Information Theoretic Limits of Cardinality Estimation: Fisher Meets Shannon*

Seth Pettie
pettie@umich.edu
University of Michigan
Computer Science and Engineering
Ann Arbor, MI, USA

Dingyu Wang
wangdy@umich.edu
University of Michigan
Computer Science and Engineering
Ann Arbor, MI, USA

ABSTRACT

Estimating the *cardinality* (number of distinct elements) of a large multiset is a classic problem in streaming and sketching, dating back to Flajolet and Martin's classic Probabilistic Counting (PCSA) algorithm from 1983.

In this paper we study the intrinsic tradeoff between the *space complexity* of the sketch and its *estimation error* in the RANDOM ORACLE model. We define a new measure of efficiency for cardinality estimators called the *Fisher-Shannon* (Fish) number $\mathcal{H}/\mathcal{I}$. It captures the tension between the limiting Shannon entropy ($\mathcal{H}$) of the sketch and its normalized Fisher information ($\mathcal{I}$), which characterizes the variance of a statistically efficient, asymptotically unbiased estimator.

Our results are as follows.

(i) We prove that all base-q variants of Flajolet and Martin's PCSA sketch have Fish-number $H_0/I_0 \approx 1.98016$ and that every base-q variant of (Hyper)LogLog has Fish-number worse than H_0/I_0, but that they tend to H_0/I_0 in the limit as $q \to \infty$. Here H_0, I_0 are precisely defined constants.

(ii) We describe a sketch called Fishmonger that is based on a *smoothed*, entropy-compressed variant of PCSA with a different estimator function. It is proved that with high probability, Fishmonger processes a multiset of $[U]$ such that *at all times*, its space is $O(\log^2 \log U) + (1 + o(1))(H_0/I_0)b \approx 1.98b$ bits and its standard error is $1/\sqrt{b}$.

(iii) We give circumstantial evidence that H_0/I_0 is the optimum Fish-number of mergeable sketches for Cardinality Estimation. We define a class of *linearizable* sketches and prove that no member of this class can beat H_0/I_0. The popular mergeable sketches are, in fact, also linearizable.

CCS CONCEPTS

• **Theory of computation → Sketching and sampling**; *Lower bounds and information complexity*.

KEYWORDS

cardinality estimation, streaming algorithm

ACM Reference Format:
Seth Pettie and Dingyu Wang. 2021. Information Theoretic Limits of Cardinality Estimation: Fisher Meets Shannon. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC '21), June 21–25, 2021, Virtual, Italy.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3406325.3451032

1 INTRODUCTION

Cardinality Estimation (aka *Distinct Elements* or F_0*-estimation*) is a fundamental problem in streaming/sketching, with widespread industrial deployments in databases, networking, and sensing. Sketches for Cardinality Estimation are evaluated along three axes: ***space complexity*** (in bits), ***estimation error***, and ***algorithmic complexity***.

In the end we want a perfect understanding of the *three-way* tradeoff between these measures, but that is only possible if we truly understand the *two-way* tradeoff between space complexity and estimation error, which is information-theoretic in nature. In this paper we investigate this two-way tradeoff in the RANDOM ORACLE model.

Prior work in Cardinality Estimation has assumed either the RANDOM ORACLE model (in which we have query access to a uniformly random hash function) or what we call the STANDARD MODEL (in which unbiased random bits can be generated, but all hash functions are stored explicitly). Sketches in the RANDOM ORACLE model typically pay close attention to constant factors in both space and estimation error [6, 13, 14, 16, 18, 24, 27–30, 32, 33, 39, 40, 45, 47, 49]. Sketches in the STANDARD MODEL [3–5, 9, 31, 35, 37] use *explicit* (e.g., $O(1)$-wise independent) hash functions and generally pay less attention to the leading constants in space and estimation error. Sketches in the RANDOM ORACLE model have had a bigger impact on the *practice* of Cardinality Estimation [34, 47, 48]; they are typically simple and have empirical performance that agrees[1] with theoretical predictions [29, 30, 34, 48].

Random Oracle Model. It is assumed that we have oracle access to a uniformly random function $h : [U] \to \{0,1\}^\infty$, where $[U]$ is the universe of our multisets and the range is interpreted as a point in $[0,1]$. (To put prior work on similar footing we assume in Table 1 that such hash values are stored to $\log U$ bits of precision.)

*For a full version of this, see [44].

[1] One reason for this is surely the non-adversarial nature of real-world data sets, but even in adversarial settings we would expect RANDOM ORACLE sketches to work well, e.g., by using a (randomly seeded) cryptographic hash function. Furthermore, since many applications maintain *numerous* Cardinality Estimation sketches, they can afford to store a single $O(n^\epsilon)$-space high-performance hash function [15], whose space-cost is negligible, being amortized over the large number of sketches.

STOC '21, June 21–25, 2021, Virtual, Italy

ACM ISBN 978-1-4503-8053-9/21/06…$15.00
https://doi.org/10.1145/3406325.3451032

For practical purposes, elements in $[U]$ and $[0,1]$ can be regarded as 64-bit integers/floats.

Problem Definition. A sequence $\mathcal{A} = (a_1, \ldots, a_N) \in [U]^N$ over some universe $[U]$ is revealed one element at a time. We maintain a b-bit sketch $S \in \{0,1\}^b$ such that if S_i is its state after seeing $(a_1, \ldots a_i)$, S_{i+1} is a function of S_i and $h(a_{i+1})$. The goal is to be able to estimate the *cardinality* $\lambda = |\{a_1, \ldots, a_N\}|$ of the set. Define $\hat{\lambda}(S) : \{0,1\}^b \to \mathbb{R}$ to be the estimation function. An estimator is (ϵ, δ)-*approximate* if $\Pr(\hat{\lambda} \notin [(1-\epsilon)\lambda, (1+\epsilon)\lambda]) < \delta$. Most results in the RANDOM ORACLE model use estimators that are almost unbiased or asymptotically unbiased (as $b \to \infty$). Given that this holds it is natural to measure the distribution of $\hat{\lambda}$ relative to λ. We pay particular attention to the *relative* variance $\frac{1}{\lambda^2}\operatorname{Var}(\hat{\lambda} \mid \lambda)$ and the *relative* standard deviation $\frac{1}{\lambda}\sqrt{\operatorname{Var}(\hat{\lambda} \mid \lambda)}$, also called the *standard error*.

Remark 1. Table 1 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.

Formally, a b-bit *sketching scheme* is defined by a state transition function $T : \{0,1\}^b \times [0,1] \to \{0,1\}^b$ where $S_{i+1} = T(S_i, h(a_{i+1}))$ is the state after seeing $\{a_1, \ldots, a_{i+1}\}$. One can decompose T into a function family $\mathcal{F} \stackrel{\text{def}}{=} \{T(\cdot, r) \mid r \in [0,1]\}$ of possible *actions* on the sketch, and a probability distribution μ over $\mathcal{F}$. I.e., if R is the hash value, uniformly distributed in $[0,1]$, then $\mu(f) = \Pr(T(\cdot, R) = f)$. For example, the (Hyper)LogLog sketch [24, 29] stores m non-negative integers $(S(0), \ldots, S(m-1))$ and can be defined by the function family $\mathcal{F} = \{f_{i,j}\}$ and distribution $\mu(f_{i,j}) = m^{-1}2^{-j}$, $i \in [m]$, $j \in \mathbb{Z}^+$, where action $f_{i,j}$ updates the ith counter to be at least j:

$$f_{i,j}(S(0), \ldots, S(m-1)) = (S(0), \ldots, S(i-1), \max\{S(i), j\}, S(i+1), \ldots, S(m-1)).$$

Suppose we process the stream $\mathcal{A} = \{a_1, \ldots, a_N\}$ using a sketching scheme $(\mathcal{F}, \mu)$. If S_0 is the initial state, and $f_{a_i} \in \mathcal{F}$ is the action of a_i determined by $h(a_i)$, the final state is

$$F_{\mathcal{A}} \stackrel{\text{def}}{=} f_{a_N} \circ \cdots \circ f_{a_1}(S_0)$$

Naturally, one wants the *distribution* of the final state $F_{\mathcal{A}}$ to depend *solely* on λ, not the identity or permutation of $\mathcal{A}$. We define a sketching scheme $(\mathcal{F}, \mu)$ to be *history independent*[2] if it satisfies

History Independence: For any two sequences $\mathcal{A}_1$ and $\mathcal{A}_2$ with $|\mathcal{A}_1| = |\mathcal{A}_2|$, $F_{\mathcal{A}_1} \stackrel{d}{\sim} F_{\mathcal{A}_2}$ (distributionally identical).

Until quite recently [14, 18, 33, 45, 49], all sketching schemes achieved history independence by satisfying a stronger property. A *commutative idempotent function family* (CIFF) $\mathcal{F}$ consists of a set of functions from $\{0,1\}^b \to \{0,1\}^b$ that satisfy

Idempotency: For all $f \in \mathcal{F}$ and $S \in \{0,1\}^b$, $(f \circ f)(S) = f(S)$.

