sum_{\gamma_2=0}^{p-1} \frac{\prod_{j=1}^{3}\left(\sum_{k=-\infty}^{\infty}\left(1 - e^{2\pi\left(X_{k;1}^{(j)}\gamma_1 + X_{k;2}^{(j)}\gamma_2\right)i/p}\right)e^{\frac{1}{3}k/m}\right)}{m^3|\Gamma(-1/3)|^3}.$$

$\psi_\cup$ estimates the number of elements with at least one non-zero attribute, i.e., the size of the union. Precisely, we have

$$\mathbb{E}\psi_\cup\left(\left(X_{k;1}^{(j)}, X_{k;2}^{(j)}\right)_{k\in\mathbb{Z},j=1,2,3}\right) = |\{v : \mathbf{x}(1)(v) \neq 0 \vee \mathbf{x}(2)(v) \neq 0\}| \cdot (1 + O(m^{-1})),$$
$$\mathbb{V}\psi_\cup\left(\left(X_{k;1}^{(j)}, X_{k;2}^{(j)}\right)_{k\in\mathbb{Z},j=1,2,3}\right) = |\{v : \mathbf{x}(1)(v) \neq 0 \vee \mathbf{x}(2)(v) \neq 0\}|^2 \cdot O(m^{-1}).$$

Linger and commit Let $[n]$ be a collection of users. Each user v has a state consisting of one integer x_v indicating how likely v is going to buy a product, and a boolean z_v indicating whether v has bought the product. The data stream now consists of three types of updates.

Like(v) $x_v \leftarrow x_v + 1$, user v became more likely to purchase the product.

Dislike(v) $x_v \leftarrow x_v - 1$, user v became less likely to purchase the product.

Commit(v) $z_v \leftarrow 1$, user v irrevocably committed a purchase.

Let us say now the marketing team wants to know how many users have either committed the purchase ($z_v = 1$) or are likely to purchase ($x_v > 0$). If the system is designed to be

strictly turnstile, i.e., $x_v \geq 0$ for all users v at all times, then this query becomes exactly the cardinality estimation problem in the M-turnstile model with $M = \mathbb{Z} \times (\{0,1\}, \mathrm{OR})$. One may use the collapsed M-PoissonTower to answer the query.

Theorem 6.7 (linger-and-commit estimator)**.** *Assume the stream is strictly turnstile and its length is at most p. Let $[n]$ be the universe of elements and the element state space be $M = \mathbb{Z}_p \times (\{0,1\}, \mathrm{OR})$. Let $(X_k^{(j)})_{k\in\mathbb{Z},j=1,2,3}$ be the triple collapsed M-PoissonTower with parameter m. Then the linger-and-commit estimator ψ defined below $(1 \pm O(1/\sqrt{m}))$-approximates the number of users have either committed the purchase ($z_v = 1$) or are likely to purchase $x_v > 0$.*

$$\psi\left(\left(X_k^{(j)}\right)_{k\in\mathbb{Z},j=1,2,3}\right) = \frac{1}{p}\sum_{\gamma=0}^{p-1} \frac{\prod_{j=1}^{3}\left(\sum_{k=-\infty}^{\infty}\left(1 - e^{2\pi X_k^{(j)}\gamma i/p}\right)e^{\frac{1}{3}k/m}\right)}{m^3|\Gamma(-1/3)|^3},$$

where $e^{\infty i}$ is defined to be 0.

Note that by Theorem 6.1 the monoid $M = \mathbb{Z}_p \times (\{0,1\}, \mathrm{OR})$ will be decomposed to $G = \mathbb{Z}_p \times \{0\}$ and $H = \mathbb{Z}_p \times \{1\}$.

6.5 A Statistical Analysis of the Finite Field Random Projection Approach

The cardinality sketches we developed for generic groups and monoids in Sections 6.2 and 6.3 are rather complicated. In this section, I will analyze a simple 0-GRA based estimator for the finite field cardinality sketch. I will consider the random-projection version which is first used by Kane et al. [110]. The goal is to investigate how the finite field size can affect the quality of estimation (in the leading constants). I will also discuss about some implications of it to the L_0-estimation.

Background and Summary In 2023, Bæk, Pagh and Walzer [7] analyzed a set sparse recovery algorithm over $\mathbb{F}_2$. They commented in the introduction that "Linear sketches over finite fields are less well-studied [in comparison to sketches over $\mathbb{Z}$ and $\mathbb{R}$], but are natural in some applications." [7]. We have a similar motivation here for $\mathbb{F}_2$-PCSA. $\mathbb{F}_2$-PCSA can be used to count the number of elements that appear an odd number of times, or the number of 1s in a boolean system. In general, for a prime p, $\mathbb{F}_p$-PCSA can be used to count the number of elements whose frequencies are *not* divisible by p.

Now consider another series of finite fields. For any positive integer k, $\mathbb{F}_{2^k}$-PCSA can be used to estimate the number of non-zero x_vs where each x_v is $\mathbb{F}_{2^k}$-valued (thus can be stored as a k-bit string). The state x_v can be used to identify a vector of k boolean attributes of the element v. Then given a stream of attribute toggles (in the form of "toggle the jth attribute of the element v"), $\mathbb{F}_{2^k}$-PCSA estimates the number of elements who have at least one nonzero attribute.

When the prime p is large, the $\mathbb{F}_p$-turnstile model approximates the original $\mathbb{Z}$-turnstile model. Thus as an application of the generic F-PCSA algorithm, $\mathbb{F}_p$-PCSA can be used to estimate L_0 when p is properly selected. By different strategies of selecting the prime p, it provides the following space complexity upper bounds (assuming a random oracle). To get a $(1 \pm \epsilon)$-estimation of L_0 with $1/3$ error probability, one needs

- $O(\epsilon^{-2} \log n \cdot (\log \epsilon^{-1} + \log \log \|x\|_\infty))$ bits, matching the state of the art L_0-estimation algorithm by Kane, Nelson and Woodruff [110];
- $O(\epsilon^{-2} \log n \cdot \log \|x\|_\infty)$ bits, which is useful when the maximum frequency is small;
- $O(\epsilon^{-2} \log n \cdot \max(\log \frac{\|x\|_1}{\|x\|_0}, \log \epsilon^{-1}))$ bits, which is useful when the average non-zero frequency is low.

Note that the upper bounds above can be made sharper by an $O(\epsilon)$-*mass discount*, i.e. ignoring the top $O(\epsilon)$-fraction of elements with non-zero frequencies. The key observation is: The worst thing that can happen in $\mathbb{F}_p$-PCSA when x_v is unusually large for some element $v \in U$ is that v gets missed in the final estimator (as if v has never appeared in the stream). It is fine to miss an $O(\epsilon)$-fraction of elements to get a $(1 \pm \epsilon)$-estimation of $\|x\|_0$, which is why in $\mathbb{F}_p$-PCSA one may simply ignore the top $O(\epsilon)$-fraction of elements with high (even unbounded) frequencies.

Not all sketches enjoy this $O(\epsilon)$-mass discount feature. For example, the first L_0-estimation algorithm [59] uses a linear combination of p-stable random variables with a small p (e.g. $p < 0.01$) to approximate L_0, which is sensitive to sparse high frequencies. To see this, let x_1 be the only non-zero entry in the vector x. For a fixed p, no matter how small p is, the estimator $\widehat{\|x\|_0}_{p\text{-stable}}$ becomes arbitrarily large as x_1 increases to infinity, while $\|x\|_0$ is just 1 by construction.

Overview of F-PCSA Since F-PCSA is similar to PCSA [83], we will first recap the definition of PCSA.

PCSA Sketch

Definition 6.6 (PCSA [83])**.** Let $U = \{1, 2, \ldots, n\}$ be the universe. The PCSA sketch stores a table of bits, BITMAP$[i, j]$ where $i \in [m]$ and $j \in \mathbb{N}$. For every (i, j), BITMAP$[i, j]$ is initialized to zero. There is a hash function $h : U \to [m] \times \mathbb{N}$ such that for any element $v \in U$, $\mathbb{P}(h(v) = (i, j)) = \frac{1}{m}2^{-j}$ for each (i, j). Upon an insertion of $v \in U$ (or receiving $(v, 1)$ in the $\mathbb{N}$-turnstile model),

$$\text{BITMAP}[h(v)] \leftarrow \text{BITMAP}[h(v)] \vee 1.$$

See Figure 4.1a for an example of PCSA's memory snapshot.

Let Q_i be the position of the leftmost zero in the ith row. It is observed in [83] that whenever the cardinality doubles, Q_i increases by 1 on average. The PCSA's estimator is thus

$$\widehat{\|x\|_0}_{\mathsf{PCSA}} = \alpha_m 2^{\frac{1}{m}\sum_{i=1}^{m} Q_i},$$

where α_m is some normalization factor.

Middle Range Assumption. We assume the cardinality is in the middle range here, i.e. $\|x\|_0 \gg 1$ and $\|x\|_0 \ll |U|$. In this case, all bits (i, j) with $j \geq \log |U|$ are zeros with high probability. Thus we only need to store bits (i, j) with $j \in [0, \log |U|]$, which requires $(m \log |U|)$ bits of memory. For the case where $\|x\|_0 = \Theta(1)$ or $\|x\|_0 = \Theta(|U|)$, separate estimation methods are needed, which are out of the scope of this thesis.

F-PCSA Sketch The algebraic structure of finite fields brings us the following properties.

Lemma 6.2. *Let F be a finite field. Then the uniform distribution over F is well-defined.*

Let $R_1, R_2, \ldots, R_k$ be i.i.d. uniform random variables over F. Then for any non-zero $a \in F$ and $k \in \mathbb{Z}_+$,

$$aR_1 \sim R_1 \tag{6.2}$$

$$\sum_{i=1}^{k} R_i \sim R_1. \tag{6.3}$$

The lemma above enables us to count over F in an almost identical way as PCSA counts over $\mathbb{N}$.

Definition 6.7 (F-PCSA)**.** Let $U = \{1, 2, \ldots, n\}$ be the universe. The F-PCSA sketch stores a table of F-values, FIELDMAP[i,j] where $i \in [m]$ and $j \in \mathbb{N}$. For every (i, j), FIELDMAP$[i, j]$ is initialized to zero (the additive identity of F). There is a hash function $h : U \to [m] \times \mathbb{N}$ such that for any $v \in U$, $\mathbb{P}(h(v) = (i, j)) = \frac{1}{m}2^{-j}$ for each (i, j). There is another hash function $g : U \to F$ such that for any $v \in U$, $g(v)$ is uniformly random over F. Upon receiving (v, k) where $v \in U, k \in F$,

$$\text{FIELDMAP}[h(v)] \leftarrow \text{FIELDMAP}[h(v)] + k \cdot g(v).$$

Together with Lemma 6.2, it is immediate that, conditioning on the hash function h, the entry FIELDMAP$[i, j]$ behaves as follows.

$$\begin{aligned}
&\text{FIELDMAP}[i, j] = 0, \quad \text{if } x_v = 0 \text{ for all } v \in U \text{ that } h(v) = (i, j) \\
&\text{FIELDMAP}[i, j] \sim R, \quad \text{if } x_v \neq 0 \text{ for some } v \in U \text{ that } h(v) = (i, j),
\end{aligned}$$

where R is a uniform random variable over F. See Figure 4.1b for a diagram of F-PCSA's memory. F-PCSA behaves statistically the same with the original PCSA with the twist that "1"s becomes "R"s. This is the intuition why it is no harder to count in the F-turnstile model than in the $\mathbb{N}$-turnstile model. One needs $\log |F|$ bits per register to store F-values and thus F-PCSA needs $(m \log |U| \cdot \log |F|)$ bits of memory.

We use the *highest index* (the index of the rightmost non-zero value) estimator for F-PCSA. Let W_i be the highest index of non-zero entries in the ith row. Then the estimator is defined as

$$\widehat{\|x\|_{0;F}}_{F\text{-PCSA}} = \beta_m 2^{\frac{1}{m}\sum_{i=1}^{m} W_i},$$

where β_m is some normalization factor. We prove, under some simplification assumptions listed in Section 6.5.1, in Theorem 6.9 that $\widehat{\|x\|_{0;F}}_{F\text{-PCSA}}$ is unbiased with $O(1/\sqrt{m})$ relative error (the leading constant depends on $|F|$). See Table 6.1 for computed normalization factors and relative errors for a list of finite fields.

Organization F-PCSA is formally analyzed in Section 6.5.1. Applications of F-PCSA on L_0-estimation are discussed in Section 6.5.2.

$\|F\|$	$\psi_{\mathbb{E}}(\|F\|)$	$\psi_{\mathbb{V}}(\|F\|)$	Normalization Factor (β_m)	Relative Error
2	-0.1100	5.588	$1.079m$	$1.638/\sqrt{m}$
3	0.5487	4.321	$0.6836m$	$1.441/\sqrt{m}$
4	0.7903	4.002	$0.5782m$	$1.387/\sqrt{m}$
5	0.9173	3.861	$0.5295m$	$1.362/\sqrt{m}$
6	not a field	not a field	not a field	not a field
7	1.049	3.732	$0.4833m$	$1.339/\sqrt{m}$
8	1.088	3.697	$0.4704m$	$1.333/\sqrt{m}$
9	1.118	3.671	$0.4607m$	$1.328/\sqrt{m}$
10	not a field	not a field	not a field	not a field
2^8	1.326	3.512	$0.3989m$	$1.299/\sqrt{m}$
2^{32}	1.333	3.507	$0.3969m$	$1.298/\sqrt{m}$
∞	1.333	3.507	$0.3969m$	$1.298/\sqrt{m}$

Table 6.1: Asymptotic constants of F-PCSA as $m \to \infty$. By Theorem 6.9, the normalization factor is $2^{-\psi_{\mathbb{E}}(|F|)}m$ and the relative error is $\sqrt{m^{-1}(\log^2 2)\psi_{\mathbb{V}}(|F|)}$, where $\psi_{\mathbb{E}}(|F|)$ and $\psi_{\mathbb{V}}(|F|)$ are defined in Theorem 6.13 (Appendix 6.5.3). The values of $\psi_{\mathbb{E}}(|F|)$ and $\psi_{\mathbb{V}}(|F|)$ in the table are obtained through numerical integration. Interestingly, as $|F|$ goes to infinity, F-PCSA estimator statistically converges to the LogLog estimator [73] and the relative error computed here ($1.298/\sqrt{m}$) matches Durand and Flajolet's computation for the LogLog's relative error, which is $m^{-1/2}\sqrt{\frac{1}{12}\log^2 2 + \frac{1}{6}\pi^2} \approx 1.29806/\sqrt{m}$.

6.5.1 Analysis of F-PCSA

Simplification Assumptions Let me recap the assumptions we have been made in Chapters 5 and 6.

Assumption 1 (random oracle)**.** There is a random oracle that provides ideal hash functions h and g.

Assumption 2 (poissonization)**.** Since we only care about the cases where $\|x\|_0$ is in the middle range (see the Middle Range Assumption in Section 6.5), we may assume the following.

- Since every element is hashed to only one table entry (through the hash function $h(\cdot)$), there are negligible negative correlations between table entries. For example, if there is only one non-zero element, then a non-zero entry necessarily implies that all other entries are zeros. However, this correlation vanishes when $\|x\|_0 \gg 1$ and thus we will assume all the entries in FIELDMAP are *strictly* independent. I.e. when a new element comes in, instead of being hashed to only one entry, now every entry (i, j) has an independent chance of $\frac{1}{m}2^{-j}$ to be hashed to.

- The probability that one hits the (i, j) entry is $\frac{1}{m}2^{-j}$ which goes to $1 - e^{-\frac{1}{m}2^{-j}}$ as j becomes large. Since $\|x\|_0 \gg 1$ and the highest index is likely to be large, we will assume the probability of hitting the (i, j) entry is *exactly* $1 - e^{-\frac{1}{m}2^{-j}}$.

- Since $\|x\|_0$ is assumed to be in the middle range, we assume that the columns are indexed by *integers*, extending infinitely to both ends. E.g. the probability of hitting the $(i, -3)$ entry is now assumed to be $1 - e^{-\frac{2^3}{m}}$.

Assumption 3 (random offsetting)**.** We assume that for each $i \in [m]$, there is a random offset Θ_i uniformly drawn from $[0, 1)$ for the ith subsketch. Then the ith subsketch randomly ignores a $2^{-\Theta_i}$-fraction of the elements in the universe. Thus now the probability of hitting the (i, j) entry is $1 - e^{-m^{-1}2^{-j-\Theta_i}}$.

With all the assumptions above, we reconstruct the F-PCSA sketch (Definition 6.7) in the following subsection.

Construction of F-PCSA With the poissonization assumptions, now F-PCSA consists of m i.i.d. subsketches (rows in FIELDMAP). We will construct one subsketch now.

Let F be any finite field. Let the *memory state* $s = (s_i)_{-\infty}^{\infty}$ be a bilaterally infinite vector over F (corresponding to one row in FIELDMAP). We will refer the state vector x in the F-turnstile model (Section 2.1) as the *stream state*. Recall that the stream state is an n-vector over F where n is the size of the universe U.

We first draw a random offset $\Theta \in [0, 1)$. Then with the (poissonized) hash functions h and g provided by the random oracle, we generate two tables of random variables $H = (H_{v,i})_{v \in U, i \in \mathbb{Z}}$ and $G = (G_{v,i})_{v \in U, i \in \mathbb{Z}}$, where for each $v \in U$ and $i \in \mathbb{Z}$,

- $H_{v,i}$ is a $\{0, 1\}$-random variable[3] with $\mathbb{P}(H_{v,i} = 0) = e^{-m^{-1}2^{-i-\Theta}}$,

- $G_{v,i}$ is a uniform random variable over F.

The entries in H and G are all independent. The two tables H and G are independent.

The sketching process can be described as follows.

- Initially, both the memory state s and the stream state x are all zeros.

- Upon receiving (v, k) for $v \in U$ and $k \in F$,

[3] 0 is the additive identity in F; 1 is the multiplicative identity in F.

- $x_v \leftarrow x_v + k$ and
- for each $i \in \mathbb{Z}$
  - $s_i \leftarrow s_i + k \cdot H_{v,i} G_{v,i}$.

Note that the process described above faithfully simulates the F-PCSA in Definition 6.7 with the assumptions listed in Section 6.5.1.

We then formalize the intuition in Section 6.5.

Characterization of F-PCSA's States

Theorem 6.8 (characterization of F-PCSA's state)**.** *Fix a stream state x and an offset $\theta \in [0, 1)$. Let the memory state be $S = (S_i)_{i \in \mathbb{Z}}$.*[4] *Then*

- *S_is are mutually independent.*
- *$\mathbb{P}(S_i \neq 0) = (1 - e^{-\|x\|_0 m^{-1} e^{-i-\theta}})(1 - |F|^{-1})$, for $i \in \mathbb{Z}$.*

Proof. By construction, the memory state S and the stream state x are linearly related:

$$S_i = \sum_{v \in U} x_v H_{v,i} G_{v,i}. \tag{6.4}$$

Consequently, S_is are independent due to the independence of entries of H and G.

Recall that $G_{v,i}$s are i.i.d. uniform random variables over F. By Lemma 6.2, we have

$$\begin{aligned} S_i = \sum_{v \in U} x_v H_{v,i} G_{v,i} &= \sum_{v \in U : x_v H_{v,i} \neq 0} x_v H_{v,i} G_{v,i} \\ &\sim \mathbb{1}\left[\exists v, x_v H_{v,i} \neq 0\right] R_i, \end{aligned}$$

where R_i is uniformly random over F. Thus we know

$$\begin{aligned} \mathbb{P}(S_i \neq 0) &= \mathbb{P}(\exists v, x_v H_{v,i} \neq 0) \mathbb{P}(R_i \neq 0) \\ &= (1 - \mathbb{P}(\forall v, x_v H_{v,i} = 0))(1 - |F|^{-1}) \\ &= (1 - \prod_{v \in U} \mathbb{P}(x_v H_{v,i} = 0))(1 - |F|^{-1}) \end{aligned} \tag{6.5}$$

[4] We capitalize the memory state s when we emphasize it as a *random vector* for a fixed stream state x.

Recall that $H_{v,i}$ is a binary random variable with probability $e^{-2^{-i/m-\theta}}$ of being 0. Since F is a field, $x_v H_{v,i} = 0$ if and only if $(H_{v,i} = 0 \vee x_v = 0)$. Thus

$$\mathbb{P}(x_v H_{v,i} = 0) = \begin{cases} 1, & x_v = 0 \\ e^{-m^{-1}2^{-i-\theta}}, & x_v \neq 0 \end{cases},$$

and then by (6.5),

$$\mathbb{P}(S_i \neq 0) = (1 - e^{-\|x\|_0 m^{-1} 2^{-i-\theta}})(1 - |F|^{-1}).$$

□

With the characterization theorem, we now analyze the estimator of F-PCSA.

Estimation of F-PCSA The estimation is based on the highest non-zero position. The analysis is overall straightforward except for a technical asymptotic analysis included in the appendix.

Definition 6.8 (highest index/position)**.** Let S be the memory state and Θ be the random offset. Then the *highest index* B is defined as

$$B = \sup\{i \in \mathbb{Z} \mid S_i \neq 0\}$$

and the *highest position*[5] is defined as

$$W = B + \Theta.$$

Definition 6.9. Let $\nu : \mathbb{R} \times [0,1] \to \mathbb{R}$ be a function. For $z \in \mathbb{R}$ and $r \in [0,1]$,

$$\nu(z, r) = (1 - e^{-2^{-z}})r \prod_{j=1}^{\infty} \left(1 - (1 - e^{-2^{-z-j}})r\right)$$

Lemma 6.3 (p.d.f. of W)**.** *Let f_W be the probability density function of W. Then*

$$f_W(z) = \nu(z - \log_2(\|x\|_0 m^{-1}), 1 - |F|^{-1}). \tag{6.6}$$

[5]The highest position is just the smoothed version of the highest index.

Proof. For $i \in \mathbb{Z}$, by the characterization theorem (Theorem 6.8),

$$\mathbb{P}(B \le i|\Theta = \theta) = \prod_{j=i+1}^{\infty} \mathbb{P}(S_j = 0) = \prod_{j=i+1}^{\infty} \left(1 - (1 - e^{-\|x\|_0 m^{-1} 2^{-j-\theta}})(1 - |F|^{-1})\right)$$

Thus

$$\begin{aligned}
&\mathbb{P}(B = i|\Theta = \theta) \\
&= \mathbb{P}(B \le i+1|\Theta = \theta) - \mathbb{P}(B \le i|\Theta = \theta) \\
&= (1 - e^{-\|x\|_0 m^{-1} 2^{-i-\theta}})(1 - |F|^{-1}) \prod_{j=i+1}^{\infty} \left(1 - (1 - e^{-\|x\|_0 m^{-1} 2^{-j-\theta}})(1 - |F|^{-1})\right) \\
&= \nu(i + \theta - \log_2(\|x\|_0 m^{-1}), 1 - |F|^{-1}). \qquad (6.7)
\end{aligned}$$

Then we have

$$\begin{aligned}
\int_{-\infty}^{z_0} f_W(z)dz = \mathbb{P}(W \le z_0) = \mathbb{P}(B + \Theta \le z_0) &= \int_0^1 \mathbb{P}(B + \theta \le z_0|\Theta = \theta)\, d\theta \\
&= \int_0^1 \sum_{i=-\infty}^{\infty} \mathbb{1}\left[i + \theta \le z_0\right] \mathbb{P}(B = i|\Theta = \theta)\, d\theta
\end{aligned}$$

by (6.7),

$$\begin{aligned}
&= \int_0^1 \sum_{i=-\infty}^{\infty} \mathbb{1}\left[i + \theta \le z_0\right] \nu(i + \theta - \log(\|x\|_0 m^{-1}), 1 - |F|^{-1})\, d\theta \\
&= \int_{-\infty}^{\infty} \mathbb{1}\left[z \le z_0\right] \nu(z - \log_2(\|x\|_0 m^{-1}), 1 - |F|^{-1})\, dz \\
&= \int_{-\infty}^{z_0} \nu(z - \log_2(\|x\|_0 m^{-1}), 1 - |F|^{-1})\, dz.
\end{aligned}$$

Differentiate both sides by z_0 and we get (6.6). □

Definition 6.10. Let $\phi : (0,1) \times [0,1] \to \mathbb{R}$ be a function. For $t \in (0,1)$ and $r \in [0,1]$,

$$\phi(t, r) = \int_{-\infty}^{\infty} 2^{tz} \nu(z, r)\, dz.$$

Lemma 6.4 (base-2 moment generating function of W)**.** *For $t \in (0,1)$ and $|F| \ge 2$,*

$$\mathbb{E}2^{tW} = \|x\|_0^t m^{-t} \phi(t, 1 - |F|^{-1}).$$

Proof. By Lemma 6.3,

$$\mathbb{E}2^{tW} = \int_{-\infty}^{\infty} f_W(z)2^{tz}\,dz = \int_{-\infty}^{\infty} \nu(z - \log_2(\|x\|_0 m^{-1}), 1 - |F|^{-1})2^{tz}\,dz$$

set $s = z - \log_2(\|x\|_0 m^{-1})$

$$= \|x\|_0^t m^{-t} \int_{-\infty}^{\infty} \nu(s, 1 - |F|^{-1})2^{ts}\,ds$$

by the definition of $\phi(\cdot,\cdot)$ (Definition 6.10)

$$= \|x\|_0^t m^{-t}\phi(t, 1 - |F|^{-1}).$$

□

Theorem 6.9 (F-PCSA)**.** *Let F be any finite field and $m \geq 3$ be the number of i.i.d. subsketches. Let $W_1, W_2, \ldots, W_m$ be the highest positions of the subsketches. Then the F-PCSA estimator is defined as*

$$\widehat{\|x\|_0} = \phi(1/m, 1 - |F|^{-1})^{-m} m 2^{\frac{1}{m}\sum_{i=1}^m W_i}.$$

The estimator is unbiased,

$$\mathbb{E}\widehat{\|x\|_0} = \|x\|_0,$$

and has variance

$$\mathbb{V}\widehat{\|x\|_0} = (\phi(1/m, 1 - |F|^{-1})^{-2m}\phi(2/m, 1 - |F|^{-1})^m - 1)\|x\|_0^2.$$

When m is large,

$$\begin{aligned}\widehat{\|x\|_0} &= (2^{-\psi_{\mathbb{E}}(|F|)} + O(m^{-1}))m2^{\frac{1}{m}\sum_{i=1}^m W_i} \\ \mathbb{V}\widehat{\|x\|_0} &= \|x\|_0^2 m^{-1}\left((\log^2 2)\psi_{\mathbb{V}}(|F|) + O(m^{-1})\right),\end{aligned}$$

where $\psi_{\mathbb{E}}$ and $\psi_{\mathbb{V}}$ are defined in Theorem 6.13 in Appendix 6.5.3. The relative error is bounded

$$\sqrt{(\log^2 2)\psi_{\mathbb{V}}(|F|)} < \sqrt{79}, \tag{6.8}$$

for all finite fields F.

Remark 6.2. The rough bound (6.8) is only to show the relative error is $O(1)$ regardless of F. The actual relative errors are much smaller. See Table 6.1 on page 263 for some computed values of relative errors.

Proof. Since W_is are independent, we have

$$\begin{aligned}\widehat{\mathbb{E}\|x\|_0} &= \phi(1/m, 1-|F|^{-1})^{-m} m \mathbb{E} 2^{\frac{1}{m}\sum_{i=1}^m W_i} \\ &= \phi(1/m, 1-|F|^{-1})^{-m} m \left(\mathbb{E} 2^{\frac{1}{m}W_1}\right)^m\end{aligned}$$

by Lemma 6.4,

$$\begin{aligned}&= \phi(1/m, 1-|F|^{-1})^{-m} m \left(\|x\|_0^{1/m} m^{-1/m} \phi(1/m, 1-|F|^{-1})\right)^m \\ &= \|x\|_0.\end{aligned}$$

Thus $\widehat{\|x\|_0}$ is unbiased. Similarly,

$$\begin{aligned}\widehat{\mathbb{E}\|x\|_0}^2 &= \phi(1/m, 1-|F|^{-1})^{-2m} m^2 \left(\mathbb{E} 2^{\frac{2}{m}W_1}\right)^m \\ &= \phi(1/m, 1-|F|^{-1})^{-2m} m^2 \left(\|x\|_0^{2/m} m^{-2/m} \phi(2/m, 1-|F|^{-1})\right)^m \\ &= \|x\|_0^2 \phi(1/m, 1-|F|^{-1})^{-2m} \phi(2/m, 1-|F|^{-1})^m.\end{aligned}$$

The variance is obtained by the formula $\mathbb{V}\widehat{\|x\|_0} = \widehat{\mathbb{E}\|x\|_0}^2 - (\widehat{\mathbb{E}\|x\|_0})^2$.

The asymptotic case when m is large is proved in Theorem 6.13 in Appendix 6.5.3.

