# HyperLogLog Hyperextended: Sketches for Concave Sublinear Frequency Statistics

Edith Cohen
Google Research, USA
Tel Aviv University, Israel
edith@cohenwang.com

## ABSTRACT

One of the most common statistics computed over data elements is the number of distinct keys. A thread of research pioneered by Flajolet and Martin three decades ago culminated in the design of optimal approximate counting sketches, which have size that is double logarithmic in the number of distinct keys and provide estimates with a small relative error. Moreover, the sketches are composable, and thus suitable for streamed, parallel, or distributed computation.

We consider here all statistics of the frequency distribution of keys, where a contribution of a key to the aggregate is concave and grows (sub)linearly with its frequency. These fundamental aggregations are very common in text, graphs, and logs analysis and include logarithms, low frequency moments, and cap statistics.

We design composable sketches of double-logarithmic size for all concave sublinear statistics. Our design combines theoretical optimality and practical simplicity. In a nutshell, we specify tailored mapping functions of data elements to output elements so that our target statistics on the data elements is approximated by the (max-) distinct statistics of the output elements, which can be approximated using off-the-shelf sketches. Our key insight is relating these target statistics to the *complement Laplace* transform of the input frequencies.

## 1 INTRODUCTION

We consider data presented as elements $e = (e.key, e.value)$ where each element has a *key* and a positive numeric *value* $> 0$. This data model is very common in streaming or distributed aggregation problems. A well-studied special case is where $e.value \equiv 1$ for all elements.

One of the most fundamental statistics over such data is the number of distinct keys: $\text{Distinct}(E) = |\{e.key \mid e \in E\}|$. Exact computation of the statistics requires maintaining a structure of size that is linear in the number of distinct keys. A pioneering design of Flajolet and Martin [14] showed that an approximate count can be obtained in a streaming model using structures ("sketches") of logarithmic size. Since then, a rich research strand proposed and analysed a diverse set of approximate counting sketches and deployed them for a wide range of applications [17].


*KDD'17, August 13–17, 2017, Halifax, NS, Canada.*
 978-1-4503-4887-4/17/08.
DOI: http://dx.doi.org/10.1145/3097983.3098020

Distinct counting sketches can be mostly classified as based on sampling (MinHash sketches) or on random projections (linear sketches). Both types of structures are mergeable/composable: This means that when the elements are partitioned, we can compute a sketch for each part separately and then obtain a corresponding sketch for the union from the sketches of each part. This property is critical for making the sketches suitable for parallel or distributed aggregation.

The original design of [14] and the leading ones used in practice use sample-based sketches. In particular, the popular Hyperloglog sketch [13] has double logarithmic size $O(\epsilon^{-2} + \log\log n)$, where $n$ is the number of distinct keys and $\epsilon$ is the target normalized root mean squared error (NRMSE). Since this size is necessary to represent the approximate count, Hyperloglog is asymptotically optimal. We note that the Hyperloglog sketch contains $\epsilon^{-2}$ registers which store exponents of the estimated count. Thus, explicit representation of the sketch has size $O(\epsilon^{-2}\log\log n)$, but one can theoretically use instead a single exponent and $\epsilon^{-2}$ constant-size offsets (e.g. [4, 20]) to bring the sketch size down to $O(\epsilon^{-2} + \log\log n)$, albeit by somewhat increasing updates complexity. Another point is that Hyperloglog uses random hash functions which have logarithmic-size representations. If we considers the hash representation to be part of the sketch [1, 20]), we get a logarithmic lower bound on sketch size. Here we follow [13, 14] 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.

We now consider other common statistics over elements. In particular, statistics expressed over a set of (key, weight) pairs, where the weight $w_x$ of a key $x$, is defined to be the sum of the values of data elements with key $x$:

$$w_x = \sum_{e \mid e.key = x} e.value\ .$$

Keys that are not active (no elements with this key) are defined to have $w_x = 0$. Note that if all elements have value equal to 1, then $w_x$ is the number of occurrences of key $x$. For a nonnegative function $f(w) \geq 0$ such that $f(0) \equiv 0$, we define the *$f$-statistics* of the data as $\sum_x f(w_x)$. We will find it convenient to work with the notation $W(w)$ for the number of keys with $w_x = w$. Equivalently, we can treat $W$ as a distribution over weights $w_x$ that is scaled by the number of distinct keys. We can then express the $f$-statistics (with a slight notation abuse) as

$$f(W) = \int_0^\infty W(w) f(w) dw\ . \tag{1}$$

The study of sketches that approximate $f$-statistics over streams of elements was formalized and popularized in a seminal paper by Alon, Matias, and Szegedy [1]. The aim is to the fundamental

tradeoff between sketch size and approximating quality for different statistics.

We focus here on functions $f$ that are concave with (sub)linear nonnegative growth-rate. Equivalently, these functions are the nonnegative *span* $\overline{\mathsf{cap}}$ (all nonnegative linear combinations) of *cap* functions

$$\mathsf{cap}_T(w) \equiv \min\{T, w\} \text{ , parameterized by } T > 0 \ .$$

Notable members of $\overline{\mathsf{cap}}$ that parametrize a spectrum between distinct count ($f(w) = 1$) and sum ($f(w) = w$) include *frequency moments* $f(w) = w^p$ in the range $p = [0, 1]$ (sum is $p = 1$ and distinct count is $p = 0$), cap functions (sum is realized by $\mathsf{cap}_\infty$ and distinct count by $\mathsf{cap}_1$ when element values are integral and by $f(w) = \mathsf{cap}_T(w)/T$ as $T \to 0$ generally), and *softcap* functions

$$\widetilde{\mathsf{cap}}_T(w) = T(1 - \exp(-w/T)) \ . \tag{2}$$

Softcap is a smooth approximation of cap: For $w \ll T$ we have $\widetilde{\mathsf{cap}}_T(w) \approx w$, for $w \gg T$ we have $\widetilde{\mathsf{cap}}_T(w) \approx T$, and for all $T, w$:

$$\forall w,\ (1 - 1/e)\mathsf{cap}_T(w) \le \widetilde{\mathsf{cap}}_T(w) \le \mathsf{cap}_T(w) \ . \tag{3}$$

Other important $\overline{\mathsf{cap}}$ members are $\log(1 + w)$ and capped moments.

Statistics in $\overline{\mathsf{cap}}$ are used in applications to decrease the impact of very frequent keys and increase the impact of rare keys. It is a common practice to weigh frequencies, say degree of nodes in a graph [24] or frequency of a term in a corpus [22, 30], by a sublinear function such as $w^p$ for $p \in (0, 1)$ or $\log(1 + w)$. In many applications, the ability to approximate the statistics over the raw data, without the cost of aggregation, can be very useful. One example is online advertising [16, 26], where data elements are opportunities to show ads to users (keys) that are interacting with various apps on different platforms. An advertisement campaign specifies a maximum number of times $T$ an ad can be displayed to the same user, so the number of qualifying impressions corresponds to $\mathsf{cap}_T$ statistics of the data. Statistics are computed over past data in order to estimate the number of qualifying impressions when designing a campaign. Another example is the computation of word embeddings, where each word has a focus and context embedding so that (a function) of the inner product captures the respective co-occurrence frequencies. Glove [27] demonstrated a significant benefit when weighting cooccurrences by $f(w) = \min\{1, w/T\}^\alpha$ for $\alpha < 1$ (instead of $f(w) = w$). Typically, the text corpus is presented as complete text documents, and elements (focus-context pairs) in arbitrary order are extracted in a distributed pass.

There is a very large body of work on the topic of approximating statistics over streamed or distributed data and it is not possible to mention it all here. Most of the prior work uses linear sketches (random linear projections). A sketch for the second moment, inspired by the JL transform [19], was presented by [1]. Indyk [18] followed with a beautiful construction based on stable distributions of sketches of size $O(\epsilon^{-2}\log^2 n)$ for moments in $p \in [0, 2]$. Braverman and Ostrovsky [2] presented an umbrella construction of polylogarithmic-size sketch structures, based on $L_2$ heavy hitter sketches, for all monotone $f$-statistics that are sketchable in polylogarithmic size. The construction is illuminating but not practical (high degree of the polylog and constant factors).

