Bayesian Sketches for Volume Estimation in Data Streams

Francesco Da Dalt
ETH Zürich
Zürich, Switzerland
fdadalt@student.ethz.ch

Simon Scherrer
ETH Zürich
Zürich, Switzerland
simon.scherrer@inf.ethz.ch

Adrian Perrig
ETH Zürich
Zürich, Switzerland
adrian.perrig@inf.ethz.ch

ABSTRACT

Given large data streams of items, each attributable to a certain key and possessing a certain volume, the aggregate volume associated with a key is difficult to estimate in a way that is both efficient and accurate. On the one hand, exact counting with dedicated counters incurs unacceptable overhead during stream processing. On the other hand, sketch algorithms, i.e., approximate-counting techniques that share counters among keys, have suffered from a trade-off between accuracy and query efficiency: Classic sketch algorithms allow to compute rough estimates in an efficient way, whereas more recent proposals yield highly accurate estimates at the cost of greatly increased computation time.

In this work, we propose three sketch algorithms that overcome this trade-off, computing highly accurate estimates with lightweight procedures. To reconcile these desiderata, we employ novel estimation methods that rely on Bayesian probability theory, counter-cardinality information, and basic machine-learning techniques. The combination of these techniques enables highly accurate estimates, which we demonstrate by both a theoretical worst-case analysis and an experimental evaluation. Concretely, our sketches allow to efficiently produce volume estimates with an average relative error of < 4%, which previous methods could only achieve with computations that are several orders of magnitude more expensive.

PVLDB Reference Format:
Francesco Da Dalt, Simon Scherrer, and Adrian Perrig. Bayesian Sketches for Volume Estimation in Data Streams. PVLDB, 14(1): XXX-XXX, 2023.
doi:XX.XX/XXX.XX

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://www.dropbox.com/s/u35xeoxwq0jglwr/bayes-sketch-revised.zip?dl=0.

1 INTRODUCTION

The analysis of data streams is a key component in numerous applications, enabling functionality as diverse as topic mining from text streams [14, 24], traffic monitoring and policing in the Internet [4, 15, 20, 30, 34], data aggregation in sensor networks [25], buffer dimensioning in VoIP networks [33, 36], and forecasting from time-series data in financial markets [1, 5]. Such stream processing includes a single iteration over a data stream, which corresponds to a sequence of *items*, each assignable to a certain *key* and possessing a certain *volume*. *Estimating the total volume associated to certain keys* (also: per-key aggregation) is an essential task, which, however, is often time-critical and therefore challenging.

Table 1: Comparison of sketches regarding accuracy, computation time and memory needed for queries (for a synthetic trace with $10k$ keys and Poisson-distributed key volumes, 6400 counters). CCB-Sketch is our algorithm. Seq-Sketch [21] is omitted as it performs worse than the PR-Sketch.

Algorithm	Rel. Error	Time [ms]	Memory [MB]
CM-Sketch [11]	99.092	10.1	27.5
C-Sketch [13]	1.264	20.1	27.5
CCB-Sketch	0.031	9.9	27.5
PR-Sketch [31]	0.024	$\sim 1.2 \cdot 10^6$	7260.1

Because of such high-speed requirements, a stream-processing algorithm must operate exclusively in low-latency memory (e.g., SRAM) to generate its analysis result. The limited availability of such low-latency memory prohibits the naive approach of keeping a counter per key and adjusting the counter whenever encountering the corresponding key, as data streams may contain billions of distinct keys. This requirement of processing efficiency has led to the development and usage of *sketching techniques*, which share counters among multiple keys and reconstruct key volumes from this compressed data structure [11, 13, 15, 21, 26, 27, 30, 31, 35]. Alas, this compression naturally causes inaccuracy in the volume estimate. In fact, classic sketch algorithms like the Count-Min Sketch [13] have been shown to deliver high accuracy (e.g., relative errors below 10%) only with impractical memory consumption [21, 30]. Hence, classic sketch algorithms are subject to an undesirable trade-off between estimation accuracy and memory efficiency.

Recent research has made significant progress in mitigating this trade-off by designing advanced *query methods* of sketch algorithms, i.e., the methods for computing a volume estimate from a synopsis. In particular, Seq-Sketch [21] employs a compressed-sensing approach, and PR-Sketch [31] relies on solving a system of linear equations, enabling both algorithms to achieve nearly zero relative error given a compact data-stream synopsis.

However, the accuracy improvements of these recent proposals come at the cost of *query efficiency*, i.e., the time and space complexity of the query methods: Both Seq-Sketch and PR-Sketch solve regularized optimization problems with potentially millions of variables, which introduces considerable complexity even with state-of-the-art numerical solvers (cf. Table 1). However, for example, high-frequency trading must be highly responsive to changes in transaction streams [3], DDoS defense systems must identify and block large flows as fast as possible to avert damage from other flows [30], and data mining from sensor-network streams relies

Proceedings of the VLDB Endowment, Vol. 14, No. 1 ISSN 2150-8097.
doi:XX.XX/XXX.XX

on tight feedback loops for efficient operation of machinery or vehicles [12]. For these usecases, the complex query operations of contemporary high-accuracy sketches are a significant impediment.

In this paper, we tackle this problem of high-accuracy sketches by proposing three sketching techniques that significantly reduce the computational cost of query execution and exhibit high accuracy both in theory and practice. Each of these sketches embodies a distinct query technique, each replacing the complex optimization-based queries of previous high-accuracy sketches with a handful of closed-form evaluations. These closed-form evaluations result from an in-depth theoretical analysis of the volume-estimation problem. To be specific, our query techniques rely on Bayesian probability theory and on counter-cardinality information, respectively.

In our first approach, we leverage Bayesian probabilistic reasoning to derive the *Count-Bayesian Sketch* (CB-Sketch). More precisely, the CB-Sketch relies on the computation of an approximate maximum a-posteriori (MAP) estimate for the volume of desired keys given the collected stream data. This estimate is captured by a closed-form solution and can be computed very efficiently while being highly accurate in general. Moreover, the Bayesian concept of priors enables a sketch operator to encode previous knowledge about the stream, and helps to achieve even higher accuracy.

In our second approach, we design the *Cardinality-Count-Average Sketch* (CCA-Sketch), where queries take into account counter-cardinality information, i.e., the number of distinct keys that were mapped to any single counter. While counter-cardinality information has been heuristically employed in previous sketch proposals, we present the first rigorous theoretical quantification of the accuracy improvements achievable by such information. These insights allow to optimally combine our two approaches into the *Cardinality-Count-Bayesian Sketch* (CCB-Sketch), enabling queries that leverage both Bayesian techniques and cardinality information.

The theoretical guarantees presented in this paper are competitive with other query-efficient sketches. In addition to this theoretical analysis, we also undertake an empirical evaluation, showing that our proposed sketches outperform their competitor algorithms in relevant settings. In particular, our experiments indicate that improvements in estimation accuracy can be considerably enhanced by the use of informed priors in the Bayesian sketches (CB- and CCB-Sketch). Moreover, both a theoretical complexity analysis and empirical evaluation confirm that our sketches enable queries that are efficient regarding computation time and memory consumption.

Our paper presents the following contributions:

- **Bayesian sketching:** We discuss how to leverage Bayesian probability theory for sketching, enabling the derivation of both the CB-Sketch and the CCB-Sketch (§4). Both these sketches arise from closed-form approximations to MAP estimates. To the best of our knowledge, the application of MAP estimates to streaming algorithms is novel.
- **Cardinality-based sketching:** We provide an in-depth theoretical analysis of the interaction between cardinality and volume information in counters. The resulting insights allow to employ counter-cardinality information in the CCA-Sketch.
- **Reconciling accuracy and query efficiency:** We present three novel streaming algorithms (CB-Sketch, CCA-Sketch and CCB-Sketch), which differ from earlier sketches by combining high estimation accuracy with high query efficiency.

Table 2: Notation

Symbol	Description
$\mathcal{I}$	Set of distinct keys in the data stream
$e_i \in \mathcal{I} \times \mathbb{R}$	i-th item in the data stream
$f_i \in \mathcal{I}$	Key associated with item e_i
$q_i \in \mathbb{R}$	Volume associated with item e_i
$a \in \mathbb{R}^{\ell_0}$	Vector of total volumes of all keys in $\mathcal{I}$
$a_f \in \mathbb{R}$	Total volume, or *size*, associated with key $f \in \mathcal{I}$
$\hat{a}_f^X \in \mathbb{R}$	Estimate of a_f by algorithm X
$\ell_0 \in \mathbb{N}$	Number of distinct keys in the data stream
$\ell_1 \in \mathbb{N}$	Total stream volume (sum over a)
$\ell_2 \in \mathbb{N}$	Squared l^2 norm of key-volume vector a ($\lvert a\rvert_2^2$)
$d \in \mathbb{N}$	Number of counter arrays
$w \in \mathbb{N}$	Size of each counter array
$V \in \mathbb{R}^{d \times w}$	Set of counter arrays storing volume information
$C \in \mathbb{R}^{d \times w}$	Set of counter arrays storing cardinality information
$(H_1, \ldots, H_d)$	Independent hash functions, $H_g : \mathcal{I} \rightarrow \{1, \ldots, w\}$

We demonstrate the properties of our algorithms with a theoretical analysis (§5) and an experimental evaluation (§6).

2 BACKGROUND

2.1 Problem Definition

In per-key volume aggregation, a data stream corresponds to a sequence of items $(e_1, e_2, \ldots, e_M)$, where each item e_i is associated to a key f_i and has a certain volume q_i. f_i is a key in $\mathcal{I}$, $|\mathcal{I}| = \ell_0$, which is the set of all distinct keys present in the data stream, whereas q_i may be any real number, positive or negative (i.e., the turnstile model). The goal of per-key volume aggregation is to estimate the aggregate volume a_f of a key f, which is defined as

$$a_f = \sum \{q_i \mid \forall i \in [1, \ldots, M]. \, f_i = f\} \, .$$

For convenience, we define $a \in \mathbb{R}^{\ell_0}$ as the vector containing the values a_f for all $f \in \mathcal{I}$ as separate dimensions. Moreover, $\hat{a}_f^X$ denotes the estimate of a_f calculated by algorithm X.

To enable volume estimation, some sketch algorithms such as SeqSketch [21] and PR-Sketch [31] involve mechanisms to collect the set $\mathcal{I}$ of distinct keys in the stream (key tracking). We do not concern ourselves with this problem, and assume that $\mathcal{I}$ is known. However, we empirically show in Sections 6.6 and 6.7 that our algorithms remain highly accurate under incomplete key tracking.

