



**HAL**  
open science

# A Formal Analysis of the Count-Min Sketch with Conservative Updates

Younes Ben Mazziane, Sara Alouf, Giovanni Neglia

► **To cite this version:**

Younes Ben Mazziane, Sara Alouf, Giovanni Neglia. A Formal Analysis of the Count-Min Sketch with Conservative Updates. IEEE INFOCOM WNA 2022 - The second Workshop on Networking Algorithms (WNA), May 2022, New York, United States. 10.1109/INFOCOMWK-SHPS54753.2022.9798146 . hal-03613957

**HAL Id: hal-03613957**

**<https://hal.science/hal-03613957v1>**

Submitted on 24 Mar 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A Formal Analysis of the Count-Min Sketch with Conservative Updates\*

Younes Ben Mazziane, Sara Alouf, Giovanni Neglia  
Université Côte d’Azur, Inria, Email: name.surname@inria.fr

## Abstract

Count-Min Sketch with Conservative Updates (CMS-CU) is a popular algorithm to approximately count items’ appearances in a data stream. Despite CMS-CU’s widespread adoption, the theoretical analysis of its performance is still wanting because of its inherent difficulty. In this paper, we propose a novel approach to study CMS-CU and derive new upper bounds on the expected value and the CCDF of the estimation error under an i.i.d. request process. Our formulas can be successfully employed to derive improved estimates for the precision of heavy-hitter detection methods and improved configuration rules for CMS-CU. The bounds are evaluated both on synthetic and real traces.

## 1 Introduction

Counting how many times a given item appears in a data stream is a basic step common to a variety of applications spanning different domains including network management. For example, routers and servers often routinely count the number of packets in each flow for troubleshooting, traffic monitoring [1], detection of denial of service attacks, etc. Similarly, caching policies often rely on content popularity estimates [10]. Counting is a deceptively simple operation: in many applications the available memory does not permit to instantiate a counter for each possible item, because the number of items is huge (e.g., catalogs of cacheable objects in content delivery networks) or because counters are updated frequently and then require expensive fast memories (e.g., for high-rate inline packet flow processing). As a consequence, these applications rely on approximate counting techniques such as sketch-based algorithms [6], among which a popular one is the Count-Min Sketch (CMS) [7]. CMS is also a building brick of more recent sketch algorithms [14, 22].

CMS achieves significant memory reduction by mapping different items to the same counters through hash functions. As different items may increment the same counter, CMS suffers from overestimation errors. When counters are only incremented, a slight modification to CMS operation, referred to as Conservative Update [11] or Minimal Increment [5], can reduce the estimation error. The Count-Min sketch with Conservative Updates (CMS-CU) is successfully employed for caching [10], heavy flows detection [21], telemarketing call detection [2], and natural language processing [13].

Although conservative updates are a minor modification to CMS operation, they heavily correlate the growth of the different counters, making CMS-CU much more difficult to study than CMS. As CMS-CU reduces CMS estimation errors, it is still possible to maintain upper bounds originally proposed for CMS [7, 8]. This approach has been adopted in some papers, for example to study CMS-CU’s trade-off between memory and accuracy [20, 21], but fails to capture the specific advantages offered by CMS-CU.

To the best of our knowledge, only three papers ventured to study CMS-CU [3, 4, 9]. The authors of [3] relied on a fluid approximation under the assumption that all counters are equally likely to be updated at each step. This assumption may be satisfied only for a large number of counters and a large number of items with similar popularities. Reference [9] modeled CMS-CU as a stack of Bloom filters and derived bounds for the error’s Complementary Cumulative Distribution Function (CCDF) when requests follow the Independent Reference Model (IRM) [12]. Unfortunately, CCDF computation in [9] is an iterative procedure whose time complexity grows quadratically with the error value. Moreover, the analysis in both papers hold for families of  $k$ -wise independent hash functions [16], where  $k$  may be arbitrarily large. But

---

\*This is the author version of Y. Ben Mazziane, S. Alouf, G. Neglia, “A formal analysis of the count-min sketch with conservative updates,” *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2022, pp. 1-6.

such families are incompatible with the memory-constrained applications that need CMS-CU, because memory requirements and computation time grow with  $k$  [18]. More recently, the authors of [4] propose an online algorithm to estimate the error.