Commutativity: For all $f, g \in \mathcal{F}$ and $S \in \{0,1\}^b$, $(f \circ g)(S) = (g \circ f)(S)$.

We define a sketching scheme $(\mathcal{F}, \mu)$ to be *commutative* if $\mathcal{F}$ is a CIFF. Clearly any commutative sketching scheme satisfies history independence, but the reverse is not true. The main virtue of commutative sketching schemes is that they are *mergeable* [2].

Mergeability: If multisets $\mathcal{A}_1$ and $\mathcal{A}_2$ are sketched as S_1 and S_2 using the same random oracle/hash function h, then the sketch S for $\mathcal{A}_1 \cup \mathcal{A}_2$ is a function of S_1 and S_2.

E.g., in the MPC[3] model we could split the multiset among M machines, sketch them separately, and estimate the cardinality of their union by combining the M sketches.

In recent years a few cardinality estimation schemes have been proposed that are history independent but *non*-commutative, and therefore suited to stream-processing on a *single* machine. The S-Bitmap [14] and Recordinality [33] sketches are history-independent but non-commutative/non-mergeable, as are all sketches derived by the Cohen/Ting [18, 49] transformation, which we call the "Martingale" transformation[4] in Table 1. Not being the focus of this paper, we discuss non-commutative sketches in the full version [44], and evaluate a non-commutative, non-history independent sketch due to Sedgewick [47] called HyperBitBit.

1.1 Survey of Cardinality Estimation

1.1.1 Commutative Algorithms in the Random Oracle Model. Flajolet and Martin [30] designed the first non-trivial sketch, called Probabilistic Counting with Stochastic Averaging (PCSA). The basic sketch S is a $\log U$-bit vector where $S_i(j) = 1$ iff some $h(a_1), \ldots, h(a_i)$ begins with the prefix 0^j1. Their estimation function $\hat{\lambda}(S)$ depends only on the least significant 0-bit $\min\{j \mid S(j) = 0\}$, and achieves a constant-factor approximation with constant probability. By maintaining m such structures they brought the standard error down to roughly $0.78/\sqrt{m}$.[5]

Flajolet [28] analyzed a sketch proposed by Wegman called AdaptiveSampling. The sketch S_i stores an index l and a list L of *all* distinct hash values among $h(a_1), \ldots, h(a_i)$ that have 0^l as a prefix. Whenever $|L| > m$, we increment l, filter L appropriately and continue. The space is thus $m \log U + \log\log U$. Flajolet proved that $\hat{\lambda}(S) \propto |L|2^l$ has standard error approaching $1.21/\sqrt{m}$.

The PCSA estimator pays attention to the least significant 0-bit in the sketch rather than the most significant 1-bit, which results in slightly better error distribution (in terms of m) but is significantly more expensive to maintain in terms of storage ($\log U$ vs. $\log\log U$ bits to store the most significant bit.) Durand and Flajolet's LogLog sketch implements this change, with stochastic averaging. The hash function $h : U \to [m] \times \mathbb{Z}^+$ produces (j,k) with probability $m^{-1}2^{-k}$.

[2] This is closely related to the definition of *history independence* from [42], which was defined as a privacy measure.

[3] (Massively Parallel Computation)

[4] Cohen [18] called these estimators "HIP" (historic inverse probability) and Ting [49] called them "Streaming" sketches to emphasize that they only work in single-stream environments.

[5] The m structures are not independent. The stream $\mathcal{A}$ is partitioned into m streams $\mathcal{A}^{(0)}, \ldots, \mathcal{A}^{(m-1)}$ u.a.r. (using h), each of which is sketched separately. They call the process of combining estimates from these m sketches *stochastic averaging*.

Table 1: Algorithms analyzed in the RANDOM ORACLE model assume oracle access to a uniformly random hash function $h : [U] \to [0,1]$. 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 (⋆) 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 (⋆⋆) is neither commutative nor history independent.

RANDOM ORACLE MODEL

MERGEABLE SKETCHES		SKETCH SIZE (BITS)	APPROXIMATION GUARANTEE
Flajolet & Martin	(PCSA) 1983	$m \log U$	Std. err. $\approx 0.78/\sqrt{m}$
Flajolet	(AdaptiveSampling) 1990	$m \log U + \log\log U$	Std. err. $\approx 1.21/\sqrt{m}$
Durand & Flajolet	(LogLog) 2003	$m \log\log U$	Std. err. $\approx 1.3/\sqrt{m}$
Giroire	(MinCount) 2005	$m \log U$	Std. err. $\approx 1/\sqrt{m}$
Chassaing & Gerin	(MinCount) 2006	$m \log U$	Std. err. $\approx 1/\sqrt{m}$
Estan, Varghase & Fisk	(Multires.Bitmap) 2006	$m \log U$	Std. err. $O(1/\sqrt{m})$
Beyer, Haas, Reinwald Sismanis & Gemulla	2007	$m \log U$	Std. err. $\approx 1/\sqrt{m}$
Flajolet, Fusy, Gandouet & Meunier	(HyperLogLog) 2007	$m \log\log U$	Std. err. $\approx 1.04/\sqrt{m}$
Lumbroso	2010	$m \log U$	Std. err. $\approx 1/\sqrt{m}$
Lang	(Compressed FM85) 2017	$\approx \log U + 1.99m$ (in expectation)	Std. err. $\approx 1/\sqrt{m}$
new	(Fishmonger) 2020	$O(\log^2 \log U) + (1 + o(1))(H_0/I_0)m$ 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. $\approx \frac{\ln(eU/m)/2}{\sqrt{m}}$	(⋆)
Helmi, Lumbroso, Martínez & Viola	(Recordinality) 2012	$(1 + o(1))m \log U$	Std. err. $\tilde{O}(1)/\sqrt{m}$	(⋆)
Cohen	(Martingale LogLog) 2014	$m \log\log U + \log U$	Std. err. $\approx 0.833/\sqrt{m}$	(⋆)
Ting	(Martingale MinCount) 2014	$(m+1)\log U$	Std. err. $\approx 0.71/\sqrt{m}$	(⋆)
Sedgewick	(HyperBitBit) 2016	134	? (See [44])	(⋆⋆)

STANDARD MODEL

Alon, Matias & Szegedy	1996	$O(\log U)$	$(\epsilon, 2/\epsilon)$-approx., $\epsilon \geq 2$
Gibbons & Tirthapura	2001	$O(\epsilon^{-2} \log U \log \delta^{-1})$	(ϵ, δ)-approx.
Bar-Yossef, Kumar & Sivakumar	2002	$O(\epsilon^{-3} \log U \log \delta^{-1})$	(ϵ, δ)-approx.
Bar-Yossef, Jayram, Kumar, Sivakumar & Trevisan	2002	$O\left(\left[\epsilon^{-2} \log\log U + \log U\right] \log \delta^{-1}\right)$	(ϵ, δ)-approx.
Kane, Nelson & Woodruff	2015	$O([\epsilon^{-2} + \log U] \log \delta^{-1})$	(ϵ, δ)-approx.
Błasiok	2018	$O(\epsilon^{-2} \log \delta^{-1} + \log U)$	(ϵ, δ)-approx.

LOWER BOUNDS

Trivial		$\Omega(\log\log U)$	$(O(1), O(1))$-approx.	(RAND. ORACLE)
Alon, Matias & Szegedy	1996	$\Omega(\log U)$	$(O(1), O(1))$-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)

After processing $\{a_1, \ldots, a_i\}$, the sketch is defined to be

$$S_i(j) = \max\{k \mid \exists i' \in \{1, \ldots, i\}, h(a_{i'}) = (j, k)\}.$$

Durand and Flajolet's estimator $\hat{\lambda}(S)$ is based on taking the *geometric mean* of the estimators derived from the individual components $S(0), \ldots, S(m-1)$, i.e.,

$$\hat{\lambda}(S) \propto m \cdot 2^{m^{-1} \cdot \sum_{j=0}^{m-1} S(j)}.$$

It is shown to have a standard error tending to $1.3/\sqrt{m}$. The HyperLogLog sketch of Flajolet, Fusy, Gandouet, and Meunier [29] differs from LogLog only in the estimation function, which uses the harmonic mean rather than geometric mean.

$$\hat{\lambda}(S) \propto m^2 \left(\sum_{j=0}^{m-1} 2^{-S(j)} \right)^{-1}.$$

They proved it has standard error tending to $\approx 1.04/\sqrt{m}$ in the limit, where $1.04 \approx \sqrt{3 \ln 2 - 1}$.

Giroire [32] considered a class of sketches (MinCount) that splits the stream into m' sub-streams, and keeps the smallest k hash values in each substream. I.e., if we interpret $h : [U] \to [0, m')$, $S_i(j)$ stores the smallest k values among $\{h(a_1), \ldots, h(a_i)\} \cap [j, j+1)$. Chassaing & Gerin [13] showed that a suitable estimator for this sketch has standard error roughly $1/\sqrt{km' - 2}$, i.e., fixing $m = km'$ we are indifferent to k and m'. Lumbroso [40] gave a detailed analysis of asymptotic distribution of errors when $k = 1$ and offered better estimators for smaller cardinalities. When $k = 1$ this is also called m-Min or Bottom-m sketches [10, 17–19] popular in measuring document/set similarity.