2.2 Trade-Offs in Previous Approaches

Traditionally, per-key volume estimation in the context of sketching techniques, i.e., the computation of $\hat{a}_f^X$, has been performed by extracting information related to key f from a data-stream synopsis, followed by some simple aggregation operation yielding $\hat{a}_f^X$. Since the estimate of each per-key volume is thus computed individually, we henceforth refer to these classic techniques as *local sketches*. The *Count Sketch* (henceforth: C-Sketch) and the *Count-Min Sketch* (henceforth: CM-Sketch) both fall into this category.

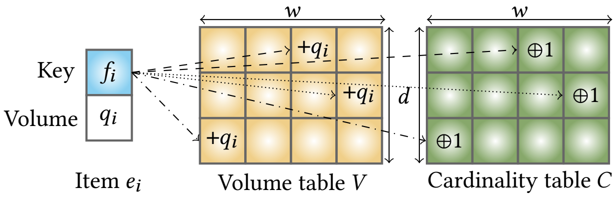


Figure 1: Illustration of data structure and update algorithm. Note that the cardinality table C may also be populated from the key set $\mathcal{I}$ at query time (retroactively).

Recently, however, sketching techniques have begun to compute estimates for all keys (i.e., the total key-volume vector a) simultaneously rather than one estimate at a time. We henceforth refer to these approaches as *global sketches*. Among these global sketches, the PR-Sketch [31] and Seq-Sketch [21] represent the most recent examples. Such global sketches have been shown to trade off unprecedented accuracy for greatly increased time and memory cost, as is illustrated by Table 1, which contains results from an experimental comparison regarding accuracy (i.e., average relative error) as well as computation time and peak memory consumption at query time. In this paper, we propose three algorithms that fill the gap between these two types of sketches, i.e., providing accuracy close to global sketches, while keeping the efficient query execution time of local sketches. We are able to reconcile these desiderata by deriving compact closed-form solutions for estimates based on Bayesian probability theory, which also allow for the injection of an accuracy-boosting prior into the estimation process.

3 DATA STRUCTURE AND ALGORITHMS

In this section, we provide a first description of the CB-Sketch, the CCA-Sketch and the CCB-Sketch. This description involves both the data structure used in the sketches (§3.1) and the algorithmic procedures that operate on the data structure (§3.2).

3.1 Data Structure

In sketching algorithms, the data structure stores all information necessary to answer key-volume queries. All three of our sketches are based on the same data structure, which consists of multiple counter arrays and is similar to the data structures used by C-Sketch [13] and CM-Sketch [11] (depicted in Figure 1). To be specific, the data structure contains a volume table $V \in \mathbb{R}^{d \times w}$, where the rows correspond to d counter arrays, each containing w counters. The d counter arrays are associated with d independent hash functions $(H_1, \ldots, H_d)$. In addition, the data structure also contains a cardinality table C, which has the same size and is associated with the same hash functions, but stores *cardinality information*, i.e., $C[i, j]$ contains the number of distinct keys mapped to counter j in counter array i: $C[i, j] = |\{f \in \mathcal{I} \mid H_i(f) = j\}|$. Furthermore, the data structure also keeps track of the total stream volume ℓ_1.

3.2 Algorithms

We separate the basic functionality of a sketch algorithm into two parts: The *update*, which modifies the data structure based on the incoming stream items e_i, and the *query* which returns a volume estimate $\hat{a}_f$ of a_f for a key $f \in \mathcal{I}$ based on the data structure.

The *update* algorithm in our sketches closely follows the update procedures in previous sketches [11, 13, 30, 31] and is briefly presented in Section 3.2.1.

In contrast, the *query* procedures of our sketches employ novel techniques. In particular, the CB-Sketch (Section 3.2.2) is inspired by Bayesian probabilistic reasoning: Its query technique arises from a stochastic optimization problem which approximates the arg max of the posterior distribution of the key-associated volume, given the data structure. In other words, the CB-Sketch performs an approximate maximum a-posteriori (MAP) estimate of any queried key, given the information collected about the observed stream.

The CCA-Sketch, which is presented in Section 3.2.3, relies on cardinality information instead of Bayesian techniques. The insights from the CCA-Sketch allow to optimally enrich the CB-Sketch with cardinality information, which produces the CCB-Sketch that relies on both Bayesian reasoning and cardinality information (presented in Section 3.2.4).

3.2.1 *Data Structure Update.*

With streaming algorithms targeting applications such as traffic supervision on network links, the *update* procedure must be extremely lightweight and time-efficient in order to reduce forwarding overhead and keep up with the item arrival rate. To that end, we employ an update procedure which is an extension of a commonly used counter-update scheme [13, 31]: For each incoming item e_i, d counter arrays are updated by adding item volume q_i to each counter to which f_i is hashed.

Regarding the total number ℓ_0 of distinct keys and the cardinality table C, this paper does not propose novel methods for estimating these quantities. This omission is conscious, as the best method of deriving ℓ_0 and C depends on the context of the stream processing. Several methods [16, 21, 30–32] have been proposed to obtain exact or approximate values of ℓ_0 and C.

3.2.2 *CB-Sketch.*

In the CB-Sketch, processing a query amounts to the computation of an approximate MAP estimate of the key volume given the information data structure. As we will justify in Section 4.1, this estimate is given as follows:

$$\hat{a}_f^{CB} = \frac{\mu_p^{CB} \cdot (\chi_p^{CB})^{-1} + w \cdot \sum_{i=1}^{d} V[i, H_i(f)] - d \cdot \ell_1}{(\chi_p^{CB})^{-1} + d \cdot (w - 1)} \tag{1}$$

where μ_p^{CB} indicates the prior mean on a_f and χ_p is linearly proportional to the prior variance on a_f. Note that ℓ_0 and C are not used by the CB-Sketch and can thus be omitted.

A rigorous derivation of the approximate MAP estimate together with an intuitive interpretation is provided in Section 4.1, whereas Section 5.1 analyzes the error bounds offered by the CB-Sketch.

3.2.3 *CCA-Sketch.*

The CCA-Sketch is inspired by considerations to combine volume and cardinality counter information in a simple algorithm to improve aggregate volume sketch accuracy. Intuitively, the volume of a key can be more accurately derived from the volume in an associated counter if the number of keys mapped to that counter is considered, given that such cardinality may vary substantially across counters. To incorporate cardinality information, the volume estimate can be based on the *ratio* between the volume and the cardinality of counters associated with a key. Such a ratio-based approach has been employed by LOFT [30], although in the context of top-k detection. To be viable for the more general problem of key-volume estimation, the CCA-Sketch departs from the LOFT algorithm in several respects. Most importantly, the CCA-Sketch applies an affine transformation to the volume-cardinality ratios. This affine transformation has been chosen based on a theoretical analysis of the statistical behavior of the volume-cardinality ratios, which is presented in Section 5.2. Concretely, the CCA-Sketch computes a volume estimate for key f as follows:

$$\hat{a}_f^{CCA} = \frac{(\ell_0 + w - 1) \cdot \sum_{i=1}^{d} \frac{V[i,H_i(f)]}{C[i,H_i(f)]} - d \cdot \ell_1}{d \cdot (w - 1)} \tag{2}$$

Despite the CCA-Sketch not stemming from a stochastic optimization problem, it is related to the CB-Sketch for subtle reasons, which will be discussed in Section 5.2.

3.2.4 *CCB-Sketch.*

So far, we have demonstrated how volume sketching can be extended with Bayesian reasoning (in the CB-Sketch) and with cardinality information (in the CCA-Sketch). Interestingly, these techniques can be combined, yielding the CCB-Sketch with the following volume estimate for a_f (cf. Section 4.2):

$$\hat{a}_f^{CCB} = \frac{\frac{\mu_p^{CCB}}{\chi_p^{CCB}} + (\ell_0 - 1) \cdot \sum_{i=1}^{d} \frac{V[i,H_i(f)]}{C[i,H_i(f)]-1} - d \cdot \ell_1}{(\chi_p^{CCB})^{-1} + \sum_{i=1}^{d} \frac{\ell_0 - C[i,H_i(f)]}{C[i,H_i(f)]-1}} \tag{3}$$

where μ_p^{CCB} and χ_p^{CCB} are the prior parameters relating to mean and variance, respectively. Same as in CB-Sketch, μ_p^{CCB} and χ_p^{CCB} can be used to add an independently selected bias to the final estimate. Said briefly, the CCB-Sketch is a direct upgrade compared to the CB-Sketch, in which the actual counter cardinality C is approximated with an expected value.

4 BAYESIAN SKETCHING

As the main contribution, this paper presents a new perspective on designing streaming algorithms, which builds on Bayesian statistics and machine learning. More concretely, we model the data structure, denoted as D, and the volume a_f of key f as random variables. Both RVs are clearly not independent and are linked by a non-trivial probability distribution. A query for an estimate of a_f can hence be refactored into the following optimization problem:

$$\begin{aligned}\hat{a}_f &= \text{argmax}_\alpha \mathbb{P}[a_f = \alpha \mid D] = \text{argmax}_\alpha \frac{\mathbb{P}[D \mid a_f = \alpha]\mathbb{P}[a_f = \alpha]}{\mathbb{P}[D]} \\ &= \text{argmax}_\alpha \log(\mathbb{P}[D \mid a_f = \alpha]) + \log(\mathbb{P}[a_f = \alpha])\end{aligned} \tag{4}$$

In other words, we model a point query of key f on a sketch data structure as a maximum-a-posteriori (MAP) estimate of a_f.

The advantages of this model are twofold. First, optimization and inference is a well understood problem for which many algorithms exist. Secondly, MAP estimates allow to insert some prior information into the estimate, i.e., $\mathbb{P}[a_f = \alpha]$. In general, priors serve to nudge estimates towards more likely values and to make predictions of extreme values unlikely.

In most scenarios, sketch operators have knowledge of some moments of the key sizes in the data stream, while their exact distribution may be unknown. A limited amount of such prior information is usually available for three reasons. First, sketch operators plausibly have domain-specific knowledge about the analyzed data streams, e.g., mean flow sizes in network settings, or typical measurements in sensor data streams. Second, even if such domain knowledge is initially unavailable, it can be practically gathered over time because streaming algorithms typically process large streams in segments, e.g., sketches in network settings handle 1-2 seconds of traffic before they are reset [30]. Hence, information collected from a previous stream segment can be used as surrogate prior information for the current stream segment. The viability of this prior information is then determined by the volatility of the data-stream statistics. Third, previous research has proposed specialized sketches that can estimate important moments [22] of a data stream and can hence yield information that can be used as a priors.