Sample-based sketches for $\overline{\mathsf{cap}}$ functions were presented by the author [6]. The sketch is a weighted sample of keys that supports approximate cap-statistics on domain queries (subsets of the keys). The framework generalizes both distinct reservoir sampling [21, 31] and the sample and hold stream sampling [7, 12, 15]. The size and quality tradeoffs of the sample are very close (within a small constant) to those of an optimal sample that can be efficiently computed over aggregated data (set of key and weight pairs). Roughly, a sample of $O(\epsilon^{-2})$ keys suffices to approximate $\mathsf{cap}_T(W)$ unbiasedly with coefficient of variation (CV) $\epsilon$. Moreover, a *multi-objective* (universal) sample (see [5, 10]) of $O(\epsilon^{-2}\log n)$ keys can approximate with CV $\epsilon$ *any* $f$-statistics for $f \in \overline{\mathsf{cap}}$. When this method is applied to sketching statistics of the full data, we can hash key identifiers to size $O(\log n)$ (to obtain uniqueness with very high probability) and obtain sketches of size $O(\epsilon^{-2}\log n)$ and a multi-objective sketches of size $O(\epsilon^{-2}\log^2 n)$. One weakness of the design is that these sketches are not fully composable: They apply on streamed elements (single pass) or take two passes over distributed data elements.

The remaining fundamental challenge was to design composable sketches of size $O(\epsilon^{-2}\log n)$ for each $\overline{\mathsf{cap}}$ statistics and a composable multi-objective sketch of size $O(\epsilon^{-2}\log^2 n)$. Given the practical significance of the problem, we seek simple and highly efficient designs. A further theoretical challenge is to design sketches that meet or approach the double-logarithmic representation-size lower bound of $O(\epsilon^{-2} + \log\log n)$.

## Contributions overview and organization

We address these challenges and make the following contributions. We show that *any* statistics in the *softcap span* $\overline{\widetilde{\mathsf{cap}}}$ can be approximated with the essential effectiveness and estimation quality of Hyperloglog. That is, we present composable sketches of size $O(\epsilon^{-2} + \log\log n)$ and estimators with NRMSE $\epsilon$ and good concentration. The softcap span $\overline{\widetilde{\mathsf{cap}}} \subset \overline{\mathsf{cap}}$ is the set of functions that can be expressed as

$$f(w) = \int_0^\infty a(t)(1 - e^{-wt})dt \ , \text{ where } a(t) \ge 0 \ . \tag{4}$$

The span includes all softcap functions, low frequency moments ($f(w) = w^p$ with $p \in (0, 1)$), and $\log(1 + w)$. We also present a composable *multi-objective* sketch for $\overline{\widetilde{\mathsf{cap}}}$. This is a single structure that is larger by a logarithmic factor than a single distinct counter and supports the approximations of *all* $\overline{\widetilde{\mathsf{cap}}}$ statistics. Finally, we consider statistics in $\overline{\mathsf{cap}}$ that are not in $\overline{\widetilde{\mathsf{cap}}}$ and show how to approximate them within small relative errors (12%) using differences of approximate $\overline{\widetilde{\mathsf{cap}}}$ statistics.

Our main component is a framework, illustrated in Figure 1, that reduces the sketching of the target statistics to sketching distinct statistics. We specify randomized functions $M$ that map data elements of the form $e = (e.key, e.value)$ to sets of *output elements*. Each output element $e' \in M(e)$ contains an *output key* (*outkey*) $e'.key$ (which generally is from a different domain than the input keys) and an optional value $e'.value \ge 0$. For a multiset of data elements $W$, we obtain a corresponding multiset of output elements

$$E = M(W) = \bigcup_{e \in W} M(e) \ .$$

The mapping functions are crafted so that the approximate statistics of the set of data elements $W$ can be obtained from approximations

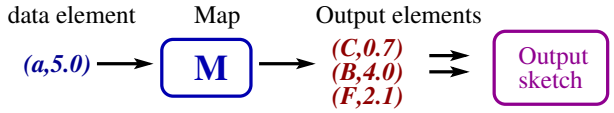


**Figure 1: Element processing framework**

of other statistics of the output elements $E$. In particular, if we have a composable sketch for the output statistics, we obtain a composable sketch for the target statistics. Note that our mapping functions are randomized, therefore the set $E$ is a random variable and so is any (exact) statistics on $E$. We refer to the value of the output statistics on the output elements as a *measurement* of $W$. When we sketch the output elements, we refer to the resulting estimate as an *approximate measurement* of $W$.

The output statistics we use are the distinct count $\mathrm{Distinct}(E)$, which allows us to leverage approximate distinct counters as black boxes, and the more general *max-distinct* statistics $\mathrm{MxDistinct}(E)$, defined as the sum over distinct keys of the maximum value of an element with the key:

$$\mathrm{MxDistinct}(E) = \sum_x m_x \,, \qquad (5)$$
$$\text{where } m_x \equiv \max_{e \in E \mid e.key = x} e.value \,,$$

which also can be sketched in double logarithmic size. Note that when all elements have value 1, $\mathrm{MxDistinct}(E) = \mathrm{Distinct}(E)$.

For multi-objective approximations we use *all-threshold* sketches that allows us to recover, for any threshold $t > 0$, an approximation of

$$\mathrm{TDistinct}_t(E) = \mathrm{Distinct}\{e \in E \mid e.value \le t\}, \qquad (6)$$

which is the number of distinct keys that appear in at least one element $e \in E$ with value $e.value \le t$. The size of the all-threshold sketch is larger by only a logarithmic factor than the basic distinct count sketch.

The paper is organized as follows. In Section 2 we define the *complement Laplace transform* $\mathcal{L}^c[W](t)$ of the frequency distribution $W$, which is its distinct count minus its Laplace transform at $t$. We have the relation

$$T\,\mathcal{L}^c[W](1/T) = \widetilde{\mathrm{cap}}_T(W) \,, \qquad (7)$$

that is, the transform at $1/T$ multiplied by $T$ is the $\widetilde{\mathrm{cap}}_T$ statistics of the data. In Section 3 we define a mapping function for any $t > 0$, so that $\mathcal{L}^c[W](t)$, and hence $\widetilde{\mathrm{cap}}_{1/t}$-statistics, is approximated by the respective Distinct measurement. We refer to this as a measurement of $\mathcal{L}^c[W]$ at point $t$.

In Section 4 we consider the span $\widetilde{\widetilde{\mathrm{cap}}}$ of softcap statistics, that is, all $f$ of the form (4). Equivalently, $a(t)$ is the inverse $\mathcal{L}^c$ transform of $f$. We derive the explicit form of the inverse transform of all frequency moments with $p \in (0,1)$ and logarithms. The statistics $f(W)$ for $f \in \widetilde{\widetilde{\mathrm{cap}}}$ can thus be expressed as

$$f(W) = \mathcal{L}^c[W][a] \equiv \int_0^\infty a(t)\,\mathcal{L}^c[W](t)dt \,.$$

This suggests that we can approximate $f(W)$ using multiple approximate point (Distinct) measurements. In section 5 we show that a single MxDistinct measurement suffices: We present element mapping functions (tailored to $f$) such that the MxDistinct statistics on output elements approximates $f(W) = \mathcal{L}^c[W][a]$. We refer to this statistics as a *combination* measurement of $\mathcal{L}^c[W]$ using $a$. A MxDistinct sketch of the output element gives us an approximation of combination measurement which approximates $f(W)$. Finally, we will review the design of HyperLoglog-like MxDistinct sketches.

In Section 6 we consider the multi-objective setting, that is, a single sketch from which we can approximate all $\widetilde{\widetilde{\mathrm{cap}}}$ statistics. We define a mapping function such that for all $t > 0$, $\mathrm{TDistinct}_t(E)$ is equivalent to a point measurement of $\mathcal{L}^c[W]$ at $t$. The output elements are processed by an *all-threshold distinct count* sketches, which can be interpreted as all-distance sketches [3, 4] and inherit their properties – In particular, the total structure size has logarithmic overhead over a single distinct counter. The all-threshold sketch allows us to obtain approximate point measurements for any $t$ and combination measurement for any $a$.

In Section 7 we consider statistics in $\overline{\mathrm{cap}}$ that may not be in $\widetilde{\widetilde{\mathrm{cap}}}$. We characterize $\overline{\mathrm{cap}}$ as the set of all concave sublinear functions and derive expressions for the *cap transform* which transforms $f \in \overline{\mathrm{cap}}$ to the coefficients of the corresponding nonnegative linear combination of cap functions. We then consider sketching these statistics $f(W)$ using approximate *signed* inverse $\mathcal{L}^c$ transform of the function $f$. We use separate combinations measurements of the positive and negative components for the approximation. We show that $\mathrm{cap}_1$ is the "hardest" function in that class in the sense that any approximate inverse transform for the function $\mathrm{cap}_1(x) = \min\{1, x\}$ can be extended (while retaining sketchability and approximation quality) to any statistics $f \in \overline{\mathrm{cap}}$, using the cap transform of $f$. We then derive some approximate transforms for $\mathrm{cap}_1(x)$, and hence for any $\overline{\mathrm{cap}}$ statistics that achieve maximum relative error of 12%.