In this paper, we propose a novel analysis of CMS-CU which leads to new upper bounds on the expected value and the CCDF of the estimation error under an IRM request process. Our methodology diverges from related work as it quantifies the error on a per-item basis, which is particularly suited for data streams with heterogeneous items' popularities. The analysis also overcomes the limitations of the previous studies as 1) it holds for pairwise independent hash functions, and 2) it provides CCDF expressions with time complexity independent of the error's value. We show that our formulas can be successfully employed to derive improved estimates for the precision of heavy-hitter detection methods and improved configuration rules for CMS-CU.

The rest of the paper is organized as follows. In Sec. 2, we provide the background and introduce the notation. The theoretical analysis is carried out in Sec. 3. Section 4 presents numerical experiments both on synthetic and real world traces.

## 2 Background, Notation and Assumptions

### 2.1 Data Stream Model

A data stream is a sequence  $S_t = (Z(s))_{s=1,\dots,t}$ , where  $Z(s)$  is an item from a universe  $I = \{1, \dots, N\}$  [17]. In general, we want to compute a function of the sequence,  $\mathcal{F}(S_t)$ , for example the number of occurrences of a given item, the set of heavy hitters (items whose number of requests exceeds a given threshold), or the top- $k$  most frequent items. Streaming algorithms aim to compute the function of interest using a few passes through the data stream (only one for the applications we consider) with an amount of memory which is sublinear in the universe's size  $N$  and the data stream size  $t$ . Even for the simple quantities mentioned above, exact computation requires a linear amount of memory and then the streaming algorithms need to settle for approximated results. In the next section, we present two popular streaming algorithms for approximate counting.

In what follows, we denote the set of integer numbers between 1 and  $d \in \mathbb{N}$  by  $[d]$ . Moreover, to lighten the notation, we do not append the sketch name to the symbols. We believe there will be no ambiguity as each sketch is presented and analyzed in a separate section.

### 2.2 Count-Min Sketch (CMS)

A Count-Min sketch is a two dimensional array with  $d$  rows, each with  $w$  counters. An item  $i$  is mapped to a counter in each row via  $d$  hash functions  $\{h_r\}_{r \in [d]}$  chosen uniformly at random from a family of pairwise independent hash functions.

$$h_r : I \rightarrow \{1, \dots, w\}, \forall r \in [d]. \quad (1)$$

We note that, once selected, the hash functions do not change during the processing of the stream  $S_t$ . We model the association between items and counters as a bipartite undirected graph  $G = (I, O, E)$ , where  $O$  is the set of counters and  $E \triangleq \{(i, h_r(i)) : i \in I, r \in [d]\}$  is the set of edges. We denote the open neighbourhood of node  $i$  in the graph as  $N_G(i) \triangleq \{c : (i, c) \in E\}$  and the value at time  $t$  of the counter in row  $r$  corresponding to item  $i$  as  $c_i^r(t)$ . When item  $i$  is requested at time  $t$ , the counters  $\{h_r(i)\}_{r \in [d]}$  are incremented by 1. Namely,