□

6.5.2 Applications to L_0-Estimation

First note that an unbiased algorithm with relative error $O(\epsilon)$ implies a $(1 \pm \epsilon)$-estimation algorithm with constant error probability by Chebyshev's inequality. Thus F-PCSA gives an $(1 \pm \epsilon)$-estimation of $\|x\|_{0;F}$ with a constant error probability in the F-turnstile model using $O(\epsilon^{-2} \log n \cdot \log |F|)$ bits of memory.

Let us show that it is possible to achieve the same space complexity of KNW-L_0 ([110]) with $\mathbb{F}_p$-PCSA using a similar p-selection trick. Instead of bounding the probabilities of all the "bad events" as in [110], for $\mathbb{F}_p$-PCSA we only need to estimate the difference between

$\|x\|_0$ and $\|x\|_{0;\mathbb{F}_p}$, where $\|x\|_0$ is the quantity we *want* to estimate and $\|x\|_{0;\mathbb{F}_p}$ is the quantity we *can* estimate accurately by $\mathbb{F}_p$-PCSA.

Theorem 6.10. *Let $\epsilon > 0$ be the desired error. Let P be a random prime uniformly drawn from the set S of the smallest $\frac{10\log\|x\|_\infty}{\epsilon}$ primes. Then for any $x \in \mathbb{Z}^n$*

$$\|x\|_0 \geq \|x\|_{0;\mathbb{F}_P}, \quad \textit{and} \quad \mathbb{P}(\|x\|_0 - \|x\|_{0;\mathbb{F}_P} > \epsilon\|x\|_0) \leq \frac{1}{10}.$$

Proof. Since for any $a \in \mathbb{Z}$, $a = 0$ implies $a = 0 \mod P$, we have $\|x\|_0 \geq \|x\|_{0;\mathbb{F}_P}$.

Imagine every element is placing balls in a set of bins indexed by S. Each element $v \in U$ will place a ball in the bin indexed by $p \in S$ if $p \mid x_v$. By randomly picking a prime P from the set S, $\|x\|_0 - \|x\|_{0;\mathbb{F}_P}$ is equal to the number of element v with $P \mid x_v$, which is equal to the number of balls in the bin indexed by P. Note that for every element $v \in U$, x_v has at most $\log\|x\|_\infty$ distinct prime factors. Thus the number of balls is at most $\|x\|_0 \log\|x\|_\infty$. As a result, the number of bins with at least $\epsilon\|x_0\|$ balls is at most $\log\|x\|_\infty/\epsilon$ (otherwise we run out of balls). We conclude that

$$\begin{aligned}\mathbb{P}(\|x\|_0 - \|x\|_{0;\mathbb{F}_P} > \epsilon\|x\|_0) &\leq \frac{\text{number of bins with at least } \epsilon\|x\|_0 \text{ balls}}{\text{number of bins}} \\ &\leq \frac{\log\|x\|_\infty/\epsilon}{10\log\|x\|_\infty/\epsilon} = 1/10.\end{aligned}$$

□

To store a prime in Theorem 6.10, one needs $(\log\epsilon^{-1} + \log\log\|x\|_\infty + O(1))$ bits of memory. Therefore, by choosing a random p with the scheme in Theorem 6.10, $\mathbb{F}_p$-PCSA gets the same space complexity $O(\epsilon^{-2}\log n(\log\epsilon^{-1} + \log\log\|x\|_\infty))$ with KNW-L_0.

It is possible to select the prime with respect to other norms of the stream, which gives other choices of space bounds.

$O(\epsilon^{-2}\log n \cdot \log\|x\|_\infty)$ **space** Let p be a prime number. Trivially, if $\|x\|_\infty < p$ then $\|x\|_0 = \|x\|_{0;\mathbb{F}_p}$. Therefore by choosing p of size $\tilde{O}(\|x\|_\infty)$, $\|x\|_0$ is perfectly estimated by $\mathbb{F}_p$-PCSA, which uses $O(\epsilon^{-2}\log n\log\|x\|_\infty)$ space. Most notably, if the system is boolean (x_vs toggle between 0 and 1) or the system's insertions and deletions are interleaved, then $\|x\|_\infty < 2$. In such case, $\mathbb{F}_2$-PCSA estimates $\|x\|_0$ perfectly with only $O(\epsilon^{-2}\log n)$ bits of memory.

$O(\epsilon^{-2}\log n \cdot \max(\log(\frac{\|x\|_1}{\|x\|_0}), \log(\epsilon^{-1})))$ **space** Observe the following.

Theorem 6.11. *Let $\epsilon > 0$ be the desired error. Let $S = \{p_1, p_2, \ldots, p_K\}$ be a set of K distinct primes such that $\min S \geq \|x\|_1/\|x\|_0$ and $K \geq 10/\epsilon$. Let P be a random prime uniformly drawn from S. Then for any $x \in \mathbb{Z}^n$,*

$$\|x\|_0 \geq \|x\|_{0;\mathbb{F}_P}, \quad \textit{and} \quad \mathbb{P}(\|x\|_0 - \|x\|_{0;\mathbb{F}_P} > \epsilon\|x\|_0) \leq \frac{1}{10}.$$

Proof. We only need to prove the second inequality. Let $y_k = |\{v : p_k | x_v, x_v \neq 0\}|$, i.e. the number of non-zero x_vs that have prime factor p_k. Without loss of generality, assume $x_1, x_2, \ldots, x_\tau$ are positive and $x_{\tau+1}, \ldots, x_n$ are zeros. Then by the AM-GM inequality,

$$\frac{\sum_{i=1}^{\tau} x_i}{\tau} \geq \left(\prod_{i=1}^{\tau} x_i\right)^{\tau^{-1}} \geq \left(\prod_{i=1}^{K} p_i^{y_i}\right)^{\tau^{-1}} \geq p_*^{\tau^{-1}\sum_{i=1}^{K} y_i},$$

where $p_* = \min S$. Thus we have $\sum_{i=1}^{K} y_i \leq \tau \frac{\log(\|x\|_1/\tau)}{\log p_*}$. Note that

$$\mathbb{P}(\|x\|_0 - \|x\|_{0;\mathbb{F}_P} \geq \epsilon\|x\|_0) = \sum_{i=1}^{K} \frac{1}{K} \mathbb{1}\left[y_k \geq \epsilon\|x\|_0\right]$$

where we use $\mathbb{1}\left[y_k \geq \epsilon\|x\|_0\right] \leq \frac{y_k}{\epsilon\|x\|_0}$,

$$\leq \sum_{i=1}^{K} \frac{1}{K\epsilon\|x\|_0} y_k = \frac{1}{K\epsilon\|x\|_0} \sum_{i=1}^{K} y_k.$$

since $\sum_{i=1}^{K} y_i \leq \tau \frac{\log(\|x\|_1/\tau)}{\log p_*}$

$$\leq \tau \frac{\log(\|x\|_1/\tau)}{\log p_*} \frac{1}{K\epsilon\|x\|_0}$$

by definition, $\tau = \|x\|_0$

$$= \frac{\log(\|x\|_1/\|x\|_0)}{\log p_*} \frac{1}{K\epsilon}$$

By assumption, $p_* = \min S \geq \|x\|_1/\|x\|_0$ and $K \geq 10/\epsilon$. We conclude that $\mathbb{P}(\|x\|_0 - \|x\|_{0;\mathbb{F}_P} \geq \epsilon\|x\|_0) \leq 1/10$. □

To store a prime in Theorem 6.11, one needs $\log(\tilde{O}(\max(\|x\|_1/\|x\|_0, \epsilon^{-1})))$ bits of space. Therefore, by choosing a random p with the scheme in Theorem 6.11, $\mathbb{F}_p$-PCSA gets a new space complexity $O(\epsilon^{-2} \log n \cdot \max(\log(\frac{\|x\|_1}{\|x\|_0}), \log(\epsilon^{-1})))$ to give a $(1 \pm \epsilon)$-estimation of L_0 with

a constant error probability.

Note that $\|x\|_1/\|x\|_0$ is precisely the *average non-zero frequency*. Thus this scheme is particularly space-saving in a system where the maximum frequency is unbounded but the average frequency is small.

Remark 6.3. Similar upper bounds can be proved in the same way with $\frac{\|x\|_1}{\|x\|_0}$ replaced by $(\frac{\sum_{i=1}^{\tau} x_i^{\gamma}}{\|x\|_0})^{\gamma^{-1}}$ for any $\gamma \in (0,1)$ by realizing that $\frac{\sum_{i=1}^{\tau} x_i^{\gamma}}{\tau} \geq (\prod_{i=1}^{\tau} x_i)^{\gamma\tau^{-1}}$.

$O(\epsilon)$-Mass Discount

Theorem 6.12. *Given a state vector $x \in \mathbb{Z}^n$, without loss of generality, we assume*

$$|x_1| \geq |x_2| \geq \ldots \geq |x_\tau| > 0 = |x_{\tau+1}| = \ldots = |x_n|,$$

where $\tau = \|x\|_0$. We define the $(\epsilon/2)$-mass discounted vector $x^{(\epsilon/2)}$ as follows.

$$\begin{aligned} x_i^{(\epsilon/2)} &= 0, \quad i \leq \epsilon\tau/2 \\ x_i^{(\epsilon/2)} &= x_i, \quad i > \epsilon\tau/2. \end{aligned}$$

Suppose P is a random prime chosen by some scheme such that

$$\mathbb{P}(\|x^{(\epsilon/2)}\|_0 - \|x^{(\epsilon/2)}\|_{0;\mathbb{F}_P} > \frac{\epsilon}{2}\|x^{(\epsilon/2)}\|_0) \leq \frac{1}{10},$$

then

$$\mathbb{P}(\|x\|_0 - \|x\|_{0;\mathbb{F}_P} > \epsilon\|x\|_0) \leq \frac{1}{10}.$$

Proof. By construction we have $\|x\|_0 - \|x^{(\epsilon/2)}\|_0 = \frac{\epsilon}{2}\|x\|_0$. Thus we have

$$\begin{aligned} &\mathbb{P}(\|x\|_0 - \|x\|_{0;\mathbb{F}_P} > \epsilon\|x\|_0) \\ &= \mathbb{P}(\|x^{(\epsilon/2)}\| - \|x\|_{0;\mathbb{F}_P} > \frac{\epsilon}{2}\|x\|_0) \end{aligned}$$

where we always have $\|x\|_{0;\mathbb{F}_P} \geq \|x^{(\epsilon/2)}\|_{0;\mathbb{F}_P}$ and $\|x\|_0 \geq \|x^{(\epsilon/2)}\|_0$, so

$$\leq \mathbb{P}(\|x^{(\epsilon/2)}\| - \|x^{(\epsilon/2)}\|_{0;\mathbb{F}_P} > \frac{\epsilon}{2}\|x^{(\epsilon/2)}\|_0) \leq 1/10.$$

□

This theorem formally proves that we may replace x by $x^{(\epsilon/2)}$ in the upper bounds we

derived earlier in this section.

6.5.3 Asymptotic Analysis

Lemma 6.5. *Let X_r be a random variable with density $\nu(z,r)$ and $r \in [1/2,1]$. Then X_r is sub-exponential. Specifically,*

$$\mathbb{E}2^{|X_r|/6} < 4.$$

Proof. Note that

$$\mathbb{E}2^{|X_r|/6} = \int_{-\infty}^{0} 2^{-z/6}\nu(z,r)\,dz + \int_{0}^{\infty} 2^{z/6}\nu(z,r)\,dz.$$

We estimate the two parts separately. Note that $\nu(z,r) \le (1-e^{-2^{-z}})$ since $r \le 1$. We first estimate

$$\int_{0}^{\infty} 2^{z/6}\nu(z,r)\,dz \le \int_{0}^{\infty} 2^{z/6}(1-e^{-2^{-z}})\,dz < 2$$

Note that since $r \in [1/2,1]$, $\nu(z,r) \le \prod_{j=1}^{\infty}(1-(1-e^{-2^{-z-j}})/2)$. Then we estimate

$$\int_{-\infty}^{0} 2^{-z/6}\nu(z,r)\,dz \le \int_{-\infty}^{0} 2^{-z/6}\prod_{j=1}^{\infty}(1-(1-e^{-2^{-z-j}})/2)\,dz < 2$$

We conclude that

$$\mathbb{E}2^{|X_r|/6} < 2+2 = 4, \tag{6.9}$$

from which we know X_r is sub-exponential. □

Lemma 6.6. *Let X_r be a random variable with density $\nu(z,r)$ and $r \in [1/2,1]$. Then*

$$\log \mathbb{E}e^{tX_r} = t\mathbb{E}X_r + \frac{t^2}{2}\mathbb{V}X_r + O(t^3), \quad t \to 0$$

Proof. Note that by the definition of the exponential function,

$$\mathbb{E}e^{tX_r} = \sum_{k=0}^{\infty}\frac{t^k}{k!}\mathbb{E}X_r^k = 1 + t\mathbb{E}X_r + \frac{t^2}{2}\mathbb{E}X_r^2 + \sum_{k=3}^{\infty}\frac{t^k}{k!}\mathbb{E}X_r^k.$$

We want to prove $\mathbb{E}e^{tX_r} = 1 + t\mathbb{E}X_r + \frac{t^2}{2}\mathbb{E}X_r^2 + O(t^3)$ as $t \to 0$. Then we need to bound the moments $\mathbb{E}|X_r|^k$. Since X_r is sub-exponential (Lemma 6.5), we can bound the moments in a standard way. For $k \geq 1$,

$$\mathbb{E}|X_r|^k = \mathbb{E}(2^{|X_r|/6} \cdot 2^{-|X_r|/6}|X_r|^k) = \int 2^{z/6} \cdot (2^{-z/6}z^k)\, d\mu_{|X_k|} \tag{6.10}$$

$$\leq \int 2^{z/6} \cdot \sup_{s\geq 0}(2^{-s/6}s^k)\, d\mu_{|X_k|} = \mathbb{E}2^{|X_r|/6} \sup_{s\geq 0}(2^{-s/6}s^k), \tag{6.11}$$

where $\sup_{s\geq 0}(2^{-s/6}s^k) = e^{-k}(6k/\log 2)^k$. Thus by Lemma 6.5 and (6.11)

$$\mathbb{E}|X_r|^k \leq 4e^{-k}(6k/\log 2)^k < 4 \cdot 3.2^k \cdot k^k, \tag{6.12}$$

Thus we have

$$\begin{aligned}|\mathbb{E}e^{tX_r} - (1 + t\mathbb{E}X_r + \frac{t^2}{2}\mathbb{E}X_r^2)| &= \left|\sum_{k=3}^{\infty} \frac{t^k}{k!}\mathbb{E}X_r^k\right| \leq \sum_{k=3}^{\infty} \frac{t^k}{k!}\mathbb{E}|X_r|^k \\ &\leq \sum_{k=3}^{\infty} \frac{t^k}{k!} 4 \cdot 3.2^k \cdot k^k\end{aligned}$$

By Stirling's formula, we know for $k \in \mathbb{N}$,

$$k! > \sqrt{2\pi k}(k/e)^k > e^{-k}k^k.$$

Thus

$$|\mathbb{E}e^{tX_r} - (1 + t\mathbb{E}X_r + \frac{t^2}{2}\mathbb{E}X_r^2)| \leq 4\sum_{k=3}^{\infty} t^k(3.2e)^k \leq 8t^3(3.2e)^3,$$

as soon as $3.2et < 1/2$. Thus we have

$$\mathbb{E}e^{tX_r} = 1 + t\mathbb{E}X_r + \frac{t^2}{2}\mathbb{E}X_r^2 + O(t^3), \quad t \to 0.$$

Note that $\log(1+x) = x - \frac{1}{2}x^2 + O(x^3)$ as $x \to 0$. Since X_r is sub-exponential (Lemma 6.5), both $|\mathbb{E}X_r|$ and $\mathbb{E}X_r^2$ are finite. Thus

$$\begin{aligned}\log \mathbb{E}e^{tX_r} &= t\mathbb{E}X_r + \frac{t^2}{2}\mathbb{E}X_r^2 - \frac{1}{2}t^2(\mathbb{E}X_r)^2 + O(t^3) \\ &= t\mathbb{E}X_r + \frac{t^2}{2}\mathbb{V}X_r + O(t^3), \quad t \to 0,\end{aligned}$$

since $\mathbb{V}X_r = \mathbb{E}X_r^2 - (\mathbb{E}X_r)^2$. $\square$

Theorem 6.13. *Define X_r to be a random variable with density $\nu(z,r)$ (see Definition 6.9). Then for $r \in [1/2, 1]$,*

$$\begin{aligned}\phi(1/m,r)^{-m} &= 2^{-\mathbb{E}X_r} + O(m^{-1}) \\ \frac{\phi(2/m,r)^m}{\phi(1/m,r)^{2m}} &= 1 + (\log^2 2)m^{-1}\mathbb{V}X_r + O(m^{-2}), \quad m \to \infty.\end{aligned}$$

Furthermore, for any $r \in [1/2, 1]$, $(\log^2 2)\mathbb{V}X_r < 79$. We define $\psi_{\mathbb{E}}(|F|) = \mathbb{E}X_{1-|F|^{-1}}$ and $\psi_{\mathbb{V}}(|F|) = \mathbb{V}X_{1-|F|^{-1}}$.

Proof. By construction, we have $\phi(t,r) = \int_{-\infty}^{\infty} 2^{tz}\nu(z,r)\,dz = \mathbb{E}2^{tX_r}$. Then we have

$$\log(\phi(1/m,r)) = \log \mathbb{E}2^{X_r/m} = \log \mathbb{E}e^{(\log 2)X_r/m}$$

by Lemma 6.6, as $m \to \infty$,

$$= ((\log 2)/m)\mathbb{E}X_r + \frac{((\log 2)/m)^2}{2}\mathbb{V}X_r + O(m^{-3}).$$

Thus we have

$$\begin{aligned}\log(\phi(1/m,r)^{-m}) &= (-m)\log(\phi(1/m,r)) \\ &= (-m)\left(((\log 2)/m)\mathbb{E}X_r + \frac{(\log(2)/m)^2}{2}\mathbb{V}X_r + O(m^{-3})\right) \\ &= -(\log 2)\mathbb{E}X_r + O(m^{-1}). \\ \log\left(\frac{\phi(2/m,r)^m}{\phi(1/m,r)^{2m}}\right) &= m\log(\phi(2/m,r)) - 2m\log(\phi(1/m,r)) \\ &= m\left((2(\log 2)/m)\mathbb{E}X_r + \frac{(2\log(2)/m)^2}{2}\mathbb{V}X_r + O(m^{-3})\right) \\ &\quad - 2m\left(((\log 2)/m)\mathbb{E}X_r + \frac{(\log(2)/m)^2}{2}\mathbb{V}X_r + O(m^{-3})\right) \\ &= (\log^2 2)m^{-1}\mathbb{V}X_r + O(m^{-2}).\end{aligned}$$

Note that $e^x = 1 + x + O(x^2)$ as $x \to 0$. Thus

$$\begin{aligned}\phi(1/m,r)^{-m} &= e^{-(\log 2)\mathbb{E}X_r + O(m^{-1})} = e^{-(\log 2)\mathbb{E}X_r}(1 + O(m^{-1})) = 2^{-\mathbb{E}X_r} + O(m^{-1}). \\ \frac{\phi(2/m,r)^m}{\phi(1/m,r)^{2m}} &= e^{(\log^2 2)m^{-1}\mathbb{V}X_r + O(m^{-2})} = 1 + (\log^2 2)m^{-1}\mathbb{V}X_r + O(m^{-2}).\end{aligned}$$

Finally, to bound $\mathbb{V}X_r$, note that

$$\mathbb{V}X_r = \mathbb{E}X_r^2 - (\mathbb{E}X_r)^2 \leq \mathbb{E}X_r^2$$

by the moment bound (6.12)

$$\mathbb{E}X_r^2 \leq 4 \cdot 3.2^2 \cdot 2^2 < 164.$$

We conclude that $(\log^2 2)\mathbb{V}X_r < 79$. □

CHAPTER 7

Approximate Counting

> I looked in many directions for a solution to the problem with the following constraints in mind. Running time was important because the program was already painfully slow. There was no significant additional space available in the machine. On the other hand, precise counts of the events were not necessary since the processing was statistical in nature and a reasonable margin of error was tolerable.
>
> Robert Morris [128], 1978

Back in the 1970s, Robert Morris wanted to count the number of each trigram in texts [123, 133] where $d = 26^3$ different trigrams are counted simultaneously. His machine had so little storage that he could only use 8 bits for each trigram, which can only count up to 255. However, the frequencies of trigrams could easily go beyond 255 and he needed to come up with a way to count beyond 255 with an 8-bit counter.

Algorithm 7.1: Morris counter

Sketch : $v \in \mathbb{N}$, initialized as 0

Increment ( ):
```
// upon increment
```
$v \leftarrow v + 1$, with probability 2^{-v}

end

Query ( ):
```
// upon query
```
return $2^v - 1$

end

In 1978, Morris [128] invented the above classic *approximate counter*. If we denote the register v in Algorithm 7.1 at the jth increment as U_j, then by construction one has $\mathbb{E}2^{U_j} - 1 = j$ for any $j \in \mathbb{N}$. Morris also showed that the estimate has a constant *relative* variance. Therefore $2^{U_j} - 1$ is a decent estimate of the true count j. If this amount of variance is fine

for the application, Morris can now use this approximate counter to count each trigram up to about $2^{255} \approx 5.8 \times 10^{76}$. In general, if a sharper estimate is needed, the Morris counter can be modified to return a $(1 \pm \epsilon)$-approximation of the true count using $\log_2 \log_2 n + O(\log_2 \epsilon^{-1})$ bits.

In the original paper [128], Morris analyzed the mean and variance of the estimates. Flajolet [80] analyzed the mean and variance of the *index* v of the Morris counter, from which a quantified space bound can be derived with Chebyshev's bound. Nelson and Yu [133] analyzed the *tail* of the index, obtaining a sharper dependence on the failure probability. Specifically, for increments up to λ and $\epsilon, \delta > 0$, if an estimate $\hat{\lambda}$ is produced with $\mathbb{P}(|\hat{\lambda} - \lambda| < \epsilon\lambda) \geq 1 - \delta$ then the Morris counter needs $O(\log \log n + \log \epsilon^{-1} + \log \log \delta^{-1})$ space, and this is proved to be optimal [133].[1] Such approximate counters are invented mainly to save space. However, it was noticed quite recently that approximate counters can also be much *faster* on modern hardware in comparison to the deterministic counter due to their low *write complexity* [153, 149, 105].

In this chapter, I will study two variants of this approximate counting problem.

- In Section 7.1, I consider the *multi-dimensional approximate counting* where for a dimension $d > 0$, each increment now chooses an axis $j \in [d]$ and the task is to $(1 \pm \epsilon)$-approximate the underlying *count vector*. I will present a simple and optimal algorithm for this task when the error metric is the Euclidean norm.

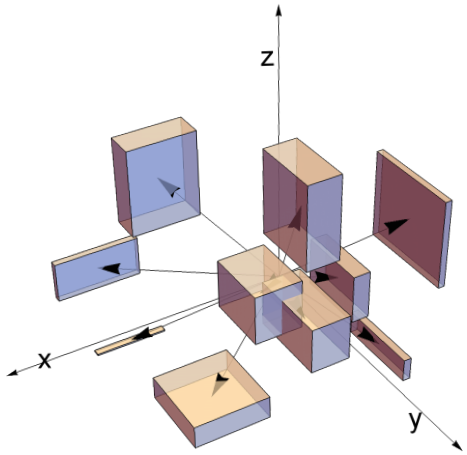
- In Section 7.2, I consider the *symmetric approximate counting* problem, where now both increments and decrements can be made to the counter. Of course, one may use two Morris counters—one for increments and one for decrements, and then take the difference. I will show that, a single Morris counter plus 1 sign-bit and 1 precision-bit can outperform the strategy of using two separate counters. And for some input sequence, the latter is *polynomially* more accurate.

7.1 Multi-dimensional Approximate Counting

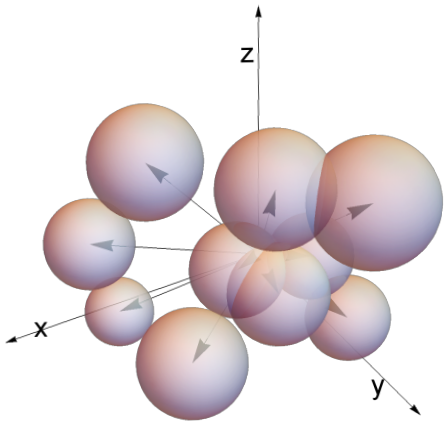
It is the common scenario in real-world applications that many different counters are maintained simultaneously. For example, the original motivation of Morris is to count the number of each trigram in texts [123, 133] where $d = 26^3$ different trigrams are counted simultaneously. Aden-Ali, Han, Nelson, and Yu [2] showed that, essentially, d independent counters

[1] Technically, one needs to keep a separate deterministic counter for small $\lambda = O(\epsilon^{-1})$ to obtain the optimal failure rate [133].

cannot be compressed as long as each counter is required to report an estimate with an error *relative to itself*. However, as we will later show, the space *can be compressed* if the error metric is the d-dimensional Euclidean norm. While the per-entry relative error requirement considered in [2] is reasonable when an approximation for *each individual entry* is needed, we remark that in applications where an approximation of the *whole count vector* is needed, the Euclidean error metric is more natural. See Fig. 7.1.



(a) Per-entry relative error box. Useful for unrelated entries—large entries won't affect the error guarantees for the small entries.



(b) Euclidean relative error box. Useful when the whole vector is a single entity such as a data embedding of the text for later use in, e.g., nearest neighbor search.

Figure 7.1: Comparison between per-entry error and Euclidean error

We now formally define the problem of d-dimensional counting in terms of Euclidean mean squared error. Let $(e_1, e_2, \ldots, e_d)$ be the standard basis of $\mathbb{R}^d$ where e_k has its kth component being one and other components being zero.

Definition 7.1 ((Euclidean) d-dimensional counting)**.** Let $n \geq 1$ be the maximum stream length, $d \geq 1$ be the dimension, and $\epsilon > 0$ be the relative error. An (n, d, ϵ)-*counter* is a randomized data structure that, for any input sequence $(w_1, \ldots, w_k) \in [d]^k$ with $k \leq n$, returns an estimate $\hat{x}$ of $x = \sum_{j=1}^{k} e_{w_j}$ with $\mathbb{E}|\hat{x} - x|^2 < \epsilon^2 |x|^2$, where $|x| = \sqrt{\sum_{j=1}^{d} x_j^2}$ is the Euclidean length. We denote the state space of the counter as Ω with $|\Omega| \geq 1$. One counter thus uses $\log_2 |\Omega|$ bits of space.

Recall that the original Morris counter [128] with parameter $a \geq 1$, denoted as Morris(a), returns an unbiased estimate of the count with relative variance $O(1/a)$. Thus by definition, Morris(a) is an $(n, 1, O(\sqrt{1/a}))$-counter. The error here is measured by the *mean squared*

error $\mathbb{E}|\hat{x} - x|^2$, which is equivalent to measure the variance if the estimator is unbiased. We note that there are, of course, many other reasonable error metrics to consider, depending on the specific applications. The mean squared error is a good starting point for a new algorithmic problem due to its simple probabilistic/geometric structures.

One natural way to construct a d-dimensional counter is to simply use d separate Morris counters, which is the original method that Morris used to count trigrams [123]. Suppose now we have d $(n, 1, \epsilon)$-counters to count $x_1, \ldots, x_d$. Let the estimates be $\widehat{x_1}, \ldots, \widehat{x_d}$ respectively. Then we have

$$\mathbb{E}|\hat{x} - x|^2 = \mathbb{E}\sum_{j=1}^{d} |\widehat{x_j} - x_j|^2 = \sum_{j=1}^{d} \mathbb{E}|\widehat{x_j} - x_j|^2$$

Since for any j, $\widehat{x_j}$ is an $(n, 1, \epsilon)$-counter for x_j, we have $\mathbb{E}|\widehat{x_j} - x_j|^2 < \epsilon^2 x_j^2$. Thus,

$$\mathbb{E}|\hat{x} - x|^2 < \sum_{j=1}^{d} \epsilon^2 x_j^2 = \epsilon^2 |x|^2.$$

We thus see d $(n, 1, \epsilon)$-counters do form an (n, d, ϵ)-counter (Definition 7.1). Each $(n, 1, \epsilon)$-counter can be implemented with a Morris$(O(\epsilon^{-2}))$ counter, taking $\log_2 \log_2 n + O(\log_2 \epsilon^{-1})$ bits. Thus the total space needed is $d \log_2 \log_2 n + O(d \log_2 \epsilon^{-1})$ bits, which is sub-optimal for this task.

The main contribution here is the construction of a simple and optimal (n, d, ϵ)-counter, with a matching lower bound.