Section 8 reports some experimental results. We conclude in Section 9. Due to page limitations, many details are omitted. A full version can be found in https://arxiv.org/abs/1607.06517.

## 2 THE LAPLACE$^C$ TRANSFORM

The *complement Laplace (Laplace$^c$) transform* $\mathcal{L}^c[W](t)$ at a point $t > 0$ is defined as

$$\mathcal{L}^c[W](t) \equiv \int_0^\infty W(w)(1 - \exp(-wt))dw$$
$$= \int_0^\infty W(w)dw - \mathcal{L}[W(w)](t) \,. \qquad (8)$$

See Figure 2 for an illustration of $\mathcal{L}^c[W](t)$ for a toy distribution $W$. The first term in (8), $\int_0^\infty W(w)dw \equiv \mathrm{Distinct}(W)$, is the "distinct count" and the second term $\mathcal{L}[W(w)]$ is the Laplace transform of our (scaled) frequency distribution $W$. Hence the name *complement Laplace* transform. Note that $\mathcal{L}^c[W](t)$ is non-decreasing with $t$. At the limit when $t$ increases, the second term vanishes and

$$\lim_{t \to +\infty} \mathcal{L}^c[W](t) = \int_0^\infty W(w)dw = \mathrm{Distinct}(W) \qquad (9)$$

is the number of distinct keys in $W$. At the limit as $t$ decreases

$$\lim_{t \to 0^+} \frac{1}{t}\,\mathcal{L}^c[W](t) = \int_0^\infty W(w)wtdw = t\,\mathrm{Sum}(W) \,, \qquad (10)$$

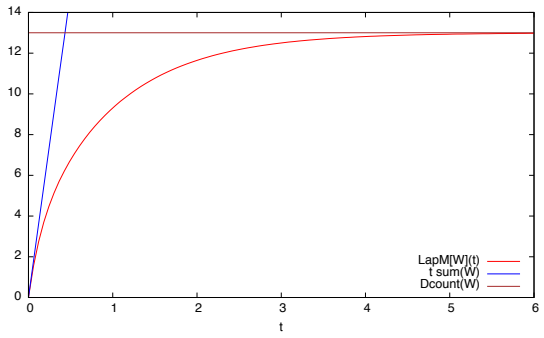


**Figure 2: Data $W$ with 10 keys with $w_x = 1$, 2 keys with $w_x = 5$, and one key with $w_x = 10$. The distinct count (number of keys) is Distinct$(W) = 13$ and the sum is Sum$(W) = 30$. We have $W(w) = 10\delta(w-1)+2\delta(w-5)+\delta(w-10)$, where $\delta$ is Dirac Delta function. The transform (plotted) is $\mathcal{L}^c[W](t) = 10(1-e^{-t})+2(1-e^{-5t})+(1-e^{-10t}) = 13-10e^{-t}-2e^{-5t}-e^{-10t}$. The plot shows the asymptotes $t$ Sum$(W)$ for small $t$ and Distinct$(W)$ for large $t$.**

where $\mathrm{Sum}(W) = \sum_{e\in W} e.value = \sum_x w_x$ is the sum of the weights of keys. More precisely:

Lemma 2.1. *For $t \le \frac{\sqrt{\epsilon}}{\max_x w_x}$ and for $t \ge \frac{-\ln\epsilon}{\min_x w_x}$, the transform is approximated within a relative error of at most $\epsilon$ by the respective limits* (10) *and* (9).

Proof. For the first claim, note that $wt \le \sqrt{\epsilon}$. Hence, using the Maclaurin expansion, $|1-e^{-wt}-wt| \approx (wt)^2/2 \le \epsilon$. For the second claim, the relative error is $\exp(-wt) \le \epsilon$. □

The Lemma implies that the fine structure of $W$ is captured by a restricted "relevant" range of $t$ values and is well approximated outside this range by the basic (and composably sketchable) Distinct and Sum statistics. The statistics Distinct$(W)$ is approximated by an off-the-shelf approximate distinct counter applied to data elements. The exact Sum$(W)$ is straightforward to compute composably with a single counter of size $O(\log \mathrm{Sum}(W))$ (assuming integral values). A classic algorithm by Morris [23] (see [4] for a composable version that can handle varying weights) uses sketches of size $O(\epsilon^{-2} + \log\log(\mathrm{Sum}(W))$.

## 3 LAPLACE$^C$ POINT MEASUREMENTS

We define a mapping function of elements such that the expectation of the (scaled) distinct count of output elements is equal to the Laplace$^c$ transform $\mathcal{L}^c[W](t)$ of $W$ at $t$. We also establish concentration around the expectation.

The basic element mapping is provided as Algorithm 1. A more efficient variant that performs computation proportional to the number of generated output elements is provided in the full version.

The mapping is parametrized by $t$ and by an integer $r \ge 1$ and uses a set of functions $H_i$ for $i \in [r]$. All we need to assume here is that for all $i$ and keys $x$, $H_i(x)$ are (nearly) unique. This can be achieved by concatenating $x$ to a string representation of $i$: $H_i(x) \equiv x \cdot \mathrm{str}(i)$). To obtain output key representation that is logarithmic in $r$ Distinct$(W)$, we can apply a random hash function to the concatenated string. An element $e$ is processed by drawing a set of $r$ independent exponential random variables $y_i \sim \mathrm{Exp}[e.value]$ with parameter $e.value$. For each $i$ such that $y_i < t$, the output key $H_i(e.key)$ is created. Note that the number of output keys returned is between 0 and $r$.

**Algorithm 1:** OutKeys$_{t,H}(e)$: Map input element to outkeys

**Input**: Element $e = (e.key, e.value)$, $t > 0$, integer $r \ge 1$, hash functions $H_i$ $i \in [r]$
**Output**: A set OutKeys of at most $r$ outkeys
OutKeys ← []  // initialize
**foreach** $i \in [r]$ **do**
  $y_i \sim \mathrm{Exp}[e.value]$  // independent exponentially distributed with parameter e.value
  **if** $y_i \le t$ **then**
    OutKeys.$append(H_i(e.key))$  // Append $H_i$(e.key) to list of output keys
**return** OutKeys

Our point measurement at $t$ is

$$\widehat{\mathcal{L}^c}[W](t) = \frac{1}{r}\,\mathrm{Distinct}\left(\bigcup_e \textsc{OutKeys}_{t,H}(e)\right), \qquad (11)$$

which is number of distinct output keys generated for all data elements, divided by $r$. We now show that for any choice of $r \ge 1$, $t$, and input data $W$, the expectation of the measurement $\widehat{\mathcal{L}^c}[W](t)$ is equal to the value of the Laplace$^c$ transform of $W$ at $t$.

Lemma 3.1.

$$E[\widehat{\mathcal{L}^c}[W](t)] = \mathcal{L}^c[W](t)$$

Proof. The number of distinct outkeys $r\widehat{\mathcal{L}^c}[W](t)$ can be expressed as the sum of $r$ Distinct$(W)$ Poisson events. For each input key $x$ and $i \in [r]$, the event is the appearance (at least once) of the outkey $H_i(x)$. The outkey $H_i(x)$ appears if the minimum $\mathrm{Exp}[e.value]$ draw over elements $e$ with key $x$ is at most $t$. The minimum of these exponential random variables is exponentially distributed with parameter equal to their sum $w_x = \sum_{e|e.key=x} e.value$. Therefore, the probability of the event is

$$p(w_x, t) = \int_0^t w_x \exp(-w_x y)dy = 1-\exp(-w_x t)\,. \qquad (12)$$

It follows that the expected contribution of a key $x$ with weight $w_x$ to the sum $r\widehat{\mathcal{L}^c}[W](t)$ is $rp(w_x, t)$. Therefore the expected value of the measurement is

$$\mathrm{E}[\widehat{\mathcal{L}^c}[W](t)] = \frac{1}{r}\int_0^\infty W(w)rp(w,t)dw \equiv \mathcal{L}^c[W](t)\,.$$

□

We next consider concentration of the measurement. When $t = +\infty$, we have $p(w,t) = 1$, and there is no measurement error. In general, we can bound the relative error by applying the Chernoff bound:

Lemma 3.2. *For $\delta < 1$,*

$$\Pr[\frac{|\widehat{\mathcal{L}^c}[W](t) - \mathcal{L}^c[W](t)|}{\mathcal{L}^c[W](t)} \geq \delta] \leq 2\exp(-r\delta^2\,\mathcal{L}^c[W](t)/3)\ .$$

The outkeys $E$ are processed by an *approximate* distinct counter which yields an approximate measurement

$$\widehat{\widehat{\mathcal{L}^c}}[W](t) = \frac{1}{r}\widehat{\text{Distinct}}(E) \tag{13}$$

equal to the approximate count of distinct output keys divided by $r$. Since there are at most $r$ Distinct$(W)$ distinct output keys, the sketch size needed for NRMSE $\epsilon$ is $O(\epsilon^{-2}+\log\log(r\,\text{Distinct}(W))) = O(\epsilon^{-2} + \log\log\text{Distinct}(W))$. Note that even a very large $r$ that is polynomial in Distinct$(W)$ will not significantly increase the sketch size. The two sources of error, due to the measurement itself and its approximation, are independent and the bottleneck one is the quality of the approximate distinct counter. When both estimates have NRMSE $\epsilon$ with concentration, the error of $\widehat{\widehat{\mathcal{L}^c}}[W](t)$ as an approximation of $\mathcal{L}^c[W](t)$ has NRMSE $\sqrt{2}\epsilon$ with concentration.

While the magnitude of $r$ has negligible effect on sketch size, it does effect element mapping computation. This can be mitigated by using a more efficient algorithm with computation that is linear in the number of generated output keys $O(r(1-\exp(-te.value))$ (see the full version). Moreover, we can bound the size of $r$ needed to guarantee that our measurement has CV of at most $\epsilon$:

Lemma 3.3. *When $t \geq \frac{\sqrt{\epsilon}}{Max(W)}$ (where $Max(W) \equiv \max_x w_x$) and*

$$r \geq \frac{3e}{e-1}\epsilon^{-2.5}\ , \tag{14}$$

*the estimator $\widehat{\mathcal{L}^c}[W](t)$ (defined in (11)) has CV at most $\epsilon$.*

Proof. From Lemma 3.2, it suffices to have

$$r\,\mathcal{L}^c[W](t) \geq 3\epsilon^{-2}\ . \tag{15}$$

From (7) and (3),

$$\mathcal{L}^c[W](t) \geq (1-\frac{1}{e})\sum_x \min\{1,\, tw_x\} \tag{16}$$

When $t \leq 1/\,\text{Max}(W)$ we have

$$\sum_x \min\{1, tw_x\} = t\,\text{Sum}(W) \geq \sqrt{\epsilon}\frac{\text{Sum}(W)}{\text{Max}(W)} \geq \sqrt{\epsilon}\ .$$

When $t \geq 1/\,\text{Max}(W)$ we have $\sum_x \min\{1, tw_x\} \geq \sum_x \min\{1, \frac{w_x}{\text{Max}(W)}\} \geq 1$. Combining, we get $\mathcal{L}^c[W](t) \geq \frac{e-1}{e}\sqrt{\epsilon}$ and the claim follows using (15). □

Recall from Lemma 2.1 that when $t < \sqrt{\epsilon}/\,\text{Max}(W)$, $\mathcal{L}^c[W](t)$ is well approximated by $t\,\text{Sum}(W)$. Since we do not know Max$(W)$ or Sum$(W)$ in advance, we use the following strategy. Our approximate point measurement algorithm computes both an approximate sum $\widehat{\text{Sum}}(W)$ and approximate count of output elements $\widehat{\text{Distinct}}(E)$ generated by Algorithm 1 with $r$ as in (14). If $\widehat{\text{Distinct}}(E) < 3\epsilon^{-2}$, we return $t\widehat{\text{Sum}}(W)$ and otherwise return (13).

We comment here that using $r$ as in (14) provides seamless worst-case quality guarantees for any $t$ and distribution $W$. In practice it is often safe to assume that Sum$(W) \gg$ Max$(W)$ and a small value of $r$ suffices. In particular when Sum$(W) \geq \epsilon^{-2.5}$ Max$(W)$ we can use $r = 1$.

# 4 THE SOFTCAP SPAN

The *softcap span* $\overline{\widehat{\text{cap}}}$ contains all functions $f$ that can be expressed as nonnegative linear combinations of $\widehat{\text{cap}}_T$ functions. Equivalently, for some $a(t) \geq 0$,

$$f(w) = \mathcal{L}^c[a](w) = \int_0^\infty a(t)(1-e^{-wt})dt\ . \tag{17}$$

Note that $a(t)$ is the inverse Laplace$^c$ transform of $f(w)$:

$$a(t) = (\mathcal{L}^c)^{-1}[f(w)](t)\ . \tag{18}$$

The following is immediate

Lemma 4.1. *The relation (17) implies that $a(t) \geq 0$ satisfies*

$$\int_0^1 a(t)tdt < \infty \ \text{ and } \ \int_1^\infty a(t)dt < \infty\ . \tag{19}$$

Table 1 lists explicit expressions for the inverse $\mathcal{L}^c$ transforms of some basic functions in the softcap span ($w^p$ for all $p \in (0,1)$, and $\ln(1+w)$). The table also includes other expressions that we will use for sketching the statistics. Our derivations utilized the following Lemma which expresses the inverse $\mathcal{L}^c$ transform of $f$ in terms of the inverse Laplace transform of the derivative of $f(w)$:

Lemma 4.2.

$$(\mathcal{L}^c)^{-1}[f(w)](t) = \frac{1}{t}\,\mathcal{L}^{-1}[\frac{\partial f(w)}{\partial w}](t)\ ,$$

*where $\mathcal{L}$ is the Laplace transform.*

Proof. We look for a solution $a(t)$ of (17). Differentiating both sides by $w$ we obtain

$$\frac{\partial f(w)}{\partial w} = \int_0^\infty ta(t)e^{-wt}dt = \mathcal{L}[a(t)t](w)\ .$$

□

In preparation for the task of sketching the statistics $f(W)$, we express it in terms of the inverse $\mathcal{L}^c$ transform $a(t)$ of $f(w)$ and the transform $\mathcal{L}^c[W]$ of the frequencies:

$$\begin{aligned} f(W) &= \int_0^\infty f(w)W(w)dw \\ &= \int_0^\infty W(w)\,\mathcal{L}^c[a](w)dw \\ &= \int_0^\infty a(t)\int_0^\infty W(w)(1-e^{-wt})dwdt \\ &= \int_0^\infty a(t)\,\mathcal{L}^c[W](t)dt\ . \end{aligned} \tag{20}$$

When the inverse transform has a discrete form, that is, $f$ is expressed using $\{a_t\}$ for $t \in Y$ as

$$f(w) = \sum_{t\in Y} a_t(1-e^{-wt})\ , \tag{21}$$

(Equivalently, $a(t)$ is a linear combination of Dirac delta functions at $t \in Y$). We can express $f(W)$ in terms of corresponding points of $\mathcal{L}^c[W]$:

$$f(W) = \sum_{t\in Y} a_t\,\mathcal{L}^c[W](t)$$

**Table 1: Inverse $\mathcal{L}^c$ transform of basic functions**

| $f(w)$ | $a(t) = (\mathcal{L}^c)^{-1}[f(w)](t)$ | $\int_\tau^\infty a(t)dt$ | $\int_0^\tau a(t)tdt$ |
|---|---|---|---|
| $\widehat{\mathrm{cap}}_T(w)$ | $T\delta_{1/T}(t)$ | $T$, when $y \le 1/T$; 0 otherwise | 0 when $\tau < 1/T$ |
| $w^p$ $(p \in (0,1))$ | $\frac{p}{\Gamma(1-p)} t^{-(1+p)}$ | $\frac{1}{\tau^p \Gamma(1-p)}$ | $\frac{p}{(1-p)\Gamma(1-p)} \tau^{1-p}$ |
| $\sqrt{w}$ | $\frac{1}{2\sqrt{\pi}} t^{-1.5}$ | $\frac{1}{\sqrt{\pi\tau}}$ | $\frac{\tau^{0.5}}{\sqrt{\pi}}$ |
| $\log(1+w)$ | $\frac{e^{-t}}{t}$ | $\int_\tau^\infty \frac{e^{-t}}{t}dt = -\tau \mathrm{Ei}(-\tau)$ | $1 - e^{-\tau}$ |

and approximate $f(W)$ using corresponding point measurements of $\mathcal{L}^c[W]$. In the next section we introduce *combination* $\mathcal{L}^c$ measurements which allow us to approximate $\mathcal{L}^c[W][a]$ for any $a(t) \ge 0$ that satisfies (19).