$$c_i^r(t) = c_i^r(t-1) + 1, \forall r \in [d]. \quad (2)$$

Let  $n_i(t)$  denote item- $i$ 's number of occurrences in the stream up to time  $t$ . Note that  $c_i^r(t)$  is updated not only by new requests for item  $i$ , but also by requests for all items that are also mapped by  $h_r$  to the same counter  $h_r(i)$ , i.e., by all items in the set  $\{j \in I : h_r(j) = h_r(i)\}$ . These items are said to *collide* with  $i$ . It follows that  $c_i^r(t) = \sum_{j: h_r(i)=h_r(j)} n_j(t)$ . As such,  $c_i^r(t)$  upper bounds  $n_i(t)$ . We denote the error resulting from using  $c_i^r(t)$  for estimating  $n_i(t)$  as  $e_i^r(t)$ , i.e.,  $e_i^r(t) \triangleq c_i^r(t) - n_i(t)$ . Since all counters' values  $\{c_i^r(t)\}_{r \in [d]}$  upper bound  $n_i(t)$ , their minimum also upper bounds  $n_i(t)$ . This minimum is the estimate of  $n_i(t)$  provided by CMS and we denote it as  $\hat{n}_i(t)$ ,

$$\hat{n}_i(t) \triangleq \min_{r \in [d]} c_i^r(t). \quad (3)$$

The estimation error is then

$$e_i(t) \triangleq \hat{n}_i(t) - n_i(t) = \min_{r \in [d]} e_i^r(t). \quad (4)$$

We also introduce  $\delta_{i,j}^r(s)$  to represent the contribution of item  $j \neq i$  to counter  $h_r(i)$  at time  $s$ . We have:

$$\delta_{i,j}^r(s) \triangleq \mathbf{1}(Z(s) = j, h_r(i) = h_r(j)), \quad (5)$$

$$e_i^r(t) = \sum_{s \in [t]} \sum_{j \in I \setminus \{i\}} \delta_{i,j}^r(s). \quad (6)$$

All quantities we defined are random variables due to the initial random choice of the hash functions. From (6) and the definition of pairwise independence [16], one can immediately conclude that  $\mathbb{E}[e_i^r(t)] = \frac{\sum_{j \neq i} n_j(t)}{w} \leq \frac{t}{w}$ . Applying (4), we obtain the following upper bound for the expected estimation error:

$$\mathbb{E}[e_i(t)] \leq \frac{t}{w}. \quad (7)$$

Moreover, the random variables  $\{e_i^r(t)\}_{r \in [d]}$  are i.i.d., and an application of the Markov inequality leads to the following upper bound on the CCDF of  $e_i(t)$ :

$$\Pr\left(\frac{e_i(t)}{t} \geq x\right) \leq \left(\frac{1}{wx}\right)^d. \quad (8)$$

Cormode and Muthukrishnan proved this result in [7, Theorem 1] for the particular value  $x = \frac{e}{w}$ .

### 2.3 Count-Min Sketch with Conservative Updates (CMS-CU)

The conservative update [11] or minimal increment [5] is an optimization of CMS that consists in incrementing only the counters that attain the minimum value. The update procedure when item  $i$  is requested at time  $t$  becomes

$$c_i^r(t) = \max\left(c_i^r(t-1), \min_{f \in [d]} c_i^f(t-1) + 1\right), \quad \forall r \in [d]. \quad (9)$$

The error  $e_i^r(t)$  in each row  $r$ , the estimation count  $\hat{n}_i(t)$ , and the estimation error  $e_i(t)$ , all depend on  $c_i^r(t)$  in the same way as in CMS. Equations (3) and (4) hold with CMS-CU. The quantities  $\{\delta_{i,j}^r(s)\}_{s \in [t], j \in I \setminus \{i\}}$  are now defined as

$$\delta_{i,j}^r(s) \triangleq \mathbf{1}(Z(s) = j, h_r(i) = h_r(j), \hat{n}_j(s-1) = c_i^r(s-1)). \quad (10)$$

Equation (6) holds for CMS-CU. With respect to (5), (10) captures the additional condition that counter  $h_r(i)$  is updated by a request for  $j$  at time  $s$  only if its current value  $c_i^r(s-1)$  coincides with the current estimate  $\hat{n}_j(s-1)$ . Because of this additional condition, CMS-CU enjoys always a smaller error than CMS. Therefore, CMS upper bounds on the expectation (7) and on the CCDF (8) also hold for CMS-CU.