Theorem 7.1. *The following statements are true.*

Upper bound *For any $n \geq 2$, $d \geq 1$, and $\epsilon \in (0, 1/3)$, there exists an (n, d, ϵ)-counter with space size $\log_2 |\Omega| = \log_2 \log_2 n + O(d \log_2 \epsilon^{-1})$.*

Lower bound *For any $n \geq e^2$, $d \geq 1$, and $\epsilon \in (0, 1/16)$, if there is an (n, d, ϵ)-counter with state space Ω, then $\log_2 |\Omega| \geq \log_2 \log_2 n + \Omega(d \log_2 \epsilon^{-1})$.*

A related data structure is the *Count-Min sketch* [62] by Cormode and Muthukrishnan. Given a count vector $x \in \mathbb{N}^d$ with $\sum_{j=1}^{d} x_j = n$, Count-Min is able to return an estimate $\widehat{x_j}$ of x_j, for any $j \in [d]$, such that $\widehat{x_j} \in [x_j, x_j + \epsilon n]$ with probability $1 - \delta$, using only $O(\epsilon^{-1} \log \delta^{-1} \log n)$ bits of space. It may seem that Count-Min with the power of hash functions has a chance to beat the lower bound in Theorem 7.1 by combining the estimates for each coordinate. However, it is hopeless to use Count-Min as a d-dimensional counter

unless $\epsilon = O(1/d)$, in which case it still lies above the lower bound in Theorem 7.1. Indeed, suppose $\widehat{x_j} \in [x_j, x_j + \epsilon n]$ holds for all j. The vector estimate $\hat{x} = (\widehat{x_1}, \ldots, \widehat{x_d})$ has error

$$|\hat{x} - x|^2 \leq d\epsilon^2 n^2 \leq d^2 \epsilon^2 |x|^2,$$

where we use the bound $|x|^2 \geq (\sum_{j=1}^{d} x_j)^2/d = n^2/d$. Therefore ϵ has to be $O(1/d)$ for $\hat{x}$ to be a multiplicative estimate of x.

7.1.1 Technical Overview

Upper Bound The upper bound in Theorem 7.1 uses the natural idea of maintaining a common *scale counter* U for all d dimensions and tracking the relative magnitude of each coordinate with a vector $V \in \mathbb{N}^d$. The estimate $\hat{x}$ is equal to $2^U V$. For demonstration, we will first describe a very simple but sub-optimal solution and then discuss how to modify it to reach optimal space-accuracy tradeoff. The transition of the states is designed similarly with the design of the Morris counter, in which the estimate is maintained *unbiased* at any moment ($\mathbb{E}2^U V = x$). Initially we set $U = 0$ and $V = (0, \ldots, 0)$.

- When the jth coordinate is incremented (i.e., $x_j \leftarrow x_j + 1$), we update $V_j \leftarrow V_j + 1$ with probability 2^{-U}, ensuring the unbiasedness of the estimate.
- Each coordinate V_j is restricted to the range $[0, a]$, so that the space to store V is $d \log_2(1 + a)$. Whenever some V_j exceeds a, we have to *scale up* to keep V_j in range. Again, we want to maintain the property that $\mathbb{E}2^U V = x$. Thus, the scale-up should be done as follows.
  - $U \leftarrow U + 1$. The scale counter is increased by 1.
  - For the estimate to be unbiased, V_j should become $V_j/2$ for every j. Here we also want V_j to always be an integer. Therefore, if V_j is even, then $V_j \leftarrow V_j/2$. If V_j is odd, then $V_j \leftarrow (V_j + \xi)/2$ where $\xi \in \{-1, 1\}$ is a freshly sampled Rademacher random variable.

The algorithm above is unbiased by design. However, the memory-accuracy tradeoff is *not* optimal. There is a fundamental problem of the design above that the parameter a has to be $\Omega(\sqrt{d})$ (with $\epsilon = O(1)$) for it to be a d-dimensional counter. Instead of analyzing the algorithm in detail, this problem can be identified by simply checking the set of *all possible*

estimates that are produced by the algorithm:

$$\mathcal{D} = \{2^U(V_1, \ldots, V_d) : U \in \mathbb{N}, \forall j \in [d], V_j \in [0, a]\}.$$

It turns out that $\mathcal{D}$ is not *dense enough* for estimating some input x. Indeed, we choose

$$x = 2^s(\underbrace{a, \ldots, a}_{r}, \underbrace{1/2, \ldots, 1/2}_{d-r}),$$

for some $r \in [0, d]$ so that both

$$y_1 = 2^s(a, \ldots, a, 1, \ldots, 1) \text{ and } y_2 = 2^{s-1}(a, \ldots, a, 1, \ldots, 1)$$

are far from x. By construction, the nearest estimate of x in $\mathcal{D}$ has to be either y_1 or y_2.[2] We choose r so that the two choices have equal distances to x. A simple calculation shows that we need to set $r = d/(1 + 4a^2)$.[3] Assume now $a < \sqrt{d}/3$ and then $r > 1$. Thus $\min_{y \in \mathcal{D}} |x - y|^2 = 2^{2s}\frac{da^2}{1+4a^2}$, while $|x|^2 = 2^{2s} \cdot 2a^2 d/(1 + 4a^2)$. Therefore, if the algorithm is an (n, d, ϵ)-counter, then $\mathbb{E}|\hat{x} - x|^2 < \epsilon^2|x|^2$, which means there is at least one $x_* \in \mathcal{D}$ such that $|x_* - x|^2 < \epsilon^2|x|^2$. This suggests that $2^{2s}a^2 d/(1 + 4a^2) < \epsilon^2 2^{2s} \cdot 2a^2 d/(1 + 4a^2)$ and therefore $\epsilon > 1/\sqrt{2}$. In other words, if $a < \sqrt{d}/3$, then any (n, d, ϵ)-counter must have $\epsilon > 1/\sqrt{2}$. We conclude that a needs to be $\Omega(\sqrt{d})$ when $\epsilon \leq 1/\sqrt{2}$, indicating an additional $O(d \log d)$ space overhead in comparison to the optimal space.

From the discussion above, we see that one has to have a dense enough set of estimates $\mathcal{D}$ for a correct (n, d, ϵ)-counter. Roughly speaking, one wants to spend more bits on large coordinates and fewer bits on small coordinates. This can be done with a simple and natural algorithmic idea: *variable-length integer encoding*. The usual binary encoding of non-negative integers automatically uses more bits on large numbers and fewer bits on small numbers. We will prove the following set of estimates is dense enough.

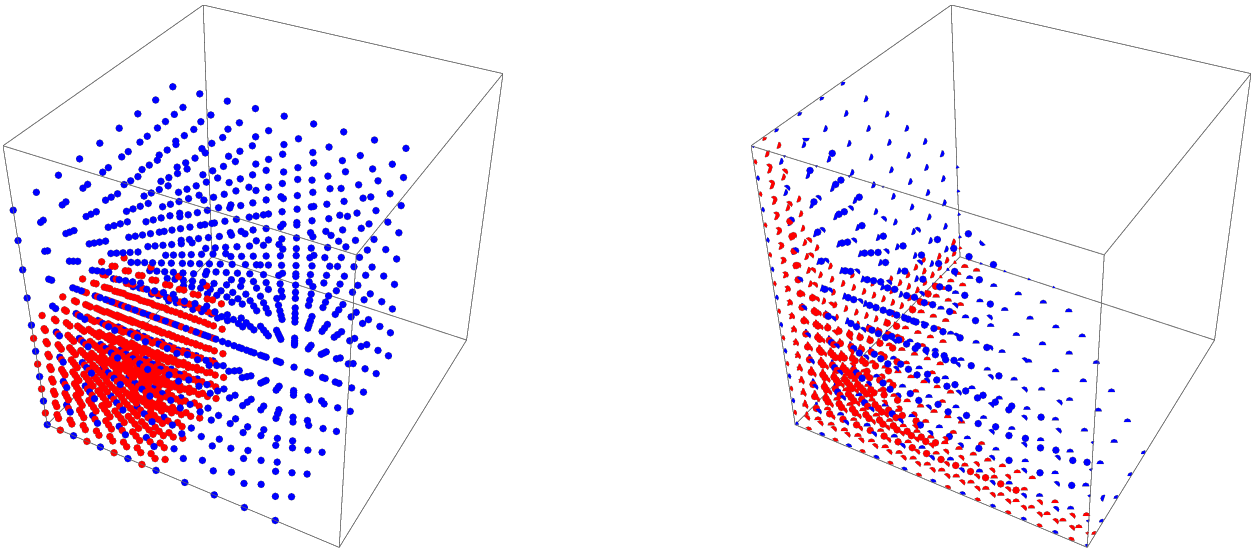
$$\mathcal{D}_{\text{compressed}} = \{2^U(V_1, \ldots, V_d) : U \in \mathbb{N}, \text{the encoding length of } V \text{ is at most } O(d \log_2 \epsilon^{-1})\},$$

This leads to our upper bound in Theorem 7.1. One may observe how this encoding idea better handles the previous counterexample $x = 2^s(a, \ldots, a, 1/2, \ldots, 1/2)$. An estimate of $2^{s-1}(2a, \ldots, 2a, 1, \ldots, 1)$ can now be returned because though each entry with value $2a$ needs

[2]Note that y_1 is one possible probabilistic rounding of x. Other roundings $\{2^s(a, \ldots, a, z_1, \ldots, z_{d-r}) \mid z_j \in \{0, 1\}\}$ all have the same distance to x. By symmetry, we only need to consider y_1. The vector y_2 is produced by decreasing the scale to 2^{s-1}, so the smaller entries can be represented precisely with the cost of underestimating the large entries.

[3]For simplicity, we only require $r > 1$ and omit the details of rounding r to an integer.

above average space to encode, each entry with value 1 uses less than average space to encode, *saving the memory space for the large entries*. See Fig. 7.2.



(a) Without compression ($\mathcal{D}$). (b) Without compression ($\mathcal{D}_{\text{compressed}}$).

Figure 7.2: Comparison between the set of estimates with and without compressing the relative vector V. Red dots are the set of estimates with the scale counter $U = r$. Blue dots are estimates with the scale counter $U = r + 1$. One observes that with compression, skewed vectors can be better represented without scaling up.

Lower Bound As we have discussed above, the set of all estimates produced by the counter has to be dense enough for the task of d-dimensional counting to be possible. The lower bound in Theorem 7.1 thus arises from estimating the size of *multiplicative space coverings*.

Definition 7.2 (multiplicative space covering)**.** For any subsets $A, R \subset \mathbb{R}^d$ and $\epsilon > 0$, we say R is a *ϵ-multiplicative covering* of A if for any $x \in A$ there exists $y \in R$ such that $|x - y| < \epsilon|x|$.

Clearly, the set of all estimates of an (n, d, ϵ)-counter will form a ϵ-multiplicative space covering of $\mathcal{X} = \{x \in \mathbb{N}^d \mid x_1 + \cdots + x_d \leq n\}$. The lower bound is thus proved by estimating the covering size of $\mathcal{X}$, which can be further reduced to the classic problem of covering spheres with smaller spheres [144].

Related Work There have been many variants and analyses of Morris counters before [115, 121, 91, 86, 133]. Nevertheless, they have only considered the one-dimensional case.

Tracking a multi-dimensional count vector (with a vector norm) is a new problem to the best of our knowledge.

In the one-dimensional case ($d = 1$), our algorithm is equivalent to the variant—*floating-point counter*—by Csűrös [65], except for that the floating-point counter restricts a to be a power of two so no probabilistic rounding is needed when halving. Nevertheless, when $d \geq 2$, probabilistic rounding is inevitable. In comparison to the classic Morris counter, the floating-point variant is much more convenient to implement on binary machines, where only simple integer operations and bit operations are needed [65]. Another advantage is that the number of random bits needed per increment is at most 2 in expectation, since the probability 2^{-U} can be simulated by generating random bits sequentially, looking for U consecutive ones [65] (stop generating random bits when a zero shows up). Such advantages are inherited by our d-dimensional counter as well.

7.1.2 Upper Bound: Variable-length Integer Encoding

We consider a specific integer encoding which is carefully designed to simplify the analysis later.

Definition 7.3 (variable-length integer encoding)**.** Let the symbol set be $\{0, 1, |\}$ where $|$ is used as a separator. We encode 0 as $|$, 1 as $0|$. For $k \geq 2$, we encode k as $[k-1]_2|$, where $[k-1]_2$ is the binary representation of $k-1$. See the following table.

integer	code	integer	code
0	$\|$	4	$11\|$
1	$0\|$	5	$100\|$
2	$1\|$	6	$101\|$
3	$10\|$	7	$110\|$

For any $k \in \mathbb{N}$, we define $\psi(k)$ to be the length of its encoding above. An integer vector $V \in \mathbb{N}^d$ is encoded by concatenating the codes for each of its component. For example $V = (3, 0, 4, 0, 1)$ is encoded as $10||11||0|$. The code length of the vector V is denoted by $\psi(V) = \sum_{j=1}^{d} \psi(V_j)$.

Remark 7.1. For clarity, the code here uses an alphabet of size 3, including a separator. If it is encoded with bits, then the length will increase by a $\log_2 3$ factor. See [135, 69] for techniques that encode m_* $\{0, 1, |\}$-symbols using $\lceil m_* \log_2 3 \rceil$ bits with constant querying time for each entry.

We now formally describe the new algorithm in Algorithm 7.2. The algorithm stores a *scale counter* $U \in \mathbb{N}$ and a *relative vector* $V = (V_1, \ldots, V_d) \in \mathbb{N}^d$. The estimator is $\hat{x} = 2^U V = (2^U V_1, \ldots, 2^U V_d)$. As discussed in the introduction, we will use the variable-length encoding in Definition 7.3 to store the vector V. Let M_* be the memory budget for V. Recall that $\psi(V)$ is the encoding length of V.

Algorithm 7.2: d-dimensional approximate counter

Sketch : $(U, V) \in \mathbb{N} \times \mathbb{N}^d$, initialized as $(0,(0,\ldots,0))$. V is compressed by Definition 7.3. $\psi(V)$ is the number of bits needed to encode V. M_* is the memory budget for V.

Increment (j)**:**
```
// upon incrementing the jth axis
V_j ← V_j + 1, with probability 2^{-U}
if ψ(V) > M_* then
    // runs out of budget for storing V; needs to scale up
    Scale-up ( )
end
```
end

Scale-up ( )**:**
```
// upon scale up
// increment the scale counter
U ← U + 1
// halve each entry of V
for k = 1, 2, ..., d do
    if V_k is even then
        V_k ← V_k/2
    else
        // V_k/2 is not representable; do probablistic rounding
        ξ ← a freashly sampled Rademacher random variable
        V_k ← (V_k + ξ)/2
    end
end
```
end

Query ( )**:**
```
// upon query
return 2^U V
```
end

Since the algorithm will execute **Scale-up** whenever the space $\psi(V)$ of storing V exceeds the budget M_*, the behavior of the algorithm will depend on the way that V is encoded. The codes in Definition 7.3 are constructed so that the space $\psi(V)$ increases by at most

one during `Increment` and decreases by at least one during **Scale-up**. This guarantees that the memory space to encode V is always bounded by M_*. We now list and prove all the properties of the encoding in Definition 7.3 that we will use in the analysis.

Lemma 7.1. *For any $V \in \mathbb{N}^d$, the following statements are true.*

- $\psi(V) \le 2d + \sum_{j=1}^{d} \log_2(1+V_j)$. *(Space bound.)*
- *For any $j \in [d]$, $\psi(V+e_j) \le \psi(V)+1$. (An increment increases the space by at most 1.)*
- $\psi(\lfloor V/2 \rfloor) \ge \psi(V) - 2d$. *(Halving the relative vector will decrease the space by at most 2d.)*
- *If there exists $j \in [d]$ such that $V_j \ge 3$, then $\psi(\lceil V/2 \rceil) \le \psi(V)-1$. (The space will decrease by at least one if not all V_js are small.)*

Remark 7.2. $\lfloor V/2 \rfloor$ and $\lceil V/2 \rceil$ are evaluated entry-wisely.

Proof. Recall that for the usual binary encoding, an integer k has length $1+\lfloor \log_2 k \rfloor$ if $k \ge 1$. Thus, by Definition 7.3, if $k \ge 2$, then $2+\lfloor \log_2(k-1) \rfloor$ symbols are needed to store the code $[k-1]_2|$. It requires one symbol to encode 0 and two symbols to encode 1. We bound the length for encoding k by $2+\log_2(1+k)$, which holds for all $k \in \mathbb{N}$. We then have $\psi(V) = \sum_{j=1}^{d} \psi(V_j) \le 2d + \sum_{j=1}^{d} \log_2(1+V_j)$.

The second statement is true by construction.

For the third statement, it suffices to prove that $\psi(\lfloor k/2 \rfloor) \ge \psi(k) - 2$ for any $k \in \mathbb{N}$. One may check that this is true for $k = 1, 2, 3, 4$. Now assume $k \ge 5$. Note that $\psi(\lfloor k/2 \rfloor) - \psi(k) = \lfloor \log_2(\lfloor k/2 \rfloor - 1) \rfloor - \lfloor \log_2(k-1) \rfloor = -2 + \lfloor \log_2(4\lfloor k/2 \rfloor - 4) \rfloor - \lfloor \log_2(k-1) \rfloor$. It suffices to prove that $4\lfloor k/2 \rfloor - 4 \ge k-1$. Indeed, $4\lfloor k/2 \rfloor - 4 \ge 2k-6 \ge k-1$ since $k \ge 5$.

For the fourth statement, since $\|V\|_\infty \ge 3$, there exists $j \in [d]$, such that $V_j \ge 3$. If V_j is even, then $V_j \ge 4$ and $\psi(V_j/2) = 2 + \lfloor \log_2(V_j/2-1) \rfloor = 1 + \lfloor \log_2(V_j-2) \rfloor \le \psi(V_j) - 1$. If V_j is odd, then $\psi(\lceil V_j/2 \rceil) = \psi((V_j+1)/2) = 2 + \lfloor \log_2((V_j+1)/2-1) \rfloor = 1 + \lfloor \log_2(V_j-1) \rfloor = \psi(V_j) - 1$. Note that ψ is non-decreasing and thus we conclude that $\psi(\lceil V/2 \rceil) \le \psi(V) - 1$. □

Next, we analyze the variance of the algorithm. Similar to the analysis of the Morris counter [128], the basic idea is to analyze the *change in variance* caused by each `Increment`, including a possible **Scale-up** step, and then compute the final variance by summing up the changes at time $k = 1, \ldots, n$. Recall that for any $r \in [d]$, e_r is the rth standard base vector of $\mathbb{R}^d$.

Theorem 7.2. *Fix any input sequence $(w_1, \dots, w_n) \in [d]^n$. Let (U, V) be the final state of the algorithm and $x = \sum_{j=1}^n e_{w_j}$ be the final count vector. Define $\hat{x} = 2^U V$. If the memory budget is $M_* = 4d + 2d \log_2(1 + a)$ with $a \geq 1$, then*

$$\mathbb{E}\hat{x} = x,$$
$$\mathbb{E}|\hat{x} - x|^2 \leq \frac{5}{6a - 2}|x|^2.$$

Furthermore for any $r \geq 1$,

$$\mathbb{P}\left(U \geq r + \log_2\left(\frac{n}{ad} + 1\right)\right) \leq 2^{-r}.$$

Proof. $\mathbb{E}\hat{x} = x$ is true by construction, since both `Increment` and **Scale-up** are designed to keep $\hat{x} = 2^U V$ unbiased. The challenge is to analyze the variance $\mathbb{E}|\hat{x} - x|^2$. We start by proving the following lemma that bounds the final variance by the distributional properties of the scale counter U at time $k = 1, \dots, n$.

Lemma 7.2 (variance decomposition)**.** *For any $k \leq n$, let $(U^{(k)}, V^{(k)})$ be the memory state at time k (i.e., after seeing $w_1, \dots, w_k$) and $x^{(k)} = \sum_{j=1}^k e_{w_j}$ be the count vector at time k. Let $\hat{x}^{(k)} = 2^{U^{(k)}} V^{(k)}$ to be the estimate at time k. Then*

$$\mathbb{E}|\hat{x}^{(n)}|^2 - |x^{(n)}|^2 \leq \sum_{k=0}^{n-1} \mathbb{E}(2^{U^{(k)}} - 1) + \frac{d}{3}\mathbb{E}(2^{2U^{(n)}} - 1).$$

Proof. We define $Y^{(k)}$ to be the intermediate vector after `Increment` before **Scale-up** at time k. By construction, we know $Y^{(k)} = V^{(k)} + e_{w_{k+1}}$ with probability $2^{-U^{(k)}}$; otherwise $Y^{(k)} = V^{(k)}$. For simplicity, we now fix $k \geq 0$ and write $U^{(k)}, V^{(k)}, x^{(k)}, w_{k+1}, Y^{(k)}$ as U, V, x, w, Y and $U^{(k+1)}, V^{(k+1)}, x^{(k+1)}$ as U', V', x' respectively. By the definition of `Increment`, we have

$$\mathbb{E}(|2^U Y|^2 - |2^U V|^2 \mid U, V) = 2^{-U} \cdot 2^{2U}(2V_w + 1) = 2^U(2V_w + 1). \tag{7.1}$$

Now if $\psi(Y) \leq M_*$, then we have $(U', V') = (U, Y)$ where no **Scale-up** is executed. If $\psi(Y) > M_*$, then we must have $\psi(Y) = M_* + 1$ because by Lemma 7.1, $\psi(Y) \leq \psi(V + e_w) \leq \psi(V) + 1$ and $\psi(V) \leq M_*$. By construction, $M_* \geq 3d$ and thus there exists $j \in [d]$ such that $Y_j \geq 3$. By Lemma 7.1, we know $\psi(\lceil Y/2 \rceil) \leq \psi(Y) - 1 = M_*$. Therefore, after the **Scale-up** step, the space will be at most M_*. When **Scale-up** is executed, $U' = U + 1$ and $V' = Y/2 + \sum_{k=1}^d \xi_k \mathbb{1}[2 \nmid Y_k]$. Here, $\xi_1, \dots, \xi_d$ are i.i.d. Rademacher random variables. Thus

we have

$$\begin{aligned}
&\mathbb{E}(|2^{U'}V'|^2 - |2^U Y|^2 \mid U, Y) \\
&= \mathbb{1}\left[\psi(Y) > M_*\right] \sum_{k=1}^{d} \mathbb{E}\left[\left(2^{2U+2}\left(\frac{Y_k + \xi_k \mathbb{1}\left[2 \nmid Y_k\right]}{2}\right)^2 - 2^{2U} Y_k^2\right) \,\middle|\, U, Y\right],
\end{aligned}$$

note that $\mathbb{E}\xi_k = 0$ and $\xi_k^2 = 1$

$$\begin{aligned}
&= \mathbb{1}\left[\psi(Y) > M_*\right] \sum_{k=1}^{d} \left(2^{2U}(Y_k^2 + \mathbb{1}\left[2 \nmid Y_k\right]) - 2^{2U} Y_k^2\right) \\
&= \mathbb{1}\left[\psi(Y) > M_*\right] 2^{2U} \sum_{k=1}^{d} \mathbb{1}\left[2 \nmid Y_k\right] \\
&\leq \mathbb{1}\left[\psi(Y) > M_*\right] 2^{2U} \cdot d. \qquad (7.2)
\end{aligned}$$

Take the total expectation and by Equation (7.1) and (7.2) we have,

$$\begin{aligned}
\mathbb{E}(|2^{U'}V'|^2 - |2^U V|^2) &= \mathbb{E}\mathbb{E}(|2^U Y|^2 - |2^U V|^2 \mid U, V) + \mathbb{E}\mathbb{E}(|2^{U'}V'|^2 - |2^U Y|^2 \mid U, Y) \\
&\leq \mathbb{E}(2^U(2V_w + 1)) + \mathbb{E}(\mathbb{1}\left[\psi(Y) > M_*\right] 2^{2U} d). \qquad (7.3)
\end{aligned}$$

Recall that $|x'|^2 - |x|^2 = (x_w + 1)^2 - x_w^2 = 2x_w + 1$ and $\mathbb{E}2^U V_w = x_w$. This implies $\mathbb{E}2^U V_w = (|x'|^2 - |x|^2 - 1)/2$. Thus by Equation (7.3) we have,

$$\mathbb{E}(|2^{U'}V'|^2 - |2^U V|^2) \leq |x'|^2 - |x|^2 + \mathbb{E}(2^U - 1) + \mathbb{E}(d2^{2U}\mathbb{1}\left[\psi(Y) > M_*\right]). \qquad (7.4)$$

Sum up the differences and we have

$$\mathbb{E}|2^{U^{(n)}}V^{(n)}|^2 = \sum_{k=0}^{n-1} \mathbb{E}(|2^{U^{(k+1)}}V^{(k+1)}|^2 - |2^{U^{(k)}}V^{(k)}|^2)$$

Recall that U', V', U, V are short for $U^{(k+1)}, V^{(k+1)}, U^{(k)}, V^{(k)}$. By Equation (7.4),

$$\begin{aligned}
&\leq \sum_{k=0}^{n-1} \left(|x^{(k+1)}|^2 - |x^{(k)}|^2 + \mathbb{E}(2^{U^{(k)}} - 1) + \mathbb{E}\left(d2^{2U^{(k)}}\mathbb{1}\left[\psi(Y^{(k)}) > M_*\right]\right)\right) \\
&= |x^{(n)}|^2 + \sum_{k=0}^{n-1} \mathbb{E}(2^{U^{(k)}} - 1) + d\mathbb{E}\left[\sum_{k=0}^{n-1} 2^{2U^{(k)}}\mathbb{1}\left[\psi(Y^{(k)}) > M_*\right]\right].
\end{aligned}$$

Note that $\mathbb{1}\left[\psi(Y^{(k)}) > M_*\right]$ is the indicator whether **Scale-up** is executed when processing

w_{k+1}. Therefore

$$\sum_{k=0}^{n-1} 2^{2U^{(k)}} \mathbb{1}\left[\psi(Y^{(k)}) > M_*\right] = \sum_{j=0}^{U^{(n)}-1} 2^{2j} = \frac{4^{U^{(n)}}-1}{4-1} = \frac{2^{2U^{(n)}}-1}{3}$$

We conclude that

$$\mathbb{E}|2^{U^{(n)}}V^{(n)}|^2 - |x^{(n)}|^2 \le \sum_{k=0}^{n-1} \mathbb{E}(2^{U^{(k)}}-1) + \frac{d}{3}\mathbb{E}(2^{2U^{(n)}}-1),$$

which gives the stated variance decomposition. □

The next step is to estimate $\mathbb{E}(2^U - 1)$ and $\mathbb{E}(2^{2U} - 1)$. The idea is to relate them with $\mathbb{E}2^U V = x$ and $\mathbb{E}|2^U V|^2 = \mathbb{E}|\hat{x}|^2$.

Lemma 7.3. *If $U \ge 1$, then $|V|^2 \ge \sum_{j=1}^d V_j \ge ad$.*

Proof. Since V_js are integer, we automatically have $|V|^2 = \sum_{j=1}^d V_j^2 \ge \sum_{j=1}^d V_j$. Thus it suffices to prove $\sum_{j=1}^d V_j \ge ad$. Note that if $U \ge 1$, then the algorithm has executed at least one **Scale-up**. Note that by Lemma 7.1, the space $\psi(V)$ increases at most by one per `Increment` and decreases at most by $2d$ per **Scale-up**. Therefore, after the first **Scale-up**, the memory usage $\psi(V)$ is between $M_* + 1 - 2d$ and M_*. Note that M_* is set to $4d + d\log_2(1+a)$. Thus we have

$$\psi(V) \ge M_* + 1 - 2d \ge 2d + d\log_2(1+a).$$

On the other hand, by Lemma 7.1 (encoding space bound), we have

$$\psi(V) \le 2d + \sum_{j=1}^d \log_2(1+V_j).$$

The bounds above imply that $\sum_{j=1}^d \log_2(1+V_j) \ge d\log_2(1+a)$. Since $\log_2(1+r)$ is convex in r, we conclude that

$$d\log_2(1+a) \le \sum_{j=1}^d \log_2(1+V_j) \le d\log_2\left(1 + \sum_{j=1}^d V_j/d\right),$$

and thus $\sum_{j=1}^d V_j \ge ad$. □

Note that if $U = 0$, then $2^U - 1 = 0$. With the lemma above, we now have

$$ad\mathbb{E}(2^U - 1) \leq \mathbb{E}(2^U - 1)\sum_{j=1}^{d} V_j \leq \mathbb{E}2^U \sum_{j=1}^{d} V_j = \sum_{j=1}^{d} x_j = k,$$

where k is the current time, i.e., the total number of increments so far. This implies $\mathbb{E}(2^U - 1) \leq k/(ad)$. Similarly, we have

$$ad\mathbb{E}(2^{2U} - 1) \leq \mathbb{E}(2^{2U} - 1)\sum_{j=1}^{d} V_j^2 \leq \mathbb{E}2^{2U} \sum_{j=1}^{d} V_j^2 = \mathbb{E}|\hat{x}|^2,$$

which implies $\mathbb{E}(2^{2U} - 1) \leq \mathbb{E}|\hat{x}|^2/(ad)$. Inserting the estimates back into Lemma 7.2, we have

$$\begin{aligned}\mathbb{E}|\hat{x}^{(n)}|^2 - |x^{(n)}|^2 &\leq \sum_{k=0}^{n-1} \mathbb{E}(2^{U^{(k)}} - 1) + \frac{d}{3}\mathbb{E}(2^{2U^{(n)}} - 1) \\ &\leq \sum_{k=0}^{n-1} k/(ad) + \frac{d}{3}\mathbb{E}|\hat{x}^{(n)}|^2/(ad)\end{aligned}$$

Note that $\sum_{k=0}^{n-1} k \leq n^2/2 \leq d|x^{(n)}|^2/2$ since $\sum_{j=1}^{d} x_j^{(n)} = n$ and $d|x^{(n)}|^2 \geq (\sum_{j=1}^{d} x_j^{(n)})^2$.

$$\leq \frac{1}{2a}|x^{(n)}|^2 + \frac{1}{3a}\mathbb{E}|\hat{x}^{(n)}|^2.$$

Reorganizing the inequality above, we have

$$\mathbb{E}|\hat{x}^{(n)}|^2 - |x^{(n)}|^2 \leq \frac{5}{6a-2}|x^{(n)}|^2.$$

Note that since the estimate is unbiased, we have $\mathbb{E}|\hat{x}^{(n)}|^2 - |x^{(n)}|^2 = \mathbb{E}|\hat{x}^{(n)} - x^{(n)}|^2$. Thus we conclude that $\mathbb{E}|\hat{x}^{(n)} - x^{(n)}|^2 \leq \frac{5}{6a-2}|x^{(n)}|^2$.