## 5 COMBINATION $\mathcal{L}^c$ MEASUREMENTS

In this section we show how to sketch $f(W)$ for any $f \in \widehat{\mathrm{cap}}$. We will use the notation

$$\mathcal{L}^c[W][a]_\tau^b \equiv \int_\tau^b a(t)\, \mathcal{L}^c[W](t)dt\ . \tag{22}$$

When the subscript or superscript are omitted, we default to $\tau = 0$ and $b = \infty$. We use the same notation with approximate measurements:

$$\widehat{\mathcal{L}^c}[W][a]_\tau^b = \int_\tau^b a(t)\, \widehat{\mathcal{L}^c}[W](t)dt\ . \tag{23}$$

From Section 4, we can equivalently present a sketch design for $\mathcal{L}^c[W][a]_0^\infty$ where $a(t) \ge 0$ satisfies (19). We define randomized mapping functions of elements, tailored to some $a(t) \ge 0$ and $\tau > 0$, such that the expectation of the (scaled) max-distinct statistics of output elements is equal to $\mathcal{L}^c[W][a]_\tau^\infty$ and establish concentration around the expectation. We estimate the contribution $\mathcal{L}^c[W][a]_0^\tau$ of the low-$t$ regime by a separate $\mathsf{Sum}(W)$ sketch.

### 5.1 Element mapping

Consider $a(t) \ge 0$. Our element processing is a simple modification of the element processing Algorithm 1 for point measurements. The algorithm inputs the function $a()$ (instead of $t$) and returns output elements (outkey and value pairs) instead of only returning outkeys. Pseudocode is provided as Algorithm 2.

**Algorithm 2:** OutElements $_{a,\tau,H}(e)$: Map of element $e$ to a set of output elements for a $\mathcal{L}^c[W][a]_\tau^\infty$ measurement

**Input**: Element $e = (e.key, e.value)$, $a(t) \ge 0$, integer $r \ge 1$, hash functions $H_i$ $i \in [r]$, $\tau > 0$
**Output**: OutElements: A set of at most $r$ output elements
OutElements $\leftarrow$ [] `// initialize`
**foreach** $i \in [r]$ **do**
  $y_i \sim \mathrm{Exp}[e.value]$ `// independent exponentially distributed with parameter e.value`
  $v \leftarrow \int_{\max\{\tau, y_i\}}^\infty a(t)dt$
  **if** $v > 0$ **then**
    OutElements.*append*$((H_i(e.key), v))$ `// New output element`
**return** OutElements

Our combination measurement is the max-distinct statistics of the output elements divided by $r$:

$$\widehat{\mathcal{L}^c[W][a]}_\tau^\infty = \frac{1}{r}\, \mathsf{MxDistinct}\left(\bigcup_{e \in W} \mathsf{OutElements}_a(e)\right)\ . \tag{24}$$

We show that the measurement has expectation equal to $\mathcal{L}^c[W][a]_\tau^\infty$ with good concentration:

Lemma 5.1.

$$E[\widehat{\mathcal{L}^c[W][a]}_\tau^\infty] = \mathcal{L}^c[W][a]_\tau^\infty$$

$$\forall \delta < 1,\ \Pr\left[\frac{|\widehat{\mathcal{L}^c[W][a]}_\tau^\infty - \mathcal{L}^c[W][a]_\tau^\infty}{\mathcal{L}^c[W][a]_\tau^\infty} \ge \delta\right] \le\ 2\exp(-r\delta^2\, \mathcal{L}^c[W][\tau]/3)\ .$$

Proof. The claim on the expectation follows from linearity of expectation and the claim for point measurements for each $t$ in Lemma 3.1. The concentration follows from Lemma 3.2 which establishes point-wise concentration at each $t$, combined with the relation

$$\min_{t \in [\tau, \infty)} \mathcal{L}^c[W](t) = \mathcal{L}^c[W](\tau)$$

which follows from monotonicity of $\mathcal{L}^c[W](t)$. □

Note that nonnegativity $a(t) \ge 0$ is necessary for correctness: It ensures monotonicity of $\int_y^\infty a(t)dt$ in $y$ which implies that the maximum indeed corresponds to minimum $y$.

We now address quality of approximation. From the lemma, quality is bounded by a function of the value of the transform at point $\tau$: $\mathcal{L}^c[W](\tau)$. From our analysis of point measurements, we know that it suffices to ensure that $\tau$ and $r$ are large enough so that (15) holds. As with point measurements, we can use $\tau = \sqrt{\epsilon}/\max(W))$ and $r$ as in (14) to obtain a concentrated measurement of $\mathcal{L}^c[W][a]_\tau^\infty$.

### 5.2 The low $t$ regime

To obtain an estimate of $\mathcal{L}^c[W][a]_0^\infty$, we need to separately estimate $\mathcal{L}^c[W][a]_0^\tau$. From Lemma 2.1, when $\tau \le \sqrt{\epsilon}/\max(W)$, we have

$$\mathcal{L}^c[W][a]_0^\tau \approx \int_0^\tau a(t)t\, \mathsf{Sum}(W)dt \approx \widehat{\mathsf{Sum}(W)} \int_0^\tau a(t)tdt\ . \tag{25}$$

The first approximation has relative error at most $\epsilon$. For the second, as discussed earlier, we can use exact $\mathsf{Sum}(W)$ or a composable Morris counter sketch that provides an estimate with a well concentrated error of $\epsilon$.

Closed expressions for $\int_0^\tau a(t)tdt$ (used in (25) and for $\int_\tau^\infty a(t)dt$ (used in Algorithm 2) for inverse transforms of basic functions are

provided in Table 1. Note that these expressions are bounded and well defined for all $a$ that are inverse $\mathcal{L}^c$ transform of a $\widehat{\overline{\mathsf{cap}}}$ function (see Lemma 4.1).

### 5.3 Putting it together

Our sketch consists of a $\widehat{\mathsf{Sum}}$ sketch applied to data elements $W$ and a $\widehat{\mathsf{MxDistinct}}$ sketch applied to output element $E$ produced by applying the element mapping Algorithm 2 to $W$. The final estimate we return is

$$\widehat{\widehat{\mathcal{L}^c}}[W][a]_0^\infty = \frac{1}{r}\widehat{\mathsf{MxDistinct}}(E) + \widehat{\mathsf{Sum}}(W)\int_0^\tau a(t)tdt\ . \quad (26)$$

There is one subtlety here that was deferred for the sake of exposition: We do not know $\mathsf{Max}(W)$ and therefore can not simply set $\tau$ prior to running the algorithm. To obtain the worst-case statistical guarantees we set $r$ as in (14) and set $\tau$ adaptively to be the $\ell = 3\epsilon^{-2}$ smallest $y$ value that is generated for a distinct output key. To do so, we extend our sketch to include another component, which we call the *sidelined* set, which is the $\ell$ distinct output keys with smallest $y$ values processed so far. Note that this sketch component is also composable. The sidelined elements are not immediately processed by the MxDistinct sketch: The processing is delayed until they are taken out of being sidelined, that is, to the point that due to sketch compositions or new arrivals, there are $\ell$ other distinct output keys with lower $y$ values. Finally, when the sketches are used to extract an estimate of the statistics, we set $\tau$ as the largest $y$ in the sidelined set, feed all the sidelined keys to the MxDistinct sketch with value $\int_\tau^\infty a(t)dt$, and apply the estimator (26). We note that explicitly maintaining the sidelined output keys and $y$ value pairs would require $\epsilon^{-2}\log n$ storage. Details on how to obtain the same effect using a double logarithmic size sketch component are provided in the full version.

### 5.4 Max-distinct sketches

For completeness, we discuss some sketch designs for max-distinct statistics. These sketches generalize MinHash sketches for distinct statistics (see overview in [4]) and are based on composable weighted sampling schemes.