### 2.4 Our Assumptions

We will assume in our analysis that the request process follows the Independent Reference Model (IRM) [12], in other words,  $\{Z(s)\}_{s \in [t]}$  are i.i.d. categorical random variables with  $\Pr(Z(s) = i) = p_i$ , for  $i \in I$ , and  $\sum_{i \in I} p_i = 1$ . We refer to  $p_i$  as the *popularity* of item  $i$ . Without loss of generality, we number items in  $I$  according to their popularity rank, hence  $p_i \geq p_{i+1}$ , for  $i \in [N-1]$ . Note that there are two sources of randomness in our setting: the hash functions' selection and the request process  $S_t$ . From now on, the expectation  $\mathbb{E}[\cdot]$  and the probability  $\Pr(\cdot)$  take both kinds of randomness into account.

## 3 Theoretical Analysis of CMS-CU

Under the IRM model, we first prove a tighter upper bound on the CCDF of  $e_i(t)$  for CMS, then we upper bound the expectation and CCDF of  $e_i(t)$  for CMS-CU.

### 3.1 CMS: CCDF of the Estimation Error

In this section we will derive a tighter bound for CMS error under the IRM assumption. As discussed in Sec. 2.3, this new bound also applies to the CMS-CU error. We first observe from (5) that  $\mathbb{E}[\delta_{i,j}^r(s)] \leq p_j/w$ , since item  $j$  is requested with probability  $p_j$  and the hash collision probability between  $i$  and  $j$  equals  $1/w$  because of pairwise independence. It readily follows from (6) that  $\mathbb{E}[e_i^r(t)] \leq (1 - p_i)t/w$ . Therefore,  $(1 - p_i)t/w$  is an upper bound on the expected error  $\mathbb{E}[e_i(t)]$ .

**Proposition 1 (Upper bound on the CCDF of  $e_i(t)/t$ )** *The CCDF of the estimation error  $e_i(t)$ , when using CMS, verifies*

$$\Pr(e_i(t)/t \geq x) \leq \mathcal{A}(x)^d, \quad (11)$$

$$\text{where } \mathcal{A}(x) \triangleq \min_{k=0, \dots, w-1} \mathcal{A}_k(x), \quad \mathcal{A}(0) = 1, \quad (12)$$

$$\text{and } \mathcal{A}_k(x) \triangleq \frac{\sum_{j>k} p_j}{(w-k)x} + \frac{k}{w}. \quad (13)$$

**Proof 1** *From (4) and the fact that the random variables  $\{e_i^r(t)\}_{r \in [d]}$  are i.i.d. when using CMS, we have  $\Pr(e_i(t)/t \geq x) = (\Pr(e_i^1(t)/t \geq x))^d$ . To prove (11) it is then sufficient to show that  $\Pr(e_i^1(t)/t \geq x) \leq \mathcal{A}_k(x)$  for  $k = 0, \dots, w-1$ . For a given  $k \neq 0$  we consider the event, called  $E_{i,k}^r$ , of no hash collision in row  $r$  between item  $i$  and any of the  $k$  most popular items (other than  $i$ , if  $i \leq k$ ). By first writing the law of total probabilities with respect to the partition  $\{E_{i,k}^1, \overline{E_{i,k}^1}\}$ , and then using the union bound to write  $\Pr(\overline{E_{i,k}^1}) \leq k/w$  and the Markov inequality to upper bound  $\Pr(e_i^1(t)/t \geq x | E_{i,k}^1)$ , we obtain*

$$\Pr(e_i^1(t)/t \geq x) \leq \mathbb{E}[e_i^1(t) | E_{i,k}^1] / (xt) + k/w \quad (14)$$

$$\leq \sum_{j>k} p_j \Pr(h_1(i) = h_1(j) | E_{i,k}^1) / x + k/w \quad (15)$$

$$\leq \sum_{j>k} p_j / ((w-k)x) + k/w \quad (16)$$

where (15) follows from (5)-(6) and (16) uses  $\Pr(\overline{E_{i,k}^1}) \leq k/w$ . By observing that (16) holds also for  $k = 0$ , we have completed the proof.