The MAP estimation problem in Equation 4 can be tackled in numerous ways, which vary in design decisions regarding the data structure, the prior distribution and the optimization method. In this paper, we present how the MAP estimate can be approximately solved for the case where the data structure is the classic data structure from Section 3, the prior distribution is a normal distribution, and the optimization method is a closed-form solution.

4.1 Constructing the CB-Sketch

The model for the CB-Sketch assumes the prior

$$\mathbb{P}[a_f = \alpha] = \mathcal{N}\left(\alpha;\ \mu_p^{CB},\ (\sigma_p^{CB})^2\right) \tag{5}$$

Furthermore, $\mathbb{P}[D \mid a_f = \alpha]$ is concretized as the probability that the volume counters in D associated to f attain their respectively stored value. Formally, this concretization results in:

$$\hat{a}_f^{CB} = \text{argmax}_\alpha \sum_{i=1}^{d} \log(\mathbb{P}[V_i = v_i \mid a_f = \alpha]) + \log(\mathbb{P}[a_f = \alpha]) \tag{6}$$

where $v_i := V[i,\ H_i(f)]$ and V_i is the RV associated to the i-th volume counter of f in the data structure.

Next, the RV V_i can be expressed as the sum over all item volumes that hash to a counter ($\mathbb{I}$ being the indicator function):

$$V_i = a_f + \sum_{g \in \mathcal{I}.\ g \neq f} \mathbb{I}_{\{H_i(f)=H_i(g)\}} a_g \tag{7}$$

Assuming that $\ell_0 w^{-1}$ is sufficiently large and that the total key volumes are i.i.d. samples of some distribution, the CLT yields the approximation:

$$V_i \mathrel{\dot\sim} \mathcal{N}\left(a_f + \frac{\ell_1 - a_f}{w},\ \frac{w-1}{w^2}\left(\ell_2 - a_f^2\right)\right) \tag{8}$$

By inserting Equation 8 into Equation 6, we obtain:

$$\begin{aligned}\hat{a}_f^{CB} \approx \text{argmax}_\alpha &\sum_{i=1}^{d} -\frac{\log\left(\ell_2 - \alpha^2\right)}{2} + \\ &\sum_{i=1}^{d} -\frac{\left(v_i - \alpha - \frac{\ell_1 - \alpha}{w}\right)^2 \cdot w^2}{2(w-1)\left(\ell_2 - \alpha^2\right)} - \frac{\left(\alpha - \mu_p^{CB}\right)^2}{2(\sigma_p^{CB})^2}\end{aligned} \tag{9}$$

In this form, the problem does not have a closed-form solution. Hence, we require a simplified surrogate problem that still yields a reasonable solution. To that end, we note that the logarithmic term in Equation 9 is dominated almost everywhere by the other terms of Equation 9, i.e., it can influence the solution to the optimization problem by at most a factor proportional to $O(w^{-1})$. It is hence reasonable to drop the logarithmic term from Equation 9 to get a tractable surrogate optimization problem, and we write:

$$\hat{a}_f^{CB} \approx \text{argmin}_\alpha \left(\ell_2 - \alpha^2\right)^{-1} \left(\sum_{i=1}^{d} \left(v_i - \alpha - \frac{\ell_1 - \alpha}{w}\right)^2 + \frac{(w-1)\left(\ell_2 - \alpha^2\right)\left(\alpha - \mu_p^{CB}\right)^2}{w^2 \cdot (\sigma_p^{CB})^2}\right) \tag{10}$$

In typical cases, we can assume $(\ell_2 - \alpha^2)$ to be approximately relatively constant for reasonable instances of α due to the assumption of ℓ_0 being very large. With that knowledge, we discard the leading factor $(\ell_2 - \alpha^2)^{-1}$ in the RHS of Equation 10 and perform a translation that simplifies the optimization problem:

$$(\sigma_p^{CB})^2 = \chi_p^{CB}\left(\ell_2 - \alpha^2\right) \implies \lim_{\ell_0 \to \infty} \hat{a}_f^{CB} \approx \text{argmin}_\alpha \frac{(w-1)\left(\alpha - \mu_p^{CB}\right)^2}{w^2 \cdot \chi_p^{CB}} + \sum_{i=1}^{d} \left(v_i - \alpha - \frac{\ell_1 - \alpha}{w}\right)^2 \tag{11}$$

Solving the optimization problem amounts to taking the derivative with respect to α and setting it to zero, which produces the CB-Sketch query:

$$\lim_{\ell_0 \to \infty} \hat{a}_f^{CB} \approx \frac{\mu_p^{CB}(\chi_p^{CB})^{-1} + w\left(\sum_{i=1}^{d} v_i\right) - d \cdot \ell_1}{(\chi_p^{CB})^{-1} + d(w-1)} \tag{12}$$

Regarding this derivation, two issues must be noted. Firstly, Equation 12 does not compute the exact MAP estimate, but instead an asymptotic approximation. Such an approximation is sub-optimal but necessary due to intractability of the exact problem. Secondly, χ_p^{CB} builds on $(\ell_2 - \alpha^2)$, which is unknown. In practice, however, it is still possible to estimate $(\ell_2 - \alpha^2)$ and to subsequently choose a matching χ_p^{CB}. Alternatively, an approach inspired from machine learning could consist of *learning* χ_p^{CB} by grid search or more sophisticated search schemes on χ_p^{CB}. While the CB-Sketch is thus based on a number of approximations, our empirical evaluation in Section 6 shows that the CB-Sketch yields highly accurate estimates, confirming the validity of the approximations..

Since the CB-Sketch arises from a solution to an optimization problem, interpreting the algorithm is non-trivial. Assuming an uninformed prior $\chi_p^{CB} = \infty$, Equation 12 is best understood the following way: It (i) uses ℓ_1 and w to formulate a "null hypothesis" regarding how big v_i would be in expectation if $a_f = 0$ (namely ℓ_1/w because the expected value of the counter is $(\ell_1 - a_f)/w + a_f$), (ii) computes the difference of the hypothesis to the actual counter value v_i, and (iii) averages these differences across the d counter arrays in order to estimate a_f. The denominator accounts for correlation between the volume counters and ℓ_1, and normalizes the numerator such that it yields an asymptotically bias-free estimate of a_f, as is shown in Section 5.1. The prior parameters μ_p^{CB} and χ_p^{CB} can add an arbitrarily strong bias to the prediction.

By relying on the difference between the hypothetical and the actual stream volume, the CB-Sketch possesses an interesting parallel to the C-Sketch [11]. The C-Sketch makes use of a key-specific signed multiplication step which causes counter values, multiplied by the sign of some key f, in expectation, to be linearly correlated to a_f and equal to 0 under the null hypothesis. Hence both C- and CB-Sketch base their estimates of a_f on the difference between an expected null hypothesis (0 and ℓ_1/w respectively) and observed data. These observations help to explain why the performance guarantees of the C-Sketch and the CB-Sketch are similar (cf. §5.1) as both sketches do conceptually the same thing and only differ in how the "null hypothesis" is found (hash-based signed multiplication and measurement of ℓ_1 respectively).

4.2 Constructing the CCB-Sketch

Same as with the CB-Sketch, we model a prior distribution over the key volumes as:

$$\mathbb{P}[a_f = \alpha] = \mathcal{N}\left(\alpha;\ \mu_p^{CCB},\ (\sigma_p^{CCB})^2\right) \tag{13}$$

However, in contrast to the preceding subsection, $\mathbb{P}[D \mid a_f = \alpha]$ indicates the joint probability of volume counters and cardinality counters attaining their respective value, assuming key f has volume α. Since counter cardinality and key volumes are independent from each other, we can move the probability of the cardinality counters attaining their respective values into the conditional:

$$\hat{a}_f^{CCB} = \text{argmax}_\alpha \sum_{i=1}^{d} \log(\mathbb{P}[V_i = v_i \mid a_f = \alpha,\ C_i = c_i]) + \log(\mathbb{P}[a_f = \alpha]) \tag{14}$$

where we abbreviate $c_i := C[i,\ H_i(f)]$ and define C_i as the RV associated with c_i.

The derivation of the approximate solution of Equation 14 is very similar to the derivation of Equation 12 and is omitted from this section to avoid repetition. The derivation of the closed-form estimate in the CCB-Sketch differs from the CB-Sketch in two high-level respects. First, the distribution of V_i conditioned on $a_f = \alpha$ and $C_i = c_i$ is modeled as an RV, which is defined as a weighted sum of entries from a multidimensional hypergeometric distribution, which is then approximated by a normal distribution. This hypergeometric distribution differs from the sum of weighted Bernoulli RVs that were used for the CB-Sketch, and stems from the conditioning on $C_i = c_i$. Correctly modeling the influence of this conditional $C_i = c_i$ is key to obtaining a superior final estimate. The second difference between derivations of the CB-Sketch and CCB-Sketch estimates concerns the prior variance, i.e., χ_p^{CB} and χ_p^{CCB}. For the CCB-Sketch, this prior variance is:

$$(\sigma_p^{CCB})^2 = \chi_p^{CCB} \frac{(\ell_0 - 1)\left(\ell_2 - \alpha^2\right) - (\ell_1 - \alpha)^2}{\ell_0 - 2} \tag{15}$$

which is a less trivial expression compared to χ_p^{CB} in Equation 11. Same as with the CB-Sketch, we require χ_p^{CCB} in order to remove the unknown term $\ell_2 - \alpha^2$ from the optimization. The prior variance is different for the two sketches because their probabilistic models yield different terms to be optimized, requiring different mathematical simplifications in order to result in a useful final estimate. For the CCB-Sketch, the approximate solution to Equation 14 is:

$$\lim_{\ell_0\to\infty} \hat{a}_f^{CCB} \approx \frac{\mu_p^{CCB}(\chi_p^{CCB})^{-1} + (\ell_0 - 1)\sum_{i=1}^{d} \frac{v_i}{c_i-1} - d\cdot\ell_1}{(\chi_p^{CCB})^{-1} + \sum_{i=1}^{d} \frac{\ell_0 - c_i}{c_i-1}} \tag{16}$$

As with the CB-Sketch, the prior parameters may be learned or estimated. Moreover, the CCB-Sketch only approximately solves Equation 14, but the empirical performance of the sketch confirms the validity of these approximations.