With bottom-$k$ sampling [8, 9, 25, 28, 29] we associate with each element (with a unique key) $e$ an independent random rank value $r_e \sim D[e.value]$ from a distribution $D$ parametrized by $e.value$ (Exponential [28] or uniform $U[0, 1/e.value]$ [25]). The sample includes the $k$ keys with minimum $r_e$ and the corresponding $e.value$. It is easy to see that this can be performed using composable sketches that contain $k = \epsilon^{-2}$ key value pairs. When the keys are not unique, we instead use ranks obtained by applying the respective inverse CDF to $u_{e.key} \sim U[0, 1]$, which is a hash function that maps strings to independent uniform random numbers. What this achieves is that minimum rank of elements with the same key $x$ has distribution $D[m_x]$ and hence the sample distribution is the same as the plain scheme applied to elements $(x, m_x)$ with unique keys. An estimator of the max-distinct statistics $\sum_x m_x$ that is applied to the sample [9, 11] is well-concentrated with CV $1/\sqrt{k-2} \approx \epsilon$. The representation of the sample amounts to storing $k$ key identifiers (or their $O(\log n)$ size hashes) and their respective $m_x$ values resulting in representation size of $O(\epsilon^{-2}\log n)$. The size can be reduced by hashing keys to a domain that is polynomial in $k$ and store approximate consistently-rounded ranks in an offset form.

A different design of max-distinct sketches of size $O(\log\log n + \epsilon^{-2}\log\epsilon^{-1})$ (assuming $1 \le m_x = O(\mathrm{poly}(n))$) builds on with-replacement weighted sampling [3] and popular distinct counting sketches [13, 14]. Our sketch contain $k$ registers that corresponds to balanced "buckets" of output keys. For balance, we can place all keys in all buckets or apply stochastic averaging and partition the output keys to buckets according to a partition of $[r]$. For an element $e$ with key that falls in bucket $i$, we compute $-\ln u_{e.key\#i}/e.value$ and if smaller than the current register, replace its value. The distribution of the minimum in the bucket is exponential with parameter equals to the sum of $m_x$ over keys in the bucket (see e.g. [3]). Since the buckets are balanced, each register contains a sample from this distribution and we can estimate the parameter by $k-1$ divided by the sum of the $k$ registers. As in Hyperloglog, we can use consistent (per key/bucket) randomized rounding to an integral power of $(1+\delta)$ and store only the negated exponent $y_i$ for register $i$. The exponent can be stored in $O(\log\log n)$ bits. For different buckets, we can store one exponent and offsets of expected size $O(1)$ per bucket. To obtain the approximate statistics from the sketch we use the estimator

$$\widehat{\mathsf{MxDistinct}}(E) = (k-1)/\sum_{i=1}^{k} b^{-y_i}\ .$$

## 6 FULL RANGE $\mathcal{L}^c$ MEASUREMENTS

In this section we present a composable sketch from which we can approximate $\mathcal{L}^c[W](t)$ for all $t$ and $\mathcal{L}^c[w][a]_\tau^\infty$ for all applicable $a() \ge 0$ and $\tau$. Concretely, consider the set of output keys $\mathsf{OutKeys}_t(e)$ generated by Algorithm 1 for input element $e$ when fixing the parameter $r$, the set of hash functions $\{H_i\}$, and the randomization $\{y_i\}$, but varying $t$. It is easy to see that the set $\mathsf{OutKeys}_t(e)$ monotonically increases with $t$ until it reaches size $r$. We can now consider all outkeys generated for input $W$ as a function of $t$

$$\mathsf{OutKeys}_t(W) \equiv \bigcup_{e\in W}\mathsf{OutKeys}_t(e)\ .$$

The number of distinct outkeys increases with $t$ until it reaches size $rn$, where $n$ is the number of distinct input keys. Our *full-range* measurement is accordingly defined as the function

$$\widehat{\mathcal{L}^c}[W](t) = \frac{1}{r}\left|\mathsf{OutKeys}_t(W)\right|\ . \quad (27)$$

The full-range element mapping Algorithm 3 returns for each input element $e$, a set of $r$ output elements $\mathsf{OutElements}_H(e)$ that are outkey and value pairs. The mapping is equivalent to Algorithm 1 except that the point $t$ is not specified and instead the output elements are always generated with value equal to the applicable threshold $t$.

### 6.1 Point from full range

We denote the set of output elements generated for all input elements $e \in W$ by $\mathsf{OutElements}(W)$. We have that

$$\mathsf{OutKeys}_t(W) = \{e.key \mid e \in \mathsf{OutElements}(W)\text{such that}e.value \le t\}\ .$$

Therefore, the point measurement for $t$ is

$$\widehat{\mathcal{L}^{\mathrm{c}}}[W](t) = \frac{1}{r}\,\mathsf{Distinct}\{e \in \mathsf{OutElements}(W) \mid e.value \le t\}\,, \tag{28}$$

which is the number of distinct keys in output elements that have value at most $t$. This measurement is identical to the one we would have obtained using Algorithm 1 with input $t$.

## 6.2 Combination from full range

A combination measurement can be computed from a full range measurement using

$$\widehat{\mathcal{L}^{\mathrm{c}}}[W][a]_\tau^\infty = \int_\tau^\infty a(t)\widehat{\mathcal{L}^{\mathrm{c}}}[W](t)dt\,. \tag{29}$$

We show that this formulation is equivalent to a combination measurement (24) obtained by a max-distinct statistics of output elements generated by Algorithm 2:

Lemma 6.1. *Consider $a(t) \ge 0$, $\tau \ge 0$ and element mappings for full-range (Algorithm 3) and combination (Algorithm 2) measurements where the mappings are performed using identical parameter $r$, hash functions $H$, and random draws $y$. Then the corresponding combination measurements* (29) *and* (24) *are equal.*

Proof. It is easy to verify that

$$\begin{aligned}\widehat{\mathcal{L}^{\mathrm{c}}[W][a]}_\tau^\infty &= \frac{1}{r}\,\mathsf{MxDistinct}\left(\bigcup_{e\in W}\mathsf{OutElements}_{a,\tau}(e)\right)\\ &= \frac{1}{r}\int_\tau^\infty a(t)|\mathsf{OutKeys}_t(W)|dt\,.\end{aligned}$$

□

## 6.3 Sketching a full range measurement

The output elements $\mathsf{OutElements}(W)$ are processed by a composable *all-threshold* distinct counting sketch. Our all-threshold sketch is a single-point All-Distance Sketch (ADS) [3–5] (universal monotone sample). We refer the reader to [4, 5] for full details and provide here an overview for completeness. The ADS summarizes data elements $E$ that are a set of key value pairs with nonnegative values. From the sketch we can approximate $\mathsf{TDistinct}_t(E)$ (6) for any $t$ and $\int_\tau^\infty a(t)\,\mathsf{TDistinct}_t(E)dt$ for any integrable $a(t) \ge 0$ and $\tau$. Respectively, we obtain from an ADS of the output elements, approximate point $\widehat{\widehat{\mathcal{L}^{\mathrm{c}}}}[W][t]$ and combination $\widehat{\widehat{\mathcal{L}^{\mathrm{c}}}}[W][a]_\tau^\infty$ measurements.

An ADS is a parametric extension of a MinHash sketch: It efficiently represents MinHash sketches for all the sets $\mathsf{OutKeys}_t(W)$ for different $t$. It retains the sketch structure of the base MinHash sketch and records all the change points $t$ at which the content of a register (of the respective MinHash sketches) changes. At the base we can use almost any MinHash sketch design including HyperLogLog [13] which uses $k = \epsilon^{-2}$ registers that store respective maximum values over processed data elements. As $t$ is increased, the values that the sketch would have stored increase. The ADS records the values and change points $t$ for each register. The total expected number of change points for all $k$ registers is well-concentrated around $k\ln(rn)$. Hence, the overhead factor of all-threshold sketching is $O(\epsilon^{-2}\ln n)$ and the representation of the breakpoints. Since $\mathcal{L}^{\mathrm{c}}[W](t)$ is smooth and Lipschitz, we can limit the precision in which the breakpoints are represented to $\log(1/\epsilon)$ significant bits and use a single exponent and offsets. We can apply the HIP estimator [4, 5] to the ADS (with respect to the swept parameter $t$) to obtain tighter estimates for all $t$.