Proposition 1 extends known results in the literature. In particular, upper bounding the right-hand side of (11) by  $(\mathcal{A}_0(x))^d$  yields (8), and then replacing  $x = e/w$ , we obtain [7, Theorem 1].

In order to highlight the importance of this proposition, we present an example where the improvement of (11) over (8) is evident. Consider a distribution where the most popular item is requested with probability  $1 - \epsilon$ , and other items have uniform popularity, i.e.,  $p_1 = 1 - \epsilon$ ,  $p_i = \epsilon/(N-1)$ ,  $\forall i > 1$ . We compare the two bounds for  $x = 1/w$ . Equation (8) provides the trivial bound  $\Pr(e_i(t)/t \geq 1/w) \leq 1$ . On the other hand, bounding the right-hand side of (11) by  $(\mathcal{A}_1(x))^d$  yields the upper bound  $(\epsilon \frac{w}{w-1} + \frac{1}{w})^d$ . For large  $w$  and  $\epsilon = o(\frac{1}{w})$ , we conclude that  $\Pr(e_i(t)/t \geq 1/w) \lesssim \frac{1}{w^d} \approx 0$ , in sharp contrast with the bound in (8).

### 3.2 CMS-CU: Expected Estimation Error

We consider now CMS-CU and derive an upper bound on the expectation of the estimation error. Because of (4), an upper bound on  $e_i^r(t)$  suffices. Thus, we turn our attention to the random variable  $\delta_{i,j}^r(s)$ . As for CMS, it is easy to prove for CMS-CU that  $\mathbb{E}[\delta_{i,j}^r(s)] \leq p_j/w$ . In the next lemma, we derive a tighter bound, in particular for  $j > i$ .

**Lemma 1 (Upper bound on  $\mathbb{E}[\delta_{i,j}^r(s)]$ )** *The expected contribution of item  $j$  to item  $i$ 's count at row  $r$  at time  $s$  satisfies (17).*

$$\begin{aligned} & \exists \alpha_{i,j} > 0, \beta_{i,j} \geq 0 : \\ & \mathbb{E}[\delta_{i,j}^r(s)] \leq \frac{p_j}{w} (\gamma_{i,j} + \beta_{i,j} \exp(-\alpha_{i,j}(s-1))), \end{aligned} \quad (17)$$

with

$$\gamma_{i,j} \triangleq \begin{cases} 1, & \forall j \leq i \\ \min(\mathcal{A}(p_i - p_j)^{d-1}, 1), & \forall j > i \end{cases} \quad (18)$$

and  $\mathcal{A}(x)$  given in (12).

**Proof 2 (Proof of Lemma 1)** We will make use of two quantities to prove Lemma 1.

$$l_j^r \triangleq \sum_{e \in N_G(h_r(j))} p_e, \quad g_j \triangleq \min_{r \in [d]} l_j^r. \quad (19)$$

For a given realization of  $G$ ,  $l_j^r$  is an upper bound on the growth rate of counter  $c_j^r(t)$  and  $g_j$  is an upper bound on the growth rate of  $\hat{n}_j(t)$ . To ease the writing, we use  $A$ ,  $B$ ,  $C$ , and  $D_j^r$  as shorthand for events “ $h_r(i) = h_r(j)$ ”, “ $\hat{n}_j(s-1) = c_i^r(s-1)$ ”, “ $g_j \geq p_i$ ”, and “ $l_j^r \geq p_i$ ”, respectively. Starting from (10) we write