Intuitively, the term $(\ell_0 - 1)\sum_{i=1}^{d} \frac{v_i}{c_i-1}$ has an elegant interpretation analogous to a term in Equation 12. In fact, this term estimates what $d\cdot\ell_1$ should have been if key f did not exist: The sub-term $v_i(c_i-1)^{-1}$ computes the average key volume in counter i, assuming the volume of that counter is attributed to only $c_i - 1$ keys, one fewer than the true number of keys c_i, which can be interpreted as non-existence of f. Those estimates are then multiplied by $(\ell_0 - 1)$, i.e., again one fewer than the number of distinct keys, in order to predict what ℓ_1 should have been. Hence, the numerator of Equation 16 represents d times the difference of what ℓ_1 is and what it should have been if key f did not exist, assuming the prior is uninformed. The denominator can be interpreted as extracting the most likely estimate of a_f from the numerator.

Both the CB-Sketch and the CCB-Sketch use the crucial information that key f can be assigned to counters with certainty; hence, the value of these counters should be deflected by a_f compared to their expected value if f did not exist. In fact, the similarities can also be shown quantitatively, as we observe that the counter-array width w in the numerator of the CB-Sketch estimate is close to the term $(\ell_0 - 1)c_i^{-1}$ in the numerator of the CCB-Sketch estimate. The same kind of relation can be observed in the denominator between $d(w-1)$ and $\sum_{i=1}^{d} \frac{\ell_0 - c_i}{c_i-1}$. Informally, the CCB-Sketch can thus be seen as a direct upgrade of the CB-Sketch, as it uses the same general recipe for estimating a_f, while substituting some information-deficient terms involving w with more precise terms based on measurements of counter cardinality c_i.

5 THEORETICAL ANALYSIS

In this section, we present and prove theorems about the theoretical performance of the CB-Sketch and CCA-Sketch, both in terms of worst-case error bounds and execution complexity.

5.1 CB-Sketch Analysis

This section presents the result of a theoretical analysis of the accuracy of the CB-Sketch, followed by comparisons to the C-Sketch and CM-Sketch. We also present a runtime analysis.

Theorem 1. *Let a_f the aggregate volume of key $f \in \mathcal{I}$, and $\hat{a}_f^{CB}$ be the estimate according to Equation 12 with a uniform prior $\chi_p^{CB} = \infty$. In the limit $\ell_0 w^{-1} \to \infty$, assuming that $\max_{g\neq f} |a_g|\kappa_f \to 0$, where*

$$\kappa_f = \sqrt{\left(\ell_2 - a_f^2\right)^{-1}(w-1)}$$

the accuracy of this estimate is bounded by:

$$\mathbb{P}\left[|\hat{a}_f^{CB} - a_f| \geq \epsilon\cdot\ell_1\right] \leq 2\exp\left(-\frac{\epsilon^2\cdot\ell_1^2\cdot d(w-1)}{2\cdot\left(\ell_2 - a_f^2\right)}\right) \tag{17}$$

Proof. Let f be any key in the set $\mathcal{I}$ and the prior on a_f be uninformed, i.e., $\chi_p^{CB} = \infty$. We abbreviate $v_i = V[i, H_i(f)]$. Since v_i is the only source of uncertainty in $\hat{a}_f^{CB}$, it is the only part required to be modeled stochastically. By definition of v_i, we know that

$$v_i = a_f + \sum_{g\in\mathcal{I},\ g\neq f} \mathbb{I}_{\{H_i(f)=H_i(g)\}}\, a_g$$

where $\mathbb{I}$ is an indicator variable and the only source of uncertainty in the model. By assumption of universal hashing, $\mathbb{I}_{\{H_i(f)=H_i(g)\}}$ are i.i.d. RVs that follow a Bernoulli distribution $Ber(w^{-1})$. The sum over $g\in\mathcal{I},\ g\neq f$ is a sum over $\ell_0 - 1$ random variables. Thus, by assumption, in the limit $\ell_0 w^{-1} \to \infty$ the CLT applies and:

$$\lim_{\ell_0 w^{-1}\to\infty} v_i \sim \mathcal{N}\left(a_f + \frac{\ell_1 - a_f}{w},\ \frac{w-1}{w^2}\left(\ell_2 - a_f^2\right)\right) \tag{18}$$

Therefore, in the limit, the estimate $\hat{a}_f^{CB}$ is distributed as:

$$\hat{a}_f^{CB} \sim \mathcal{N}\left(a_f,\ \frac{\ell_2 - a_f^2}{d(w-1)}\right) \tag{19}$$

For Equation 19, we used the fact that all hash functions are pairwise independent and hence $\forall i.\ v_i$ are i.i.d.. This shows in primis that for a uniform prior, in the limit, the CB-Sketch implements an unbiased estimator. To bound the estimation error $|\hat{a}_f^{CB} - a_f|$, we use the sub-gaussian concentration bound on the normal distribution:

$$\lim_{\ell_0 w^{-1}\to\infty} \mathbb{P}\left[|\hat{a}_f^{CB} - a_f| \geq \epsilon\cdot\ell_1\right] \leq 2\exp\left(-\frac{\epsilon^2\cdot\ell_1^2\cdot d(w-1)}{2\cdot\left(\ell_2 - a_f^2\right)}\right) \tag{20}$$

which concludes the proof. □

5.1.1 *Comparison to CM-Sketch.*

The CM-Sketch [13] has the following characteristics:

$$\mathbb{P}\left[|\hat{a}_f^{CM} - a_f| \geq \epsilon\cdot\ell_1\right] \leq \exp\left(-\frac{\epsilon\cdot d\cdot w}{e}\right) \tag{21}$$

At first sight, the CM-Sketch might seem to dominate the CB-Sketch because the linear ϵ reduces the exponential function in Equation 21 more quickly than the squared ϵ in Equation 17. However, the ratio of ℓ_1^2 to $(\ell_2 - a_f^2)$ in Equation 17 compensates this effect, as for practical scenarios, this term behaves linearly in the number of keys ℓ_0. Since $\ell_0 \gg \epsilon^{-1}$, the additional ϵ in the exponent of the CB-Sketch bound is counteracted. In practice, the CB-Sketch effectively dominates the CM-Sketch as ϵ^{-1} is sub-linear in ℓ_0.

5.1.2 *Comparison to C-Sketch.*

The performance guarantees of the C-Sketch [11] are given by:

$$\mathbb{P}\left[|\hat{a}_f^{C} - a_f| \geq \epsilon\cdot\ell_1\right] \leq \exp\left(-\frac{\epsilon^2\cdot\ell_1^2\cdot d\cdot w}{3\cdot\left(\ell_2 - a_f^2\right)}\right) \tag{22}$$

From this, we observe that the C-Sketch has strikingly similar performance guarantees compared to the CB-Sketch, which may also be superior depending on sketch dimensioning and stream characteristics. This partially superior accuracy may stem from (1) approximations made in the computation of the CB-Sketch estimate, and (2) a more sophisticated update procedure of the C-Sketch. Nevertheless, in our experimental evaluation (§6), we demonstrate that the CB-Sketch outperforms the C-Sketch in terms of accuracy in most scenarios.

5.1.3 Time Complexity. Equation 12 indicates that the time required to compute the estimate $\hat{a}_f^{CB}$ is in $O(d)$. Hence, the CB-Sketch has the same asymptotic query complexity as the CM-Sketch. Importantly, however, the C-Sketch query requires the computation of a median of a vector of length d. While this median can be computed in asymptotically linear time by the Quickselect algorithm [19], in practice this algorithm is more expensive than a single list iteration, and has poor worst-case performance. Moreover, we emphasize that the update procedure of the C-Sketch applies double the number of hash functions than the CB-Sketch, which constitutes a significant performance impediment [26].

5.2 CCA-Sketch Analysis

The error analysis of the CCA-Sketch is considerably more involved compared to the CB-Sketch, mainly due to the introduction of cardinality information. However, although the CCA-Sketch is quite different from the CB-Sketch, the CCA-Sketch analysis uncovers an equivalence regarding accuracy between the sketches.

Theorem 2. *Let a_f be the aggregate volume of key $f \in \mathcal{I}$, and $\hat{a}_f^{CCA}$ be the estimate according to Equation 2. In the limit $\ell_0 w^{-1} \rightarrow \infty$, assuming that $\max_{g \neq f} |a_g| \kappa_f \rightarrow 0$, where*

$$\kappa_f = \sqrt{\left(\ell_2 - a_f^2\right)^{-1} (w-1)}$$

the accuracy of this estimate is bounded by:

$$\lim_{\ell_0 w^{-1} \rightarrow \infty} \mathbb{P}\left[|\hat{a}_f^{CCA} - a_f| \geq \epsilon \ell_1\right] \leq 2 \exp\left(-\frac{\epsilon^2 \cdot \ell_1^2 \cdot d(w-1)}{2\left(\ell_2 - a_f^2\right)}\right) \tag{23}$$

which equals the bound on $\hat{a}_f^{CB}$ from Theorem 1.