1.1.2 Commutative Algorithms in the Standard Model. In the Standard Model one must explicitly account for the space of every hash function. Specifically, a k-wise independent function $h : [D] \to [R]$ requires $\Theta(k \log(DR))$ bits. Typically an ϵ-approximation ($\hat{\lambda} \in [(1-\epsilon)\lambda, (1+\epsilon)\lambda]$) is guaranteed with constant probability, and then amplified to $1 - \delta$ probability by taking the median of $O(\log \delta^{-1})$ trials. The following algorithms are commutative *in the abstract*, meaning that they are commutative if certain events occur, such as a hash function being injective on a particular set.

Gibbons and Tirthapura [31] rediscovered AdaptiveSampling [28] and proved that it achieves an (ϵ, δ)-guarantee using an $O(\epsilon^{-2} \log U \log \delta^{-1})$-bit sketch and $O(1)$-wise independent hash functions. The space was improved [4] to $O((\epsilon^{-2} \log\log U + \log U) \log \delta^{-1})$. Kane, Nelson, and Woodruff [37] designed a sketch that has size $O((\epsilon^{-2} + \log U) \log \delta^{-1})$, which is optimal when $\delta^{-1} = O(1)$ as it meets the $\Omega(\epsilon^{-2})$ lower bound of [35] (see also [11]) and the $\Omega(\log U)$ lower bound of [3]. Using more sophisticated techniques, Błasiok [9] derived an optimal sketch for all (ϵ, δ) with space $O(\epsilon^{-2} \log \delta^{-1} + \log U)$, which meets the $\Omega(\epsilon^{-2} \log \delta^{-1})$ lower bound of Jayram and Woodruff [36].

1.2 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 [21, 43, 46]. The idea of sketch compression goes back to the original Flajolet-Martin paper [30], 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 [23, 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 [9, 37] algorithms use this property, and further show that a compressed representation of these counters can be updated in $O(1)$ time [8].

The practical efforts to compress (Hyper)LogLog have used fixed-length codes rather than variable length codes. Xiao, Chen, Zhou, and Luo [52] 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; see [44]. An implementation of HyperLogLog in Apache *DataSketches* [48] 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 [29].

A recent proposal of Sedgewick [47] called HyperBitBit can also be construed as a lossy compression of LogLog. It has constant relative bias and variance, independent of sketch length; see [44].

Scheuemann and Mauve [46] experimented with compression of PCSA and HyperLogLog sketches to their entropy bounds via arithmetic coding, and noted that, with the usual estimation functions [29, 30], Compressed-PCSA is slightly smaller than Compressed-HLL for the same standard error. Lang [38] 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-HLL+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 [48]. 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 [48].

To sum up, the idea of compressing sketches has been studied since the beginning, heuristically [30, 47, 52], experimentally [46, 48], and numerically [38], but to our knowledge never analytically.

1.3 New Results

Our goal is to understand the intrinsic tradeoff between space and accuracy in Cardinality Estimation. This question has been answered up to a large constant factor in the Standard Model with matching upper and lower bounds of $\Theta(\epsilon^{-2} \log \delta^{-1} + \log U)$ [9,

35–37]. However, in the RANDOM ORACLE model we can aspire to understand this tradeoff precisely.

To answer this question 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 [12, 50].

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 [12, 50] 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:

$$\lim_{m\to\infty} \sqrt{m}\left(\hat{\lambda}(S) - \lambda\right) \sim \mathcal{N}\left(0, \frac{1}{I_{S(0)}(\lambda)}\right).$$

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 $\mathcal{N}\left(\lambda, \frac{1}{I_S(\lambda)}\right)$ with variance $1/I_S(\lambda) = 1/(m \cdot I_{S(0)}(\lambda))$. (See Section 2.)

Suppose for the moment that $I_S(\lambda)$ is scale-free, 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,[6] i.e., lower Fish-numbers are superior. The actual definition of Fish (Section 3.4) 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

$$H_0 = \frac{1}{\ln 2} + \sum_{k=1}^{\infty} \frac{1}{k} \log_2\left(1 + 1/k\right),$$
$$I_0 = \zeta(2) = \frac{\pi^2}{6}.$$

The constant H_0/I_0 is very close to Lang's [38] numerical calculations of 2-PCSA's entropy and mean squared error. Let q-LL be the natural base-q analogue of LogLog = 2-LL. Whereas the Fisher information for q-PCSA is expressed in terms of the *Riemann zeta* function ($\zeta(2)$), the Fisher information of q-LL is expressed 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-LL is always worse than PCSA, but approaches the efficiency of PCSA in the limit, i.e.,

$$\forall q > 1, \mathsf{Fish}(q\text{-LL}) > H_0/I_0,$$
$$\text{but } \lim_{q\to\infty} \mathsf{Fish}(q\text{-LL}) = H_0/I_0.$$

(2) The results of (1) should be thought of as *lower bounds* on implementing compressed representations of q-PCSA and q-LL. We give a new sketch called Fishmonger whose space, at all times, is $O(\log^2 \log U) + (1 + o(1))(H_0/I_0)b \approx 1.98b$ bits and whose standard error, at all times, is $1/\sqrt{b}$, with probability $1 - 1/\mathrm{poly}(b)$.[7]

(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.

1.4 Related Work

As mentioned earlier, Scheuermann and Mauve [46] and Lang [38] explored entropy-compressed PCSA and LogLog sketches experimentally. Maximum Likelihood Estimators (MLE) for MinCount were studied by Chassaing and Gerin [13] and Clifford and Cosma [16]. Clifford and Cosma [16] and Ertl [25] studied the computational complexity of MLE in LogLog sketches. Lang [38] experimented with MLE-type estimators for 2-PCSA and 2-LogLog. Cohen, Katzir, and Yehezkel [20] looked at MLE estimators for estimating the cardinality of set intersections.

1.5 Organization

In Section 2 we review Shannon entropy, Fisher information, and the asymptotic efficiency of Maximum Likelihood Estimation.

In Section 3.2 we define a notion of *base-q scale-invariance* for a sketch, meaning its Shannon entropy and normalized Fisher information are invariant when changing the cardinality by multiples of q. Under this definition Shannon entropy and normalized Fisher information are *periodic* functions of $\log_q \lambda$. In Section 3.3 we define *average* entropy/information and show that the average behavior of any base-q scale-invariant sketch can be realized by a generic *smoothing* mechanism. Section 3.4 defines the Fish number of a

[6] Set m such that $b = \mathcal{I}(S) = m \cdot \mathcal{I}(S(0))$. The expected space required is $m \cdot \mathcal{H}(S(0)) = b(\mathcal{H}(S(0))/\mathcal{I}(S(0))) = b \cdot \mathsf{Fish}(S)$.

[7] This sketch was developed before we were aware of Lang's technical report [38]. 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 $\log U + (H_0/I_0 + \epsilon)b$ and the standard error at most $1/\sqrt{b}$, for some small constant $\epsilon > 0$ (see Section 3.3 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 $O(\log^2 \log U) + (1 + o(1))(H_0/I_0)b$ bits *at all times*, with high probability $1 - 1/\mathrm{poly}(b)$.

scale-invariant sketch in terms of average entropy and average information.

Section 4 analyzes the Fish numbers of base-q generalizations of PCSA and LogLog. Section 5 defines the class of *linearizable* sketches and proves that no such sketch has Fish-number smaller than H_0/I_0. We conclude and highlight some open problems in Section 6.

The Fishmonger sketch is described and analyzed in the full version [44]. In [44], we also survey non-commutative sketching. All missing proofs appear in the full version [44].

2 PRELIMINARIES

2.1 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

$$H(X_1 \mid R_1) = \mathbb{E}\left(-\log_2 f(X_1 \mid R_1)\right),$$

which is interpreted as the *average* entropy of X_1 after knowing R_1.

Theorem 1 (chain rule for entropy [22]). *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 [22] 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 [41, 51].

2.2 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* [12, 50].

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:[8]

$$I_X(\lambda_0) = \mathbb{E}\left(\frac{\frac{\partial}{\partial\lambda} f_\lambda(X)}{f_\lambda(X)}\right)^2 \Big|_{\lambda=\lambda_0}.$$

The *conditional Fisher information* of X_1 given X_0 at $\lambda = \lambda_0$ is defined as

$$I_{X_1|X_0}(\lambda_0) = \mathbb{E}\left(\frac{\frac{\partial}{\partial\lambda} f_\lambda(X_1 \mid X_0)}{f_\lambda(X_1 \mid X_0)}\right)^2 \Big|_{\lambda=\lambda_0}.$$

Similar to Shannon's entropy, we also have a chain rule for Fisher information numbers.