Finally we estimate the tail of U. We already know that $\mathbb{E}(2^{U^{(k)}} - 1) \leq k/(ad)$ and thus $\mathbb{E}2^{U^{(k)}} \leq 1 + k/(ad)$. Then for any $z > 0$, by Markov,

$$\mathbb{P}(2^{U^{(k)}} \geq z) \leq \frac{\mathbb{E}2^{U^{(k)}}}{z} \leq \left(\frac{k}{ad} + 1\right)/z$$

Setting $z = 2^r(k/(ad) + 1)$ and we have

$$\mathbb{P}\left(U^{(k)} \geq r + \log_2\left(\frac{k}{ad} + 1\right)\right) \leq 2^{-r}.$$

We thus have proved the tail bound for U. □

Theorem 7.2 characterizes the mean and variance of the estimator. We now analyze the space. The vector V is encoded with length at most M_* by construction. We are left to analyze the memory space needed for the scale counter U.

Corollary 7.1. *For $\epsilon \in (0, 1/3)$, there exists an (n, d, ϵ)-counter with state space size $\log_2 |\Omega| = \log_2 \log_2 n + O(d \log_2 \epsilon^{-1})$.*

Proof. If $n \leq \epsilon^{-1}$, we may use d deterministic counters taking $d \log_2 n \leq d \log_2 \epsilon^{-1}$ space which surely form a (n, d, ϵ)-counter. We thus assume $n > \epsilon^{-1}$.

With a fixed space budget for storing the scale counter U, there is a chance that U grows too large to be stored. In such case, the algorithm simply enters a fail state and returns $(0, \ldots, 0)$ upon query. For $\tau > 0$, we store U only up to $U_* = \log_2(\tau^{-1}) + \log_2(n/(ad) + 1)$. By design U is non-decreasing in time and it suffices to analyze U at the end of the stream. By Theorem 7.2, we know $\mathbb{P}(U \geq U_*) \leq \tau$. The estimate $\hat{x}$ can be written as $\mathbb{1}[U < U_*] 2^U V$. Its mean squared error is

$$\begin{aligned}\mathbb{E}\left|\mathbb{1}[U < U_*] 2^U V - x\right|^2 &= \mathbb{E}\left(\mathbb{1}[U < U_*] |2^U V - x|^2 + \mathbb{1}[U \geq U_*] |x|^2\right) \\ &\leq \mathbb{E}|2^U V - x|^2 + \mathbb{P}(U \geq U_*)|x|^2\end{aligned}$$

Note that $\mathbb{E}|2^U V - x|^2 \leq \frac{5}{6a-2}|x|^2$ by Theorem 7.2 and $\mathbb{P}(U \geq U_*) \leq \tau$ by construction.

$$\leq \left(\frac{5}{6a - 2} + \tau\right) |x|^2$$

We set $\tau = \epsilon^2/2$ and $a = 2\epsilon^{-2}$ and have

$$\frac{5}{6a - 2} + \tau = \frac{5}{12\epsilon^{-2} - 2} + \epsilon^2/2 \leq \epsilon^2,$$

since $\epsilon < 1$. In this setting, we have $\mathbb{E}|\mathbb{1}[U < U_*] 2^U V - x|^2 \leq \epsilon^2 |x|^2$ which implies the algorithm is an (n, d, ϵ)-counter. We now compute its space usage. By construction, the space used by U is at most $\log_2(\log_2(\tau^{-1}) + \log_2(n/(da) + 1))) \leq \log_2 \log_2 n + O(1)$ bits since we assumed $\tau^{-1} < n$. The space used by the vector V is by construction $M_* =$

$4d + d\log_2(1+a) = O(d\log\epsilon^{-1})$. Note that V is encoded with a alphabet set of size 3 ($\{0,1,|\}$) and M_* is the maximum number of symbols. One may encode V using bits by adding an additional $\log_2 3$ factor to the leading constant. We conclude that the total space needed to store (U,V) is $\log_2\log_2 n + O(d\log\epsilon^{-1})$. □

7.1.3 Lower Bound: Multiplicative Space Covering

Recall that by Definition 7.2, for any subset $A, R \subset \mathbb{R}^d$ and $\epsilon > 0$, we say R is an ϵ-multiplicative covering of A if for any $x \in A$ there exists $y \in R$ such that $|x-y| < \epsilon|x|$.

We now prove that any correct (n,d,ϵ)-counter will induce an $O(\epsilon)$-multiplicative covering of the space. The proof here will be slightly more complicated than what we described in §7.1.1. Previously we assumed the algorithm returns a fixed estimate for a given state and therefore the set of estimates naturally form a covering. In general, a probabilistic algorithm is allowed to produce a randomized answer even if the final memory state is fixed. We show that even when estimates are randomized for a given state, an (n,d,ϵ)-counter will still induce a covering of the space.

Lemma 7.4. *If Ω is the state space of an (n,d,ϵ)-counter with $\epsilon < 1/3$, then there exists a 4ϵ-multiplicative covering R of $\{x \in \mathbb{R}^d_+ \mid \sqrt{d}\epsilon^{-1} \le |x| \le n/\sqrt{d}\}$ with $|R| \le |\Omega|$.*

Proof. Let $g : \Omega \to \mathbb{R}^d$ be the randomized function that returns the estimate $\hat{x} = g(S)$ based on the final state $S \in \Omega$. Let $\mathcal{X} = \{x \in \mathbb{N}^d \mid \sum_{j=1}^d x_j \le n\}$ be the space of all possible count vectors. By definition of the (n,d,ϵ)-counter, for any $x \in \mathcal{X}$, we have

$$\mathbb{E}|g(S) - x|^2 < \epsilon^2|x|^2.$$

For any $x \in \mathcal{X}$, define $s_x = \arg\min_{s\in\Omega} \mathbb{E}|g(s) - x|^2$. Thus we have

$$\mathbb{E}|g(s_x) - x|^2 \le \mathbb{E}|g(S) - x|^2 < \epsilon^2|x|^2. \tag{7.5}$$

This implies $\mathbb{E}|g(s_x) - x| \le \sqrt{\mathbb{E}|g(s_x)-x|^2} < \epsilon|x|$. Suppose for some $y \in \mathcal{X}$ with $s_y = s_x$. By using the same randomness for $g(s_x)$ and $g(s_y)$, we have

$$|x-y| = |g(s_x) - x - (g(s_y) - y)| \le |g(s_x) - x| + |(g(s_y) - y)| \tag{7.6}$$

By Equation (7.5), we know $\mathbb{E}|g(s_x) - x| < \epsilon|x|^2$ and $\mathbb{E}|g(s_y) - y| < \epsilon|y|^2$. Take expectation on both sides of Equation (7.6) and we have

$$|x - y| \leq \mathbb{E}|g(s_x) - x| + \mathbb{E}|(g(s_y) - y)| < \epsilon(|x| + |y|). \tag{7.7}$$

We now group the count vectors x by the value s_x. For any $s \in \Omega$, let $A_s = \{x \in \mathcal{X} \mid s_x = s\}$ and $\{A_s | s \in \Omega\}$ form a partition of $\mathcal{X}$. For each s that $A_s \neq \emptyset$, we choose a representation $x_*^{(s)} \in A_s$ according to the lexicographic order. Next we are going to show that $R = \{x_*^{(s)} \mid s \in \Omega, A_s \neq \emptyset\}$ forms a good multiplicative covering of $\{x \in \mathbb{R}_+^d \mid d\epsilon^{-1} < |x| < n/d\}$.

For any $x \in \mathbb{R}_+^d$, denote $\lfloor x \rfloor = (\lfloor x_1 \rfloor, \ldots, \lfloor x_d \rfloor)$. We have $|x - \lfloor x \rfloor| < \sqrt{d}$. Note that

$$\sum_{j=1}^{d} \lfloor x_j \rfloor \leq \sqrt{d}|\lfloor x \rfloor| \leq \sqrt{d}|x|.$$

If $|x| \leq n/\sqrt{d}$, then $\sum_{j=1}^{d} \lfloor x_j \rfloor \leq \sqrt{d}|x| \leq n$, which implies that $\lfloor x \rfloor \in \mathcal{X}$. Set $y = x_*^{(s_{\lfloor x \rfloor})} \in R$. Since $s_y = s_{\lfloor x \rfloor}$, we have by Equation (7.7),

$$|y - \lfloor x \rfloor| < \epsilon(|y| + |\lfloor x \rfloor|) \leq \epsilon(|\lfloor x \rfloor| + |y - \lfloor x \rfloor| + |\lfloor x \rfloor|)$$

Reorganizing,

$$|y - \lfloor x \rfloor| < \frac{2\epsilon}{1 - \epsilon}|\lfloor x \rfloor|.$$

Thus we have

$$\begin{aligned} |y - x| &\leq |y - \lfloor x \rfloor| + |x - \lfloor x \rfloor| \\ &< \frac{2\epsilon}{1 - \epsilon}|\lfloor x \rfloor| + \sqrt{d} \\ &\leq \frac{2\epsilon}{1 - \epsilon}|x| + \sqrt{d}, \end{aligned}$$

If $|x| \geq \epsilon^{-1}\sqrt{d}$, then $\sqrt{d} \leq \epsilon|x|$. Thus we have

$$|y - x| < \frac{2\epsilon}{1 - \epsilon}|x| + \epsilon|x| = \frac{3\epsilon - \epsilon^2}{1 - \epsilon}|x|.$$

Since $\epsilon < 1/3$, we have $\frac{3\epsilon - \epsilon^2}{1 - \epsilon} \leq 4\epsilon$. Thus $|y - x| < 4\epsilon|x|$. Note that the above argument holds for any x with $|x| \in [\epsilon^{-1}\sqrt{d}, n/\sqrt{d}]$. Therefore R is a 4ϵ-multiplicative cover of $\{x \in \mathbb{R}_+^d \mid \epsilon^{-1}\sqrt{d} \leq |x| \leq n/\sqrt{d}\}$. Finally, note that the covering $R = \{x_*^{(s)} \mid s \in \Omega, A_s \neq \emptyset\}$ has size at

most $|\Omega|$. $\square$

Lemma 7.4 shows that the state space of an (n, d, ϵ)-counter will be at least the size of an optimal 4ϵ-multiplicative covering of the shell[4] $\{x \in \mathbb{R}_+^d \mid \sqrt{d}\epsilon^{-1} \leq |x| \leq n/\sqrt{d}\}$. We now bound the size of the shell covering. The first step is to bound the size of the sphere covering. Covering a sphere with smaller spheres is a classic problem [144]. We prove the following lemma based on [16].

Lemma 7.5. *For any $\epsilon \in (0, 1/3)$, if $R \subset \mathbb{R}^d$ is an ϵ-multiplicative cover of $\{x \in \mathbb{R}_+^d \mid |x| = 1\}$, then $|R| \geq 2^{-d}\epsilon^{-(d-1)}$.*

Proof. We first consider the case $d = 1$. The target set $\{x \in \mathbb{R}_+ \mid |x| = 1\} = \{1\}$ has size 1 and thus $|R| = 1 > 2^{-1}$. We now assume $d \geq 2$.

Let $\mathbb{S}^{d-1}$ be the unit sphere in $\mathbb{R}^d$. Let $|\mathbb{S}^{d-1}|$ be the surface area of the sphere. Let the maximum area on $\mathbb{S}^{d-1}$ that can be covered by a single ball with radius ϵ be η_ϵ. By Corollary 3.2 in [16],

$$\frac{|\mathbb{S}^{d-1}|}{\eta_\epsilon} \geq \sqrt{2\pi(d-1)}\frac{\sqrt{1-\epsilon^2}}{\epsilon^{(d-1)}},$$

if $\sqrt{1-\epsilon^2} \geq 1/\sqrt{d}$, which is true since we assumed $\epsilon < 1/3$ and $d \geq 2$.

On the other hand, by symmetry, $|\mathbb{S}^{d-1} \cap \mathbb{R}_+^d| = 2^{-d}|\mathbb{S}^{d-1}|$. Therefore, the number of points in R is at least

$$\frac{|\mathbb{S}^{d-1} \cap \mathbb{R}_+^d|}{\eta_\epsilon} \geq 2^{-d}\sqrt{2\pi(d-1)}\frac{\sqrt{1-\epsilon^2}}{\epsilon^{(d-1)}} \geq 2^{-d}\epsilon^{-(d-1)},$$

since $d \geq 2$ and $\sqrt{2\pi(1-1/9)} \geq 1$. $\square$

With the bound for the sphere covering in hand, the next step is to count the number of *layers* in the shell so that the coverings for different layers are disjoint. Define $\mathbb{S}_+^{d-1} = \mathbb{S}^{d-1} \cap \mathbb{R}_+^d$, $\alpha\mathbb{S}_+^{d-1} = \{x \in \mathbb{R}_+^d \mid |x| = \alpha\}$, and $[\alpha, \beta]\mathbb{S}_+^{d-1} = \{x \in \mathbb{R}_+^d \mid |x| \in [\alpha, \beta]\}$.

Corollary 7.2. *For $0 < \alpha < \beta$ and $\epsilon \in (0, 1/3)$, if R is an ϵ-multiplicative cover of $[\alpha, \beta]\mathbb{S}_+^{d-1}$, then $|R| \geq \frac{1}{3}2^{-d}\epsilon^{-d}\log(\beta/(e\alpha))$.*

Proof. For any $x \in \mathbb{R}^d$, if $|x - y| < \epsilon|x|$, then $|y| \in ((1-\epsilon)|x|, (1+\epsilon)|x|)$. Now define

[4]Technically, the shell intersected with the positive cone.

$b_k = (1+3\epsilon)^k$ for $k \in \mathbb{N}$. Define

$$R_k = R \cap \{x \in \mathbb{R}^d_+ : |x| \in ((1-\epsilon)b_k, (1+\epsilon)b_k)\}$$

for any $k \in \mathbb{N}$. It is straightforward to check that $b_k(1+\epsilon) < b_{k+1}(1-\epsilon)$ since $\epsilon < 1/3$. Therefore R_ks are disjoint for different ks. By construction, if $b_k \in [\alpha, \beta]$, then R_k covers $b_k \mathbb{S}^{d-1}_+$. By Lemma 7.5, we know $|R_k| \geq 2^{-d}\epsilon^{-(d-1)}$ (the *multiplicative* covering of a sphere does not depend on the radius.) Thus

$$|R| \geq \sum_{k:b_k \in [\alpha,\beta]} |R_k| \geq 2^{-d}\epsilon^{-(d-1)} \sum_{k:b_k \in [\alpha,\beta]} 1,$$

and we only need to count the number of b_ks that are in $[\alpha, \beta]$. Note that

$$b_k \in [\alpha, \beta] \iff k \log(1+3\epsilon) \in [\log\alpha, \log\beta] \iff k \in [\frac{\log\alpha}{\log(1+3\epsilon)}, \frac{\log\beta}{\log(1+3\epsilon)}].$$

We conclude that

$$|R| \geq 2^{-d}\epsilon^{-(d-1)}(\frac{\log\beta - \log\alpha}{\log(1+3\epsilon)} - 1)$$

relaxing $1 + 3\epsilon < e^{3\epsilon}$ and $1 \leq 1/(3\epsilon)$

$$\begin{aligned} &> 2^{-d}\epsilon^{-(d-1)}(\frac{1}{3\epsilon}\log(\beta/\alpha) - \frac{1}{3\epsilon}) \\ &= \frac{1}{3}2^{-d}\epsilon^{-d}\log(\beta/(e\alpha)), \end{aligned}$$

which lower bounds the size of ϵ-multiplicative covers of $[\alpha, \beta]\mathbb{S}^{d-1}_+$. □

We now finish the proof of the space lower bound of (n, d, ϵ)-counters.

Corollary 7.3. *If Ω is the state space of an (n, d, ϵ)-counter with $\epsilon < 1/3$, then*

$$|\Omega| \geq \frac{1}{3}2^{-d}(4\epsilon)^{-d}\log(n/(e\epsilon^{-1}d)).$$

This implies that for $n \geq e^2$, $d \geq 1$, and $\epsilon < 1/16$,

$$\log_2|\Omega| = \log_2\log_2 n + \Omega(d\log\epsilon^{-1}).$$

Proof. By Lemma 7.4, we know there exists a 4ϵ-multiplicative covering R of $[\sqrt{d}\epsilon^{-1}, n/\sqrt{d}]$ with $|R| \leq |\Omega|$. By Corollary 7.2, we know $|R| \geq \frac{1}{3}2^{-d}(4\epsilon)^{-d}\log(n/(ed\epsilon^{-1}))$. This proves the

first statement. For the second statement, note that

$$\log_2 |\Omega| \geq d \log_2 \epsilon^{-1} - 3d - \log_2 3 + \log_2 \log(n/(e\epsilon^{-1}d)).$$

If $\epsilon < 1/16$ then $\log_2 \epsilon^{-1} > 4$ and thus $3 \leq \frac{3}{4} \log_2 \epsilon^{-1}$. This implies that $d \log_2 \epsilon^{-1} - 3d \geq \frac{1}{4} d \log_2 \epsilon^{-1}$. Furthermore, if $\epsilon^{-1} \geq n^{1/6}$ or $d \geq n^{1/6}$, then $\log_2 \log_2 n = o(d \log \epsilon^{-1})$. Otherwise, we have $n/(d\epsilon^{-1}) \geq n^{1/3}$. Note that $e \leq n^{1/2}$ by assumption. Thus we have $\log_2 \log(n/(e\epsilon^{-1}d)) \geq \log_2 \log n^{1/6} = \log_2 \log_2 n + \log_2(\frac{1}{6} \log 2)$. We conclude that

$$\begin{aligned} \log_2 |\Omega| &= \log_2 \log_2 n + \Omega(d \log \epsilon^{-1}) - O(1) \\ &= \log_2 \log_2 n + \Omega(d \log \epsilon^{-1}), \end{aligned}$$

when $n \geq e^2$, $d \geq 1$, and $\epsilon < 1/16$. □

7.1.4 Sample Runs

For demonstration, we run the d-dimension counter defined in §7.1.2 with $d = 4$ and memory budget $M_* = 12$. The input is generated randomly at each step with probability distribution $(1/2, 1/4, 1/8, 1/8)$. The table below tracks the state of the algorithm whenever (U, V) is changed. Recall that x is the input count vector, the estimator is $\hat{x} = 2^U V$, and the encoding of V is defined in Definition 7.3. We make the following remarks.

- When $U = 0$, V tracks x deterministically.
- When x is incremented, the encoding length $\psi(V)$ of V increases by at most one. When $\psi(V)$ reaches $M_* + 1$, the encoding length decreases by at least 1 and at most $2d$ during the **Scale-up**. when $U \geq 1$, i.e., after the first **Scale-up**, the encoding length is between $M_* + 1 - 2d$ and M_*. These are the key properties we have used in the proof of Theorem 7.2.
- Since the rates for inserting the entries are skewed, the space used to encode each entry is different. Larger entries are expected to take more space.

U	V	estimate	x	encoded V
0	0 0 0 0	0 0 0 0	0 0 0 0	||||
0	0 1 0 0	0 1 0 0	0 1 0 0	|0|||
0	0 1 0 1	0 1 0 1	0 1 0 1	|0||0|

0	1 1 0 1	1	1	0	1	1	1	0	1	0|0||0|
0	2 1 0 1	2	1	0	1	2	1	0	1	1|0||0|
0	2 1 0 2	2	1	0	2	2	1	0	2	1|0||1|
0	2 2 0 2	2	2	0	2	2	2	0	2	1|1||1|
0	3 2 0 2	3	2	0	2	3	2	0	2	10|1||1|
0	4 2 0 2	4	2	0	2	4	2	0	2	11|1||1|
0	4 3 0 2	4	3	0	2	4	3	0	2	11|10||1|
0	5 3 0 2	5	3	0	2	5	3	0	2	100|10||1|
0	5 3 1 2	5	3	1	2	5	3	1	2	100|10|0|1|
0	6 3 1 2	6	3	1	2	6	3	1	2	101|10|0|1|
0	6 4 1 2	6	4	1	2	6	4	1	2	101|11|0|1|
0	6 4 2 2	6	4	2	2	6	4	2	2	101|11|1|1|
1	3 3 1 1	6	6	2	2	6	5	2	2	10|10|0|0|
1	3 3 1 2	6	6	2	4	6	5	2	3	10|10|0|1|
1	4 3 1 2	8	6	2	4	7	5	2	3	11|10|0|1|
1	5 3 1 2	10	6	2	4	9	5	2	3	100|10|0|1|
1	6 3 1 2	12	6	2	4	11	5	2	3	101|10|0|1|
1	6 3 2 2	12	6	4	4	13	6	3	3	101|10|1|1|
1	7 3 2 2	14	6	4	4	14	6	3	3	110|10|1|1|
1	7 4 2 2	14	8	4	4	14	7	3	3	110|11|1|1|
1	8 4 2 2	16	8	4	4	15	7	3	3	111|11|1|1|
2	4 2 2 1	16	8	8	4	16	7	4	3	11|1|1|0|
2	4 3 2 1	16	12	8	4	19	11	4	4	11|10|1|0|
2	4 4 2 1	16	16	8	4	19	12	4	4	11|11|1|0|
2	4 5 2 1	16	20	8	4	23	13	6	5	11|100|1|0|
2	4 6 2 1	16	24	8	4	25	14	7	5	11|101|1|0|
3	3 3 1 0	24	24	8	0	29	14	7	5	10|10|0||
3	4 3 1 0	32	24	8	0	30	14	7	5	11|10|0||
3	5 3 1 0	40	24	8	0	31	14	7	5	100|10|0||
3	5 4 1 0	40	32	8	0	36	16	7	6	100|11|0||
3	5 4 2 0	40	32	16	0	45	20	9	7	100|11|1||
3	6 4 2 0	48	32	16	0	50	25	11	8	101|11|1||
3	7 4 2 0	56	32	16	0	52	28	12	8	110|11|1||
3	7 4 3 0	56	32	24	0	52	28	13	8	110|11|10||
3	8 4 3 0	64	32	24	0	57	33	14	12	111|11|10||
4	4 2 1 0	64	32	16	0	64	34	15	14	11|1|0||
4	5 2 1 0	80	32	16	0	76	43	17	17	100|1|0||
4	6 2 1 0	96	32	16	0	84	46	19	18	101|1|0||

4	7 2 1 0	112	32	16	0	90	49	20	18	110|1|0||
4	7 3 1 0	112	48	16	0	90	50	20	18	110|10|0||
4	8 3 1 0	128	48	16	0	93	51	22	19	111|10|0||
4	9 3 1 0	144	48	16	0	94	53	22	19	1000|10|0||
4	10 3 1 0	160	48	16	0	125	63	25	25	1001|10|0||
4	10 4 1 0	160	64	16	0	134	69	28	29	1001|11|0||
5	5 2 1 1	160	64	32	32	137	69	28	30	100|1|0|0|
5	5 3 1 1	160	96	32	32	139	73	29	30	100|10|0|0|
5	5 4 1 1	160	128	32	32	154	76	32	30	100|11|0|0|
5	5 4 2 1	160	128	64	32	216	102	52	44	100|11|1|0|
5	6 4 2 1	192	128	64	32	234	105	55	46	101|11|1|0|
5	7 4 2 1	224	128	64	32	236	105	55	46	110|11|1|0|
5	8 4 2 1	256	128	64	32	277	118	62	55	111|11|1|0|
6	4 2 1 0	256	128	64	0	279	118	63	55	11|1|0||
6	4 2 1 1	256	128	64	64	315	133	70	64	11|1|0|0|
6	5 2 1 1	320	128	64	64	322	139	73	65	100|1|0|0|
6	6 2 1 1	384	128	64	64	345	154	79	66	101|1|0|0|
6	6 2 1 2	384	128	64	128	374	176	88	76	101|1|0|1|
6	7 2 1 2	448	128	64	128	399	186	99	87	110|1|0|1|
6	8 2 1 2	512	128	64	128	435	195	105	90	111|1|0|1|
6	9 2 1 2	576	128	64	128	478	225	120	107	1000|1|0|1|
7	4 1 0 1	512	128	0	128	593	300	134	136	11|0||0|
7	5 1 0 1	640	128	0	128	599	302	139	138	100|0||0|
7	5 1 1 1	640	128	128	128	634	321	146	146	100|0|0|0|
7	6 1 1 1	768	128	128	128	686	351	153	156	101|0|0|0|
7	6 2 1 1	768	256	128	128	687	353	155	157	101|1|0|0|
7	6 3 1 1	768	384	128	128	732	377	170	174	101|10|0|0|
7	7 3 1 1	896	384	128	128	785	419	183	189	110|10|0|0|
7	7 4 1 1	896	512	128	128	869	464	203	216	110|11|0|0|
7	8 4 1 1	1024	512	128	128	969	522	235	236	111|11|0|0|
7	8 4 1 2	1024	512	128	256	994	536	244	240	111|11|0|1|
8	4 2 0 1	1024	512	0	256	1058	570	259	252	11|1||0|
8	4 3 0 1	1024	768	0	256	1102	596	269	262	11|10||0|
8	5 3 0 1	1280	768	0	256	1138	623	282	274	100|10||0|
8	6 3 0 1	1536	768	0	256	1217	660	304	295	101|10||0|
8	7 3 0 1	1792	768	0	256	1222	665	307	296	110|10||0|
8	7 4 0 1	1792	1024	0	256	1344	718	339	321	110|11||0|
8	7 5 0 1	1792	1280	0	256	1521	812	377	366	110|100||0|

8	7	5	0	2	1792	1280	0	512	1581	850	393	384	110|100||1|
9	3	3	0	1	1536	1536	0	512	1713	907	427	423	10|10||0|
9	4	3	0	1	2048	1536	0	512	1743	922	430	427	11|10||0|
9	5	3	0	1	2560	1536	0	512	1755	930	434	434	100|10||0|
9	5	4	0	1	2560	2048	0	512	1849	987	460	453	100|11||0|
9	6	4	0	1	3072	2048	0	512	2215	1195	572	548	101|11||0|
9	6	4	1	1	3072	2048	512	512	2451	1293	630	607	101|11|0|0|
9	7	4	1	1	3584	2048	512	512	2481	1311	640	615	110|11|0|0|
9	8	4	1	1	4096	2048	512	512	2663	1396	691	660	111|11|0|0|
9	8	4	2	1	4096	2048	1024	512	2751	1433	719	686	111|11|1|0|
10	4	2	1	1	4096	2048	1024	1024	2817	1458	733	695	11|1|0|0|
10	4	2	2	1	4096	2048	2048	1024	3437	1781	876	836	11|1|1|0|
10	4	2	2	2	4096	2048	2048	2048	4039	2071	1030	979	11|1|1|1|
10	4	3	2	2	4096	3072	2048	2048	4217	2145	1071	1021	11|10|1|1|

7.2 Symmetric Approximate Counting

Now let us switch back to the one-dimensional case of counting. The Morris counter, at its core, is to use a *sparse random walk* to track a *dense deterministic walk*. With the constraint that the estimation is unbiased, an algorithm designer is free to choose various kinds of sparse sets as the support of the random walk.

Definition 7.4. An *approximate counter* is defined by an increasing sequence $b(0), b(1), \ldots$ such that

- $b(0) = 0$ and $b(1) = 1$;
- $b(k) - b(k-1) \geq 1$ for $k \in \mathbb{Z}_+$;
- $b(k+1) - b(k) \geq b(k) - b(k-1)$ for $k \in \mathbb{Z}_+$.