Proof. In the following, f is any key in the set $\mathcal{I}$, and we abbreviate $v_i = V[i, H_i(f)]$ and $c_i = C[i, H_i(f)]$. To analyze the CCA-Sketch, we require the distribution of v_i/c_i. To that end, we first observe that

$$v_i \mid (c_i = c) \sim [a_1, \ldots, a_{\ell_0}] \cdot \mathit{MDHGD}\left(c-1,\ \mathbf{1}_{\ell_0} - \vec{e}_f\right) + a_f \tag{24}$$

where $\mathit{MDHGD}(x,\ y)$ is a multidimensional hypergeometric distribution with x draws from the "bins" defined by the vector y, $\vec{e}_f$ is the f'th unit vector and $\mathbf{1}_{\ell_0}$ is a vector of 1s of length ℓ_0. By making use of the known moments of the *MDHG* distribution in Equation 24, we can derive

$$\begin{aligned} \mathbb{E}\left[\frac{v_i}{c_i} \mid (c_i = c)\right] &= \frac{1}{c}\mathbb{E}[v_i \mid (c_i = c)] = \frac{(c-1)\left(\ell_1 - a_f\right)}{c(\ell_0 - 1)} + \frac{a_f}{c} \\ \mathbb{V}\left[\frac{v_i}{c_i} \mid (c_i = c)\right] &= \frac{1}{c^2}\mathbb{V}[v_i \mid (c_i = c)] = \frac{(c-1)(\ell_0 - c)m}{c^2(\ell_0-1)^2(\ell_0-2)} \end{aligned} \tag{25}$$

with $m = (\ell_0 - 1)\left(\ell_2 - a_f^2\right) - \left(\ell_1 - a_f\right)^2$

which results from collapsing the matrix-vector multiplications of mean and variance in terms of ℓ_2 and ℓ_1. These moments enable expressing the distribution of $v_i \cdot c_i^{-1}$ as a mixture distribution of $v_i \cdot c^{-1}$, where c itself is distributed according to c_i which is an affine binomial distribution:

$$c_i \sim 1 + Bin\left(\ell_0 - 1, w^{-1}\right) \tag{26}$$

Thanks to the law of total variance and the law of total expectation, the moments of $v_i \cdot c_i^{-1}$ are given by

$$\begin{aligned} \mathbb{E}\left[\frac{v_i}{c_i}\right] &= \mathbb{E}_{c \sim c_i}\left[\frac{(c-1)\left(\ell_1 - a_f\right)}{c(\ell_0-1)} + \frac{a_f}{c}\right] \\ \mathbb{V}\left[\frac{v_i}{c_i}\right] &= \mathbb{E}_{c \sim c_i}\left[\frac{(c-1)(\ell_0-c)m}{c^2(\ell_0-1)^2(\ell_0-2)}\right] + \\ &\quad \mathbb{V}_{c \sim c_i}\left[\frac{(c-1)(\ell_1 - a_f)}{c(\ell_0-1)} + \frac{a_f}{c}\right] \end{aligned} \tag{27}$$

The expressions in Equations 27 have no exact closed-form solutions. Hence, we use an approximation which is inspired by a work on confidence intervals for ratios of binomial distributions [23]. This work was expanded upon to be applicable to powers of RVs that approximately follow a normal distribution. More precisely, our approximation scheme involves (1) a log-transformation on the ratio of RVs, which yield a difference of logarithms of RVs, and (2) a number of linear approximations to remove the logarithm operation. The result of applying this approximation scheme, whose error converges to 0 for $\ell_0 w^{-1} \rightarrow \infty$, to Equations 27 leads to the following result in the limit $\ell_0 w^{-1} \rightarrow \infty$:

$$\mathbb{E}\left[\frac{v_i}{c_i}\right] = \mu_U = \frac{(\ell_1 - a_f) + w \cdot a_f}{\ell_0 + w - 1} = \frac{\ell_1 + (w-1) \cdot a_f}{\ell_0 + w - 1} \tag{28}$$

$$\mathbb{V}\left[\frac{v_i}{c_i}\right] = \sigma_U^2 = -\frac{\begin{pmatrix} (w-1)\Big(-\left(2(\ell_0-1)^2 w + (\ell_0-1)^3 - w^2\right)m \\ -\left((\ell_0-2)w^2(\ell_1 - (\ell_0-1)a_f)^2\right)\Big) \end{pmatrix}}{(\ell_0-2)(\ell_0-1)(\ell_0+w-1)^4} \tag{29}$$

The bias-free CCA-Sketch estimate, as presented in Equation 2, follows from the solution of Equation 28.

In similar fashion, it is possible to show that $v_i \cdot c_i^{-1}$ is approximately normally distributed: After applying a log-transform on $v_i \cdot c_i^{-1}$ and a linear approximation to the random logarithmic terms, we observe that both v_i and c_i can be described as a sum of numerous independent RVs, where v_i and c_i share some co-variance. Hence, $\log(v_i \cdot c_i^{-1})$ is approximately normally distributed. Finally, since the the log-normal distribution of $v_i \cdot c_i^{-1}$ can be shown to converge to a normal distribution in the limit $\ell_0 w^{-1} \rightarrow \infty$, in the sense that the difference of the CDFs converges to zero, we conclude the analysis by stating that in the limit $\ell_0 w^{-1} \rightarrow \infty$:

$$\begin{aligned} \frac{v_i}{c_i} \sim \mathcal{N}\left(\mu_U, \sigma_U^2\right) \implies \hat{a}_f^{CCA} &\sim d^{-1}\mathcal{N}\left(d \cdot a_f, \frac{d \cdot \sigma_U^2(\ell_0 + w - 1)^2}{(w-1)^2}\right) \\ &= \mathcal{N}\left(a_f, \frac{\sigma_U^2(\ell_0 + w - 1)^2}{d(w-1)^2}\right) \end{aligned} \tag{30}$$

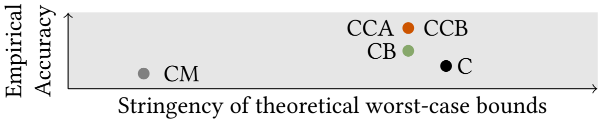


Figure 2: Accuracy characterization of sketch algorithms discussed in this paper. Note that the characterizations of the CB- and the CCB-Sketch relate to the *untrained* versions; the trained versions achieve higher average accuracy.

With the most important characteristics of $v_i c_i^{-1}$, we again apply a concentration bound on $\hat{a}_f^{CCA}$:

$$\lim_{\ell_0 w^{-1} \to \infty} \mathbb{P}\left[|\hat{a}_f^{CCA} - a_f| \geq \epsilon \ell_1\right] \leq 2 \exp\left(-\frac{\epsilon^2 \cdot \ell_1^2 \cdot d(w-1)^2}{2\sigma_U^2 (\ell_0 + w - 1)^2}\right) \tag{31}$$

By inserting Equation 29 into Equation 31 and taking the limit $\ell_0 w^{-1} \to \infty$ of the RHS, the bound behaves asymptotically like:

$$\lim_{\ell_0 w^{-1} \to \infty} \mathbb{P}\left[|\hat{a}_f^{CCA} - a_f| \geq \epsilon \ell_1\right] \leq 2 \exp\left(-\frac{\epsilon^2 \cdot \ell_1^2 \cdot d(w-1)}{2\left(\ell_2 - a_f^2\right)}\right) \tag{32}$$

which concludes the proof. □

Theorem 2 has two goals. First, the theorem shows that the CCA-Sketch shares worst-case accuracy guarantees with the CB-Sketch, and is hence superior to the CM-Sketch (cf. Section 5.1.1). Secondly, the results of the CCA-Sketch analysis may also apply to the CCB-Sketch, which itself has proven difficult to treat analytically. This similarity is plausible from both an algorithmic and an empirical perspective. From an algorithmic perspective, the estimate $\hat{a}_f^{CCB}$ revolves around $\sum_{i=1}^{d} v_i (c_i - 1)^{-1}$, while the estimate of the CCA-Sketch is based on $\sum_{i=1}^{d} v_i c_i^{-1}$, which are very similar quantities, especially if ℓ_0 is large. From an empirical perspective, the CCA-Sketch and the CCB-Sketch share similar performance characteristics, as is evidenced by empirical results in Section 6. Hence, we conjecture that the two sketches also share similar theoretical worst-case performance bounds.

Conjecture 3. *CCA-Sketch and CCB-Sketch share the same worst-case performance bounds.*

Importantly, while our proposed sketches may thus provide the same theoretical asymptotic worst-case guarantees, their effective accuracy in experiments differs considerably, with the cardinality-based sketches generally outperforming the CB-Sketch. Hence, cardinality information can improve accuracy in most scenarios, although these improvements are not visible in the bounds (cf. Figure 2).

Since the CCA-Sketch is equivalent to the CB-Sketch regarding its lower accuracy bounds, the comparison to the C-Sketch and CM-Sketch follows from Sections 5.1.1 and 5.1.2.

5.2.1 *Time Complexity.* Like the CB-Sketch, the query procedure of the CCA-Sketch requires time $O(d)$, which is also the asymptotic complexity of the C-Sketch and the CM-Sketch. However, the CCA-Sketch introduces a constant-factor runtime cost to obtain the cardinality information which its query relies on. More precisely, if such cardinality information is not yet collected in the update procedure (with cardinality estimators such as HyperLogLog [16]), the query procedure must reconstruct the cardinality counters by mapping all keys to their respective counters in all counter arrays, effectively requiring $\ell_0 \cdot d$ hash-function evaluations. However, this reconstruction work is independent of the number of performed volume estimates and is thus amortised if a large number of estimates is conducted. If the volume of every key is estimated, the query procedure requires exactly only one additional iteration over the data structure per estimate. Note that the CCB-Sketch shares the same complexity as the CCA-Sketch.

6 EXPERIMENTAL ANALYSIS

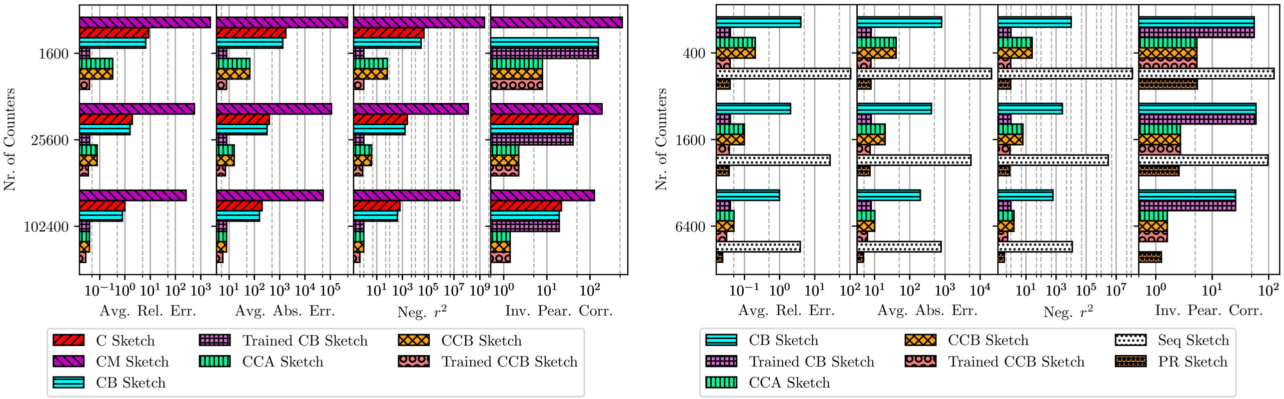
In this section, we complement the theoretical analysis from Section 5 with an experimental evaluation, which is valuable for several reasons. First, the preceding theoretical analysis focuses on guarantees at certain probability levels, which are not necessarily informative about the average accuracy of sketches in realistic settings. Second, the theoretical analysis provides little insight into the performance of the CCB-Sketch, although Section 5.2 alluded that the CCB-Sketch is a direct upgrade to the CB-Sketch. Third, this section demonstrates how to improve the performance of the CB-Sketch and CCB-Sketch by injecting informative priors into the estimation process. Fourth, none of the sketches presented in this paper have yet been compared to global sketches such as the PR-Sketch [31] and Seq-Sketch [21]. Both of these sketches are expected to provide higher accuracy compared to the other sketches mentioned in this paper, because these sketches reconstruct not single key sizes a_f at a time, but the entire vector a at once. Hence, these global sketches can capture the inter-dependency of key sizes in the reconstruction. However, global-sketching techniques can also be expected to come at the cost of considerably increased time and space requirements for key-query operations.