Theorem 2 (chain rule for Fisher information [53]). *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 $I_{\mathbf{X}}(\lambda) = m \cdot I_{X_0}(\lambda)$.*

The celebrated Cramér-Rao lower bound [12, 50] states that, under mild regularity conditions (see Section 2.3), for any unbiased estimator $\hat{\lambda}(\mathbf{X})$ with finite variance,

$$\mathrm{Var}(\hat{\lambda} \mid \lambda) \geq \frac{1}{I_{\mathbf{X}}(\lambda)}.$$

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.,

$$\lim_{m\to\infty} \sqrt{m}(\hat{\lambda} - \lambda) \sim \mathcal{N}\left(0, \frac{1}{I_{X_0}(\lambda)}\right),$$

or equivalently, $\hat{\lambda} \sim \mathcal{N}\left(\lambda, \frac{1}{I_{\mathbf{X}}(\lambda)}\right)$ 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 $\left(\lambda^2 \cdot I_{\mathbf{X}}(\lambda)\right)^{-1}$. We define the *normalized* Fisher information number of λ with respect to the observation $\mathbf{X}$ to be $\lambda^2 \cdot I_{\mathbf{X}}(\lambda)$.

2.3 Regularity Conditions and Poissonization

The asymptotic normality of MLE and the Cramér-Rao lower bound depend on various regularity conditions [1, 7, 53], 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.[9] 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 [24, 29]. Consider the following two processes.

Discrete counting process. Starting from time 0, an element is inserted at every time $k \in \mathbb{N}$.

[8] Since in this paper the parameter is always the cardinality, the parameter λ is omitted in the notation $I_X(\lambda_0)$.

[9] 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 [12].

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 $[0, \infty)$.

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 $1 - 1/\text{poly}(\lambda)$. Since we are concerned with the asymptotic efficiency of sketches, we are indifferent between these two models.[10]

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 state space is independent of λ, and f_λ will always be differentiable w.r.t. λ. Henceforth, we use the terms "time" and "cardinality" interchangeably.

3 SCALE-INVARIANCE AND Fish NUMBERS

We are destined to measure the efficiency of observations in terms of entropy (H) and normalized information ($\lambda^2 \times I$), 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). In Section 3.1 we switch from the functional view of sketches (as CIFFs) to a distributional interpretation, then in Section 3.2 define a weak notion of *scale-invariance* for sketches. In Section 3.3 we give a generic method to iron out periodic behavior in scale-invariant sketches, and in Section 3.4 we formally define the Fish number of a sketch.

3.1 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 1 (Induced Distribution Family). Let A be the name of a sketch having a countable state space $\mathcal{M}$. The *Induced Distribution Family (IDF)* of A is a parameterized distribution family

$$\Psi_A = \{\psi_{A,\lambda} : \mathcal{M} \to [0,1] \mid \lambda > 0\},$$

where $\psi_{A,\lambda}(x)$ is the probability of A being in state x at cardinality λ. Define $X_{A,\lambda} \sim \psi_{A,\lambda}$ to be a random state drawn from $\psi_{A,\lambda}$.

We can now directly characterize existing sketches as IDFs.[11] For example, the state-space of a single LogLog (2-LL) sketch [24][12] is $\mathcal{M} = \mathbb{N}$ and Ψ_{LL} contains, for each $\lambda > 0$, the function[13]

$$\psi_{\text{LL},\lambda}(k) = e^{-\frac{\lambda}{2^{k+1}}} - e^{-\frac{\lambda}{2^k}}.$$

We usually consider just the basic version of each sketch, e.g., a single bit-vector for PCSA or a single counter for LL. When we apply the machinery laid out in Section 2 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* [26, 29, 30], which, after Poissonization, yields the same sketch with cardinality $\lambda' = m\lambda$.

3.2 Weak Scale-Invariance

Consider a basic sketch A with IDF Ψ_A, and let A^m denote a vector of m independent A-sketches. From the Cramér-Rao lower bound we know the variance of an unbiased estimator is at least $\frac{1}{I_{A^m}(\lambda)} = \frac{1}{m \cdot I_A(\lambda)}$. (Here $I_{A^m}(\lambda)$ is short for $I_{X_{A^m,\lambda}}(\lambda)$, where $X_{A^m,\lambda}$ is the observed final state of A^m at time λ.) The memory required to store it is at least $H(X_{A^m,\lambda}) = m \cdot H(X_{A,\lambda})$. Thus the product of the memory and the *relative* variance is lower bounded by

$$\frac{H(X_{A,\lambda})}{\lambda^2 \cdot I_A(\lambda)},$$

which only depends on the distribution family Ψ_A and the unknown parameter λ. However, ideally it would depend only on Ψ_A.

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 2 (Weak Scale-Invariance). Let A be a sketch with induced distribution family Ψ_A and $q > 1$ be a real number. We say A is *weakly scale-invariant with base q* if for any $\lambda > 0$, we have

$$H(X_{A,\lambda}) = H(X_{A,q\lambda}) \quad \text{and} \quad I_A(\lambda) = q^2 \cdot I_A(q\lambda).$$

Remark 2. For example, the original (Hyper)LogLog and PCSA sketches [24, 29, 30] are, after Poissonization, base-2 weakly scale-invariant.

Observe that if a sketch A is weakly scale-invariant with base q, then the ratio

$$\frac{H(X_{A,q\lambda})}{(q\lambda)^2 \cdot I_A(q\lambda)} = \frac{H(X_{A,\lambda})}{\lambda^2 \cdot I_A(\lambda)}$$

becomes multiplicatively periodic with period q. See Figure 1 for illustrations of the periodicity of the entropy (H) and normalized information ($\lambda^2 I$) of the base-q LogLog sketch.

3.3 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 Figures 1 and 2. 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

[10] 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.

[11] It is still required that the sketches be effected by a CIFF family, but this does not influence how IDFs are defined.

[12] In any real implementation it would be truncated at some finite maximum value, typically 64.

[13] It would be $\psi_{\text{LL},\lambda}(k) = (1 - \frac{1}{2^{k+1}})^\lambda - (1 - \frac{1}{2^k})^\lambda$ without Poissonization.

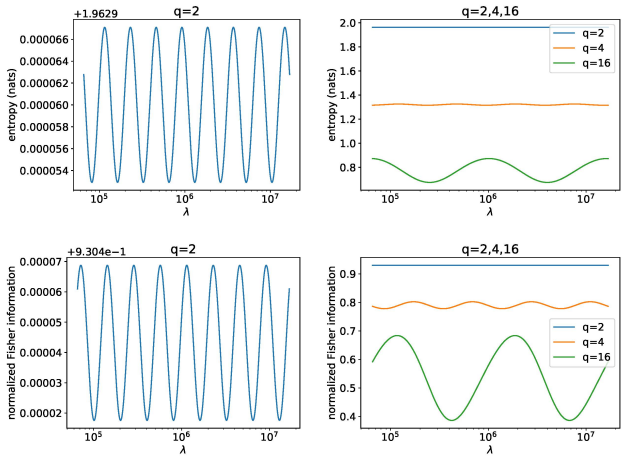

Figure 1: Entropy and normalized Fisher information number for q-LogLog sketches for $\lambda \in [2^{16}, 2^{24}]$. See Section 4.2 for the precise definitions. Left: At a sufficiently small scale, the oscillations in entropy (top) and normalized information (bottom) of 2-LL become visible. Right: At higher values of $q \in \{2, 4, 16\}$, the oscillations in entropy (top) and normalized information (bottom) of q-LL 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. [29]. This highlights how little room for improvement there is in the HyperLogLog analysis.

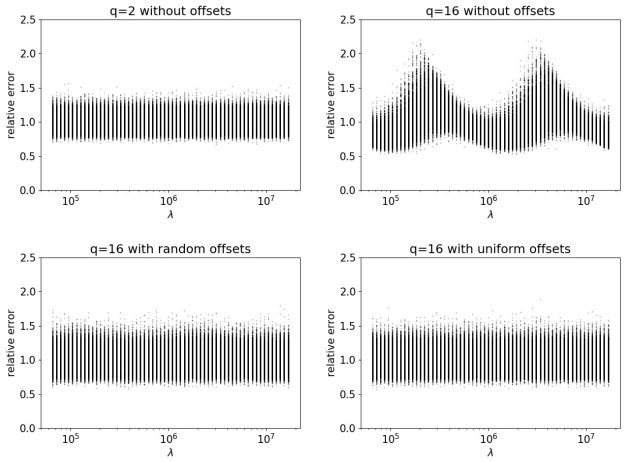

Figure 2: The empirical relative error ($\hat{\lambda}/\lambda$) distribution (for $\lambda \in [2^{16}, 2^{24}]$) of q-LogLog for four cases: (1) $q = 2$ without offsets; (2) $q = 16$ without offsets; (3) $q = 16$ with random offsets; (4) $q = 16$ with uniform offsets. All use $m = 128$ and the number 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.

values.[14] Figure 2 illustrates the effectiveness of this smoothing operation for reasonably small values of $q = 16$ and $m = 128$.