**Algorithm 3:** $\mathsf{OutElements}_H(e)$: Map of element $e$ to a set of output elements for full range measurement

**Input**: Element $e = (e.key, e.value)$, integer $r \ge 1$, hash functions $\{H_i\}$ $i \in [r]$
**Output**: OutElements: A set of $r$ output elements
OutElements ← []
**foreach** $i \in [r]$ **do**
  $y_i \sim \mathsf{Exp}[e.value]$ `// independent exponentially distributed with parameter e.value`
  OutElements.*append*$((H_i(e.key), y_i))$ `// New output element`
**return** OutElements

# 7 THE CAP SPAN

In this section we consider sketching statistics for concave sublinear $f$ that may not be in $\overline{\overline{\mathsf{cap}}}$. We show that all concave sublinear $f$ are in the nonnegative span $\overline{\mathsf{cap}}$ of cap functions. We define *cap transform* of $f$ as a function $a(t) \ge 0$ and $A_\infty \ge 0$ such that $\int_0^\infty a(t)dt < \infty$ and

$$f(x) = A_\infty x + \int_0^\infty a(t)\mathsf{cap}_t(x)dt\,.$$

We can express the transform as follows:

Theorem 7.1. *Let $f : [0,\infty]$ be a nonnegative, continuous, concave, and monotone non-decreasing function such that $f(0) = 0$ and $\partial_+ f(0) < \infty$. Then $f \in \overline{cap}$ with the cap transform:*

$$a(x) = \begin{cases} -\partial^2 f(x) & \text{when } \partial_-^2 f(x) = \partial_+^2 f(x) \\ (\partial_- f(x) - \partial_+ f(x))\delta_x & \text{otherwise}\end{cases} \tag{30}$$

$$A_\infty = \partial f(\infty) \tag{31}$$

*where $\delta$ is the Dirac delta function and $\partial f(\infty) \equiv \lim_{t\to\infty}\partial f(t)$.*

## 7.1 Sketching with signed inverse transforms

We consider sketching statistics for $f$ such that $a(t) = \mathcal{L}^{\mathrm{c}-1}[f(w)](t)$ (17) is *signed*. We use the notation $a(t) = a_+(t) - a_-(t)$ where

$$a_+(t) = \max\{a(t), 0\} \text{ and } a_-(t) = \max\{-a(t), 0\}\,. \tag{32}$$

We define $f_+(w) = \mathcal{L}^{\mathrm{c}}[a_+](w)$ and $f_-(w) = \mathcal{L}^{\mathrm{c}}[a_-](w)$. Note that for all $w$, $f(w) = f_+(w) - f_-(w)$ and in particular $f(W) = f_+(W) - f_-(W)$. Since $a_+, a_- \ge 0$, we can obtain a good approximations $\hat{f}_+(W)$ and $\hat{f}_-(W)$ for each of $f_+(W) = \mathcal{L}^{\mathrm{c}}[W][a_+]$ and $f_-(W) = \mathcal{L}^{\mathrm{c}}[W][a_-]$ using approximate full-range, two combination, or several point measurements when $a$ is discrete and small. We approximate $f(W)$ using the difference $\hat{f}(W) = \hat{f}_+(W) - \hat{f}_-(W)$.

The quality of this estimate depends on a parameter $\rho$:

$$\rho(a) \equiv \max_w \max\frac{\mathcal{L}^{\mathrm{c}}[a_+](w)}{\mathcal{L}^{\mathrm{c}}[a](w)}, \frac{\mathcal{L}^{\mathrm{c}}[a_-](w)}{\mathcal{L}^{\mathrm{c}}[a](w)} \tag{33}$$

Lemma 7.2. *For all $W$, $f(w) = \mathcal{L}^c[a](w)$,*

$$\frac{|f(W) - \hat{f}(W)|}{f(W)} \leq \rho(a)\left(\frac{|\mathcal{L}^c[W][a_+] - \widehat{\mathcal{L}^c}[W][a_+]|}{\mathcal{L}^c[W][a_+]} + \frac{|\mathcal{L}^c[W][a_-] - \widehat{\mathcal{L}^c}[W][a_-]|}{\mathcal{L}^c[W][a_-]}\right)$$

That is, when the components $f_+(W)$ and $f_-(W)$ are estimated within relative error $\epsilon$, then our estimate of $f(W)$ has relative error at most $\epsilon\rho$. In particular, the concentration bound in Lemma 5.1 holds with $\rho\delta$ replacing $\delta$ and the sketch size has $\epsilon\rho$ replacing $\epsilon$.

When the exact inverse transform $a$ of $f$ is not defined or has a large $\rho(a)$, we look for an *approximate* inverse transform $a$ such that $\rho(a)$ is small and $f(w) \approx \mathcal{L}^c[a](w)$:

$$\mathsf{relerr}(f, \mathcal{L}^c[a]) = \max_{w>0} \frac{|f(w) - \mathcal{L}^c[a](w)|}{f(w)} \leq \epsilon \quad (34)$$

## 7.2 From $\mathsf{cap}_1$ to $\overline{\mathsf{cap}}$

We show that from an approximate signed inverse transform of $\mathsf{cap}_1$ we can obtain one with the same quality for any $f \in \overline{\mathsf{cap}}$.

Theorem 7.3. *Let $f \in \overline{cap}$ and let $a(t)$ be the cap transform of $f$. Let $\alpha(x)$ be such that $\mathcal{L}^c[\alpha](w)$ is an approximation of $cap_1$. Then*

$$c(x) = \int_0^\infty a(T)\alpha(x/T)dT\ , \quad (35)$$

*is an approximate inverse transform of $f$ that satisfies*

$$\rho(c) \leq \rho(\alpha) \quad (36)$$

$$relerr(f, \mathcal{L}^c[c]) \leq relerr(cap_1, \mathcal{L}^c[\alpha]) \quad (37)$$

It follows that to approximate $f(W)$ we do as follows. We compute the cap transform of $f$ and use an approximate inverse transform of $\mathsf{cap}_1$, from which we obtain an approximate inverse transform $c$ of $f$. We then perform two combination measurements (with respect to the negative and positive components $c_+$ and $c_-$). Alternatively, we can use one full-range measurement to estimate both.

## 7.3 Sketching $\mathsf{cap}_1$

We consider approximate inverse transforms of $\mathsf{cap}_1$. The simplest approximation (see (3)) is to approximate $\mathsf{cap}_T$ statistics by $\widetilde{\mathsf{cap}}_T$. The worst-case error of this approximation is $\mathsf{relerr}(\mathsf{cap}_T, \widetilde{\mathsf{cap}}_T) = 1/e \approx 0.37$. Note, however, that the relative error is maximized at $w = T$, but vanishes when $w \ll T$ and $w \gg T$. This means that only distributions that are heavy with keys of weight approximately $T$ would have significant error. Noting that $\widetilde{\mathsf{cap}}_T$ is an underestimate of $\mathsf{cap}_T$, we can decrease the worst-case relative error using the approximation

$$\frac{2e}{2e-1}\widetilde{\mathsf{cap}}_T(w) \quad (38)$$

and obtain $\mathsf{relerr}(\mathsf{cap}_T, \frac{2e}{2e-1}\widetilde{\mathsf{cap}}_T) = \frac{1}{2e-1} \approx 0.23$ . This improvement, however, comes at the cost of spreading the error, that otherwise dissipated for very large and very small frequencies $w$, across all frequencies.

We derive tighter approximations of $\mathsf{cap}_1$ using a signed approximate inverse transform $\alpha()$. We first specify properties of $\alpha()$ so that $\mathcal{L}^c[\alpha]$ has desirable properties as an approximation of $\mathsf{cap}_1$. To have the error vanish for $w \gg 1$, that is, have $\mathcal{L}^c[\alpha](w) \to 1$ when $w \to +\infty$, we must have

$$\int_0^\infty \alpha(t)dt = 1\ . \quad (39)$$

To have the error vanish for $w \ll 1$, that is, have $\mathcal{L}^c[\alpha](w)/w \to 1$ when $w \to 0$, we must have

$$\int_0^\infty t\alpha(t)dt = 1\ . \quad (40)$$

We show (see full version) that using $\alpha$ of the form

$$\alpha(t) = (A+1)\delta_1(t) - \alpha_1\delta_{\beta_1}(t) - \alpha_2\delta_{\beta_2}(t)\ ,$$

where $\delta$ is the Dirac delta function, we obtain the following:

| $A$ | $\beta_1$ | $\beta_2$ | relerr $\approx$ | $\rho(\alpha) <$ |
|---|---|---|---|---|
| 10.0 | 0.9 | 3.75 | 0.115 | 12.4 |
| 1.5 | 0.6 | 7.97 | 0.14 | 2.9 |

Figure 3 shows $\mathsf{cap}_1(w)$ and various approximations. The single point measurement approximations: $\widetilde{\mathsf{cap}}_1$ and scaled $\widetilde{\mathsf{cap}}_1$ and the two 3-point approximations from the table. One plot shows the functions and the other shows their ratio to $\mathsf{cap}_1$ which corresponds to the relative error as a function of $w$. The graphs show that the error vanishes for small and large values of $w$ for all but the scaled $\widetilde{\mathsf{cap}}_1$ (38). We can also see the smaller error for the 3-point approximations.