$$\begin{aligned} \mathbb{E}[\delta_{i,j}^r] &= p_j \Pr(A \cap B) \\ &= p_j \left[ \Pr(A \cap B \cap C) + \Pr(A \cap B \cap \bar{C}) \right] \\ &\leq p_j \left[ \Pr(A \cap C) + \Pr(A \cap \bar{C}) \Pr(B | A, \bar{C}) \right] \\ &\leq p_j \left[ \Pr(A \cap (\cap_{e \in [d], e \neq r} D_j^e)) + \Pr(A) \Pr(B | A, \bar{C}) \right] \\ &\leq p_j \Pr(A) \left[ \Pr(D_j^1)^{d-1} + \Pr(B | A, \bar{C}) \right] \\ &\leq \frac{p_j}{w} \left[ \gamma_{i,j} + \beta_{i,j} \exp(-\alpha_{i,j}(s-1)) \right] \end{aligned}$$

where  $\gamma_{i,j}$  is given in (18). We obtained the last step by using (for  $j > i$ )  $\Pr(D_j^1) = \Pr(l_j^1 - p_j \geq p_i - p_j) \leq \mathcal{A}(p_i - p_j)$ , that can be derived following the steps in (14)-(16). The inequality

$$\Pr(B | A, \bar{C}) \leq \beta_{i,j} \exp(-\alpha_{i,j}(s-1)) \quad (20)$$

also used in the last step, requires more explanations. Due to space constraints we only sketch its derivation. For  $z = s-1$ , we define the random variable  $y_j(z)$  as,

$$y_j(z) \triangleq \sum_{e: h_{r_0}(j) = h_{r_0}(e)} n_e(z) : r_0 = \arg \min_{r \in [d]} l_j^r. \quad (21)$$

It follows that  $\hat{n}_j(z) \leq y_j(z)$ . Furthermore, since  $c_i^r(z) \geq n_i(z)$ , we get  $\Pr(B | A, \bar{C}) \leq 1 - \Pr(F | A, \bar{C})$ , where  $F = \{n_i(z) > y_j(z)\}$ . Since  $n_i(z)$  and  $y_j(z)$  are negatively associated [15], we have:

$$\Pr(B | A, \bar{C}) \leq 1 - \Pr(L) \Pr(J | A, \bar{C}) \quad (22)$$

Where  $L = \{n_i(z) > m(z)\}$ ,  $J = \{y_j(z) < m(z)\}$  and  $m(z) = (p_i + g_j)z/2$ . Following (21), for every fixed graph realization of  $G$ ,  $y_j(z)$  is the sum of negatively associated random variables [15] and has an expected value of  $g_j z < m(z)$  under the conditioning  $g_j < p_i$ , thus using Chernoff bounds on events  $L$  and  $J$  we get (20).

**Comments on Lemma 1** We recall that using CMS, this expectation is upper bounded by  $p_j/w$ , thus the term  $\gamma_{i,j} + \beta_{i,j} \exp(-\alpha_{i,j}(s-1))$  is an attenuation term taking into account the conservative update. As  $s \rightarrow \infty$ , this attenuation term converges to  $\gamma_{i,j}$ . The larger the difference between  $i$  and  $j$  probabilities, the smaller is  $\gamma_{i,j}$ . This is expected, as the larger the difference in popularity between two items  $i, j$ , the more likely that  $c_i^r(t) > \hat{n}_j(t)$ .

We now state the main result of this section.

**Theorem 1 (Upper bound on  $\mathbb{E}[e_i(t)]$ )** The error experienced by item  $i$  is upper bounded as follows

$$\exists B_i \in \mathbb{R}^+ : \mathbb{E} \left[ \frac{e_i(t)}{t} \right] \leq \frac{1}{w} \sum_{j \in I \setminus \{i\}} p_j \gamma_{i,j} + \frac{B_i}{t}, \quad (23)$$

where  $\gamma_{i,j}$  is defined in (18).

**Proof 3** We prove the upper bound on  $\mathbb{E}[e_i^r(t)]$  and because of (4) this upper bound holds for  $\mathbb{E}[e_i(t)]$  too. An upper bound on  $\mathbb{E}[e_i^r(t)]$  is readily found by linearity of the expectation, using (6) and Lemma 1. We find

$$\mathbb{E}[e_i^r(t)] \leq \sum_{j \in I \setminus \{i\}} \frac{p_j \gamma_{i,j}}{w} t + \frac{\beta_i}{w(1 - \exp(-\alpha_i))}, \quad (24)$$

where  $\alpha_i = \min_{j \in I} \alpha_{i,j}$  and  $\beta_i = \max_{j \in I} \beta_{i,j}$  and  $\alpha_{i,j}$  and  $\beta_{i,j}$  are the constants in Lemma 1.