Throughout this section we let A be a weakly scale-invariant sketch with base q, having state-space $\mathcal{M}$, and IDF Ψ_A. Let $(R_1, Y_1) \in [0, 1) \times \mathcal{M}$ be a pair where R_1 is uniformly random in $[0, 1)$, and Y_1 is the state of A after seeing λq^{-R_1} distinct insertions.[15] Then

$$\Pr(Y_1 = y_1 \mid R_1 = r_1, \lambda) = \psi_{A,\lambda q^{-r_1}}(y_1).$$

Thus the joint density function is

$$f_\lambda(r_1, y_1) = \psi_{A,\lambda q^{-r_1}}(y_1).$$

Lemma 1. *Fix the unknown cardinality (parameter) λ. The Fisher information of λ with respect to (R_1, Y_1) is equal to*

$$\frac{1}{\lambda^2} \int_0^1 q^{2r} I_A(q^r)\, dr.$$

Lemma 2. *Fix the unknown cardinality (parameter) λ. The conditional entropy $H(Y_1 \mid R_1)$ is equal to*

$$\int_0^1 H(X_{A,q^r})\, dr.$$

[14] As m goes to infinity, using the set of uniform offsets $(0, \ldots, \frac{m-1}{m})$ will also work.
[15] 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.

In conclusion, with random offsetting we can transform any weakly scale-invariant sketch A so that the product of the memory and the relative variance is

$$\frac{\int_0^1 H(X_{A,q^r})\, dr}{\lambda^2 \cdot \frac{1}{\lambda^2} \int_0^1 q^{2r} I_A(q^r)\, dr} = \frac{\int_0^1 H(X_{A,q^r})\, dr}{\int_0^1 q^{2r} I_A(q^r)\, dr},$$

and hence independent of the cardinality λ.

3.4 The Fisher-Shannon Number of a Sketch

Let A_q be a weakly scale-invariant sketch with base q. The *Fisher-Shannon* (Fish) *number* of A_q captures the maximum performance we can potentially extract out of A_q, after applying random offsets (Section 3.3), optimal estimators (Section 2.2), and compression to the entropy bound (Section 2.1), as $m \to \infty$. In particular, any sketch composed of independent copies of A_q with standard error $\frac{1}{\sqrt{b}}$ must use at least $\mathsf{Fish}(A_q) \cdot b$ bits. Thus, smaller Fish-numbers are better.

Definition 3. Let A_q be a weakly scale-invariant sketch with base q. The Fish *number* of A_q is defined to be $\mathsf{Fish}(A_q) \stackrel{\text{def}}{=} \mathcal{H}(A_q)/\mathcal{I}(A_q)$, where

$$\mathcal{H}(A_q) \stackrel{\text{def}}{=} \int_0^1 H(X_{A_q,q^r})\, dr \quad \text{and} \quad \mathcal{I}(A_q) \stackrel{\text{def}}{=} \int_0^1 q^{2r} I_{A_q}(q^r)\, dr.$$

4 Fish NUMBERS OF PCSA AND LL

In this section, we will find the Fish numbers of generalizations of PCSA [30] and (Hyper)LogLog [24, 29]. The results are expressed in terms of two important constants, H_0 and I_0.

Definition 4. Let $h(x) = -x\ln x - (1-x)\ln(1-x)$ and $g(x) = \frac{x^2}{e^x-1}$. We define

$$H_0 \stackrel{\text{def}}{=} \frac{1}{\ln 2}\cdot\int_{-\infty}^{\infty} h\left(e^{-e^w}\right)\,dw \quad \text{and} \quad I_0 \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} g\left(e^w\right)\,dw.$$

Lemma 3 derives simplified expressions for H_0 and I_0. All missing proofs from this section can be found in the full version [44].

Lemma 3.

$$H_0 = \frac{1}{\ln 2} + \sum_{k=1}^{\infty}\frac{1}{k}\log_2(1+1/k) \quad \textit{and} \quad I_0 = \zeta(2) = \frac{\pi^2}{6},$$

where $\zeta(s) = \sum_{n=1}^{\infty}\frac{1}{n^s}$ is the Riemann zeta function.

4.1 The Fish Numbers of q-PCSA Sketches

In the discrete counting process, a natural base-q generalization of PCSA (q-PCSA) maintains 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,[16] where $\Pr(x_i = 1) = q^{-i}$, then update $\mathbf{b} \leftarrow \mathbf{b} \vee \mathbf{x}$, where $\vee$ is bit-wise OR.[17] 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 2.

Definition 5 (IDF of q-PCSA Sketches). For any base $q > 1$, the state space[18] of q-PCSA $\mathcal{M}_{\text{PCSA}} = \{0,1\}^{\mathbb{Z}}$ and the induced distribution for cardinality λ is

$$\psi_{q\text{-PCSA},\lambda}(\mathbf{b}) = \prod_{k=-\infty}^{\infty} e^{-\frac{\lambda(1-b_k)}{q^k}}\left(1-e^{-\frac{\lambda}{q^k}}\right)^{b_k}.$$

Theorem 3. *For any $q > 1$, q-PCSA is weakly scale-invariant with base q. Furthermore, we have*

$$\mathcal{H}(q\text{-PCSA}) = \frac{H_0}{\ln q} \quad \textit{and} \quad \mathcal{I}(q\text{-PCSA}) = \frac{I_0}{\ln q},$$

and hence $\mathsf{Fish}(q\text{-PCSA}) = \frac{H_0}{I_0} \approx 1.98016$.

The proof of Theroem 3 can be found in the full version [44].

[16] If we are interested in cardinalities $\ll U$, we would truncate the hash at $\log U$ bits.
[17] We can simplify this scheme with the same two levels of stochastic averaging used by Flajolet and Martin [30], 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.
[18] Strictly speaking the state-space is not countable. However, it suffices to consider only states with finite Hamming weight.

4.2 The Fish Numbers of q-LogLog Sketches

In a discrete counting process, the natural base-q generalization of the (Hyper)LogLog sketch (q-LL) works as follows. Let $Y = \min_{a\in\mathcal{A}} h(a) \in [0,1]$ be the minimum hash value seen. The q-LL sketch stores the integer $S = \left\lfloor -\log_q Y \right\rfloor$, so when the cardinality is λ,

$$\begin{aligned}\Pr(S = k) &= \Pr(q^{-k} \le Y < q^{-k+1}) = (1-q^{-k})^\lambda - (1-q^{-k+1})^\lambda \\ &\approx e^{-\lambda/q^k} - e^{-\lambda/q^{k-1}}.\end{aligned}$$

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 6 (IDF of q-LL sketches). For any base $q > 1$, the state space of q-LL is $\mathcal{M}_{\text{LL}} = \mathbb{Z}$ and the induced distribution for cardinality λ is

$$\psi_{q\text{-LL},\lambda}(k) = e^{-\lambda/q^k} - e^{-\lambda/q^{k-1}}.$$

In Lemma 5 we express the Fish number of q-LL in terms of two quantities $\phi(q)$ and $\rho(q)$, defined as follows.

Definition 7.

$$\phi(q) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} -(e^{-e^r} - e^{-e^r q})\log_2(e^{-e^r} - e^{-e^r q})\,dr.$$

$$\rho(q) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} \frac{(-e^r e^{-e^r} + e^r q e^{-e^r q})^2}{e^{-e^r} - e^{-e^r q}}\,dr.$$

Lemma 4 gives simplified expressions for $\phi(q)$ and $\rho(q)$.

Lemma 4. *Let $\zeta(s,t) = \sum_{k\ge 0}(k+t)^{-s}$ be the Hurwitz zeta function. Then ϕ and ρ can be expressed as:*

$$\phi(q) = \frac{1-1/q}{\ln 2} + \sum_{k=1}^{\infty}\frac{1}{k}\log_2\left(\frac{k+\frac{1}{q-1}+1}{k+\frac{1}{q-1}}\right).$$

$$\rho(q) = \zeta\left(2, \frac{q}{q-1}\right) = \sum_{k=0}^{\infty}\frac{1}{(k+\frac{q}{q-1})^2}.$$

The following lemma calculates the entropy and normalized information of q-LL.

Lemma 5. *For any $q > 1$, q-LL is weakly scale-invariant with base q. Furthermore, we have*

$$\mathcal{H}(q\text{-LL}) = \frac{\phi(q)}{\ln q} \quad \textit{and} \quad \mathcal{I}(q\text{-LL}) = \frac{\rho(q)}{\ln q}.$$

Theorem 4. *For any $q > 1$, the* Fish *number of q-LL is*

$$\mathsf{Fish}(q\text{-LL}) > \frac{H_0}{I_0}.$$

Furthermore, we have

$$\lim_{q\to\infty}\mathsf{Fish}(q\text{-LL}) = \frac{H_0}{I_0}.$$

Proof. We prove the second statement first. By Lemma 5, we have

$$\lim_{q\to\infty} \mathsf{Fish}(q\text{-LL}) = \lim_{q\to\infty} \frac{\mathcal{H}(q\text{-LL})}{\mathcal{I}(q\text{-LL})}$$
$$= \lim_{q\to\infty} \frac{\frac{1-1/q}{\ln 2} + \sum_{k=1}^{\infty} \frac{1}{k} \log_2\left(\frac{k+\frac{1}{q-1}+1}{k+\frac{1}{q-1}}\right)}{\sum_{k=1}^{\infty} \frac{1}{(k+\frac{1}{q-1})^2}}$$
$$= \frac{\frac{1}{\ln 2} + \sum_{k=1}^{\infty} \frac{1}{k} \log_2\left(\frac{k+1}{k}\right)}{\sum_{k=1}^{\infty} \frac{1}{k^2}} = \frac{H_0}{I_0}.$$

The first statement follows from Lemmas 6 and 7. Refer to the full version [44] for proof. □

Lemma 6. $\mathsf{Fish}(q\text{-LL})$ *is strictly decreasing for* $q \geq 1.4$.