We call the sequence $b(0), b(1), \ldots$ the *ticks* of the counter. The associated algorithm stores an index S, initially 0. Upon an increment, with probability $1/(b(S+1) - b(S))$, $S \leftarrow S+1$. Upon query, it returns $b(S)$, i.e., the current tick.

For any $a > 0$, the *Morris counter* with parameter a [128] is an approximate counter with ticks $b^{(a)}(k) = a((1 + 1/a)^k - 1)$.[5]

[5]One may check that the ticks of the Morris counter satisfy all three conditions: $b^{(a)}(0) = 0$, $b^{(a)}(1) = 1$,

Denote S_k as the index after k increments. The essence of the approximate counter is to replace a deterministic increment of step size 1 with a probabilistic increment with a step size $b(S_k+1)-b(S_k)$ and probability $1/(b(S_k+1)-b(S_k))$. Therefore by construction $b(S_k)-k$ is a *martingale* with mean 0 [145] and $b(S_k)$ is always an unbiased estimator for k, i.e., $\mathbb{E}b(S_k)=k$.

In comparison to the deterministic counter, one fundamental constraint of the Morris counter is that it only allows increments. It is not too surprising that there have been few investigation into generalizing Morris counter to allow decrements. On the one hand, it is hopeless to guarantee any constant relative error in the worst case by modifying Morris counters to incorporate decrements. On the other hand, in the *bounded deletion model* [103] where $|\lambda_+ - \lambda_-| = \Omega(\lambda_+ + \lambda_-)$ (λ_+ and λ_- are the number of increments and decrements respectively), an estimate with a guaranteed relative error can be returned with the trivial solution to use two Morris counters to track increments and decrements separately and take the difference of the estimates. There is, however, a natural extension of Morris counter that is left to be analyzed, i.e., the *symmetric Morris counter*. The symmetric Morris counter symmetrizes increments and decrements which probabilistically *decreases* the index in the presence of decrements.

Definition 7.5 (symmetric Morris counter)**.** A *symmetric approximate counter* extends the ticks of an approximate counter (Definition 7.4) from $\mathbb{N}$ to $\mathbb{Z}$ by setting $b(-k)=-b(k)$ for each $k\in\mathbb{Z}_+$. The associated algorithm stores an index S, initially 0. It supports the following three operations.

Increment: With probability $1/(b(S+1)-b(S))$, $S\leftarrow S+1$.

Decrement: With probability $1/(b(S)-b(S-1))$, $S\leftarrow S-1$.

Query: Return $b(S)$.

The *symmetric Morris counter* with parameter a uses $b^{(a)}$ as the ticks (Definition 7.4).

By construction, the symmetric Morris counter also has the correct mean but now the variance will depend on the number of increments, the number of decrements, and the way they are interleaved, which is challenging to characterize. Another challenge is to bound the memory space of the symmetric Morris counter. For the original Morris counter where

and $b^{(a)}(k)-b^{(a)}(k-1)=(1+1/a)^{k-1}$ is at least 1 and increasing. We note that there have been different conventions to parameterize Morris counters. We have used the original parameterization by Morris [128], where the larger a is, the finer the ticks are. Flajolet [80] sets $b(k)=(a'^k-1)(a'-1)$ which effectively has $a'=1+a$. Nelson and Yu [133] sets $b(k)=a''^{-1}((1+a'')^k-1)$ which has $a''=1/a$.

only increments are present, the maximum index ever reached is always the final index and it suffices to analyze the distribution of the final index to bound the space. The index of the symmetric counter can move left and right throughout the execution and one needs to bound the maximum absolute value of the index that has been *ever* reached throughout the whole process to bound its memory space.

My main contribution is to prove that for the task of counting with both increments and decrements (or equivalently tracking the difference of two incremental counters), using one symmetric Morris counter *completely dominates* using two Morris counters that track increments and decrements separately, in terms of the space-accuracy tradeoff. In comparison to storing increments and decrements separately, the symmetric Morris counter uses 50% of the space with a matching worst-case[6] variance. For some input sequences, the symmetric counter is *polynomially* more accurate. We note that while usually the constants in space complexity are not of primary concerns, it is well-motivated to optimize the constant for problems that are encountered in practice.[7] For such a fundamental problem, a 50% memory saving is meaningful and significant.

Theorem 7.3. *Fix any $a > 0$ as the Morris parameter and $n \geq 4$ be the input length. For any input sequence $w = (w_1, w_2, \ldots, w_n) \in \{-1, 1\}^n$, define $\lambda_+ = \sum_{j\in[n]} \mathbb{1}[w_j = 1]$ and $\lambda_- = \sum_{j\in[n]} \mathbb{1}[w_j = -1]$ to be the number of increments and decrements respectively. We compare the following two strategies.*

- *Use a symmetric Morris counter with parameter $2a$. Define $(S^{\pm,(2a)}_{w_1,\ldots,w_k})_{k\in[n]}$ to be the index history. The final estimate is $b^{(2a)}(S^{\pm,(2a)}_{w_1,\ldots,w_n})$.*
- *Use two Morris counters with parameter a and count increments and decrements separately. Define $S^{+,(a)}_{\lambda_+}$ and $S^{-,(a)}_{\lambda_-}$ to be the final indices of the two counters. The final estimate is $b^{(a)}(S^{+,(a)}_{\lambda_+}) - b^{(a)}(S^{-,(a)}_{\lambda_-})$.*

For any input sequence $w \in \{-1, 1\}^n$,

$$\mathbb{E}b^{(2a)}(S^{\pm,(2a)}_{w_1,\ldots,w_n}) = \mathbb{E}\left(b^{(a)}(S^{+,(a)}_{\lambda_+}) - b^{(a)}(S^{-,(a)}_{\lambda_-})\right) = \lambda_+ - \lambda_- \quad \text{(both are unbiased)}$$

$$\mathbb{V}b^{(2a)}(S^{\pm,(2a)}_{w_1,\ldots,w_n}) \leq (1 + 2/n)\mathbb{V}\left(b^{(a)}(S^{+,(a)}_{\lambda_+}) - b^{(a)}(S^{-,(a)}_{\lambda_-})\right)$$

(matching worst-case variance)

[6]For fixed λ_+ and λ_-, there is a fixed distribution of the estimate produced by tracking λ_+ and λ_- separately. However, for the symmetric counter, the variance will depend on how the increments and decrements are interleaved.

[7]One example is the problem of `COUNT-DISTINCT` studied in Chapter 5, which has a long line of work [80, 73, 82, 151, 117, 137] improving the constants in the memory-accuracy trade-off. The constant-optimized sketches such as HyperLogLog [82] remain extremely impactful in practice till today.

$$\mathbb{P}(\max_{k\in[n]} |S^{\pm,(2a)}_{w_1,\ldots,w_k}| \geq 2r) \leq \mathbb{P}(S^{(a)}_n \geq r), \qquad \text{(halved memory space)}$$

for any $r \in \mathbb{N}$, where $S^{(a)}_n$ is the final index of the Morris counter with parameter a after n increments. Furthermore, for any partition of λ_+ and λ_- with $\lambda_+ + \lambda_- = n$, there exists an input sequence $w \in \{-1,1\}^n$ such that

$$\mathbb{V}b^{(2a)}(S^{\pm,(2a)}_{w_1,\ldots,w_n}) \leq O\left(\frac{(\lambda_+ - \lambda_-)^2}{n^2}\right) \mathbb{V}\left(b^{(a)}(S^{+,(a)}_{\lambda_+}) - b^{(a)}(S^{-,(a)}_{\lambda_-})\right)$$

(polynomially better in some cases)

We remark that $\mathbb{P}(\max_{k\in[n]} |S^{\pm,(2a)}_{w_1,\ldots,w_k}| \geq 2r) \leq \mathbb{P}(S^{(a)}_n \geq r)$ implies that any probabilistic space bound of the Morris counter with parameter a will apply to the symmetric counter with parameter $2a$ with two additional bits. In particular, suppose for any r and $\delta > 0$ that $\mathbb{P}(\log_2 S^{(a)}_n \geq \log_2 r) < \delta$, then $\mathbb{P}(\log_2 \max_{k\in[n]} |S^{\pm,(2a)}_{w_1,\ldots,w_k}| \geq \log_2 r + 1) < \delta$. Thus we need one bit to match the probability bound of the maximum index (in magnitude) and an another additional bit to store the sign of index. Therefore, the symmetric counter saves 50% of the space asymptotically.

The main technique we use to prove the main theorem (Theorem 4.1) is *coupling*, which can "compare" random processes without massive computation on the distributional properties (see, e.g., Flajolet's computation on the mean and variance of the index [80] and Nelson and Yu's computation on the tail of the index [133]). As a powerful probabilistic tool, coupling arguments have been used to analyze the running time of algorithmic Lovász local lemma [129] and to prove the optimality of 2-party contention resolution protocols [156].

7.2.1 Proof of Theorem 7.3

We need three lemmas to prove the main theorem. The first lemma constructs a coupling of the execution of a counter under two different input cases. The first is an arbitrary length-n input $w \in \{-1,1\}^n$ and the second is n increments. For any approximate counter $b : \mathbb{N} \to \mathbb{R}_+$, we define $(S^{\pm}_{w_1,\ldots,w_k})_{k\in\mathbb{N}}$ to be the *index history* of the symmetric counter (using ticks b) with input stream $w_1, w_2, \ldots$, where $w_j \in \{-1,1\}$ for each j (Definition 7.5). We define $(S_k)_{k\in\mathbb{N}}$ to be the index history of the approximate counter (using ticks b) over an infinite stream of increments (Definition 7.4).

Lemma 7.6 (coupling)**.** *For any input sequence $w \in \{-1,1\}^\infty$, there exists a coupled process $(S^{\pm}_{w_1,\ldots,w_k}, S_k)_{k\in\mathbb{N}}$ such that $|S^{\pm}_{w_1,\ldots,w_k}| \leq S_k$ for any $k \in \mathbb{N}$.*

Proof. The construction is given as follows. Initialize $S^{\pm}_{\emptyset} = S_0 = 0$ and for each time $k \in [n]$,

- Freshly sample a uniform random variable U_k over $(0, 1)$.
- $S_k \leftarrow S_{k-1} + \mathbb{1}\left[U_k < (b(S_{k-1}+1) - b(S_{k-1}))^{-1}\right]$.
- $S^{\pm}_{w_1,\ldots,w_k} \leftarrow S^{\pm}_{w_1,\ldots,w_{k-1}} + w_k \mathbb{1}\left[U_k < \left|b(S^{\pm}_{w_1,\ldots,w_{k-1}} + w_k) - b(S^{\pm}_{w_1,\ldots,w_{k-1}})\right|^{-1}\right]$.

The resulting processes $(S_k)_{k\in\mathbb{N}}$ and $(S^{\pm}_{w_1,\ldots,w_k})_{k\in\mathbb{N}}$ clearly simulate the history of the incremental counter and the symmetric counter. We now prove that $|S^{\pm}_{w_1,\ldots,w_k}| \le S_k$ by induction on $k \in \mathbb{N}$. By construction, we have $|S^{\pm}_{w_1}| = S_0 = 1$ since we have $b(1)-b(0) = b(0)-b(-1) = 1$. The claim is true for $k = 1$. Suppose $|S^{\pm}_{w_1,\ldots,w_k}| \le S_k$ for some $k \ge 1$. If $S^{\pm}_{w_1,\ldots,w_k} = 0$, we immediately have $|S^{\pm}_{w_1,\ldots,w_{k+1}}| \le 1 \le S_k$ (note that $S_k \ge 1$ for any $k \ge 1$). By symmetry, without loss of generality we may assume $S^{\pm}_{w_1,\ldots,w_k} \ge 1$. The only case we need to prove is when $S^{\pm}_{w_1,\ldots,w_k} = S_k$ and $w_{k+1} = 1$. In this case, by the coupling above, they will both increase by one or keep unchanged since they share a same U_{k+1}. This completes the induction. □

Next, we decompose the variance for any input sequence along the index history as how [145, 65] decomposes the variance in the incremental setting.

Lemma 7.7 (decomposition of variance)**.** *For any input sequence $w \in \{-1, 1\}^n$,*

$$\mathbb{V}b(S^{\pm}_{w_1,\ldots,w_n}) = \sum_{k=1}^{n} w_k \mathbb{E}(b(S^{\pm}_{w_1,\ldots,w_{k-1}} + w_k) - b(S^{\pm}_{w_1,\ldots,w_{k-1}})) - n$$

Remark 7.3. Note that by construction (Definitions 7.4 and 7.5), one always has $w_k\mathbb{E}(b(S^{\pm}_{w_1,\ldots,w_{k-1}} + w_k) - b(S^{\pm}_{w_1,\ldots,w_{k-1}})) \ge 1$ and therefore the computed variance is always non-negative.

Remark 7.4. By setting $w_k = 1$ for all $k \in [n]$, one gets the variance decomposition for the incremental counter, as follows.

$$\mathbb{V}b(S_n) = \sum_{k=1}^{n} \mathbb{E}(b(S_{k-1}+1) - b(S_{k-1})) - n.$$

Proof. We compute

$$\mathbb{E}(b(S^{\pm}_{w_1,\ldots,w_k})^2 - b(S^{\pm}_{w_1,\ldots,w_{k-1}})^2 | S^{\pm}_{w_1,\ldots,w_{k-1}}) = \frac{b(S^{\pm}_{w_1,\ldots,w_{k-1}} + w_k)^2 - b(S^{\pm}_{w_1,\ldots,w_{k-1}})^2}{\left|b(S^{\pm}_{w_1,\ldots,w_{k-1}} + w_k) - b(S^{\pm}_{w_1,\ldots,w_{k-1}})\right|}$$

$$= w_k(b(S^{\pm}_{w_1,\dots,w_{k-1}} + w_k) + b(S^{\pm}_{w_1,\dots,w_{k-1}})),$$

since $b(j)$ is increasing over $\mathbb{Z}$. Take the total expectation and we have

$$\mathbb{E}b(S^{\pm}_{w_1,\dots,w_k})^2 - \mathbb{E}b(S^{\pm}_{w_1,\dots,w_{k-1}})^2 = w_k\mathbb{E}(b(S^{\pm}_{w_1,\dots,w_{k-1}} + w_k) + b(S^{\pm}_{w_1,\dots,w_{k-1}}))$$

By construction (Definition 7.5) we have

$$\mathbb{E}b(S^{\pm}_{w_1,\dots,w_k}) - \mathbb{E}b(S^{\pm}_{w_1,\dots,w_{k-1}}) = w_k,$$

which implies

$$\begin{aligned}(\mathbb{E}b(S^{\pm}_{w_1,\dots,w_k}))^2 - (\mathbb{E}b(S^{\pm}_{w_1,\dots,w_{k-1}}))^2 &= w_k\mathbb{E}(b(S^{\pm}_{w_1,\dots,w_k}) + b(S^{\pm}_{w_1,\dots,w_{k-1}})) \\ &= w_k\mathbb{E}(2b(S^{\pm}_{w_1,\dots,w_{k-1}}) + w_k)\end{aligned}$$

Thus we have

$$\begin{aligned}&\mathbb{V}b(S^{\pm}_{w_1,\dots,w_k}) - \mathbb{V}b(S^{\pm}_{w_1,\dots,w_{k-1}}) \\ &= \mathbb{E}b(S^{\pm}_{w_1,\dots,w_k})^2 - \mathbb{E}b(S^{\pm}_{w_1,\dots,w_{k-1}})^2 - \left((\mathbb{E}b(S^{\pm}_{w_1,\dots,w_k}))^2 - (\mathbb{E}b(S^{\pm}_{w_1,\dots,w_{k-1}}))^2\right) \\ &= w_k\mathbb{E}(b(S^{\pm}_{w_1,\dots,w_{k-1}} + w_k) - b(S^{\pm}_{w_1,\dots,w_{k-1}}) - w_k).\end{aligned}$$

Note that $S^{\pm}_{\emptyset} = 0$ with zero variance. The proof is thus completed by aggregating the variance increments from $k = 1$ to n. □

The third lemma is to compare the space of the Morris counter with different parameters.

Lemma 7.8 (memory comparison)**.** *Let $b : \mathbb{N} \to \mathbb{R}_+$ and $b' : \mathbb{N} \to \mathbb{R}_+$ be two approximate counters. If there exists $\alpha \in \mathbb{Z}_+$ such that for any $j \in \mathbb{Z}_+$ $b(\alpha j+1)-b(\alpha j) \geq b'(j+1)-b'(j)$. Let $(S_k)_{k\in\mathbb{N}}$ and $(S'_k)_{k\in\mathbb{N}}$ be the history of the two counters with a stream of increments. Then there exists a coupled process $(S_k, S'_k)_{k\in\mathbb{N}}$ such that $S_k \leq \alpha S'_k$ for each $k \in \mathbb{N}$.*

Proof of Lemma 7.8. The coupling is constructed as follows. Initialize $S_0 = S'_0 = 0$ and for each time $k \in \mathbb{Z}_+$,

- Freshly sample a uniform random variable U_k over $(0, 1)$.
- $S_k \leftarrow S_{k-1} + \mathbb{1}\left[U_k < (b(S_{k-1} + 1) - b(S_{k-1}))^{-1}\right]$.

- $S'_k \leftarrow S'_{k-1} + \mathbb{1}\left[U_k < (b'(S'_{k-1}+1) - b'(S'_{k-1}))^{-1}\right]$.

The resulting processes $(S_k)_{k\in\mathbb{N}}$ and $(S'_k)_{k\in\mathbb{N}}$ clearly simulate the two counters over an infinite stream of increments (Definition 7.4). We now prove that $S_k \le \alpha S'_k$ by induction on $k \in \mathbb{Z}_+$. By construction, we have $S_1 = S'_1 = 1$ and thus it is true for $k = 1$. Suppose $S_k \le \alpha S'_k$ for some $k \ge 1$. Since $S_{k+1} \le S_k + 1$, it suffices to consider the case where $S_k = \alpha S'_k$. By assumption we have $b(\alpha j + 1) - b(\alpha j) \ge b'(j+1) - b'(j)$ for any $j \in Z_+$. Thus we have

$$b'(S'_k + 1) - b'(S'_k) \le b(\alpha S'_k + 1) - b(\alpha S'_k)$$

since we assumed $S_k = \alpha S'_k$

$$= b(S_k + 1) - b(S_k). \tag{7.8}$$

Since both S_{k+1} and S'_{k+1} are coupled with the same randomness U_{k+1}, we have

$$S_{k+1} = S_k + \mathbb{1}\left[U_{k+1} < (b(S_k+1) - b(S_k))^{-1}\right]$$

we assumed $S_k = \alpha S'_k$ and by Eq. (7.8)

$$\le \alpha S'_k + \mathbb{1}\left[U_{k+1} < (b'(S'_k+1) - b'(S'_k))^{-1}\right]$$

note that $\alpha \ge 1$

$$\le \alpha S'_k + \alpha\mathbb{1}\left[U_{k+1} < (b'(S'_k+1) - b'(S'_k))^{-1}\right]$$
$$= \alpha S'_{k+1},$$

which completes the induction. □

Recall that the ticks for Morris counter (Definition 7.4) are $b^{(a)}(k) = a((1+1/a)^k - 1)$ and we know $b^{(a)}(k+1) - b^{(a)}(k) = (1+1/a)^k$. Note that for any $\alpha \in \mathbb{Z}_+$ and $a > 0$, $(1+1/a) \le (1+1/(\alpha a))^\alpha$. Therefore, we have

$$b^{(\alpha a)}(\alpha k + 1) - b^{(\alpha a)}(\alpha k) = (1+1/(\alpha a))^{\alpha k} \ge (1+1/a)^k = b^{(a)}(k+1) - b^{(a)}(k). \tag{7.9}$$

Thus by Lemma 7.8 we know, by changing the parameter a to αa, the increase in memory is $\log_2 \alpha$ in the strictest sense. We now proceed to prove the main theorem.

Proof of Theorem 4.1. By design, the ticks are the unbiased estimates for symmetric coun-

ters. We now compare the variance. By Lemma 7.7, we have

$$\mathbb{V}b^{(2a)}(S^{\pm,(2a)}_{w_1,\dots,w_n}) = \sum_{k=1}^{n} \mathbb{E}w_k(b(S^{\pm,(2a)}_{w_1,\dots,w_{k-1}} + w_k) - b(S^{\pm,(2a)}_{w_1,\dots,w_{k-1}})) - n$$

Note that for any $s \in \mathbb{Z}$ and $w_k \in \{-1, 1\}$, by Definition 7.4, we know $w_k(b(s+w_k)-b(s)) \leq b(|s|+1) - b(|s|)$.

$$\leq \sum_{k=1}^{n} \mathbb{E}(b(|S^{\pm,(2a)}_{w_1,\dots,w_{k-1}}| + 1) - b(|S^{\pm,(2a)}_{w_1,\dots,w_{k-1}})|) - n$$

We use the coupling $(S^{\pm,(2a)}_{w_1,\dots,w_k}, S^{(2a)}_k)_{k\in[m]}$ constructed in Lemma 7.6 in which $|S^{\pm,(2a)}_{w_1,\dots,w_{k-1}}| \leq S^{(2a)}_{k-1}$ for each $k \in [n]$. Also note that by Definition 7.4, $b(|s|+1) - b(|s|)$ is increasing in $|s|$. Thus

$$\begin{aligned}
&\leq \sum_{k=1}^{n} \mathbb{E}(b(S^{(2a)}_{k-1} + 1) - b(S^{(2a)}_{k-1})) - n \\
&= \mathbb{V}b^{(2a)}(S^{(2a)}_n) && \text{(Remark 7.4)}
\end{aligned}$$

It is proved by Morris [128] that $\mathbb{V}S^{(a)}_n = \frac{n(n-1)}{2a}$. Thus

$$\mathbb{V}b^{(2a)}(S^{(2a)}_n) = \frac{n^2-n}{4a} \leq (1+2/n)\frac{n^2-2n}{4a}, \tag{7.10}$$

where we assumed $n \geq 4$. Recall $\lambda_+ + \lambda_- = n$ and therefore

$$\mathbb{V}\left(b^{(a)}(S^{+,(a)}_{\lambda_+}) - b^{(a)}(S^{-,(a)}_{\lambda_-})\right) = \mathbb{V}b^{(a)}(S^{+,(a)}_{\lambda_+}) + \mathbb{V}b^{(a)}(S^{-,(a)}_{\lambda_-})$$

by Morris [128]

$$\begin{aligned}
&= \frac{\lambda_+(\lambda_+-1)}{2a} + \frac{\lambda_-(\lambda_--1)}{2a} \\
&= \frac{(\lambda_+-\lambda_-)^2 + n^2 - 2n}{4a}.
\end{aligned} \tag{7.11}$$

Compare Eq. (7.10) and Eq. (7.11) and we have $\mathbb{V}b^{(2a)}(S^{\pm,(2a)}_{w_1,\dots,w_n}) \leq (1+2/n)\mathbb{V}(b^{(a)}(S^{+,(a)}_{\lambda_+}) - b^{(a)}(S^{-,(a)}_{\lambda_-}))$. To prove the memory bound, we note that by the coupling in Lemma 7.6,

$$\mathbb{P}(\max_{k\in[n]} |S^{\pm,(2a)}_{w_1,\dots,w_k}| \geq 2r) \leq \mathbb{P}(S^{(2a)}_n \geq 2r)$$

by the coupling in Lemma 7.8,

$$\leq \mathbb{P}(S_n^{(a)} \geq r).$$

Finally, without loss of generality, we assume $\lambda_+ \geq \lambda_-$ and construct the input sequence as follows.

$$w^* = (\underbrace{1, -1, 1, -1, \ldots, 1, -1}_{2\lambda_-}, \underbrace{1, 1, \ldots, 1}_{\lambda_+ - \lambda_-}).$$

For this sequence, the symmetric Morris counter (with parameter $2a$) will switch between $S = 0$ and $S = 1$ at first and then start counting $(\lambda_+ - \lambda_-)$ increments, whose variance equals $\frac{(\lambda_+ - \lambda_-)(\lambda_+ - \lambda_- - 1)}{4a}$. On the other hand, if increments and decrements are collected separately, the variance is $\frac{(\lambda_+ - \lambda_-)^2 + n^2 - 2n}{4a}$, given by Eq. (7.11). Thus we conclude that for the input sequence w^*, the variance ratio between the two strategies is $O((\lambda_+ - \lambda_-)^2/n^2)$. $\square$

CHAPTER 8

Fraud Detection for Streaming Sketches

Data sketches for counting *distinct elements* (aka *cardinality estimation*) have many applications, one of which is to detect distributed denial of service (DDoS) attacks [42, 45, 31, 41]. Therefore, it is natural to expect that the *data sketches themselves* could be a target for attacks [143]. These sketches have provable guarantees so long as (i) they have access to truly random bits [13, 110] (used to build hash functions) or oracle access to random hash functions [80, 82, 137, 160], and (ii) the input is selected independent of the internal randomness/hash functions.

Reviriego and Ting [143] observed that protecting the internal randomness of data sketches is more complicated than one might think. One of the most valuable features of cardinality/distinct elements sketches is *mergeability*, i.e., if $\mathcal{A}, \mathcal{B} \subseteq U$ are sets over a universe U, $\text{sk}(\mathcal{A} \cup \mathcal{B})$ can be computed from $\text{sk}(\mathcal{A}), \text{sk}(\mathcal{B})$, *but only if* $\text{sk}(\mathcal{A}), \text{sk}(\mathcal{B})$ *were constructed using the same hash functions.* Mergeability is critical for parallel/distributed processing: it allows huge data sets to be partitioned arbitrarily among different machines and later combined. It is also essential for estimating the size of unions and intersections [152], and computing set-similarity [48, 27].

Any organization that wishes to support merging of data sketches and union/intersection queries must distribute the *same* hash function $\hbar$ across all relevant machines and use it exclusively for all sketching tasks. Nonetheless, most sketching will still be done by a single machine operating on a single data stream. Given such a broad attack surface, it is reasonable to be skeptical about the *secrecy* of the hash function. In this section we shall simply assume that the relevant hash function $\hbar$ has been completely compromised, and is *public* randomness. If there is an adversary, it can choose inputs with desired hash-values,[1] and has the goal of forcing the sketch to report inaccurate statistical estimates.

Is it possible to quickly detect whether the input data was selected by such an adversary?

[1] Choosing a cryptographic hash function $\hbar$ does not help much. Attacks are feasible if $\hbar$ can be evaluated on arbitrary inputs. It is not necessary to find preimages in $\hbar$.

In other words, in the process of sketching a data stream $\mathcal{A}$ as $\text{sk}_h(\mathcal{A})$, can we differentiate between a *good* $\mathcal{A}$ chosen independent of h, and a *bad* $\mathcal{A}$ chosen with knowledge of h, and designed to yield inaccurate estimates?

Summary of Contributions In this section we present a proof-of-concept that it *is* possible to detect this type of data fraud, by using a *private* sketch sk_g that depends on a *private* hash function g. (In the distributed setup outlined above, one should imagine a different g being selected by each machine, and whenever a new stream is processed. Merging and union/intersection queries would still only operate on sk_h.) We analyze the situation where sk_h and sk_g are K-Min and k-Min sketches (aka MinHash [27]), though the setup should work for other popular cardinality sketches like HyperLogLog and PCSA [83, 82, 160]. Here we regard $h, g : U \to [0, 1]$ as truly random hash functions and let $\text{sk}_h(\mathcal{A})$ and $\text{sk}_g(\mathcal{A})$ be the K-smallest and k-smallest hash values according to h and g, respectively.

Our scheme answers several natural qualitative questions. Note that standard errors[2] for the public and private sketches are $\Theta(1/\sqrt{K})$ and $\Theta(1/\sqrt{k})$. One might think that we would only be able to detect attacks that introduce relative errors of $\Omega(1/\sqrt{k})$. This is not true, and in fact there is a tradeoff between the private sketch size k, the *severity* of the attack (ϵ), and the *latency* (N) of detecting an attack after it has commenced. We give a formal statement of the tradeoff between k, ϵ, and N in Section 8.1.

Related Work Reviriego and Ting [143] first considered the problem of detecting compromised hash functions in the HyperLogLog cardinality estimator, and also proposed using a second sketch sk_g based on a private hash function g.[3] They noted that in the absence of adversarial manipulation, the estimators $\hat{\lambda}_h$ and $\hat{\lambda}_g$ are close to normally distributed (as k, K get large), with standard error tending to $c\sqrt{1/K}$ and $c\sqrt{1/k}$, for a certain constant c. Thus, $\xi = (\hat{\lambda}_h - \hat{\lambda}_g)/\hat{\lambda}_g$ should be close to normally distributed with mean and standard deviation close to 0 and $c\sqrt{1/K + 1/k}$, respectively. If the empirical distribution[4] of ξ is far from normal, then it is safe to assume the stream is controlled by an adversary with knowledge of h. To make this scheme rigorous, one would need to characterize the distribution of $\hat{\lambda}_g$, which is only close to normal, and analyze how much empirical data from ξ is necessary to make a statistically sound accusation of data fraud. Reviriego and Ting [143] also considered the feasibility of several attacks on HyperLogLog.