6.1 Evaluation Set-Up

All experiments presented in this section have been performed in a common C++ testbed [2] which fed the data stream to the data structures and computed error and performance statistics based on the various estimates $\hat{a}^X$ returned by the different sketches. All algorithms were implemented in C++, where the local sketches were all implemented manually, while Seq-Sketch made use of `Armadillo` [28, 29] and the compressive-sensing library `Kl1p` [17]. PR-Sketch was implemented with the help of `Armadillo` by computing the pseudo-inverse of the system matrix by means of sparse matrix SVD. Manual implementations were parallelized with `std::thread` and the other algorithms made use of `Armadillo`'s `LAPACK` powered parallelized linear algebra operations.

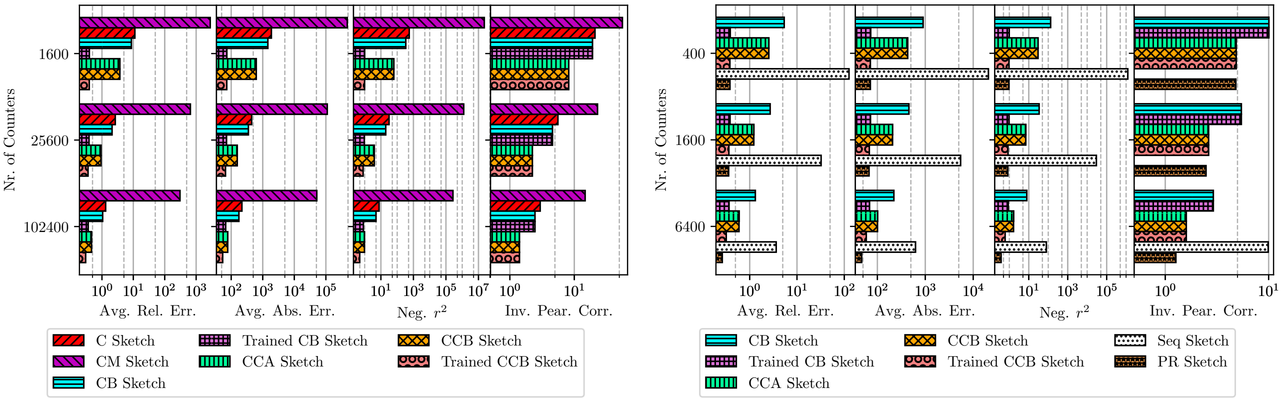
All algorithms compared in this section were given the same amount of information and memory. In the case of Seq-Sketch, this equalization implies that Seq-Sketch does not need to detect which keys are present in the data stream, which is the goal of a dedicated component in Seq-Sketch. Instead, Seq-Sketch can focus on reconstructing the known keys as well as possible.

In terms of evaluation metrics, we present performance and error statistics evaluated on both synthetic and real data-stream traces. We mainly use four statistics to compare the fitness of a sketch:



(a) Comparison to local sketches (Trace of $100k$ keys). **(b) Comparison to global sketches (Trace of $10k$ keys).**

Figure 3: Comparison based on Poisson trace.



(a) Comparison to local sketches ($100k$ keys). **(b) Comparison to global sketches ($10k$ keys).**

Figure 4: Comparison based on geometric trace.

- **Average relative error:** $\ell_0^{-1} \sum_{f \in \mathcal{I}} |\hat{a}_f^X - a_f| a_f^{-1}$
- **Average absolute error:** $\ell_0^{-1} \sum_{f \in \mathcal{I}} |\hat{a}_f^X - a_f|$
- **Negative r^2:** $1 - r^2$, where r^2 is the coefficient of determination
- **Inverted Pearson correlation:** Inverted sample Pearson correlation between $\hat{a}^X$ and a

Furthermore, we show data for experiments under different memory constraints. To that end, the *Nr. of Counters* denotes the total number of counters used for the data structure, i.e., $w \cdot d$.

Relevant for experiments relating to runtime and memory consumption of the various algorithms in Section 6.8, the machine on which the experiments were run is a 2 + 8 core Apple Silicon M1 Pro with 16 GB of memory.

6.2 Poisson Trace

In the first set of experiments, we compare the accuracy of our sketches to previous proposals on the basis of synthetic Poisson traces, i.e., traces in which the key sizes in the data stream were sampled i.i.d. from a distribution $100 + Poi(100)$.

For the untrained versions of the CB-Sketch and the CCB-Sketch, the priors are uninformed, i.e., $\chi_p^{CCB} = \chi_p^{CB} = \infty$. For the trained versions of CB-Sketch and the CCB-Sketch, we rely on a training trace which is synthesized identically to the test trace, both regarding length and statistics. Given this training trace, we use a simple log-scale grid search in order to find the best prior parameters χ_p^{CCB} and χ_p^{CB}, where the optimization goal was the *average absolute error*. We set the prior means μ_p^{CCB} and μ_p^{CB} to be the empirical mean on the training trace. Details on the cost of training are elaborated in Section 6.5.

6.2.1 Observations. Figure 3a shows that the CB-Sketch is considerably superior to the CM-Sketch, which can be expected from the theoretical analysis in Section 5.1.1. More surprisingly, the CB-Sketch also narrowly outperforms C-Sketch, although the theoretical guarantees of the CB-Sketch are slightly less strong than those of the C-Sketch (cf. Section 5.1.2). Furthermore, the trained and untrained sketches clearly differ in the estimation errors for both the CB-Sketch and the CCB-Sketch, where the trained versions clearly outperform the untrained versions. This error-reducing effect of training aligns with the expectations based on the favorable distribution of the trace. We note that training the priors with respect to the *average relative error* also improves the *average absolute error* and the r^2 *score* while leaving the *correlation* unaffected. In fact, the correlation and ordering of $\hat{a}_f^X$ is guaranteed to be invariant to the prior in all cases.

With regards to the global sketches, we observe from Figure 3b that the PR-Sketch dominates both trained and untrained versions of the proposed sketches, although only by a minuscule margin for certain memory constraints. This high accuracy of the PR-Sketch is partially to be expected, especially for lax memory constraints since the PR-Sketch approaches the interpolation regime, where estimates inevitably lie within machine precision. Surprisingly, the Seq-Sketch is inferior to our proposed sketches in terms of errors and correlation. Finally, we observe that the CCB-Sketch and the CCA-Sketch are qualitatively indistinguishable regarding their accuracy, which supports Conjecture 3.

6.3 Geometric Trace

To explore the case in which the priors of our Bayesian sketches (i.e., normal distributions) poorly match the distribution of aggregated key-volumes, we perform the experiments in the same way as in the preceding section, except that the Poisson distribution is substituted by a Geometric distribution and the prior training is conducted with the negative r^2 score as optimization goal.

6.3.1 Observations. Figure 4a suggests that the proposed sketches dominate both the C-Sketch and CM-Sketch even though the prior distribution significantly differs from the actual distribution of key sizes. The likely reason for such high accuracy is that the geometric distribution has a short tail and hence normal approximations still work out sufficiently well. As is displayed in Figure 4b, the PR-Sketch dominates all proposed algorithms in this experiment, whereas the Seq-Sketch is again consistently inferior. Also, we once again observe qualitatively identical performance statistics for the CCB- and CCA-Sketch, yielding additional empirical evidence for Conjecture 3.

6.4 CAIDA Trace

To complement the synthetic-trace experiments in the preceding sections, we repeat the experiments with a CAIDA trace [10], which involves network-packet data from a commercial backbone link and contains roughly 1 million unique flows (i.e., keys). For the sake of keeping the scale of experiments manageable, we use a random subset of *100k* and *10k* keys to compare the proposed algorithms to local and global sketches, respectively. The priors are determined with a training trace which was sampled randomly from the whole CAIDA trace, and with average relative error as optimization goal.

6.4.1 Observations. Figure 5a shows that the proposed methods dominate the CM-Sketch for all choices of data-structure size, which is in line with the results of the theoretical analysis and the synthetic-trace experiments. In turn, our proposed sketches are dominated by the C-Sketch if uninformed priors are used. However, the trained methods dominate all local sketches in terms of *average relative* and *average absolute error* by a considerable margin.

Regarding global sketches, shown in Figure 5b, we again observe that the PR-Sketch outperforms the untrained sketches, whereas the Seq-Sketch is inferior to them. In contrast to previous experiments, however, the trained methods still manage to beat the PR-Sketch in terms of *average relative* and *absolute error*. This surprising result shows that the injection of a prior can vastly improve the accuracy of the sketch, in some cases even outperforming global sketches.