Lemma 7. $\mathsf{Fish}(q\text{-LL}) > \mathsf{Fish}(2\text{-LL})$ *for* $q \in (1, 1.4]$.

5 A SHARP LOWER BOUND ON LINEARIZABLE SKETCHES

In Section 5.1 we introduce the *Dartboard* model, which is essentially the same as Ting's *area-cutting process* [49], with some minor differences.[19] In Section 5.2 we define the class of *Linearizable* sketches, and in Section 5.3 we prove that no Linearizable sketch has Fish-number strictly smaller than H_0/I_0.

5.1 The Dartboard Model

Define the *dartboard* to be a unit square $[0,1]^2$, partitioned into a set C of *cells* of various sizes. A *state space* is a set $\mathcal{S} \subseteq 2^C$. Each state $\sigma \in \mathcal{S}$ partitions the cells into *occupied* cells (σ) and *free* cells ($C\backslash\sigma$). We process a stream of elements from some multiset. When a *new* element arrives we throw a *dart* at the dartboard and update the state.[20] The probability that a cell $c_i \in C$ 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 8. 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 $h : [U] \to [U]$. The dartboard is partitioned into $[U]$ 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. 3.

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.

5.2 Linearizable Sketches

Informally, a sketch in the dartboard model is called *linearizable* if there is a fixed permutation of cells $(c_0, c_1, \ldots, c_{|C|-1})$ 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

$$Y_i = Z_i \vee \phi(\mathbf{Y}_{i-1}), \qquad \text{where } \mathbf{Y}_{i-1} = (Y_0, \ldots, Y_{i-1}).$$

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 9. All linearizable sketches are commutative (and hence mergeable).

Thus we have

$$\text{all sketches} \supseteq \text{dartboard sketches} \supseteq \text{commutative sketches} \supseteq \text{linearizable sketches}$$

All of these containments are *strict* (see Figure 4), but most popular commutative sketches are linearizable. For example, PCSA-type sketches [27, 30] 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 MinCount), 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 [27], Discrete MaxCount [49], and Curtain [45].

Strictly speaking AdaptiveSampling [28, 31] 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

[19] Ting's definition does not fix the state-space *a priori*, and in its full generality allows for non-deterministic sketching algorithms.
[20] 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.

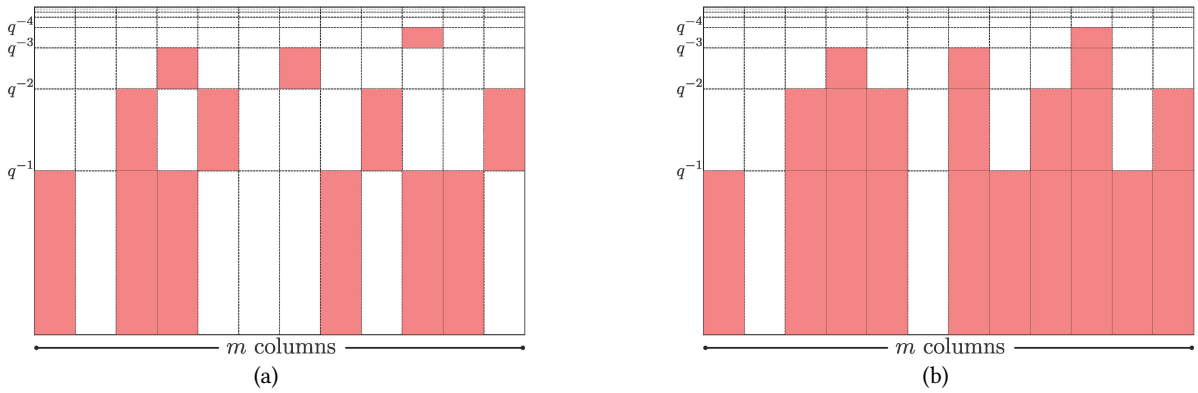

Figure 3: The cell partition used by q-PCSA and q-LL. (a) A possible state of PCSA. Occupied (red) cells are precisely those containing darts. (c) The corresponding state of LogLog. Occupied (red) cells contain a dart, or lie below a cell in the same column that contains a dart.

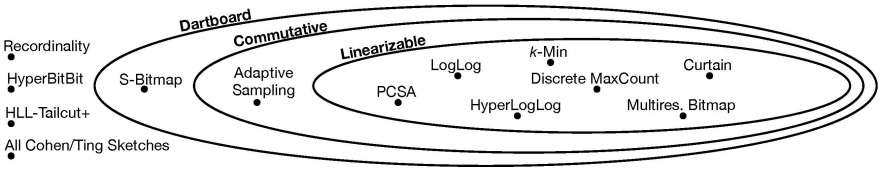

Figure 4: A classification of sketching algorithms for cardinality estimation.

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 [14].

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 [18, 49] 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. The other type are heuristic sketches that violate Rule (R3) (occupied cells stay occupied), like HyperBitBit [47] and HLL-Tailcut+ [52]. Rule (R3) is critical if the sketch is to be (asymptotically) unbiased; see [44].

5.3 The Lower Bound

When phrased in terms of the dartboard model, our analysis of the Fish-number of PCSA (Section 4) 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 $c_i \in C$ 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 10 are useful for describing these contributions.[21]

Lemma 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

$$\dot{H}(t) \stackrel{\text{def}}{=} \frac{1}{\ln 2}\left(te^{-t} - (1-e^{-t})\ln(1-e^{-t})\right),$$
$$\dot{I}(t) \stackrel{\text{def}}{=} \frac{t^2}{e^t - 1}.$$

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 $\dot{H}, \dot{I}$.

Still fixing $c_i \in C$ 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 $\phi(\mathrm{Y}_{i-1})$ 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 11.

$$\int_{-\infty}^{\infty} \dot{H}(e^x)dx = H_0 \quad \text{and} \quad \int_{-\infty}^{\infty} \dot{I}(e^x)dx = I_0.$$

In other words, if we let cell c_i "live" forever (fix $\phi(\mathrm{Y}_{i-1}) = 0$ 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 $t_i = p_i\lambda$?

Figure 5 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 12.

Lemma 12. *$\dot{H}(t)/\dot{I}(t)$ is decreasing in t on $(0, \infty)$.*

[21] See [44] for the missing proofs in this section.

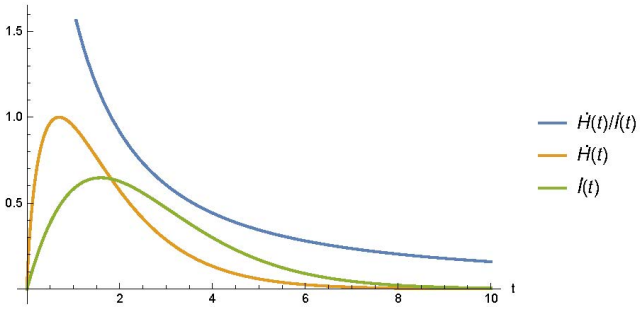

Figure 5: $\dot{H}(t), \dot{I}(t)$ **and** $\dot{H}(t)/\dot{I}(t)$

Lemma 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.[22] 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 $\mathbf{Y} = \mathbf{Y}_{|C|-1} = (Y_0, \ldots, Y_{|C|-1})$ be the vector of indicators encoding the state of the sketch and $\mathbf{Y}_{[\lambda]} = (Y_{0,\lambda}, \ldots, Y_{|C|-1,\lambda})$ refer to $\mathbf{Y}$ at time λ.

Proposition 1. *For any linearizable sketch and any $c_i \in C$, $\Pr(\phi(\mathbf{Y}_{i-1,\lambda}) = 0)$ is non-increasing with λ.*

Proof. Follows from Rule (R3) and the monotonicity of ϕ. □

The proof depends on *linearizability* mainly through Lemma 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 1, is non-increasing over time.

Lemma 13. *For any linearizable sketch and any $\lambda > 0$, we have*

$$H(\mathbf{Y}_{[\lambda]}) = \sum_{i=0}^{|C|-1} \dot{H}(p_i\lambda)\Pr(\phi(\mathbf{Y}_{i-1,\lambda}) = 0),$$
$$\lambda^2 \cdot I_{\mathbf{Y}}(\lambda) = \sum_{i=0}^{|C|-1} \dot{I}(p_i\lambda)\Pr(\phi(\mathbf{Y}_{i-1,\lambda}) = 0).$$

Definition 8 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 $\lambda \in [e^a, e^b]$.