There is a large body of work on *adaptive* adversaries, which we can think of as attempting

[2]Suppose the actual and estimate cardinalities of $\mathcal{A}$ are $\lambda, \hat{\lambda}$. Then the *standard error* is $\lambda^{-1}\sqrt{\mathbb{V}(\hat{\lambda}/\lambda)}$, i.e., the standard deviation normalized by the true cardinality λ.

[3]They proposed implementing g by "salting" the (known) h with random seed s, i.e., $g(x) = h(x, s)$.

[4]Presumably we sample ξ whenever sk_g changes state.

to learn $\hbar$ by using the input-output interface of the data structure, i.e., adaptively inserting elements from U, interleaved with querying statistics. An early result in this model is Naor and Yogev's [131] version of Bloom filters robust to adaptive inputs. See [44, 10, 166, 21, 92] for many recent results on adaptive adversaries in streaming. Ajtai et al. [4] studied the *white-box* streaming model, in which the adversary has knowledge of the internal state of the sketch, i.e., there is no private randomness.

The *privacy* of data sketches has also been considered, where the "attack" is to determine, for some individual $x \in U$, whether $x \in \mathcal{A}$ or $x \notin \mathcal{A}$ by looking at $\text{sk}(\mathcal{A})$. See [67, 148, 68, 93] for positive and negative results.

8.1 Technical Overview

The input stream $\mathcal{A} = (x_1, x_2, x_3, \ldots, x_z) \in U^z$ is presented one element at a time. The K-Min sketch $\text{sk}_\hbar$ stores the K smallest distinct hash-values among $\{\hbar(x_1), \hbar(x_2), \ldots, \hbar(x_z)\}$, where $\hbar : U \to [0,1]$. Suppose they are $0 < v_1 < v_2 < \ldots < v_K$ in sorted order. Here are some simple observations about K-Min sketches. The state of the sketch is purely a function of the *set* of distinct elements in $\mathcal{A}$ (duplicates have no effect), and assuming $\hbar$ is a uniformly random hash function, the state is only a function of the cardinality $\lambda = |\mathcal{A}|$. The probability the sketch changes state upon insertion of a distinct new element is v_K. Moreover, v_K is a *sufficient statistic* for the K-Min sketch (w.r.t. any parameter of interest) since once we condition on the value of v_K, $\{v_1, \ldots, v_{K-1}\}$ are independent and uniformly distributed in $[0, v_K)$. Thus, we can assume without loss of generality that any cardinality estimator for $|\mathcal{A}|$ is a function of v_K. Two K-Min sketches $(v_1, \ldots, v_K), (v'_1, \ldots, v'_K)$ of $\mathcal{A}, \mathcal{A}'$ can clearly be merged, simply by retaining the minimum K values in their union.[5]

Our strategy is to monitor a K-Min sketch $\text{sk}_\hbar$ that uses a possibly insecure shared hash function $\hbar$ with a k-Min sketch sk_g using a secure, private hash function g. Naturally we want to *periodically* compare $\text{sk}_\hbar$ and sk_g, but what does *periodically* mean? If we have processed $x_1, \ldots, x_t$, it does not make sense to call t the "time" since the sketch is insensitive to duplicates, and the majority of the stream may be duplicates. The most sensible definition is to let the *cardinality* of $\{x_1, \ldots, x_t\}$ itself define the time after processing x_t, but the cardinality is not precisely known. This is the first subtle technical issue that makes this problem unusual, as most Markov chains have a more accessible notion of *time*.

[5]Note, however, that the sufficiency argument does not apply to two related sketches. For example, when estimating $|\mathcal{A} \cap \mathcal{A}'|$ from $\text{sk}_\hbar(\mathcal{A}), \text{sk}_\hbar(\mathcal{A}')$, the *entire* sketch contains useful information, not just the Kth smallest value; see Ting [152].

We could mark time by looking at *state changes* in $\mathrm{sk}_\hbar$ or sk_g. It is probably a bad idea to pay attention to $\mathrm{sk}_\hbar$, as this is potentially controlled by the adversary. Using sk_g also presents problems as the process of *marking* time would be correlated with our *estimate* of time, and may therefore introduce a small bias. Our solution is to introduce a third probabilistic *metronome* sketch that is used to mark time independent of $\mathrm{sk}_\hbar, \mathrm{sk}_g$. That is, we have:

- A public K-Min sketch $\mathrm{sk}_\hbar$, which is subject to merges and queries involving multiple sketches.
- A private k-Min sketch sk_g.
- A private 1-Min probabilistic metronome sketch sk_m.

Consider N times (cardinalities) $T_1, \ldots, T_N$ generated by state changes of the metronome sk_m. Let $P_{n;K}$ and $P_{n;k}$ be the Kth largest and kth largest hash-values of $\mathrm{sk}_\hbar$ and sk_g at time T_n, respectively. We prove that $N^{-1}\log(\prod_{n=1}^N P_{n;k}T_n)$ is concentrated around its mean μ_k. Thus, in the absence of adversarial manipulation,

$$N^{-1}\left(\log\left(\prod_{n=1}^N P_{n;k}T_n\right) - \log\left(\prod_{n=1}^N P_{n;K}T_n\right)\right) = N^{-1}\log\left(\frac{\prod_{n=1}^N P_{n;k}}{\prod_{n=1}^N P_{n;K}}\right)$$

is concentrated around $\mu_k - \mu_K$. (The cancellation of the $\{T_n\}$ here is essential to our approach since we have no way to know these values exactly!)

The upshot is that we are able to detect *any* adversarial manipulation that affects the product $\prod_{n=1}^N P_{n;K}T_n$ over a sufficiently large number N of observations. For example, if one wanted to suppress or boost the cardinality estimator $\hat{\lambda}_\hbar$ by an e^ϵ factor, one would see a sequence $(P_{n;K})$ of values that are too large/too small by an $\approx e^\epsilon$-factor.[6] Theorem 8.1 shows that an *attack of magnitude* ϵ (that perturbs $N^{-1}(\log(\prod_{n=1}^N P_{n;K}T_n)$ by ϵ off its expectation) can be detected with false alarm rate δ after $N = \tilde{O}(\mathrm{poly}(\epsilon^{-1}, k, \log K)\log\delta^{-1})$ observations, assuming $k \le K$.

Theorem 8.1. *There are some $C_1, C_2, \mu_k > 0$ such that for any $\epsilon \in (0,1), \beta \in [1,k]$ and*

[6]The details would depend on the precise cardinality estimator used. Several cardinality estimators for K-Min sketches have been proposed [90, 35, 122, 151]. Most estimators are designed to be unbiased w.r.t. $\lambda = |\mathcal{A}|$, i.e., $\mathbb{E}(\hat{\lambda} - \lambda) = 0$, though it is equally reasonable to be unbiased w.r.t. $\log\lambda$, i.e., $\mathbb{E}(\log\hat{\lambda} - \log\lambda) = 0$. This two notions are the same in the limit, as $K \to \infty$, but are formally distinct for fixed K.

$N > C_1\beta^{1.01}\log^7 k$,

$$\mathbb{P}\left(\left|\frac{1}{N}\log\left(\prod_{n=1}^{N} P_{n;k}T_n\right) - \mu_k\right| \geq \epsilon\right) \leq (2+2N)\exp\left(-C_2\epsilon^2 N^{1/3}\beta^{2/3}\right).$$

Proof. This is a simplification of Corollary 8.5. □

Thus we now define our detector as follows.

$$\begin{aligned} D_N &= \left(\frac{1}{N}\log\left(\prod_{n=1}^{N} P_{n;k}T_n\right) - \mu_k\right) - \left(\frac{1}{N}\log\left(\prod_{n=1}^{N} P_{n;K}T_n\right) - \mu_K\right) \\ &= \frac{1}{N}\log\left(\frac{\prod_{n=1}^{N} P_{n;k}}{\prod_{n=1}^{N} P_{n;K}}\right) - (\mu_k - \mu_K), \end{aligned}$$

where the unobservable (T_n) are canceled.

Corollary 8.1 (False Alarm Probability)**.** *Assume $k \leq K$. There are some $C_1, C_2 > 0$ such that for any $\epsilon \in (0,1)$ and $N > C_1 k^{1.01}\log^7 K$,*

$$\mathbb{P}\left(|D_N| \geq 2\epsilon\right) \leq (4+4N)\exp\left(-C_2\epsilon^2 N^{1/3}k^{2/3}\right).$$

Proof. Apply Theorem 8.1 to both the public sketch and the private sketch with $\beta = k$, since we assumed $k \leq K$. □

Corollary 8.1 says that to detect any type of data fraud of magnitude ϵ, we should continually monitor D_N (which depends on the last N state changes of the metronome), and declare corruption whenever $|D_N| > 2\epsilon$. So long as $N = \tilde{O}(\text{poly}(\epsilon^{-1}, k, \log K)\log\delta)$, the probability of a false alarm ($|D_N| > 2\epsilon$ by chance) is δ. Since we do not know the magnitude ϵ, we should think of running this detector for all ϵ simultaneously.

Corollary 8.2 (Probability of No Detection)**.** *If an adversary makes the event*

$$A : \left|\frac{1}{N}\log\left(\prod_{n=1}^{N} P_{n;K}T_n\right) - \mu_K\right| \geq 3\epsilon$$

happen, then the probability that we do not make a fraud detection alarm is

$$\begin{aligned} \mathbb{P}(|D_N| \leq 2\epsilon | A) &\leq \mathbb{P}\left(\left|\frac{1}{N}\log\left(\prod_{n=1}^{N} P_{n;k}T_n\right) - \mu_k\right| \geq \epsilon\right) \\ &\leq (2+2N)\exp\left(-C_2\epsilon^2 N^{1/3}k^{2/3}\right), \end{aligned}$$

for $\epsilon \in (0,1)$ and $N > C_1 k^{1.01} \log^7 k$ where C_1 and C_2 are some constants.

Proof. ($|D_N| \le 2\epsilon$) and the event A necessarily imply the event

$$\left| \frac{1}{N} \log \left(\prod_{n=1}^{N} P_{n;k} T_n \right) - \mu_k \right| \ge \epsilon.$$

The tail bound is then obtained by applying Theorem 8.1 with $\beta = k$. □

Organization In Section 8.2 we formalize the evolution of a k-Min sketch in terms of a Poisson point process in the plane. Analyzing the progression of $\mathrm{sk}_{\mathcal{h}}, \mathrm{sk}_{\mathcal{g}}, \mathrm{sk}_{\mathcal{m}}$ through time turns out to be much more complicated than considering their reverse-chronological evolution. We analyze the backwards progression in Section 8.2.2. Section 8.3 proves the concentration inequality for the Markov chain $(P_{n;k}T_n)$ appearing in Theorem 8.1.

8.2 Derivation of the $k/1$-Chain

8.2.1 Poissonized k-Min Process

The k-Min sketch is clearly insensitive to duplicates and the identities of the elements. Thus, without loss of generality, we will ignore duplicates and simply assume elements $1, 2, 3, \ldots$ are inserted in order. In this discrete view, element t is inserted at time t. It is more convenient to consider an essentially equivalent *continuous* notion of time. A *standard exponential* is an exponential random variable with mean 1.

Definition 8.1 (continuous insertion process)**.** Let $X_0, X_1, \ldots$ be i.i.d. standard exponentials. Define $T_0 = 0$ and $T_{n+1} = T_n + X_n$ for $n \in \mathbb{N}$. Then the element i is inserted at time T_i. Let $H_1, H_2, \ldots$ be i.i.d. uniform random variables over $[0,1]$ where H_i is the hash value of the element i. The continuous *insertion process* is the random process $\mathsf{H} = (T_i, H_i)_{i \in \mathbb{N}_+}$. See Figure 8.2a for a visualization.

Now we define a continuous version of the k-Min process as a statistic of the insertion process.

Definition 8.2 (continuous k-Min process)**.** Let $\mathsf{H} = (T_i, H_i)_{i \in \mathbb{N}_+}$ be an insertion process. For a finite set $A \subset \mathbb{R}$, define $k\text{-min}(A)$ to be the kth minimum element in A where $k\text{-min}(A)$

is 1 if $|A| < k$. Then we define the *continuous k-Min process* as $S = (S_t)_{t>0}$ where for any $t > 0$,

$$S_t = k\text{-min}(\{H_i \mid T_i \leq t, i = 1, 2, \ldots\}).$$

Note that the k-Min process only depends on the *set* of the points $\{(T_i, H_i) \mid i \in \mathbb{N}_+\}$. Thus the ordering of H is unnecessary to simulate the k-Min process and we only need a *Poisson point process*.

Definition 8.3 (Poisson point process on $\mathbb{R}_+^2$)**.** A Poisson point process N on $\mathbb{R}_+^2$ is a random process $\{N(B) \mid B \text{ is a Borel set of } \mathbb{R}_+^2\}$. For any Borel set $B \subset \mathbb{R}_+^2$, $N(B)$ is a Poisson random variable with mean $|B|$, representing the number of points in region B. Furthermore, for any $w \in \mathbb{N}$ and disjoint Borel sets $B_1, \ldots, B_w$, $N(B_1), \ldots, N(B_w)$ are mutually independent.

Definition 8.4 (Poissonized k-Min process)**.** Let N be a Poisson point process on $\mathbb{R}_+^2$. Then we define the *poissonized k-Min process* as $S = (S_t)_{t>0}$ where for any $t > 0$,

$$S_t = \sup\{s \mid N([0, t] \times [0, s]) \leq k - 1\}.$$

Theorem 8.2. *Let $S = (S_t)_{t>0}$ be a continuous k-Min process and $S' = (S'_t)_{t>0}$ be a poissonized k-Min process. Then $(\mathsf{Min}(S'_t, 1))_{t>0}$ simulates $(S_t)_{t>0}$.*

Proof. See Figure 8.1 for a simple visualization of the proof. The set of points $\{(T_i, H_i) \mid i \in \mathbb{N}_+\}$ occurring in the insertion process simulates a Poisson point process N in the plane restricted to the strip $[0, \infty) \times [0, 1]$, where S_t is the kth minimum h-value in the rectangle $[0, t) \times [0, 1]$ and 1 if there are less than k points in it. Note that $S'_t = \sup\{s \mid N([0, t] \times [0, s])$ is exactly the kth minimum h-value of the points in the whole strip $[0, t] \times [0, \infty)$. When $S'_t > 1$, it is necessary that there are less than k points in $[0, t) \times [0, 1]$. When $S'_t \leq 1$, then S'_t and S_t coincide. We conclude that $S_t = \mathsf{Min}(S'_t, 1)$ for any $t > 0$ through this coupling. □

This theorem suggests that the poissonized k-Min process coincides with the continuous k-Min process once $t \gg k$. Since we only care about the case when t is large, we will work with the poissonized k-Min process from now on.

We will now first characterize the distribution of S_t for any $t > 0$.

Proposition 8.1. *Let $(S_t)_{t>0}$ be a poissonized k-Min process. Then for any $t > 0$, tS_t is a* Gamma$(k, 1)$ *random variable.*

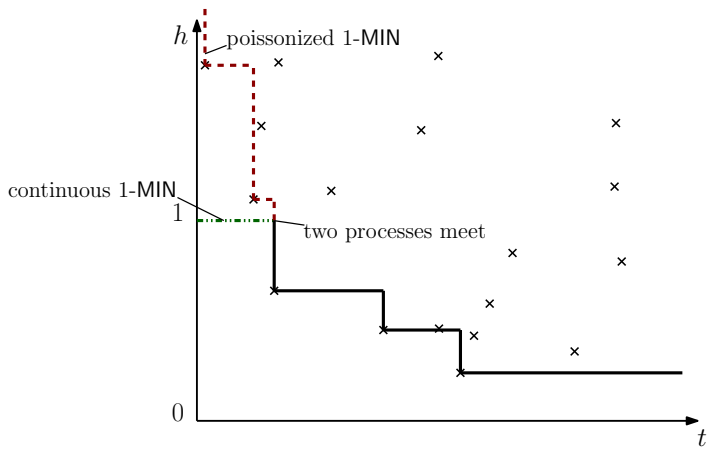


Figure 8.1: Visualization of the coupling between continuous 1-Min and poissonized 1-Min processes

Proof. The continuous k-Min process simulates a Poisson point process restricted to $[0,\infty)\times[0,1]$ *from left to right*. Now we simulate the Poisson point process restricted to $[0,t]\times[0,\infty)$ *from bottom to top*. See Figure 8.2b.

The h-values of the points in $[0,t]\times[0,\infty)$ form a Poisson point process of density t. Thus the vertical distance between consecutive points distributes as $t^{-1}X$ where X is a standard exponential. By definition, S_t is the vertical position of the kth minimum point. Thus

$$S_t \sim t^{-1}X_1' + t^{-1}X_2' + \ldots + t^{-1}X_k'.$$

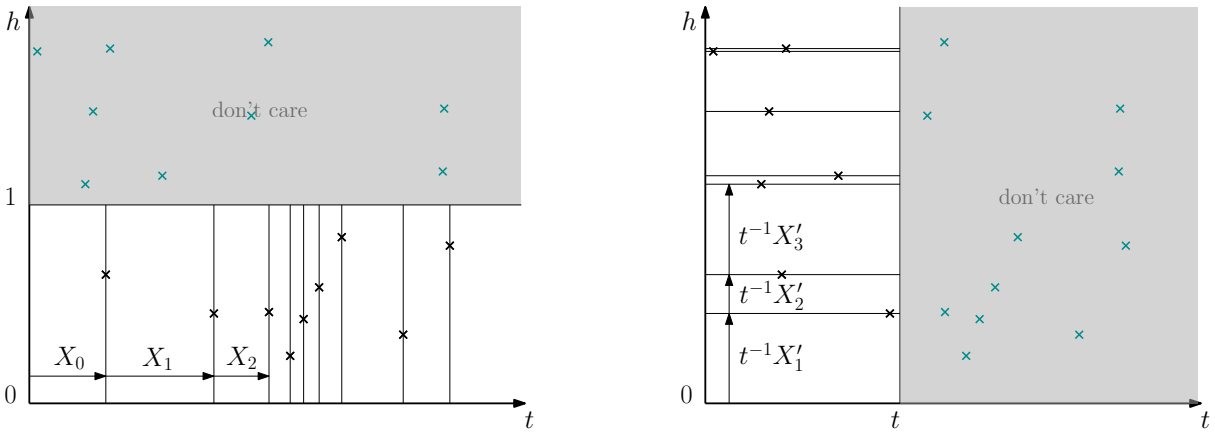
Thus tS_t is a sum of k i.i.d. standard exponentials, which is Gamma$(k,1)$. □

8.2.2 Backwards Evolution of the k-Min Process

Recall that we will observe the public and private sketches (sk_h and sk_g) when the *metronome* 1-Min sketch sk_m changes state. It is more convenient to consider these times in reverse-chronological order.

Definition 8.5 (backwards jumping times of 1-Min process)**.** Let $(S_t)_{t>0}$ be a 1-Min process. Let the *current time* be $t_0 > 0$. Define the *backwards jumping process* $(J_n)_{n\in\mathbb{N}}$ from time t_0 as follows.

$$J_0 = t_0$$



(a) insertion process

(b) bottom to top simulation

Figure 8.2: Two ways of simulating a Poisson point process. (a) and (b) simulate overlapping regions of a same underlying Poisson point process. See Figure 8.1 or 8.3 for the underlying point process. X_is and X'_is are i.i.d. standard exponentials.

$$J_{n+1} = \sup\{t' \leq J_n \mid S_{t'} \neq S_{J_n}\}, \forall n \in \mathbb{N}.$$

Theorem 8.3 (Markov chain of backwards jumping process)**.** *The backwards jumping process* $(J_n)_{n\in\mathbb{N}}$ *of 1-*Min *at time* t_0 *is a Markov chain which can be simulated as follows.*

$$J_0 = t_0$$
$$J_{n+1} = J_n U_n^*, \forall n \in \mathbb{N}$$

where the $\{U_n^*\}$ *are i.i.d. uniform random variables over* $[0, 1)$*.*

Proof. We reveal the randomness strip by strip. The horizontal position of the minimum point in an unrevealed strip $[0, J_n) \times [S_{J_n}, \infty)$ is uniformly random over $[0, t)$. Thus $J_{n+1} \sim J_n U_n^*$. See Figure 8.3 for a visualization of the proof. □

The backwards jumping times are the times generated by a private 1-Min at which we observe other independent k-Min processes. The sequence of observed sketch states with specific time sequence is characterized as follows.

Theorem 8.4 (Markov chain of backwards observation process)**.** *Let* $t_0 > t_1 > \ldots$ *be a series of* observation times*, listed in reverse-chronological order. Then for* $n \in \mathbb{N}$*,*

$$S_{t_{n+1}} = S_{t_n} + \sum_{j=1}^{k} \mathbb{1}\left[U_{n,j} > \frac{t_{n+1}}{t_n}\right] \frac{1}{t_{n+1}} X_{n,j}, \tag{8.1}$$

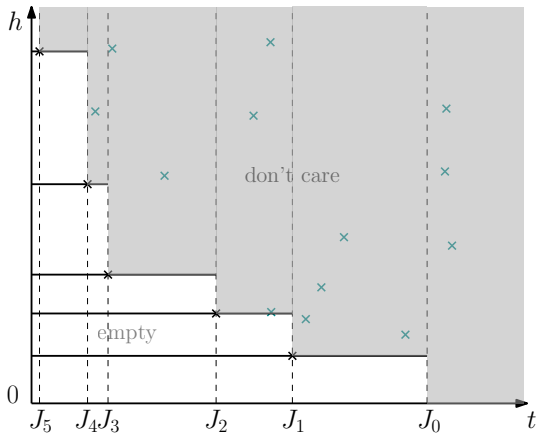


Figure 8.3: Visualization of the backwards jumping times of 1-Min. The points on the solid lines are uniformly at random on their corresponding lines.

where the $\{U_{n,j}\}$ *are i.i.d. uniform random variables over* $[0,1)$ *and the* $\{X_{n,j}\}$ *are i.i.d. standard exponentials.*

Proof. At time t_n, the kth minimum value is S_{t_n}, which means there are k points in the rectangle $[0,t_n] \times [0,S_{t_n}]$ whose t-values are i.i.d. uniformly distributed, denoted as $U_{n,j}t_n$ for $j = 1,\ldots,k$. Let the number of points in the rectangle $(t_{n+1},t_n] \times [0,S_{t_n}]$ be $R = \sum_{i=1}^{k} \mathbb{1}[U_{n,j}t_n > t_{n+1}]$. Then $S_{t_{n+1}}$ is the value of the Rth minimum value in the strip $[0,t_{n+1}] \times (S_{t_n},\infty)$. Thus we have the formula (8.1). See Figure 8.4 for a visualization.

Finally note that in the argument we assume the k minimum values are horizontally random, conditioning on the information we have already revealed. This is true since the $k-R$ points are still random within the rectangle $[0,t_{n+1}] \times [0,S_{t_n}]$ and the R new values remain unrevealed horizontally. For the former we only reveal how many of them are in the rectangle, for the latter we only reveal their vertical positions. Neither reveals any information about their horizontal positions. Thus the k minimum points at time t_{n+1} remain horizontally random. □

Suppose the current time is t_0. Consider a k-Min process $(S_t)_{t>0}$ that we observe in reverse-chronological order, at the jumping times $(J_n)_{n\in\mathbb{N}}$ of an independent 1-Min process, where $J_0 = t_0$. The following theorem shows that $(S_{J_n}J_n)_{n\in\mathbb{N}}$ forms a *Markov chain.*

Theorem 8.5. *Define a random process* $(Z_n)_{n\in\mathbb{N}}$ *as follows.*

$$Z_0 \sim \text{Gamma}(k,1)$$

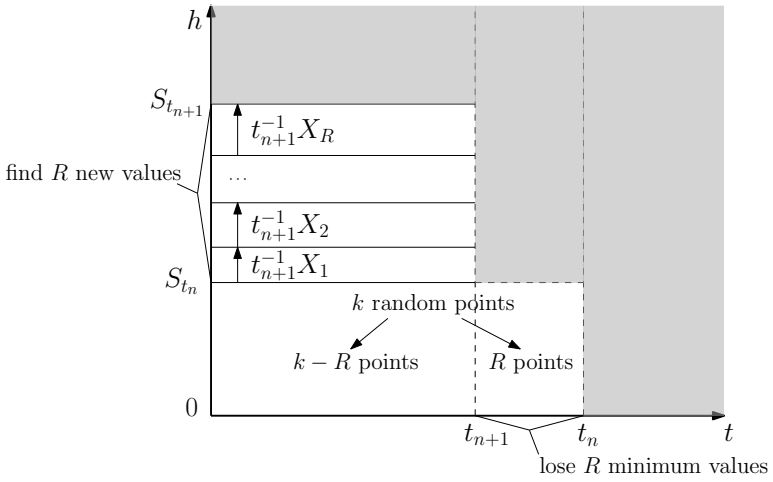


Figure 8.4: Backwards observation process, stepping from t_n to t_{n+1}. X_is are i.i.d. standard exponentials.

$$Z_{n+1} = Z_n U_n^* + \sum_{j=1}^{k} \mathbb{1}\left[U_{n;j} > U_n^*\right] X_{n;j}, \quad \forall n \in \mathbb{N}$$

where the $\{U_{n,j}\}$ *and* $\{U_n^*\}$ *are i.i.d. uniform random variables over* $[0,1)$*, and the* $\{X_{n,j}\}$ *are i.i.d. standard exponentials. Then* $(Z_n)_{n\in\mathbb{N}}$ *simulates* $(S_{J_n}J_n)_{n\in\mathbb{N}}$*. Furthermore, it is stationary, i.e.,* $Z_n \sim \text{Gamma}(k,1)$ *for any* $n \in \mathbb{N}$*.*

Proof. Follows directly from Proposition 8.1 and Theorem 8.3 and 8.4. □

We call the stationary Markov chain $(Z_n)_{n\in\mathbb{N}}$ the *backwards k-Min/1-Min Markov chain* or just *k/1-chain*. As its name suggests, it simulates the states of a k-Min observed backwards at the jumping times of an independent 1-Min.

Next, we are going to show that $\frac{1}{N}\sum_{n=1}^{N} \log Z_n$ concentrates.

8.3 Concentration of the Time Average of $\log Z_n$

In this section we present a concentration inequality for the time average $\frac{1}{N}\sum_{n=1}^{N} \log Z_n$ where $(Z_n)_{n\in\mathbb{N}}$ is a $k/1$-chain. The proof can be outlined as follows.

- We first construct a coupling $(Z_n, \tilde{Z}_n)_{n\in\mathbb{N}}$ where $(\tilde{Z}_n)_{n\in\mathbb{N}}$ is a *capped Markov process* where $\tilde{Z}_n \in [\tau, \tau']$ for all $n \in \mathbb{N}$ with some parameters τ, τ'. At the same time, with

high probability, the two processes coincide *in the range of interest*, i.e. $Z_n = \tilde{Z}_n$ for all $n = 1, \dots N$.

- We then prove $\frac{1}{N}\sum_{n=1}^{N} \log \tilde{Z}_n$ concentrates as N increases by Azuma's inequality [71], since $\tilde{Z}_n$ is capped between $[\tau, \tau']$ now and the martingale differences can be bounded by a constant depending on τ and τ', which is the core technical step.
- By choosing proper parameters τ and τ', we finally conclude that with high probability $\frac{1}{N}\sum_{n=1}^{N} \log Z_n$ concentrates, since both the event $\frac{1}{N}\sum_{n=1}^{N} \log \tilde{Z}_n$ concentrates and the event $Z_n = \tilde{Z}_n$ for all $n = 1, \dots N$ happen with high probability.

8.3.1 The Capped Markov Process

Let $(Z_n)_{n\in\mathbb{N}}$ be a $k/1$-chain. Fix $0 < \tau < \tau'$. We construct a coupling $(Z_n, \tilde{Z}_n)_{n\in\mathbb{N}}$ as follows

$$\begin{aligned}\tilde{Z}_0 &= \mathrm{cap}(Z_0, [\tau, \tau']) \\ \tilde{Z}_{n+1} &= \mathrm{cap}\left(\tilde{Z}_n U_n^* + \sum_{j=1}^{K} \mathbb{1}\left[U_{n;j} \geq U_n^*\right] X_{n;j}, [\tau, \tau']\right), \quad \forall n \in \mathbb{N}\end{aligned}$$

where the cap function is defined as, for $x \in \mathbb{R}$,

$$\mathrm{cap}(x, [\tau, \tau']) = \begin{cases} \tau, & x < \tau, \\ x, & x \in [\tau, \tau'], \\ \tau', & x > \tau'. \end{cases}$$

Remark 8.1. Note that $(\tilde{Z}_n)_{n\in\mathbb{N}}$ is itself a Markov chain. One the other hand, the directly capped process $(\mathrm{cap}(Z_n, [\tau, \tau']))_{n\in\mathbb{N}}$ is *not* Markovian.