Comparing the proposed sketches among each other, we find that the CB-Sketch, CCA-Sketch and CCB-Sketch have very similar performances in this experiment, where the latter has marginally better performance compared to the former two, again supporting Conjecture 3

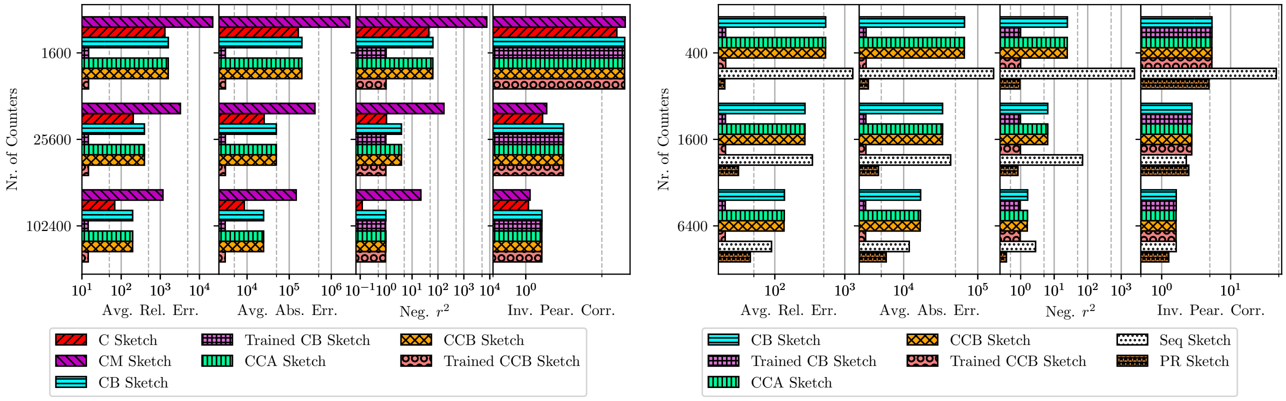
6.5 Training and Prior Tuning

From the previous sections, the value of learning suitable priors for the Bayesian sketches becomes apparent. To illustrate the trade-offs in tuning the prior parameters for the trainable sketches, we present Figure 6, which shows data collected on a Poisson trace. In terms of all error statistics, we observe the following behaviors: The CCB-Sketch yields strictly better performance compared to the CB-Sketch, no matter which common prior is chosen. Secondly, we note that the CCB-Sketch attains its optimum at higher values of χ_p^X compared to the CB-Sketch. This higher reliance of the CB-Sketch on narrow localization of the prior (i.e., low prior variance χ_p^X) is expected as the CCB-Sketch incorporates more information into its estimates compared to the CB-Sketch. Therefore, the CCB-Sketch is in general less reliant on prior knowledge. Thirdly, in all settings, the error function has a unique minimum, which is neither $\chi_p^X = 0$ nor $\chi_p^X = \infty$. Hence, χ_p^X can always be chosen such that the resulting estimates are better than both predicting the mean and predicting with uniform priors. The highest accuracy is thus achieved through an appropriate combination of prior knowledge and data stream information. In particular, when the prior mean is close to the average key size in the data stream, both the CCB-Sketch and the CB-Sketch allow a selection of χ_p^X such that the r^2 score is higher than 0, i.e., the estimation is better than simply taking the average key size as estimate. Lastly, we also see that $\mu_p^X = 0$ can be used effectively for error reduction, although Figure 6 suggests only slim gains for the CCB-Sketch.

In all experiments presented in this paper, the training method was a simple exponential grid search with 200 steps that minimized a given error statistic averaged over all training keys in the stream. Hence, the time required for training was roughly 200 times the time required for evaluating the sketch on all training keys in the data stream. This training cost thus depends on how dense the grid search is, and how costly the error-statistic computation is. Moreover, we note that the training of our sketches is only rarely performed (potentially only once for certain use-cases), and the cost of training can therefore be amortized over many sketch evaluations. Additionally, training is temporally and spatially detached from testing and can thus be scheduled at advantageous times on less critical hardware.

6.6 Noisy Cardinality Information

In practice, the full set of keys $\mathcal{I}$ (and thus the cardinality counters) cannot always be collected without error. Therefore, the effect of faulty information on the accuracy of the CCB-Sketch deserves further attention. To quantify this effect, we evaluate the accuracy of the CCB-Sketch on a Poisson trace for different *noise levels*,



(a) Comparison to local sketches (Subsample of *100k* keys).

(b) Comparison to global sketches (Subsample of *10k* keys).

Figure 5: Comparison based on CAIDA trace.

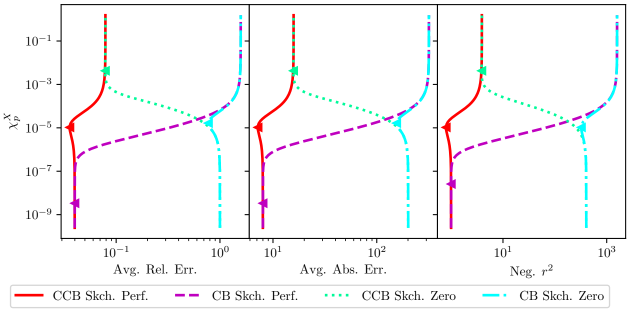


Figure 6: Statistics of trained sketches as a function of tuning parameter χ_p^X. "Zero" refers to the setting $\mu_p^X = 0$ and "Perf." refers to the case where μ_p^X is set to the training trace empirical mean. The triangles indicate the minima of the plotted curves.

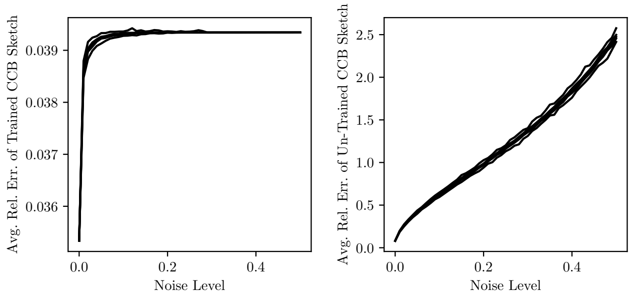


Figure 7: The curves plotted are the minimum, 25th, 50th and 75th percentile and maximum of average relative errors recorded across 50 simulations for different *noise levels*, i.e., the fraction of keys missing during the computation of the cardinality information. The y-axis in the left plot is truncated.

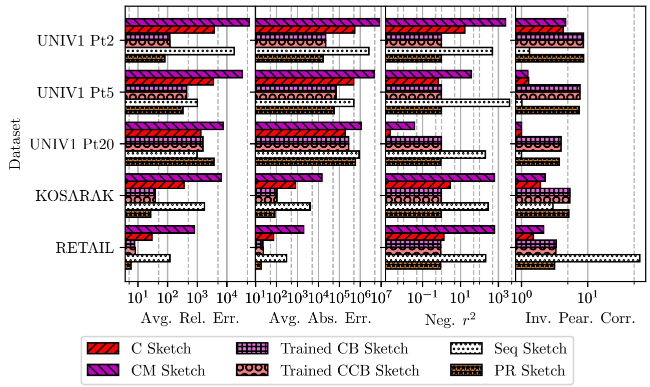


Figure 8: System simulations on various real datasets

i.e., random fractions of keys that are unknown during query execution. Figure 7 shows how the performance of the CCB-Sketch degrades with increasing noise level. However, we also observe that the trained version suffers a lot less from increasing noise levels because it can compensate for the poor data stream information by choosing an adequate prior (Note that the training was also conducted on a training trace equally noisy as the test trace). Given high noise levels, the prior becomes dominant in the trained CCB sketch, always making the sketch predict more or less the mean μ_p^{CCB}. However, modern key-tracking mechanisms achieve noise levels of lower than 5% [31], for which the CCB-Sketch still achieves high accuracy.

6.7 Stress-Testing Simulations

To further evaluate the CB- and CCB-Sketches in realistic settings, we perform simulations in an environment with three challenging properties. First, we use the UNIV1 [6, 8], KOSARAK [7], and RETAIL [9] datasets, which in parts heavily conflict with the normal assumptions of the Bayesian sketches. Second, to account for concept drift, we introduce temporal distance between training and testing: For KOSARAK and RETAIL, training and testing is done on the first and last 10% of the datasets, respectively; and for UNIV1, the first part of the trace is used for training, while testing is performed on 3 increasingly distant parts (2, 5 and 20). Third, realistic noise in cardinality information is induced by using a Bloom filter as key tracker (as in Seq-Sketch).

For UNIV1, RETAIL, and KOSARAK, the sample kurtosis of the total key volumes well exceeds 10k in all cases and visual tests suggest that the variances are not necessarily well defined, which makes learning priors difficult. While the largest single key volume in parts 1 and 2 of UNIV1 makes up about 5% of the total volume, in parts 5 and 20 the ratio of maximal key volume to total trace volume increases to 20% and 85% percent, respectively, which breaks asymptotic approximations in the proposed sketches.

6.7.1 Observations. Despite unfavorable trace characteristics, especially in KOSARAK, RETAIL, and UNIV Pt2, the learned priors give the CB- and CCB-Sketch a clear advantage over other local sketches, despite considerable temporal distance between training and testing. We also observe that the proposed sketches struggle with parts 5 and 20 of UNIV1. This is due to two factors: an increasing discrepancy between the training and testing statistics, and an increasingly extreme heaviness of the tail. It is worth noting that PR-Sketch also appears to suffer considerably from increasing heaviness of the tail, which would suggest the relevance of that distribution property. The performance difference between CB- and CCB-Sketch is minuscule and slightly favors the former over the latter. Noisy cardinality information plays a role, but the bigger factor, considering simulations from previous sections, seems to be the heavy tail of the key-volume distribution, which leads to misinterpretation of measurements under the normal assumption of the sketches. This effect is stronger for the CCB-Sketch, which is more receptive, and hence more exposed, to measurement data.

6.8 Time and Space Complexity

The goal of this work is to devise sketches that combine the accuracy of global sketches with the query efficiency of local sketches. While the preceding sections demonstrate that the trained Bayesian sketches are competitive with local sketches, we focus on the query efficiency of our sketches in this section.

To demonstrate the differences in space and time complexity of the various algorithms, Figure 9 shows data collected from an experiment with 5k keys in the data stream and compares the time and peak memory necessary to query every $\hat{a}_f^X$ for $f \in \mathcal{I}$. All local sketches have approximately the same peak memory usage since they all only require information about one flow f in order to estimate a_f. The computation of $\hat{a}_f^X$ does not incur substantial additional memory allocation in the case of local sketches since the computation only amounts to a form of weighted aggregation. Among the local sketches, only the C-Sketch stands out, as the computation of a median is more expensive than a simple cumulative aggregation over the array. Also note that the data structure itself consumes negligible memory compared to memory required for query evaluation; hence, the CCA-Sketch and the CCB-Sketch are not noticeably more expensive than other local sketches, despite additionally keeping a cardinality table.

More importantly, we see that the global sketches are much more expensive regarding memory and runtime costs to all local sketches. The PR-Sketch is the most expensive evaluated sketch, as its peak memory consumption and execution time increases rapidly due to the SVD computation, despite being optimized for sparse matrices. Similarly, the Seq-Sketch suffers from rapidly increasing peak memory consumption and time complexity, although memory consumption is asymptotically lower than that of the PR-Sketch.

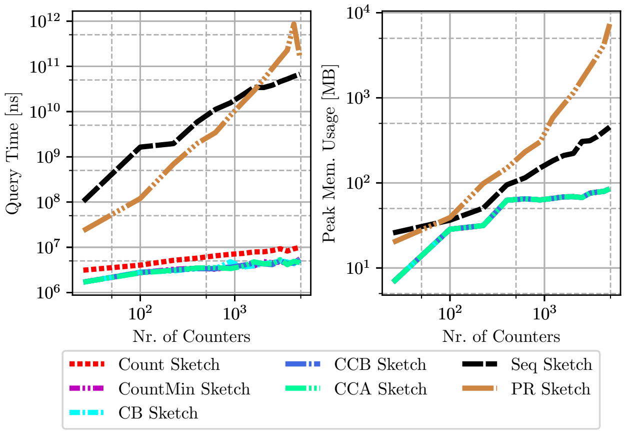


Figure 9: Query time [ns] and peak memory usage [MB] of a query operation on all keys in the data stream.