[22] Clifford and Cosma [16] calculated the optimal Fisher information for Bernoulli observables when λ was known to lie in a small range.

Definition 8. Fix a linearizable sketch. Let $C \subset \mathcal{C}$ be a collection of cells and $W \subset \mathbb{R}$ be an interval of the reals. Define:

$$\mathbf{H}(C \to W) = \int_W \sum_{c_i \in C} \dot{H}(p_i e^x)\Pr(\phi(\mathbf{Y}_{i-1,e^x}) = 0)dx,$$
$$\mathbf{I}(C \to W) = \int_W \sum_{c_i \in C} \dot{I}(p_i e^x)\Pr(\phi(\mathbf{Y}_{i-1,e^x}) = 0)dx.$$

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 C 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 C, ϕ.

In Theorem 5 we assume that such a linearizable sketching scheme has produced C, ϕ 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 1) is that all scale-invariant linearizable sketches have Fish-number at least H_0/I_0.

Theorem 5. *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

$$\frac{\tilde{H}}{\tilde{I}} \geq \frac{H_0}{I_0} \frac{1 - \max(8d^{-1/4}, 5e^{-d/2})}{1 + \frac{(344+4\sqrt{d})}{d}\frac{H_0}{I_0}\left(1 - \max(8d^{-1/4}, 5e^{-d/2})\right)}$$
$$= (1 - o_d(1))\frac{H_0}{I_0}.$$

The expression for this $1 - o_d(1)$ factor arises from the following two technical lemmas.

Lemma 14. *For any $d > 0$ and $t \geq \frac{1}{2}\ln d$,*

$$\frac{\int_{-\infty}^{-t} \dot{H}(e^x)dx}{\int_{-\infty}^{-t+d} \dot{H}(e^x)dx} \leq \max(8d^{-1/4}, 5e^{-d/2}).$$

Lemma 15. *Let $d = b - a > 1$, $\Delta = \frac{1}{2}\ln d$ and $C^* = \{c_i \in C \mid p_i < e^{-a-\Delta}\}$. Assume that for all $\lambda > 0$, $H(\mathbf{Y}_{[\lambda]}) \leq \tilde{H}$ (the first condition of Theorem 5). Then we have*

$$\mathbf{I}(C \setminus C^* \to [a,b]) \leq (344 + 4e^{\Delta})\tilde{H}.$$

Corollary 1. *Let A_q be any linearizable, weakly scale-invariant sketch with base q. Then* $\mathsf{Fish}(A_q) \geq H_0/I_0$.

6 CONCLUSION

We introduced a natural metric (Fish) for sketches that consist of statistical observations of a data stream. It captures the tension between the encoding length of the observation (Shannon entropy) and its value for statistical estimation (Fisher information).

The constant $H_0/I_0 \approx 1.98016$ is fundamental to the Cardinality Estimation problem. It is the Fish-number of PCSA [30], and achievable up to a $(1 + o(1))$-factor with the Fishmonger sketch

([44]), i.e., roughly $(1+o(1))(H_0/I_0)m$ bits suffice to get standard error $1/\sqrt{m}$. These two facts were foreshadowed by Lang's [38] numerical and experimental investigations into compressed sketches and MLE-type estimators.

We defined a natural class of commutative (mergeable) sketches called *linearizable* sketches, and proved that no such sketch can beat H_0/I_0. The most well known sketches are linearizable, such as PCSA, (Hyper)LogLog, MinCount/k-Min/Bottom-k, and Multres. Bitmap.

We highlight two open problems.

- Shannon entropy and Fisher information are both subject to data processing inequalities, i.e., no deterministic transformation can increase entropy/information. Our lower bound (Section 5) can be thought of as a specialized data processing inequality for Fish, with two notable features. First, the deterministic transformation has to be of a certain type (the *linearizability* assumption). Second, we need to measure $\mathcal{H}/\mathcal{I}$ over a sufficiently long period of *time*. The second feature is essential to the H_0/I_0 lower bound. The open question is whether the first feature can be relaxed. We conjecture that H_0/I_0 is a lower bound on *all commutative/mergeable sketches*.[23]
- Our lower bound provides some evidence that Fishmonger is optimal up to a $(1+o(1))$-factor among commutative/mergeable sketches. However, it is not particularly fast nor elegant, and must be decompressed/recompressed between updates. This can be mitigated in practice, e.g., by storing the first column containing a 0-bit[24] or buffering insertions and only decompressing when the buffer is full. The CPC sketch in Apache DataSketches uses the latter strategy [38, 48]. Is it possible to design a *conceptually simple* mergeable sketch (i.e., without resorting to entropy compression) that can be updated in $O(1)$ *worst-case time* and occupies space $(H_0/I_0+c)m$ (with standard error $1/\sqrt{m}$) for some *reasonably small* $c>0$?

ACKNOWLEDGMENTS

We thank Liran Katzir for suggesting references [16, 25, 26] and an anonymous reviewer for bringing the work of Lang [38] and Scheuermann and Mauve [46] to our attention. The first author would like to thank Bob Sedgewick and Jérémie Lumbroso for discussing the cardinality estimation problem at Dagstuhl 19051. This work was supported by NSF grants CCF-1637546 and CCF-1815316.

REFERENCES

[1] Felix Abramovich and Ya'acov Ritov. 2013. *Statistical theory: a concise introduction.* CRC Press.
[2] Pankaj K. Agarwal, Graham Cormode, Zengfeng Huang, Jeff M. Phillips, Zhewei Wei, and Ke Yi. 2013. Mergeable summaries. *ACM Trans. Database Syst.* 38, 4 (2013), 26:1–26:28. https://doi.org/10.1145/2500128
[3] Noga Alon, Yossi Matias, and Mario Szegedy. 1999. The Space Complexity of Approximating the Frequency Moments. *J. Comput. Syst. Sci.* 58, 1 (1999), 137–147. https://doi.org/10.1006/jcss.1997.1545
[4] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, D. Sivakumar, and Luca Trevisan. 2002. Counting Distinct Elements in a Data Stream. In *Proceedings 6th International Workshop on Randomization and Approximation Techniques (RANDOM) (Lecture Notes in Computer Science, Vol. 2483)*. 1–10. https://doi.org/10.1007/3-540-45726-7_1
[5] Ziv Bar-Yossef, Ravi Kumar, and D. Sivakumar. 2002. Reductions in streaming algorithms, with an application to counting triangles in graphs. In *Proceedings 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 623–632.
[6] Kevin S. Beyer, Rainer Gemulla, Peter J. Haas, Berthold Reinwald, and Yannis Sismanis. 2009. Distinct-value synopses for multiset operations. *Commun. ACM* 52, 10 (2009), 87–95. https://doi.org/10.1145/1562764.1562787
[7] P Bickel and K Doksum. 2001. Mathematical statistics: Basic ideas and selected topics. 2d. ed. vol. 1 prentice hall. *Upper Saddle River, NJ* (2001).
[8] Daniel K. Blandford and Guy E. Blelloch. 2008. Compact dictionaries for variable-length keys and data with applications. *ACM Trans. Algorithms* 4, 2 (2008), 17:1–17:25. https://doi.org/10.1145/1361192.1361194
[9] Jarosław Błasiok. 2020. Optimal Streaming and Tracking Distinct Elements with High Probability. *ACM Trans. Algorithms* 16, 1 (2020), 3:1–3:28. https://doi.org/10.1145/3309193
[10] Andrei Z. Broder. 1997. On the resemblance and containment of documents. In *Proceedings of Compression and Complexity of SEQUENCES*. 21–29. https://doi.org/10.1109/SEQUEN.1997.666900
[11] Joshua Brody and Amit Chakrabarti. 2009. A Multi-Round Communication Lower Bound for Gap Hamming and Some Consequences. In *Proceedings 24th Annual IEEE Conference on Computational Complexity (CCC)*. 358–368. https://doi.org/10.1109/CCC.2009.31
[12] G. Casella and R. L. Berger. 2002. *Statistical Inference, 2nd Ed.* Brooks/Cole, Belmont, CA.
[13] Philippe Chassaing and Lucas Gerin. 2006. 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. *Discrete Mathematics & Theoretical Computer Science*.
[14] Aiyou Chen, Jin Cao, Larry Shepp, and Tuan Nguyen. 2011. Distinct Counting With a Self-Learning Bitmap. *J. Amer. Statist. Assoc.* 106, 495 (2011), 879–890. https://doi.org/10.1198/jasa.2011.ap10217
[15] Tobias Christiani, Rasmus Pagh, and Mikkel Thorup. 2015. From Independence to Expansion and Back Again. In *Proceedings 47th Annual ACM Symposium on Theory of Computing (STOC)*. 813–820. https://doi.org/10.1145/2746539.2746620
[16] Peter Clifford and Ioana A. Cosma. 2012. A Statistical Analysis of Probabilistic Counting Algorithms. *Scandinavian Journal of Statistics* 39, 1 (2012), 1–14. https://doi.org/10.1111/j.1467-9469.2010.00727.x
[17] Edith Cohen. 1997. Size-Estimation Framework with Applications to Transitive Closure and Reachability. *J. Comput. Syst. Sci.* 55, 3 (1997), 441–453. https://doi.org/10.1006/jcss.1997.1534
[18] Edith Cohen. 2015. All-Distances Sketches, Revisited: HIP Estimators for Massive Graphs Analysis. *IEEE Trans. Knowl. Data Eng.* 27, 9 (2015), 2320–2334. https://doi.org/10.1109/TKDE.2015.2411606
[19] Edith Cohen and Haim Kaplan. 2008. Tighter estimation using bottom k sketches. *Proc. VLDB Endow.* 1, 1 (2008), 213–224. https://doi.org/10.14778/1453856.1453884
[20] Reuven Cohen, Liran Katzir, and Aviv Yehezkel. 2017. 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)*. 95–103. https://doi.org/10.1145/3097983.3097999
[21] Jeffrey Considine, Feifei Li, George Kollios, and John W. Byers. 2004. Approximate Aggregation Techniques for Sensor Databases. In *Proceedings of the 20th International Conference on Data Engineering (ICDE)*. 449–460. https://doi.org/10.1109/ICDE.2004.1320018
[22] Thomas M. Cover and Joy A. Thomas. 2006. *Elements of Information Theory (Second Edition)*. Wiley.
[23] Marianne Durand. 2004. *Combinatoire analytique et algorithmique des ensembles de données. (Multivariate holonomy, applications in combinatorics, and analysis of algorithms)*. Ph.D. Dissertation. Ecole Polytechnique X.
[24] Marianne Durand and Philippe Flajolet. 2003. Loglog Counting of Large Cardinalities. In *Proceedings 11th Annual European Symposium on Algorithms (ESA) (Lecture Notes in Computer Science, Vol. 2832)*. Springer, 605–617. https://doi.org/10.1007/978-3-540-39658-1_55
[25] Otmar Ertl. 2017. New Cardinality Estimation Methods for HyperLogLog Sketches. *CoRR* abs/1706.07290 (2017). arXiv:1706.07290 http://arxiv.org/abs/1706.07290
[26] Otmar Ertl. 2018. BagMinHash - Minwise Hashing Algorithm for Weighted Sets. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. 1368–1377. https://doi.org/10.1145/3219819.3220089
[27] Cristian Estan, George Varghese, and Michael E. Fisk. 2006. Bitmap algorithms for counting active flows on high-speed links. *IEEE/ACM Trans. Netw.* 14, 5 (2006), 925–937. https://doi.org/10.1145/1217709
[28] Philippe Flajolet. 1990. On adaptive sampling. *Computing* 43, 4 (1990), 391–400. https://doi.org/10.1007/BF02241657