We estimate the probability that $\tilde{Z}_n$ coincides with Z_n for every $n = 1, \dots, N$.

Theorem 8.6. *Let $N \in \mathbb{N}$. Then we have*

$$\mathbb{P}(Z_n = \tilde{Z}_n, \forall n \in [N]) \geq 1 - N(\tau^k + ke^{-\tau'/k}). \tag{8.2}$$

Proof. Note that Z_n is a Gamma$(k, 1)$ random variable.

$$\mathbb{P}(Z_n < \tau) = \int_0^\tau \frac{1}{\Gamma(k)} e^{-x} x^{k-1}\, dx$$

we bound e^{-x} by 1 and then note $\Gamma(k) = (k-1)!$

$$\leq \frac{1}{\Gamma(k)} \int_0^\tau x^{k-1}\, dx = \frac{1}{\Gamma(k)} \left. \frac{x^k}{k} \right|_0^\tau \leq \frac{\tau^k}{k!} \leq \tau^k.$$

For the upper tail, we replace Z_n with the sum of k standard exponential random variables Y_is, which gives

$$\mathbb{P}(Z_n > \tau') \leq \sum_{i=1}^{k} \mathbb{P}(Y_i > \tau'/k) = ke^{-\tau'/k}.$$

Thus by a union bound, for any $k \geq 1$,

$$\mathbb{P}(Z_n \in [\tau, \tau'], \forall n \in [N]) \geq 1 - N(\tau^k + ke^{-\tau'/k}),$$

where $(Z_n \in [\tau, \tau'], \forall n \in [N])$ necessarily implies $(Z_n = \tilde{Z}_n, \forall n \in [N])$. □

8.3.2 Martingale Difference

Consider the standard martingale decomposition. Define

- $M = \sum_{n=1}^{N} \log \tilde{Z}_n$,
- $\mathcal{F}_n$ is the σ-algebra generated by $(U_{i;j}, X_{i;j})_{i \leq n, j \in [K]}$ and $(U_i^*)_{i \leq n}$,
- $M_n = \mathbb{E}(M|\mathcal{F}_n)$, and thus $(M_n)_{n \in \mathbb{N}}$ is a martingale,
- $\Delta_n = M_n - M_{n-1}$ is the martingale difference.

The martingale difference can be bounded as follows.

$$|\Delta_n| = |M_n - M_{n-1}| = \left| \mathbb{E}\left(\sum_{i=1}^{N} \log \tilde{Z}_i - \mathbb{E}\left(\sum_{i=1}^{N} \log \tilde{Z}_i | \mathcal{F}_{n-1} \right) | \mathcal{F}_n \right) \right|$$

note that $\tilde{Z}_i$ is known from $\mathcal{F}_{n-1}$ for every $i \leq n$

$$= \left| \mathbb{E}\left(\sum_{i=n+1}^{N} \log \tilde{Z}_i - \mathbb{E}\left(\sum_{i=n+1}^{N} \log \tilde{Z}_i | \tilde{Z}_n \right) | \tilde{Z}_{n+1}, \tilde{Z}_n \right) \right|$$
$$= \left| \mathbb{E}\left(\sum_{i=0}^{N-n-1} \log \tilde{Z}_i^{(1)} - \sum_{i=0}^{N-n-1} \log \tilde{Z}_i^{(2)} | \tilde{Z}_0^{(1)} = \tilde{Z}_{n+1}, \tilde{Z}_{-1}^{(2)} = \tilde{Z}_n \right) \right|,$$

where $(\tilde{Z}_n^{(1)}, \tilde{Z}_n^{(2)})_{n\in\mathbb{N}}$ are arbitrarily coupled copies of $(\tilde{Z}_n)_{n\in\mathbb{N}}$

$$\leq \mathbb{E}\left(\sum_{i=0}^{N-n-1}\left|\log\tilde{Z}_i^{(1)} - \log\tilde{Z}_i^{(2)}\right| | \tilde{Z}_0^{(1)} = \tilde{Z}_{n+1}, \tilde{Z}_{-1}^{(2)} = \tilde{Z}_n\right),$$
$$\leq \mathbb{E}\left(\sum_{i=0}^{\infty}\left|\log\tilde{Z}_i^{(1)} - \log\tilde{Z}_i^{(2)}\right| | \tilde{Z}_0^{(1)} = \tilde{Z}_{n+1}, \tilde{Z}_{-1}^{(2)} = \tilde{Z}_n\right), \tag{8.3}$$

We now define a specific coupling. Random processes like $(Z_n)_{n\in\mathbb{N}}$ and $(U_{n,j})$ will be written as Z and U if it is clear in the context.

Definition 8.6 (coupling difference for capped $k/1$-chain)**.** Define

$$\tilde{\phi}(x,y) = \mathbb{E}\left(\sum_{n=0}^{\infty}\left|\log(\tilde{Z}_n^{(1)}) - \log(\tilde{Z}_n^{(2)})\right| \;\middle|\; \tilde{Z}_0^{(1)} = x, \tilde{Z}_0^{(2)} = y\right),$$

where $\tilde{Z}^{(1)}, \tilde{Z}^{(2)}$ are copies of $\tilde{Z}$ that share the same randomness except for having different initial states, i.e. they use the same U, X, U^*.

With this coupling, (8.3) gives

$$|\Delta_n| \leq \mathbb{E}(\tilde{\phi}(\tilde{Z}_0^{(1)}, \tilde{Z}_0^{(2)}) | \tilde{Z}_0^{(1)} = \tilde{Z}_{n+1}, \tilde{Z}_{-1}^{(2)} = \tilde{Z}_n)$$

by construction, $\tilde{Z}_0^{(1)}, \tilde{Z}_0^{(2)} \in [\tau, \tau']$, thus we have

$$\leq \sup_{x,y\in[\tau,\tau']} \tilde{\phi}(x,y), \tag{8.4}$$

which is a uniform bound for all $|\Delta_n|$s, which makes the Azuma's inequality applicable.

However, due to the cap function in the capped Markov process, it is not convenient to analyze $\tilde{\phi}(x,y)$. On the other hand, the same quantity $\phi(x,y)$ in the original process is tractable.

Definition 8.7 (coupling difference for $k/1$-chain)**.** Define

$$\phi(x,y) = \mathbb{E}\left(\sum_{n=0}^{\infty}\left|\log(Z_n^{(1)}) - \log(Z_n^{(2)})\right| \;\middle|\; Z_0^{(1)} = x, Z_0^{(2)} = y\right),$$

where $Z^{(1)}, Z^{(2)}$ are copies of Z that share the same randomness except for having different initial states, i.e. they use the same U, X, U^*.

We now prove that $\tilde{\phi}(x,y)$ can be bounded by $\phi(x,y)$.

Lemma 8.1. *Let $0 < \tau < \tau'$. Let $a_n, b_n \geq 0$ be two sequences. For $x > 0$ and $n \in \mathbb{N}$, define $f(x,n)$ and $g(x,n)$ recursively as follows.*

$$\begin{aligned} f(x,0) &= x \\ f(x,n+1) &= f(x,n)a_n + b_n \\ g(x,0) &= \mathrm{cap}(x,[\tau,\tau']) \\ g(x,n+1) &= \mathrm{cap}(g(x,n)a_n + b_n, [\tau,\tau']). \end{aligned}$$

Then for any $0 < x \leq y$ and $n \in \mathbb{N}$, one of the following two happens.

- *$g(x,n) = g(y,n)$, or*
- *$f(x,n) \leq g(x,n) \leq g(y,n) \leq f(y,n)$.*

This implies that for any $0 < x \leq y$ and any $n \in \mathbb{N}$,

$$|\log(g(x,n)) - \log(g(y,n))| \leq |\log(f(x,n)) - \log(f(y,n))|.$$

Proof. Note that if $g(x,n) = g(y,n)$ for some $n \in \mathbb{N}$. Then $g(x,k) = g(y,k)$ for all $k \geq n$.

Consider the case $n = 0$. If $x \leq y \leq \tau$, then $g(x,0) = g(y,0) = \tau$. If $\tau' \leq x \leq y$, then $g(x,0) = g(y,0) = \tau'$. In both cases the lemma holds. Thus now $x < \tau'$ and $y > \tau$. $x < \tau'$ implies that $f(x,0) \leq g(x,0)$. $y > \tau$ implies that $g(y,0) \leq f(y,0)$. Also, trivially we have $g(x,0) \leq g(y,0)$ since $x \leq y$. Thus the lemma holds for $n = 0$.

Now assume the lemma holds for $n = k$. Consider the case $n = k+1$. If $g(x,k) = g(y,k)$, we are done. By inductive hypothesis, we have $f(x,n) \leq g(x,n) \leq g(y,n) \leq f(y,n)$. Consider the following cases.

- If $g(x,n)a_n + b_n \geq \tau'$, then necessarily $g(y,n)a_n + b_n \geq \tau'$. Thus $g(x,n+1) = g(y,n+1)$.
- If $g(y,n)a_n + b_n \leq \tau$, then necessarily $g(x,n)a_n + b_n \leq \tau$. Thus $g(x,n+1) = g(y,n+1)$.
- If neither of the above two happen, then $f(x,n+1) \leq g(x,n+1) \leq g(y,n+1) \leq f(y,n+1)$.

This finishes the induction. Finally note that $\log(\cdot)$ is an increasing function and thus we have

$$\log(f(x,n)) \leq \log(g(x,n)) \leq \log(g(y,n)) \leq \log(f(y,n)),$$

and thus $|\log(g(x,n)) - \log(g(y,n))| \le |\log(f(x,n)) - \log(f(y,n))|$. □

Corollary 8.3. *For any $x, y \in [\tau, \tau']$, $\tilde{\phi}(x,y) \le \phi(x,y)$.*

Proof. Now we couple $(Z_n^{(1)}, Z_n^{(2)}, \tilde{Z}_n^{(1)}, \tilde{Z}_n^{(2)})$ together using the same randomness except for the initial states where

- $Z_n^{(1)} = \tilde{Z}_n^{(1)} = x$,
- $Z_n^{(2)} = \tilde{Z}_n^{(2)} = y$.

Set $a_n = U_n^*$ and $b_n = \sum_{j=1}^k \mathbb{1}\left[U_{n;j} > U_n^*\right] X_{n;j}$, then we have by construction for any $n \in \mathbb{N}$,

$$(Z_n^{(1)}, Z_n^{(2)}, \tilde{Z}_n^{(1)}, \tilde{Z}_n^2) = (f(x,n), f(y,n), g(x,n), g(y,n))$$

. Then Lemma 8.1 implies that under any realization of U, X, U^*, for any $n \in \mathbb{N}$, we have

$$|\log \tilde{Z}_n^{(1)} - \log \tilde{Z}_n^{(2)}| \le |\log Z_n^{(1)} - \log Z_n^{(2)}|.$$

Thus we conclude that for $x, y \in [\tau, \tau']$.

$$\begin{aligned}
\tilde{\phi}(x,y) &= \mathbb{E}\left(\sum_{n=0}^{\infty} \left|\log(\tilde{Z}_n^{(1)}) - \log(\tilde{Z}_n^{(2)})\right| \;\middle|\; \tilde{Z}_0^{(1)} = x, \tilde{Z}_0^{(2)} = y\right) \\
&\le \mathbb{E}\left(\sum_{n=0}^{\infty} \left|\log(Z_n^{(1)}) - \log(Z_n^{(2)})\right| \;\middle|\; Z_0^{(1)} = x, Z_0^{(2)} = y\right) \\
&= \phi(x,y).
\end{aligned}$$

□

We now estimate $\phi(x,y)$.

Theorem 8.7 (coupling difference)**.** *There exists constant $C_1 > 0$ such that for any $x, y > 0$,*

$$\phi(x,y) \le C_1(|\log x| + \log^2(y+1) + 1),$$

Proof. By direct induction, we know

$$Z_n = Z_0 \prod_{j=0}^{n-1} U_j^* + \sum_{i=0}^{n-1} \sum_{j=1}^{k} \mathbb{1}\left[U_{i;j} > U_i^*\right] X_{i;j} \prod_{j=i+1}^{n-1} U_j^*.$$

Thus we have

$$\mathbb{E}\left(\left|\log(Z_n^{(1)}) - \log(Z_n^{(2)})\right| \,|Z_0^{(1)} = x, Z_0^{(2)} = y\right)$$
$$= \mathbb{E}\log\frac{y\prod_{j=0}^{n-1} U_j^* + \sum_{i=0}^{n-1}\sum_{j=1}^{k} \mathbb{1}\left[U_{i;j} > U_i^*\right] X_{i;j}\prod_{j=i+1}^{n-1} U_j^*}{x\prod_{j=0}^{n-1} U_j^* + \sum_{i=0}^{n-1}\sum_{j=1}^{k} \mathbb{1}\left[U_{i;j} > U_i^*\right] X_{i;j}\prod_{j=i+1}^{n-1} U_j^*}.$$

Since $y > x$, the expression above becomes smaller when a same positive amount is removed from both numerator and denominator. Let $U_1, \ldots, U_i$ be i.i.d. uniform random variables and $X_1, \ldots, X_i$ be i.i.d. standard exponentials. We then have

$$\leq \mathbb{E}\log\frac{y\prod_{j=0}^{n-1} U_j^* + \sum_{i=0}^{n-1} \mathbb{1}\left[U_i > U_i^*\right] X_i\prod_{j=i+1}^{n-1} U_j^*}{x\prod_{j=0}^{n-1} U_j^* + \sum_{i=0}^{n-1} \mathbb{1}\left[U_i > U_i^*\right] X_i\prod_{j=i+1}^{n-1} U_j^*} \leq \log\frac{y}{x}.$$

We want to prove that $\mathbb{E}\left(\left|\log(Z_n^{(1)}) - \log(Z_n^{(2)})\right| \;\middle|\; Z_0^{(1)} = x, Z_0^{(2)} = y\right)$ decays sub-exponentially as n increases. The key step is to prove that the following event happens with high probability. Consider the event E_n where

$$\text{Event } E_n\text{:} \qquad \sum_{i=0}^{n-1} \mathbb{1}\left[U_i > U_i^*\right] X_i \prod_{j=i+1}^{n-1} U_j^* \geq e^{\alpha n}\prod_{j=0}^{n-1} U_j^*,$$

where $\alpha > 0$ is some constant to be determined. Note that $\mathbb{P}(\mathbb{1}\left[U_i > U_i^*\right] = 0) = \frac{1}{2}$. For event E_n to happen, we need $\mathbb{1}\left[U_i > U_i^*\right] = 1$ for large enough number of is. Let A_n be the event that

$$\text{Event } A_n\text{:} \qquad W_n \stackrel{\text{def}}{=} \sum_{i=(1-\beta)n}^{n-1} \mathbb{1}\left[U_i > U_i^*\right] \geq \kappa\beta n,$$

where $\beta \in (0,1)$ and $\kappa \in (0, 1/2)$ are constants to be determined. Note that W_n is a binomial random variable with βn trials and probability $1/2$ to be successful. We use the following concentration result.

Lemma 8.2 (by Theorem 4 in [43])**.** *Let X be a binomial random variable with n trials and successful rate $1/2$. Then for $\delta \geq 0$,*

$$\mathbb{P}(X \leq n/2 - \delta) \leq e^{-\frac{\delta^2}{n}}.$$

By the lemma we immediately have

$$\mathbb{P}(\overline{A_n}) = \mathbb{P}(W_n < \kappa\beta n) = \mathbb{P}(W_n < \beta n/2 - (1/2-\kappa)\beta n) \le \exp\left(-(1/2-\kappa)^2\beta n\right).$$

Then let $Y_1, \ldots, Y_{\kappa\beta n}$ be i.i.d. standard exponentials.

$$\begin{aligned}\mathbb{P}(\overline{E_n} \cap A_n) &\le \mathbb{P}\left(\sum_{i=1}^{\kappa\beta n} Y_i \prod_{j=(1-\beta)n+1}^{n-1} U_j^* < e^{\alpha n} \prod_{j=0}^{n-1} U_j^*\right) \\ &\le \mathbb{P}\left(\left(\sum_{i=1}^{\kappa\beta n} Y_i\right)^{-1} \prod_{j=0}^{(1-\beta)n} U_j^* > e^{-\alpha n}\right)\end{aligned}$$

by Markov's inequality,

$$\le \mathbb{E}\left(\left(\sum_{i=1}^{\kappa\beta n} Y_i\right)^{-1} \prod_{j=0}^{(1-\beta)n} U_j^*\right) e^{\alpha n},$$

where $\mathbb{E}(\sum_{i=1}^{\kappa\beta n} Y_i)^{-1} = \frac{1}{\kappa\beta n - 1}$ (when $\kappa\beta n \ge 2$) and $\mathbb{E}U_j^* = 1/2$,

$$= \frac{1}{\kappa\beta n - 1} 2^{-(1-\beta)n-1} e^{\alpha n}.$$

We then have for $n \ge 2/(\kappa\beta)$,

$$\begin{aligned}\mathbb{P}(\overline{E_n}) &\le \mathbb{P}(\overline{E_n} \cap A_n) + \mathbb{P}(\overline{A_n}) \\ &\le 2^{-(1-\beta)n-1} e^{\alpha n} + \exp\left(-(1/2-\kappa)^2\beta n\right)\end{aligned}$$

and therefore that

$$\begin{aligned}&\mathbb{E}\left(\left|\log(Z_n^{(1)}) - \log(Z_n^{(2)})\right| \, | Z_0^{(1)} = x, Z_0^{(2)} = y\right) \\ &\le \mathbb{P}(E_n) \log \frac{y + e^{\alpha n}}{x + e^{\alpha n}} + \mathbb{P}(\overline{E_n}) \log \frac{y}{x} \\ &\le \log \frac{y + e^{\alpha n}}{x + e^{\alpha n}} + (2^{-(1-\beta)n-1} e^{\alpha n} + \exp\left(-(1/2-\kappa)^2\beta n\right)) \log \frac{y}{x}. \qquad (8.5)\end{aligned}$$

The bound (8.5) is indeed sub-exponential. We next sum the bound (8.5) over $n \in \mathbb{N}$. Note that

$$\sum_{n=0}^{\infty} \log \frac{y + e^{\alpha n}}{x + e^{\alpha n}} \le \sum_{n=0}^{\infty} \log \frac{y + e^{\alpha n}}{e^{\alpha n}}$$

For the terms in which $e^{\alpha n} \le y$, we use $\log \frac{y+e^{\alpha n}}{e^{\alpha n}} \le \log(y+1)$; for the terms in which $e^{\alpha n} > y$, we use $\log \frac{y+e^{\alpha n}}{e^{\alpha n}} \le ye^{-\alpha n}$. The number of n for which $e^{\alpha n} \le y$ is at most $\frac{1}{\alpha}\log(y+1)$ and when $e^{\alpha n} > y$, $ye^{-\alpha n}$ is a geometric sequence with initial value smaller than 1. Then we have

$$\le \frac{1}{\alpha}\log^2(y+1) + \sum_{j=0}^{\infty} e^{-\alpha j} = \frac{1}{\alpha}\log^2(y+1) + \frac{1}{1-e^{-\alpha}}. \tag{8.6}$$

Also

$$\sum_{n=0}^{\infty}(2^{-(1-\beta)n-1}e^{\alpha n} + \exp\left(-(1/2-\kappa)^2\beta n\right)) = \frac{1/2}{1-e^{-(1-\beta)\log 2+\alpha}} + \frac{1}{1-e^{-(1/2-\kappa)^2\beta}}. \tag{8.7}$$

Finally, when $n < 2/\kappa\beta$, we bound $\mathbb{E}\left(\left|\log(Z_n^{(1)}) - \log(Z_n^{(2)})\right| |Z_0^{(1)} = x, Z_0^{(2)} = y\right)$ by $\log \frac{y}{x}$. Thus we add one more $\frac{2}{\kappa\beta}\log\frac{y}{x}$ to the final estimation. Together with (8.5), (8.6) and (8.7), we conclude that

$$\begin{aligned}
&\mathbb{E}\left(\sum_{n=0}^{\infty}\left|\log(Z_n^{(1)}) - \log(Z_n^{(2)})\right| |Z_0^{(1)} = x, Z_0^{(2)} = y\right) \\
&= \sum_{n=0}^{\infty}\mathbb{E}\left(\left|\log(Z_n^{(1)}) - \log(Z_n^{(2)})\right| |Z_0^{(1)} = x, Z_0^{(2)} = y\right) \\
&\le \underbrace{\frac{2}{\kappa\beta}\log\frac{y}{x}}_{\text{when } n < 2/\kappa\beta} + \underbrace{\frac{1}{\alpha}\log^2(y+1) + \frac{1}{1-e^{-\alpha}}}_{\sum_{n=0}^{\infty}\mathbb{P}(E_n)\log\frac{y+e^{\alpha n}}{x+e^{\alpha n}}} + \underbrace{\left(\frac{1/2}{1-e^{-(1-\beta)\log 2+\alpha}} + \frac{1}{1-e^{-(1/2-\kappa)^2\beta}}\right)\log\frac{y}{x}}_{\sum_{n=0}^{\infty}\mathbb{P}(\overline{E_n})\log\frac{y}{x}}.
\end{aligned}$$

The theorem is proved by choosing $\alpha = 0.05, \kappa = 0.2, \beta = 0.5$ and the expression above is bounded by $26\log\frac{y}{x} + 20\log^2(y+1) + 21 \le 26(\log\frac{y}{x} + \log^2(y+1) + 1)$. □

Corollary 8.4. *Let $0 < \tau < \tau'$ such that $\tau' > 2$ and $|\log\tau| > |\log\tau'|^2$. Then there exists constant $C_2 > 0$ such that for any $x, y \in [\tau, \tau']$,*

$$\tilde{\phi}(x,y) \le C_2|\log\tau|. \tag{8.8}$$

Furthermore, for any $n \in \mathbb{N}$.

$$|\Delta_n| \le C_2|\log\tau|. \tag{8.9}$$

Proof. By Theorem 8.7 shows that $\phi(x,y) \le C_1(|\log x| + \log^2(y+1) + 1)$ for some $C_1 > 0$. Since we assume $\tau' > 2$ and $|\log\tau| > \log^2\tau'$, we have $2\log\tau' > 1$ and $\tau' + 1 \le \tau'^2$. We then have

- $|\log x| \le \max(|\log \tau|, \log \tau') \le |\log \tau| + \log \tau' \le |\log \tau| + 2\log^2 \tau' < 3|\log \tau|$.
- $\log^2(y+1) \le \log^2(\tau'+1) \le \log^2(\tau'^2) \le 4\log^2(\tau') < 4|\log \tau|$.
- $1 < (2\log(\tau'))^2 < 4|\log \tau|$.

Thus $\phi(x,y) \le C_1(|\log x| + \log^2(y+1)+1) \le 11C_1|\log \tau|$. Finally, by Corollary 8.3, we have $\tilde{\phi}(x,y) \le \phi(x,y) \le 11C_1|\log \tau|$. (8.9) then follows directly from (8.8) and (8.4). □

8.3.3 Concentration of The Capped Process

Theorem 8.8 (Azuma–Hoeffding inequality)**.** *If $(M_n)_{0\le n\le N}$ is a martingale such that there is a sequence $c_1, \dots, c_N \ge 0$ that for any n,*

$$|M_n - M_{n-1}| \le c_n,$$

then for $\epsilon > 0$,

$$\mathbb{P}(|M_n - M_0| \ge \epsilon) \le 2\exp\left(-\frac{\epsilon^2}{2\sum_{n=1}^N c_n^2}\right).$$

Theorem 8.9. *Let $0 < \tau < \tau'$ where $\tau' > 2$ and $|\log \tau| > |\log \tau'|^2$. Then let $\mu = \frac{1}{N}\mathbb{E}\sum_{n=1}^N \log \tilde{Z}_n$.*

$$\mathbb{P}\left(\left|\frac{1}{N}\sum_{n=1}^N \log \tilde{Z}_n - \mu\right| > \epsilon\right) \le 2\exp\left(-\frac{\epsilon^2 N}{C_3|\log \tau|^2}\right),$$

where C_3 is some positive constant.

Proof. Corollary 8.4 shows that $|\Delta_n| \le C_2|\log \tau|$ for some $C_2 > 0$. By Azuma's inequality, we have for $t \ge 0$,

$$\begin{aligned}\mathbb{P}(|M - \mathbb{E}M| \ge N\epsilon) &\le 2\exp\left(-\frac{N^2\epsilon^2}{2NC_2^2|\log \tau|^2}\right) \\ &\le 2\exp\left(-\frac{N\epsilon^2}{2C_2^2|\log \tau|^2}\right).\end{aligned}$$

□

Corollary 8.5. *Let $\mu = \frac{1}{N}\mathbb{E}\sum_{n=1}^{N}\log\tilde{Z}_n$. There are some constants $C_4, C_5 > 0$ such that for any $\epsilon \in (0,1)$, $\beta \in [1,k]$, and $N > C_4\beta^{1.01}\log^7 k$,*

$$\mathbb{P}\left(\left|\frac{1}{N}\sum_{n=1}^{N}\log Z_n - \mu\right| > \epsilon\right) \le 2(1+N)\exp\left(-C_5\epsilon^2 N^{1/3}\beta^{2/3}\right).$$

Proof. By union bound,

$$\mathbb{P}\left(\left|\frac{1}{N}\sum_{n=1}^{N}\log Z_n - \mu\right| > \epsilon\right) \le \mathbb{P}(Z_n \neq \tilde{Z}_n, \exists n \in [N]) + \mathbb{P}\left(\left|\frac{1}{N}\sum_{n=1}^{N}\log \tilde{Z}_n - \mu\right| > \epsilon\right)$$

by Theorem 8.6 and 8.9

$$\le N(\tau^k + ke^{-\tau'/k}) + 2\exp\left(-\frac{\epsilon^2 N}{C_3|\log\tau|^2}\right),$$

where $C_3 > 0$ is some constant and we need to satisfy the following conditions to apply Theorem 8.9.

- $\tau' > 2$ and $|\log\tau| > |\log\tau'|^2$.

Let $\beta \le k$. Choose $\tau = e^{-(N/\beta)^{1/3}}$ and $\tau' = (N\beta^2)^{1/3}k^{1.01}$. We may choose a large enough C_4 so that $N > C_4\beta^{1.01}\log^7 k$ implies the condition above. Thus

$$\mathbb{P}\left(\left|\frac{1}{N}\sum_{n=1}^{N}\log S_n - \mu\right| > \epsilon\right) \le 2\exp\left(-\frac{\epsilon^2 N}{C_3(N/\beta)^{2/3}}\right) + N(e^{-k\beta^{-1/3}N^{1/3}} + e^{-N^{1/3}\beta^{2/3}})$$

since $\beta \le k$, $k\beta^{-1/3} \ge \beta^{2/3}$

$$\le 2(1+N)\exp\left(-\min(C_3^{-1}\epsilon^2, 1)N^{1/3}\beta^{2/3}\right)$$

□

8.4 Summary of Fraud Detection

We define *data fraud* in the context of a Monte Carlo randomized data structure to be any input that is selected as a function of the *internal randomness* of the data structure—which we assume to have been completely compromised—in order to induce incorrect behavior. In distributed systems it is often important to support *mergeability* among data sketches,

which necessitates sharing hash functions (internal randomness) across many machines, and across time. For this reason, we regard the security of the hash functions used to support mergeability as *much* weaker than randomness generated locally, and on the fly.

In this chapter we gave what we believe to be the first rigorous proof-of-concept that detecting *data fraud* of cardinality sketches is feasible using a small, private "sanity check" sketch.[7] Our results show a tradeoff between the *severity* of the attack (ϵ), the size of the sanity-check sketch (k), and the *latency* (N) of detecting data fraud.

This direction leaves several specific and general open questions, among them:

- What is the optimum tradeoff among N, k, ϵ for detecting data fraud attacks on cardinality sketches, that persistently causes errors on the order of e^{ϵ}? Our detector measured the ratio of the probability-of-state-change between the public and private sketches. It may be more efficient to also consider second or higher moments.

- The cardinality estimation problem is defined on *insertion-only* streams, in which there is a natural definition of "time" (the cardinality of the set so far). This notion of time played a big part in designing our fraud detector. Many data sketches are linear projections of a large vector $\mathbf{x} \in \mathbb{Z}^n$ subject to coordinate wise updates $(i, \Delta) \in [n] \times \mathbb{Z}$, indicating that $x(i) \leftarrow x(i) + \Delta$. (All linear sketches are mergeable, and therefore subject to the same pressure to share randomness across machines to support merges and queries involving multiple sketches.) It would be interesting to define a coherent notion of *data fraud* for linear sketches, and to develop fraud detectors for important problems, e.g., heavy hitters, L_p-estimation, etc.

[7] Reviriego and Ting [143] proposed a general method for detecting such attacks, but it was never fully specified nor analyzed. See the related work paragraph.