# 8 EXPERIMENTS

We performed experiments aimed at demonstrating the practicality of our schemes and providing some insights on parameter selection. We implemented the element mapping functions in Python and generated synthetic data elements with values equal to 1 and Zipf distributed keys with parameter $\alpha \in [1, 2]$. Figure 4 shows the measurement error as a function of the parameter $r$ for point measurements and for a combination measurement.

# 9 CONCLUSION

We presented a novel elegant framework for composable sketching of concave sublinear statistics. We obtain state-of-the-art asymptotic bounds on sketch size together with highly efficient practical solution.

We leave open the intriguing questions of fully understanding the limits of our approach. In particular, whether the scope of sample-based sketching is limited to (sub)linear statistics (we suspect it does) and to precisely quantify the attainable approximation tradeoff for $\mathsf{cap}_1$ (and hence for any $\overline{\mathsf{cap}} \setminus \widetilde{\overline{\mathsf{cap}}}$ function)

# REFERENCES


[1] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *J. Comput. System Sci.*, 58:137–147, 1999.
[2] V. Braverman and R. Ostrovsky. Zero-one frequency laws. In *STOC*. ACM, 2010.
[3] E. Cohen. Size-estimation framework with applications to transitive closure and reachability. *J. Comput. System Sci.*, 55:441–453, 1997.
[4] E. Cohen. All-distances sketches, revisited: HIP estimators for massive graphs analysis. *TKDE*, 2015.
[5] E. Cohen. Multi-objective weighted sampling. In *HotWeb*. IEEE, 2015. full version: http://arxiv.org/abs/1509.07445.
[6] E. Cohen. Stream sampling for frequency cap statistics. In *KDD*. ACM, 2015. full version: http://arxiv.org/abs/1502.05955.
[7] E. Cohen, N. Duffield, H. Kaplan, C. Lund, and M. Thorup. Algorithms and estimators for accurate summarization of unaggregated data streams. *J. Comput. System Sci.*, 80, 2014.

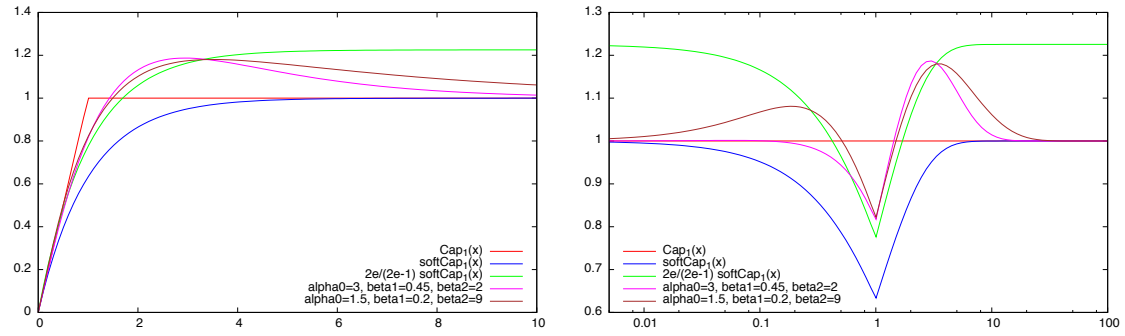


**Figure 3: Left: $\mathrm{cap}_1(w)$ function and different approximations: With a single $\mathcal{L}^c$ point measurement we can use the softcap function $\widetilde{\mathrm{cap}}_1(w)$ or scaling it $\frac{2e}{2e-1}\widetilde{\mathrm{cap}}_1(w)$ to minimize the worst-case relative error. We also show two 3 point measurements approximations with different parameters. Right: The corresponding ratio to $\mathrm{cap}_1$ which shows the relative error of the different approximations.**

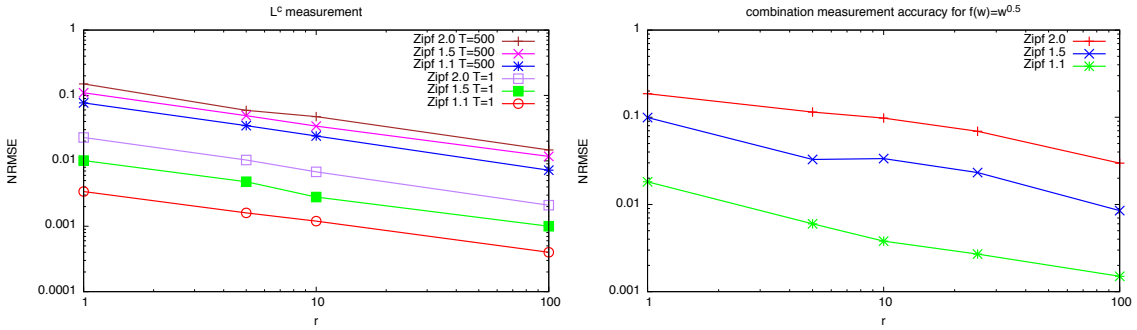


**Figure 4: Experiments on $10^5$ elements with Zipf-distributed keys (averaged over 200 repetitions), showing measurement error NRMSE as a function of $r$ Left: point measurements $\widehat{\mathcal{L}^c}[W](1/T)$. Right: combination measurement for $f(w) = \sqrt{w}$.**


[8] E. Cohen and H. Kaplan. Summarizing data using bottom-k sketches. In *ACM PODC*, 2007.

[9] E. Cohen and H. Kaplan. Tighter estimation using bottom-k sketches. In *Proceedings of the 34th VLDB Conference*, 2008.

[10] E. Cohen, H. Kaplan, and S. Sen. Coordinated weighted sampling for estimating aggregates over multiple weight assignments. *VLDB*, 2(1–2), 2009. full: `http://arxiv.org/abs/0906.4560`.

[11] N. Duffield, M. Thorup, and C. Lund. Priority sampling for estimating arbitrary subset sums. *J. Assoc. Comput. Mach.*, 54(6), 2007.

[12] C. Estan and G. Varghese. New directions in traffic measurement and accounting. In *SIGCOMM*. ACM, 2002.

[13] P. Flajolet, E. Fusy, O. Gandouet, and F. Meunier. Hyperloglog: The analysis of a near-optimal cardinality estimation algorithm. In *Analysis of Algorithms (AofA)*. DMTCS, 2007.

[14] P. Flajolet and G. N. Martin. Probabilistic counting algorithms for data base applications. *J. Comput. System Sci.*, 31:182–209, 1985.

[15] P. Gibbons and Y. Matias. New sampling-based summary statistics for improving approximate query answers. In *SIGMOD*. ACM, 1998.

[16] Google. *Frequency capping: AdWords help*, December 2014. https://support.google.com/adwords/answer/117579.

[17] S. Heule, M. Nunkesser, and A. Hall. HyperLogLog in practice: Algorithmic engineering of a state of the art cardinality estimation algorithm. In *EDBT*, 2013.

[18] P. Indyk. Stable distributions, pseudorandom generators, embeddings and data stream computation. In *Proc. 41st IEEE Annual Symposium on Foundations of Computer Science*, pages 189–197. IEEE, 2001.

[19] W. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Math.*, 26, 1984.

[20] D. M. Kane, J. Nelson, and D. P. Woodruff. An optimal algorithm for the distinct elements problem. In *PODS*, 2010.

[21] D. E. Knuth. *The Art of Computer Programming, Vol 2, Seminumerical Algorithms*. Addison-Wesley, 1st edition, 1968.

[22] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[23] R. Morris. Counting large numbers of events in small registers. *Comm. ACM*, 21, 1977.

[24] S. Nandanwar and N. N. Murty. Structural neighborhood based classification of nodes in a network. In *KDD*. ACM, 2016.

[25] E. Ohlsson. Sequential poisson sampling. *J. Official Statistics*, 14(2):149–162, 1998.

[26] M. Osborne. *Facebook Reach and Frequency Buying*, October 2014. http://citizennet.com/blog/2014/10/01/facebook-reach-and-frequency-buying/.

[27] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

[28] B. Rosén. Asymptotic theory for successive sampling with varying probabilities without replacement, I. *The Annals of Mathematical Statistics*, 43(2):373–397, 1972.

[29] B. Rosén. Asymptotic theory for order sampling. *J. Statistical Planning and Inference*, 62(2):135–158, 1997.

[30] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513 – 523, 1988.

[31] J.S. Vitter. Random sampling with a reservoir. *ACM Trans. Math. Softw.*, 11(1):37–57, 1985.