[23] In particular, one would need to consider monotone "forced occupation" functions $\phi(\mathbf{Y}_{-i}) \in \{0,1\}$ that depend on $\mathbf{Y}_{-i}$, i.e., all cells except for c_i.
[24] (allowing us to summarily ignore the vast majority of elements without decompression)

[29] Philippe Flajolet, Éric Fusy, Olivier Gandouet, and Frédéric Meunier. 2007. HyperLogLog: the analysis of a near-optimal cardinality estimation algorithm, In Proceedings of the 18th International Meeting on Probabilistic, Combinatorial, and Asymptotic Methods for the Analysis of Algorithms (AofA). *Discrete Mathematics & Theoretical Computer Science*, 127–146.

[30] Philippe Flajolet and G. Nigel Martin. 1985. Probabilistic Counting Algorithms for Data Base Applications. *J. Comput. Syst. Sci.* 31, 2 (1985), 182–209. https://doi.org/10.1016/0022-0000(85)90041-8

[31] Phillip B. Gibbons and Srikanta Tirthapura. 2001. Estimating simple functions on the union of data streams. In *Proceedings 13th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*. 281–291. https://doi.org/10.1145/378580.378687

[32] Frédéric Giroire. 2009. Order statistics and estimating cardinalities of massive data sets. *Discret. Appl. Math.* 157, 2 (2009), 406–427. https://doi.org/10.1016/j.dam.2008.06.020

[33] Ahmed Helmi, Jérémie Lumbroso, Conrado Martínez, and Alfredo Viola. 2012. 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). *Discrete Mathematics & Theoretical Computer Science*, 323–338.

[34] Stefan Heule, Marc Nunkesser, and Alexander Hall. 2013. HyperLogLog in practice: algorithmic engineering of a state of the art cardinality estimation algorithm. In *Proceedings 16th International Conference on Extending Database Technology (EDBT)*. 683–692. https://doi.org/10.1145/2452376.2452456

[35] Piotr Indyk and David P. Woodruff. 2003. Tight Lower Bounds for the Distinct Elements Problem. In *Proceedings 44th IEEE Symposium on Foundations of Computer Science (FOCS), October 2003, Cambridge, MA, USA, Proceedings*. 283–288. https://doi.org/10.1109/SFCS.2003.1238202

[36] T. S. Jayram and David P. Woodruff. 2013. Optimal Bounds for Johnson-Lindenstrauss Transforms and Streaming Problems with Subconstant Error. *ACM Trans. Algorithms* 9, 3 (2013), 26:1–26:17. https://doi.org/10.1145/2483699.2483706

[37] Daniel M. Kane, Jelani Nelson, and David P. Woodruff. 2010. An optimal algorithm for the distinct elements problem. In *Proceedings 29th ACM Symposium on Principles of Database Systems (PODS)*. 41–52. https://doi.org/10.1145/1807085.1807094

[38] Kevin J. Lang. 2017. Back to the Future: an Even More Nearly Optimal Cardinality Estimation Algorithm. *CoRR* abs/1708.06839 (2017). arXiv:1708.06839

[39] Aleksander Łukasiewicz and Przemysław Uznański. 2020. Cardinality estimation using Gumbel distribution. *CoRR* abs/2008.07590 (2020). arXiv:2008.07590 https://arxiv.org/abs/2008.07590

[40] Jérémie Lumbroso. 2010. 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)*. 489–504.

[41] Alistair Moffat, Radford M. Neal, and Ian H. Witten. 1998. Arithmetic Coding Revisited. *ACM Trans. Inf. Syst.* 16, 3 (1998), 256–294. https://doi.org/10.1145/290159.290162

[42] Moni Naor and Vanessa Teague. 2001. Anti-presistence: history independent data structures. In *Proceedings 33rd Annual ACM Symposium on Theory of Computing (STOC)*. 492–501. https://doi.org/10.1145/380752.380844

[43] Suman Nath, Phillip B. Gibbons, Srinivasan Seshan, and Zachary R. Anderson. 2008. Synopsis diffusion for robust aggregation in sensor networks. *ACM Trans. Sens. Networks* 4, 2 (2008), 7:1–7:40. https://doi.org/10.1145/1340771.1340773

[44] Seth Pettie and Dingyu Wang. 2020. Information theoretic limits of cardinality estimation: Fisher meets Shannon. *arXiv preprint arXiv:2007.08051* (2020).

[45] Seth Pettie, Dingyu Wang, and Longhui Yin. 2020. Simple and Efficient Cardinality Estimation in Data Streams. *CoRR* abs/2008.08739 (2020). arXiv:2008.08739 https://arxiv.org/abs/2008.08739

[46] Björn Scheuermann and Martin Mauve. 2007. Near-Optimal Compression of Probabilistic Counting Sketches for Networking Applications. In *Proceedings of the 4th International Workshop on Foundations of Mobile Computing (DIALM-POMC)*.

[47] Robert Sedgewick. [n.d.]. Cardinality Estimation. ([n. d.]). https://www.cs.princeton.edu/~rs/talks/Cardinality.pdf Presentation delivered at AofA (2016), Knuth-80 (2018), and Dagstuhl 19051 (2019). https://www.cs.princeton.edu/~rs/talks/Cardinality.pdf.

[48] The Apache Foundation. 2019. Apache DataSketches: A software library of stochastic streaming algorithms. https://datasketches.apache.org/. (2019). https://datasketches.apache.org/

[49] Daniel Ting. 2014. Streamed approximate counting of distinct elements: beating optimal batch methods. In *Proceedings 20th ACM Conference on Knowledge Discovery and Data Mining (KDD)*. 442–451. https://doi.org/10.1145/2623330.2623669

[50] A. W. van der Vaart. 1998. *Asymptotic Statistics*. Cambridge University Press. https://doi.org/10.1017/CBO9780511802256

[51] Ian H. Witten, Radford M. Neal, and John G. Cleary. 1987. Arithmetic Coding for Data Compression. *Commun. ACM* 30, 6 (1987), 520–540. https://doi.org/10.1145/214762.214771

[52] Qingjun Xiao, Shigang Chen, You Zhou, and Junzhou Luo. 2020. Estimating Cardinality for Arbitrarily Large Data Stream With Improved Memory Efficiency. *IEEE/ACM Trans. Netw.* 28, 2 (2020), 433–446. https://doi.org/10.1109/TNET.2020.2970860

[53] Pablo Zegers. 2015. Fisher information properties. *Entropy* 17, 7 (2015), 4918–4939.