BIBLIOGRAPHY

[1] Felix Abramovich and Ya'acov Ritov. *Statistical theory: a concise introduction*. CRC Press, 2013.

[2] Ishaq Aden-Ali, Yanjun Han, Jelani Nelson, and Huacheng Yu. On the amortized complexity of approximate counting. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, pages 33:1–33:17, 2024.

[3] Yuqing Ai, Wei Hu, Yi Li, and David P Woodruff. New characterizations in turnstile streams with applications. In *31st Conference on Computational Complexity (CCC)*, 2016.

[4] Miklós Ajtai, Vladimir Braverman, T. S. Jayram, Sandeep Silwal, Alec Sun, David P. Woodruff, and Samson Zhou. The white-box adversarial data stream model. In *Proceedings of the 41st ACM Symposium on Principles of Database Systems (PODS)*, pages 15–27, 2022.

[5] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *Journal of Computer and System Sciences*, 58(1):137–147, 1999.

[6] Charles W. Bachman. The programmer as navigator. *Commun. ACM*, 16(11):653–658, November 1973.

[7] Jakob Bæk Tejs Houen, Rasmus Pagh, and Stefan Walzer. Simple set sketching. In *Proceedings of the 6th SIAM Symposium on Simplicity in Algorithms (SOSA)*, pages 228–241, 2023.

[8] Ziv Bar-Yossef, Thathachar S Jayram, Ravi Kumar, and D Sivakumar. An information statistics approach to data stream and communication complexity. *Journal of Computer and System Sciences*, 68(4):702–732, 2004.

[9] Ran Ben-Basat, Gil Einziger, Shir Landau Feibish, Jalil Moraney, and Danny Raz. Network-wide routing-oblivious heavy hitters. In *Proceedings of the 2018 Symposium on Architectures for Networking and Communications Systems (ANCS)*, pages 66–73, 2018.

[10] Omri Ben-Eliezer, Rajesh Jayaram, David P Woodruff, and Eylon Yogev. A framework for adversarially robust streaming algorithms. *Journal of the ACM*, 69(2):1–33, 2022.

[11] P Bickel and K Doksum. Mathematical statistics: Basic ideas and selected topics. 2d. ed. vol. 1 prentice hall. *Upper Saddle River, NJ*, 2001.

[12] Daniel K. Blandford and Guy E. Blelloch. Compact dictionaries for variable-length keys and data with applications. *ACM Trans. Algorithms*, 4(2):17:1–17:25, 2008.

[13] Jarosław Błasiok. Optimal streaming and tracking distinct elements with high probability. *ACM Trans. Algorithms*, 16(1):3:1–3:28, 2020.

[14] Jarosław Błasiok, Vladimir Braverman, Stephen R Chestnut, Robert Krauthgamer, and Lin F Yang. Streaming symmetric norms via measure concentration. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 716–729, 2017.

[15] Guillaume Boglioni Beaulieu. *Pairwise versus mutual independence: visualization, actuarial applications and central limit theorems*. PhD thesis, UNSW Sydney, 2023.

[16] Károly Böröczky and Gergely Wintsche. Covering the sphere by equal spherical balls. *Discrete and Computational Geometry: The Goodman-Pollack Festschrift*, pages 235–251, 2003.

[17] George E. P. Box. Science and statistics. *Journal of the American Statistical Association*, 71(356):791–799, 1976.

[18] Vladimir Braverman and Stephen R. Chestnut. Universal sketches for the frequency negative moments and other decreasing streaming sums. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, pages 591–605, 2015.

[19] Vladimir Braverman, Stephen R Chestnut, David P Woodruff, and Lin F Yang. Streaming space complexity of nearly all functions of one variable on frequency vectors. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS)*, pages 261–276, 2016.

[20] Vladimir Braverman, Dan Feldman, Harry Lang, and Daniela Rus. Streaming coreset constructions for M-estimators. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, 2019.

[21] Vladimir Braverman, Avinatan Hassidim, Yossi Matias, Mariano Schain, Sandeep Silwal, and Samson Zhou. Adversarial robustness of streaming algorithms through importance sampling. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 3544–3557, 2021.

[22] Vladimir Braverman, Jonathan Katzman, Charles Seidell, and Gregory Vorsanger. An optimal algorithm for large frequency moments using $O(n^{1-2/k})$ bits. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, 2014.

[23] Vladimir Braverman and Rafail Ostrovsky. Zero-one frequency laws. In *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC)*, pages 281–290, 2010.

[24] Vladimir Braverman and Rafail Ostrovsky. Approximating large frequency moments with pick-and-drop sampling. In *International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX)*, pages 42–57, 2013.

[25] Vladimir Braverman and Rafail Ostrovsky. Generalizing the layering method of Indyk and Woodruff: Recursive sketches for frequency-based vectors on streams. In *International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX)*, pages 58–70, 2013.

[26] Vladimir Braverman, Rafail Ostrovsky, and Alan Roytman. Zero-one laws for sliding windows and universal sketches. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, 2015.

[27] Andrei Z. Broder. On the resemblance and containment of documents. In *Proceedings of Compression and Complexity of SEQUENCES*, pages 21–29, 1997.

[28] Florian Brück, Jan-Frederik Mai, and Matthias Scherer. Exchangeable min-id sequences: Characterization, exponent measures and non-decreasing id-processes. *Extremes*, 26(1):183–239, 2023.

[29] Thilina Buddhika, Matthew Malensek, Sangmi Lee Pallickara, and Shrideep Pallickara. Synopsis: A distributed sketch over voluminous spatiotemporal observational streams. *IEEE Trans. Knowl. Data Eng.*, 29(11):2552–2566, 2017.

[30] G. Casella and R. L. Berger. *Statistical Inference, 2nd Ed.* Brooks/Cole, Belmont, CA, 2002.

[31] Y. Chabchoub, R. Chiky, and B Dogan. HyperLogLog and EWMA detect port scan attacks in IP traffic? *J. Info. Security*, 5, 2014.

[32] Ivan Chajda, Radomír Halaš, and Jan Kühr. *Semilattice structures*, volume 30. Heldermann Lemgo, 2007.

[33] 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 IEEE Annual Conference on Computational Complexity (CCC)*, pages 107–117, 2003.

[34] Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. *Theoretical Computer Science*, 312(1):3–15, 2004.

[35] Philippe Chassaing and Lucas Gerin. Efficient estimation of the cardinality of large data sets. In *Proceedings of the 4th Colloquium on Mathematics and Computer Science Algorithms, Trees, Combinatorics and Probabilities*, 2006.

[36] Aiyou Chen, Jin Cao, Larry Shepp, and Tuan Nguyen. Distinct counting with a self-learning bitmap. *Journal of the American Statistical Association*, 106(495):879–890, 2011.

[37] Justin Y. Chen, Piotr Indyk, and David P. Woodruff. Space-optimal profile estimation in data streams with applications to symmetric functions. In *Proceedings of the 15th Innovations in Theoretical Computer Science Conference (ITCS)*, pages 32:1–32:22, 2024.

[38] Min Chen, Shigang Chen, and Zhiping Cai. Counter tree: A scalable counter architecture for per-flow traffic measurement. *IEEE/ACM Trans. Netw.*, 25(2):1249–1262, 2017.

[39] Stephen Robert Chestnut. *Stream sketches, sampling, and sabotage*. PhD thesis, Johns Hopkins University, 2015.

[40] Pern Hui Chia, Damien Desfontaines, Irippuge Milinda Perera, Daniel Simmons-Marengo, Chao Li, Wei-Yen Day, Qiushi Wang, and Miguel Guevara. KHyperLogLog: Estimating reidentifiability and joinability of large data at scale. In *Proceedings of the 2019 IEEE Symposium on Security and Privacy*, pages 350–364, 2019.

[41] Balarengadurai Chinnaiah. CEP support for detection of application layer attacks. *International Journal of Applied Engineering Research*, 13(18):13683–13688, 2018.

[42] Balarengadurai Chinnaiah. Protection of DDoS attacks at the application layer: HyperLogLog (HLL) cardinality estimation. In *Cognitive Informatics and Soft Computing*, pages 595–604, 2021.

[43] Fan Chung and Linyuan Lu. Concentration inequalities and martingale inequalities: a survey. *Internet Mathematics*, 3(1):79–127, 2006.

[44] David Clayton, Christopher Patton, and Thomas Shrimpton. Probabilistic data structures in adversarial environments. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 1317–1334, 2019.

[45] Vera Clemens, Lars-Christian Schulz, Marten Gartner, and David Hausheer. DDoS detection in P4 using HYPERLOGLOG and COUNTMIN sketches. In *Proceedings of the 2023 IEEE/IFIP Network Operations and Management Symposium (NOMS)*, pages 1–6, 2023.

[46] Alfred H Clifford and Gordon B Preston. The algebraic theory of semigroups, vol. 1. *Mathematical surveys*, 7, 1961.

[47] Peter Clifford and Ioana A. Cosma. A statistical analysis of probabilistic counting algorithms. *Scandinavian Journal of Statistics*, 39(1):1–14, 2012.

[48] Edith Cohen. Size-estimation framework with applications to transitive closure and reachability. *Journal of Computer and System Sciences*, 55(3):441–453, 1997.

[49] Edith Cohen. All-distances sketches, revisited: HIP estimators for massive graphs analysis. *IEEE Trans. Knowl. Data Eng.*, 27(9):2320–2334, 2015.

[50] Edith Cohen. Hyperloglog hyperextended: Sketches for concave sublinear frequency statistics. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 105–114, 2017.

[51] Edith Cohen. Stream sampling framework and application for frequency cap statistics. *ACM Transactions on Algorithms*, 14(4):1–40, 2018.

[52] Edith Cohen and Ofir Geri. Sampling sketches for concave sublinear functions of frequencies. In *In Proceedings of the 32nd Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 1361–1371, 2019.

[53] Edith Cohen and Haim Kaplan. Tighter estimation using bottom-k sketches. *Proceedings of the VLDB Endowment*, 1(1):213–224, 2008.

[54] Edith Cohen, Rasmus Pagh, and David P. Woodruff. WOR and p's: Sketches for ℓ_p-sampling without replacement. In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[55] Reuven Cohen, Liran Katzir, and Aviv Yehezkel. A minimal variance estimator for the cardinality of big data set intersection. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 95–103, 2017.

[56] Jeffrey Considine, Feifei Li, George Kollios, and John W. Byers. Approximate aggregation techniques for sensor databases. In *Proceedings of the 20th International Conference on Data Engineering (ICDE)*, pages 449–460, 2004.

[57] Stephen A. Cook and Robert A. Reckhow. Time bounded random access machines. *Journal of Computer and System Sciences*, 7(4):354–375, 1973.

[58] Don Coppersmith and Ravi Kumar. An improved data stream algorithm for frequency moments. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 151–156, 2004.

[59] Graham Cormode, Mayur Datar, Piotr Indyk, and Shanmugavelayutham Muthukrishnan. Comparing data streams using Hamming norms (how to zero in). *IEEE Transactions on Knowledge and Data Engineering*, 15(3):529–540, 2003.

[60] Graham Cormode and Donatella Firmani. A unifying framework for ℓ_0-sampling algorithms. *Distributed Parallel Databases*, 32(3):315–335, 2014.

[61] Graham Cormode and Marios Hadjieleftheriou. Methods for finding frequent items in data streams. *The VLDB Journal*, 19:3–20, 2010.

[62] Graham Cormode and Shan Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithms*, 55(1):58–75, 2005.

[63] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Second Edition)*. Wiley, 2006.

[64] Noel Cressie. A note on the behaviour of the stable distributions for small index α. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 33(1):61–64, 1975.

[65] Miklós Csűrös. Approximate counting with a floating-point counter. In *International Computing and Combinatorics Conference*, pages 358–367, 2010.

[66] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[67] Damien Desfontaines, Andreas Lochbihler, and David A. Basin. Cardinality estimators do not preserve privacy. *Proc. Priv. Enhancing Technol.*, 2019(2):26–46, 2019.

[68] Charlie Dickens, Justin Thaler, and Daniel Ting. Order-invariant cardinality estimators are differentially private. In *Proceedings of 35th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[69] Yevgeniy Dodis, Mihai Patrascu, and Mikkel Thorup. Changing base without losing space. In *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC)*, pages 593–602, 2010.

[70] David L Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[71] D. P. Dubhashi and A. Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, 2009.

[72] Marianne Durand. *Combinatoire analytique et algorithmique des ensembles de données. (Multivariate holonomy, applications in combinatorics, and analysis of algorithms)*. PhD thesis, Ecole Polytechnique X, 2004.

[73] Marianne Durand and Philippe Flajolet. Loglog counting of large cardinalities. In *Proceedings 11th Annual European Symposium on Algorithms (ESA)*, volume 2832 of *Lecture Notes in Computer Science*, pages 605–617. Springer, 2003.

[74] Rick Durrett. *Probability: theory and examples*. Cambridge University Press, 2019.

[75] R. A. Leo Elworth, Qi Wang, Pavan K. Kota, C. J. Barberan, Benjamin Coleman, Advait Balaji, Gaurav Gupta, Richard G. Baraniuk, Anshumali Shrivastava, and Todd J. Treangen. To petabytes and beyond: recent advances in probabilistic and signal processing algorithms and their application to metagenomics. *Nucleic Acids Research*, 48(10):5217–5234, 2020.

[76] David Eppstein, Zvi Galil, and Giuseppe F Italiano. Dynamic graph algorithms. *Algorithms and Theory of Computation Handbook*, 1:9–1, 1999.

[77] Otmar Ertl. New cardinality estimation methods for HyperLogLog sketches. *CoRR*, abs/1706.07290, 2017.

[78] Otmar Ertl. Ultraloglog: A practical and more space-efficient alternative to hyperloglog for approximate distinct counting. *Proceedings of the VLDB Endowment*, 17(7):1655–1668, 2024.

[79] 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.

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

[81] Philippe Flajolet. On adaptive sampling. *Computing*, 43(4):391–400, 1990.

[82] Philippe Flajolet, Éric Fusy, Olivier Gandouet, and Frédéric Meunier. Hyperloglog: the analysis of a near-optimal cardinality estimation algorithm. In *Discrete Mathematics and Theoretical Computer Science*, pages 137–156, 2007.

[83] Philippe Flajolet and G Nigel Martin. Probabilistic counting algorithms for data base applications. *Journal of Computer and System Sciences*, 31(2):182–209, 1985.

[84] Gerald B. Folland. *A course in abstract harmonic analysis*, volume 29. CRC press, 2016.

[85] Michael J. Freitag and Thomas Neumann. Every row counts: Combining sketches and sampling for accurate group-by result estimates. In *Proceedings of the 9th Biennial Conference on Innovative Data Systems Research (CIDR)*, 2019.

[86] Michael Fuchs, Chung-Kuei Lee, and Helmut Prodinger. Approximate counting via the Poisson-Laplace-Mellin method. In *Discrete Mathematics and Theoretical Computer Science*, pages 13–28. Discrete Mathematics and Theoretical Computer Science, 2012.

[87] Sumit Ganguly. Estimating frequency moments of data streams using random linear combinations. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, pages 369–380, 2004.

[88] Sumit Ganguly, Mohit Bansal, and Shruti Dube. Estimating hybrid frequency moments of data streams. *Journal of Combinatorial Optimization*, 23(3):373–394, 2012.

[89] Phillip B. Gibbons and Srikanta Tirthapura. Estimating simple functions on the union of data streams. In *Proceedings 13th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 281–291, 2001.

[90] Frédéric Giroire. Order statistics and estimating cardinalities of massive data sets. *Discret. Appl. Math.*, 157(2):406–427, 2009.

[91] André Gronemeier and Martin Sauerhoff. Applying approximate counting for computing the frequency moments of long data streams. *Theory of Computing Systems*, 44:332–348, 2009.

[92] Avinatan Hassidim, Haim Kaplan, Yishay Mansour, Yossi Matias, and Uri Stemmer. Adversarially robust streaming algorithms via differential privacy. In *Proceedings 33rd Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[93] Jonathan Hehir, Daniel Ting, and Graham Cormode. Sketch-flip-merge: Mergeable sketches for private distinct counting. In *Proceedings International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pages 12846–12865. PMLR, 2023.

[94] Ahmed Helmi, Jérémie Lumbroso, Conrado Martínez, and Alfredo Viola. Data Streams as Random Permutations: the Distinct Element Problem. In *Proceedings of the 23rd International Meeting on Probabilistic, Combinatorial, and Asymptotic Methods for the Analysis of Algorithms (AofA)*, pages 323–338, 2012.

[95] Stefan Heule, Marc Nunkesser, and Alexander Hall. Hyperloglog in practice: Algorithmic engineering of a state of the art cardinality estimation algorithm. In *Proceedings of the 16th International Conference on Extending Database Technology*, pages 683–692, 2013.

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

[97] Piotr Indyk. Sketching, streaming and sublinear-space algorithms (graduate course notes). https://people.csail.mit.edu/indyk/ita-web.pdf, 2007.

[98] Piotr Indyk and David Woodruff. Optimal approximations of the frequency moments of data streams. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 202–208, 2005.

[99] Piotr Indyk and David P. Woodruff. Tight lower bounds for the distinct elements problem. In *Proceedings of the 44th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 283–288, 2003.

[100] Svante Janson, Jérémie Lumbroso, and Robert Sedgewick. Bit-array-based alternatives to HyperLogLog. In *Proceedings of the 35th International Conference on Probabilistic, Combinatorial and Asymptotic Methods for the Analysis of Algorithms (AofA)*, volume 302, pages 5:1–5:19, 2024.

[101] Rajesh Jayaram. *Sketching and Sampling Algorithms for High-Dimensional Data*. PhD thesis, Carnegie Mellon University Pittsburgh, PA, 2021.

[102] Rajesh Jayaram and David Woodruff. Perfect l_p sampling in a data stream. *SIAM Journal on Computing*, 50(2):382–439, 2021.

[103] Rajesh Jayaram and David P Woodruff. Data streams with bounded deletions. In *Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS)*, pages 341–354, 2018.

[104] Rajesh Jayaram, David P Woodruff, and Samson Zhou. Truly perfect samplers for data streams and sliding windows. In *Proceedings of the 41st ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS)*, pages 29–40, 2022.

[105] Rajesh Jayaram, David P Woodruff, and Samson Zhou. Streaming algorithms with few state changes. *Proceedings of the ACM on Management of Data*, 2(2):1–28, 2024.

[106] William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 26(189-206):1, 1984.

[107] Hossein Jowhari, Mert Sağlam, and Gábor Tardos. Tight bounds for l_p samplers, finding duplicates in streams, and related problems. In *Proceedings of the 30th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*, pages 49–58, 2011.

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

[109] Daniel M. Kane, Jelani Nelson, and David P. Woodruff. On the exact space complexity of sketching and streaming small norms. In *Proceedings of the 2010 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1161–1178, 2010.

[110] 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.

[111] Richard M Karp. On-line algorithms versus off-line algorithms: How much. In *Algorithms, Software, Architecture: Information Processing 92: Proceedings of the IFIP 12th World Computer Congress*, volume 1, page 416, 1992.

[112] Matti Karppa and Rasmus Pagh. Hyperlogloglog: cardinality estimation with one log more. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 753–761, 2022.

[113] Sato Ken-Iti. *Lévy processes and infinitely divisible distributions*, volume 68. Cambridge university press, 1999.

[114] Donald E. Knuth. Theory and practice, 1991.

[115] Joseph B Kruskal and Albert G. Greenberg. A flexible way of counting large numbers approximately in small registers. *Algorithmica*, 6:590–596, 1991.

[116] Abhishek Kumar, Minho Sung, Jun Xu, and Jia Wang. Data streaming algorithms for efficient and accurate estimation of flow size distribution. *ACM SIGMETRICS Performance Evaluation Review*, 32(1):177–188, 2004.

[117] Kevin J. Lang. Back to the future: an even more nearly optimal cardinality estimation algorithm. *CoRR*, abs/1708.06839, 2017.

[118] Ping Li. Estimators and tail bounds for dimension reduction in l_α $(0 < \alpha \leq 2)$ using stable random projections. In *Proceedings 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 10–19, 2008.

[119] Yi Li, Huy L. Nguyen, and David P. Woodruff. Turnstile streaming algorithms might as well be linear sketches. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC)*, pages 174–183, 2014.

[120] Zaoxing Liu, Antonis Manousis, Gregory Vorsanger, Vyas Sekar, and Vladimir Braverman. One sketch to rule them all: Rethinking network flow monitoring with univmon. In *Proceedings of the 2016 ACM SIGCOMM Conference*, pages 101–114, 2016.

[121] Guy Louchard and Helmut Prodinger. Generalized approximate counting revisited. *Theoretical Computer Science*, 391(1):109–125, 2008.

[122] Jérémie Lumbroso. An optimal cardinality estimation algorithm based on order statistics and its full analysis. In *Proceedings of the 21st International Meeting on Probabilistic, Combinatorial, and Asymptotic Methods in the Analysis of Algorithms (AofA)*, pages 489–504, 2010.

[123] Jérémie O Lumbroso. The story of HyperLogLog: How Flajolet processed streams with coin flips. *arXiv preprint arXiv:1805.00612*, 2018.

[124] Samuel Madden and Michael J Franklin. Fjording the stream: An architecture for queries over streaming sensor data. In *Proceedings of the 18th International Conference on Data Engineering (ICDE)*, pages 555–566, 2002.

[125] Guillaume Marçais, Brad Solomon, Rob Patro, and Carl Kingsford. Sketching and sublinear data structures in genomics. *Annual Review of Biomedical Data Science*, 2(1):93–118, 2019.

[126] Andrew McGregor. Graph stream algorithms: a survey. *ACM SIGMOD Record*, 43(1):9–20, 2014.

[127] Alistair Moffat, Radford M. Neal, and Ian H. Witten. Arithmetic coding revisited. *ACM Trans. Inf. Syst.*, 16(3):256–294, 1998.

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

[129] Robin A Moser and Gábor Tardos. A constructive proof of the general Lovász local lemma. *Journal of the ACM*, 57(2):1–15, 2010.

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

[131] Moni Naor and Eylon Yogev. Bloom filters in adversarial environments. *ACM Trans. Algorithms*, 15(3):35:1–35:30, 2019.

[132] Suman Nath, Phillip B. Gibbons, Srinivasan Seshan, and Zachary R. Anderson. Synopsis diffusion for robust aggregation in sensor networks. *ACM Trans. Sens. Networks*, 4(2):7:1–7:40, 2008.

[133] Jelani Nelson and Huacheng Yu. Optimal bounds for approximate counting. In *Proceedings of the 41st ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS)*, pages 119–127, 2022.

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

[135] Mihai Pătraşcu. Succincter. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 305–313, 2008.

[136] KA Penson and K Górska. Exact and explicit probability densities for one-sided lévy stable distributions. *Physical Review Letters*, 105(21):210604, 2010.

[137] Seth Pettie and Dingyu Wang. Information theoretic limits of cardinality estimation: Fisher meets Shannon. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 556–569, 2021.

[138] Seth Pettie and Dingyu Wang. Sketching, moment estimation, and the Lévy-Khintchine representation theorem. In *16th Innovations in Theoretical Computer Science Conference (ITCS)*, pages 77:1–77:23, 2025.

[139] Seth Pettie and Dingyu Wang. Universal perfect samplers for incremental streams. In *Proceedings of the 36th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3409–3422, 2025.

[140] Seth Pettie, Dingyu Wang, and Longhui Yin. Non-mergeable sketching for cardinality estimation. In *Proceedings of the 48th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 104:1–104:20, 2021.

[141] Ninh Pham. Hybrid LSH: faster near neighbors reporting in high-dimensional space. In *Proceedings of the 20th International Conference on Extending Database Technology (EDBT)*, pages 454–457, 2017.

[142] Eric Price and David P. Woodruff. Applications of the Shannon-Hartley theorem to data streams and sparse recovery. In *2012 IEEE International Symposium on Information Theory Proceedings*, pages 2446–2450, 2012.

[143] Pedro Reviriego and Daniel Ting. Security of HyperLogLog (HLL) cardinality estimation: Vulnerabilities and protection. *IEEE Commun. Lett.*, 24(5):976–980, 2020.

[144] CA Rogers. Covering a sphere with spheres. *Mathematika*, 10(2):157–164, 1963.

[145] Walter A. Rosenkrantz. Approximate counting: a martingale approach. *Stochastics*, 20(2):111–120, 1987.

[146] Björn Scheuermann and Martin Mauve. Near-optimal compression of probabilistic counting sketches for networking applications. In *Proceedings of the 4th International Workshop on Foundations of Mobile Computing (DIALM-POMC)*, 2007.

[147] Robert Sedgewick. Cardinality estimation. Presentation delivered at AofA (2016), Knuth-80 (2018), and Dagstuhl 19051 (2019). https://www.cs.princeton.edu/~rs/talks/Cardinality.pdf, 2019.

[148] Adam D. Smith, Shuang Song, and Abhradeep Thakurta. The Flajolet-Martin sketch itself preserves differential privacy: Private counting with minimal space. In *Proceedings Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[149] Guy L Steele Jr and Jean-Baptiste Tristan. Adding approximate counters. *ACM Transactions on Parallel Computing*, 4(1):1–45, 2017.

[150] The Apache Foundation. Apache DataSketches: A software library of stochastic streaming algorithms. https://datasketches.apache.org/. 2019.

[151] Daniel Ting. Streamed approximate counting of distinct elements: beating optimal batch methods. In *Proceedings 20th ACM Conference on Knowledge Discovery and Data Mining (KDD)*, pages 442–451, 2014.

[152] Daniel Ting. Towards optimal cardinality estimation of unions and intersections with sketches. In *Proceedings of the 22nd International ACM Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1195–1204, 2016.

[153] Jean-Baptiste Tristan, Joseph Tassarotti, and Guy L Steele Jr. Efficient training of LDA on a GPU by mean-for-mode Gibbs sampling. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, volume 37, 2015.

[154] A. W. van der Vaart. *Asymptotic Statistics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1998.

[155] Jeffrey S Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software*, 11(1):37–57, 1985.

[156] Dingyu Wang. Optimal protocols for 2-party contention resolution. In *Proceedings of the 23rd International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS)*, pages 456–468, 2021.

[157] Dingyu Wang. Probabilistic counting in generalized turnstile models. *arXiv preprint arXiv:2310.14977*, 2023.

[158] Dingyu Wang. Harmonic decomposition in data sketches. In *Proceedings of the 57th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, 2025.

[159] Dingyu Wang. Multi-dimensional approximate counting. In *Proceedings of the 8th SIAM Symposium on Simplicity in Algorithms (SOSA)*, pages 144–157, 2025.

[160] Dingyu Wang and Seth Pettie. Better cardinality estimators for HyperLogLog, PCSA, and beyond. In *Proceedings of the 42nd ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS)*, pages 317–327, 2023.

[161] Kyu-Young Whang, Brad T Vander-Zanden, and Howard M Taylor. A linear-time probabilistic counting algorithm for database applications. *ACM Transactions on Database Systems*, 15(2):208–229, 1990.

[162] Jake Wires, Stephen Ingram, Zachary Drudi, Nicholas J. A. Harvey, and Andrew Warfield. Characterizing storage workloads with counter stacks. In *Proceedings of the 11th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pages 335–349, 2014.

[163] Ian H. Witten, Radford M. Neal, and John G. Cleary. Arithmetic coding for data compression. *Commun. ACM*, 30(6):520–540, 1987.

[164] David Woodruff. Optimal space lower bounds for all frequency moments. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, page 167–175, 2004.

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

[166] David P. Woodruff and Samson Zhou. Tight bounds for adversarially robust streams and sliding windows via difference estimators. In *Proceedings 62nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 1183–1196, 2021.

[167] Qingjun Xiao, Shigang Chen, You Zhou, Min Chen, Junzhou Luo, Tengli Li, and Yibei Ling. Cardinality estimation for elephant flows: A compact solution based on virtual register sharing. *IEEE/ACM Trans. Netw.*, 25(6):3738–3752, 2017.

[168] Qingjun Xiao, Shigang Chen, You Zhou, and Junzhou Luo. Estimating cardinality for arbitrarily large data stream with improved memory efficiency. *IEEE/ACM Trans. Netw.*, 28(2):433–446, 2020.

[169] Yuri Yakubovich. A simple proof of the Lévy–Khintchine formula for subordinators. *Statistics & Probability Letters*, 176:109136, 2021.

[170] Pablo Zegers. Fisher information properties. *Entropy*, 17(7):4918–4939, 2015.

[171] Jianshu Zhao, Jean Pierre Both, Luis M Rodriguez-R, and Konstantinos T Konstantinidis. Gsearch: ultra-fast and scalable genome search by combining k-mer hashing with hierarchical navigable small world graphs. *Nucleic Acids Research*, 52(16):e74–e74, 07 2024.

[172] Yunpeng Zhao. A note on new Bernstein-type inequalities for the log-likelihood function of Bernoulli variables. *Statistics & Probability Letters*, 163:108779, 2020.