For real-world applications involving a large number of keys, querying the PR-Sketch would likely require several hundreds of GB and up to an hour of computation time on a high-end mainframe, whereas the CCB-Sketch can be queried in near-real time on relatively modest hardware.

7 CONCLUSION

In this paper, we present the derivation, analysis and evaluation of three novel stream-processing algorithms. Both our theoretical analysis (§5) and experimental results (§6) show that these sketches combine the strengths of lightweight but not very accurate sketches (e.g., C-Sketch) and more accurate but significantly costlier methods (e.g., PR-Sketch). Hence, our sketches enable high accuracy at low query cost: Typically, the proposed CCB-Sketch is orders of magnitude more accurate compared to the C-Sketch while significantly cheaper than the PR-Sketch. This is also the case when modeled prior and actual data stream statistics belong to different parametric families.

This reconciliation of estimation accuracy and query efficiency decisively propels real-world applications that rely on stream processing. For example, emerging QoS systems based on bandwidth reservation [18] must quickly identify flows that overuse their reservation, and require the highly efficient and highly accurate flow-size estimation which only the sketches in this paper can guarantee.

Moreover, we emphasize that our paper is only an initial exploration of the research opportunities that are opened up by combining Bayesian techniques with sketching algorithms. In future work, it will be of interest to investigate the value of different priors (e.g., exponential distributions), other variational-inference techniques (e.g., MCMC), and hybrid approaches based on constrained optimization (as in PR-Sketch) and Bayesian techniques.

REFERENCES

[1] Rakesh Agrawal, Ramakrishnan Srikant, et al. 1994. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, Vol. 1215. Citeseer, 487–499.

[2] Anonymous. 2022. Accompanying Code. https://www.dropbox.com/s/u35xeoxwq0jglwr/bayes-sketch-revised.zip?dl=0.

[3] Pablo Basanta-Val, Norberto Fernandez-Garcia, Luis Sánchez-Fernández, and Jesus Arias-Fisteus. 2017. Patterns for distributed real-time stream processing. *IEEE Transactions on Parallel and Distributed Systems* 28, 11 (2017), 3243–3257.

[4] Ran Ben Basat, Gil Einziger, Roy Friedman, and Yaron Kassner. 201720. Randomized admission policy for efficient top-k and frequency estimation. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 1–9.

[5] Michael Bender and Slobodan Simonovic. 1994. Time-series modeling for long-range stream-flow forecasting. *Journal of Water Resources Planning and Management* 120, 6 (1994), 857–870.

[6] Theophilus Benson, Aditya Akella, and David Maltz. 2010. Network Traffic Characteristics of Data Centers in the Wild. *Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC*, 267–280. https://doi.org/10.1145/1879141.1879175

[7] Ferenc Bodon. 2003. KOSARAK dataset. http://fimi.uantwerpen.be/data/kosarak.dat.gz.

[8] Ferenc Bodon. 2010. UNIV1 dataset. https://pages.cs.wisc.edu/~tbenson/IMC10_Data.html.

[9] Tom Brijs. 2003. RETAIL dataset. http://fimi.uantwerpen.be/data/retail.dat.gz.

[10] CAIDA. 2018. The CAIDA UCSD Anonymized Internet Traces - Oct. 18th. http://www.caida.org/data/passive/passive_dataset.xml

[11] Moses Charikar, Kevin Chen, and Martin Farach-Colton. 2002. Finding frequent items in data streams. In *International Colloquium on Automata, Languages, and Programming*.

[12] Lior Cohen, Gil Avrahami-Bakish, Mark Last, Abraham Kandel, and Oscar Kipersztok. 2008. Real-time data mining of non-stationary data streams from sensor networks. *Information Fusion* 9, 3 (2008), 344–353.

[13] Graham Cormode and S. Muthukrishnan. 2005. An Improved Data Stream Summary: The Count-Min Sketch and Its Applications. *J. Algorithms* 55, 1 (apr 2005), 58–75. https://doi.org/10.1016/j.jalgor.2003.12.001

[14] Ryohei Ebina, Kenji Nakamura, and Shigeru Oyanagi. 2011. A real-time burst detection method. In *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*. IEEE, 1040–1046.

[15] Cristian Estan and George Varghese. 2003. New Directions in Traffic Measurement and Accounting: Focusing on the Elephants, Ignoring the Mice. *ACM Transactions on Computer Systems* 21, 3 (2003), 270–313.

[16] Philippe Flajolet, Éric Fusy, Olivier Gandouet, and Frédéric Meunier. 2007. Hyperloglog: the analysis of a near-optimal cardinality estimation algorithm. In *Discrete Mathematics and Theoretical Computer Science*. Discrete Mathematics and Theoretical Computer Science, 137–156.

[17] René Gebel. 2012. KL1p - a portable C++ library for compressed sensing.

[18] Giacomo Giuliari, Dominik Roos, Marc Wyss, Juan A. Garcia-Pardo, Markus Legner, and Adrian Perrig. 2021. Colibri: A Cooperative Lightweight Inter-domain Bandwidth-Reservation Infrastructure. *Proceedings of ACM CoNEXT* (2021).

[19] Charles AR Hoare. 1961. Algorithm 65: find. *Commun. ACM* 4, 7 (1961), 321–322.

[20] Thomas Holterbach, Edgar Costa Molero, Maria Apostolaki, Alberto Dainotti, Stefano Vissicchio, and Laurent Vanbever. 2019. Blink: Fast connectivity recovery entirely in the data plane. In *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*. 161–176.

[21] Qun Huang, Siyuan Sheng, Xiang Chen, Yungang Bao, Rui Zhang, Yanwei Xu, and Gong Zhang. 2021. Toward nearly-zero-error sketching via compressive sensing. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*. 1027–1044.

[22] Daniel M. Kane, Jelani Nelson, Ely Porat, and David P. Woodruff. 2010. Fast Moment Estimation in Data Streams in Optimal Space. *CoRR* abs/1007.4191 (2010). arXiv:1007.4191 http://arxiv.org/abs/1007.4191

[23] D. Katz, J. Baptista, S. P. Azen, and M. C. Pike. 1978. Obtaining Confidence Intervals for the Risk Ratio in Cohort Studies. *Biometrics* 34, 3 (1978), 469–474. http://www.jstor.org/stable/2530610

[24] Jon Kleinberg. 2003. Bursty and hierarchical structure in streams. *Data mining and knowledge discovery* 7, 4 (2003), 373–397.

[25] George Kollios, John W Byers, Jeffrey Considine, Marios Hadjieleftheriou, and Feifei Li. 2005. Robust Aggregation in Sensor Networks. *IEEE Data Eng. Bull.* 28, 1 (2005), 26–32.

[26] Zaoxing Liu, Ran Ben-Basat, Gil Einziger, Yaron Kassner, Vladimir Braverman, Roy Friedman, and Vyas Sekar. 2019. Nitrosketch: Robust and general sketch-based monitoring in software switches. In *Proceedings of the ACM Special Interest Group on Data Communication*. 334–350.

[27] Z. Liu, A. Manousis, G. Vorsanger, V. Sekar, and V. Braverman. 2016. One Sketch to Rule Them All: Rethinking Network Flow Monitoring with UnivMon. In *ACM SIGCOMM*. https://doi.org/10.1145/2934872.2934906

[28] Conrad Sanderson and Ryan Curtin. 2016. Armadillo: A template-based C++ library for linear algebra. *Journal of Open Source Software* 1 (07 2016), 26. https://doi.org/10.21105/joss.00026

[29] Conrad Sanderson and Ryan Curtin. 2020. An Adaptive Solver for Systems of Linear Equations. In *2020 14th International Conference on Signal Processing and Communication Systems (ICSPCS)*. IEEE. https://doi.org/10.1109/icspcs50536.2020.9309998

[30] Simon Scherrer, Che-Yu Wu, Yu-Hsi Chiang, Benjamin Rothenberger, Daniele E. Asoni, Arish Sateesan, Jo Vliegen, Nele Mentens, Hsu-Chun Hsiao, and Adrian Perrig. 2021. Low-Rate Overuse Flow Tracer (LOFT): An Efficient and Scalable Algorithm for Detecting Overuse Flows. arXiv:2102.01397 [cs.NI]

[31] Siyuan Sheng, Qun Huang, Sa Wang, and Yungang Bao. 2021. PR-Sketch: Monitoring per-Key Aggregation of Streaming Data with Nearly Full Accuracy. *Proc. VLDB Endow.* 14, 10 (jun 2021), 1783–1796. https://doi.org/10.14778/3467861.3467868

[32] Haibo Wang, Chaoyi Ma, Olufemi O Odegbile, Shigang Chen, and Jih-Kwon Peir. 2021. Randomized Error Removal for Online Spread Estimation in Data Streaming. *Proc. VLDB Endow.* 14, 6 (feb 2021), 1040–1052. https://doi.org/10.14778/3447689.3447707

[33] Chen-Chi Wu, Kuan-Ta Chen, Chun-Ying Huang, and Chin-Laung Lei. 2009. An empirical evaluation of VoIP playout buffer dimensioning in Skype, Google talk, and MSN Messenger. In *Proceedings of the 18th international workshop on Network and operating systems support for digital audio and video*. 97–102.

[34] Hao Wu, Hsu-Chun Hsiao, and Yih-Chun Hu. 2014. Efficient Large Flow Detection over Arbitrary Windows: An Algorithm Exact Outside an Ambiguity Region. In *Proceedings of the 2014 Conference on Internet Measurement Conference (IMC)*. ACM, 209–222.

[35] Tong Yang, Jie Jiang, Peng Liu, Qun Huang, Junzhi Gong, Yang Zhou, Rui Miao, Xiaoming Li, and Steve Uhlig. 2018. Elastic sketch: Adaptive and fast network-wide measurements. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*. ACM, 561–575.

[36] Zheng Zhong, Shen Yan, Zikun Li, Decheng Tan, Tong Yang, and Bin Cui. 2021. BurstSketch: Finding Bursts in Data Streams. In *Proceedings of the 2021 International Conference on Management of Data*. 2375–2383.