As  $t \rightarrow \infty$ , the term  $B_i$  can be ignored. Note that the bound depends on the item's rank  $i$ . As discussed before,  $\gamma_{i,j}$  is a decreasing function of  $p_i - p_j$ , thus, of  $j$ . A necessary and sufficient condition to improve over the bound in (7) for a given item  $i$  is then to have  $\gamma_{i,N} < 1$ . At the same time,  $\gamma_{i,j}$  is an increasing function of  $i$ . Therefore, the more popular the item, the smaller the bound (23), which is always smaller than or equal to the bound (7) when neglecting  $B_i$ . While previous studies [3, 9] bounded the error uniformly across items, our analysis provides error bounds depending on item's popularity. In particular, our work is the first to support analytically the experimental evidence that the most popular items barely experience any error [3].

To highlight the improvement of our bound over the CMS bound  $\mathbb{E}[e_i(t)] \leq (1 - p_i)/w$  (see the beginning of Sec. 3.1), we consider the same example as in Sec. 3.1, i.e.,  $p_1 = 1 - \epsilon$ ,  $p_i = \epsilon/(N - 1)$  for  $i > 1$ , and focus on the most popular item. According to CMS analysis,  $\mathbb{E}[e_1(t)/t] \leq \epsilon/w$ , whereas (24) yields a bound for the most popular item that is  $\epsilon \mathcal{A} \left(1 - \epsilon \frac{N}{N-1}\right)^{d-1} / w \approx \epsilon \mathcal{A} (1 - \epsilon)^{d-1} / w$ . This bound is smaller than  $\epsilon \mathcal{A}_0 (1 - \epsilon)^{d-1} / w = \epsilon (w(1 - \epsilon))^{1-d} / w$ . By choosing  $\epsilon = \frac{1}{2}$ , we get an improvement by a factor of  $(w/2)^{d-1}$ .

Having analysed the expectation of the estimation error, we turn our attention to its CCDF, which is studied in the next section.

### 3.3 CMS-CU: CCDF of the Estimation Error

**Proposition 2 (Upperbound on the CCDF of  $e_i(t)/t$ )** *The CCDF of the estimation error  $e_i(t)$ , when using CMS-CU, is upper bounded as follows:*

$$\Pr \left( \frac{e_i(t)}{t} \geq x \right) \leq \min(\mathcal{A}(x)^d, \mathcal{B}(x, i, t)), \quad (25)$$

where

$$\mathcal{B}(x, i, t) \triangleq \min_{k=0, \dots, w-1} \frac{1}{x} \left( \frac{1}{w-k} \sum_{j>k} p_j \gamma_{i,j} + \frac{B_i(k)}{t} \right) + \frac{k}{w}, \quad (26)$$

and  $B_i(k)$  are constants that depend only on item  $i$  and on  $k$ .

**Proof 4 (Sketch of the proof)** *The bounds that are valid with CMS are also valid with CMS-CU, thus by Proposition 1, the CCDF with CMS-CU is less than  $\mathcal{A}(x)^d$ . To prove the other part, we rely on the same arguments used in the proofs of Proposition 1 and Lemma 1. We first write (14) for the random variable  $e_i^r(t)$ . To bound  $\mathbb{E}[e_i^r(t) | E_{i,k}^r]$ , we repeat the derivations in the proof of Lemma 1 to bound the conditional expectation of  $\delta_{i,j}^r(s)$ , and then using (6) we obtain  $\mathbb{E}[e_i^r(t) | E_{i,k}^r] \leq \sum_{j>k} p_j \gamma_{i,j} t / (w - k) + B_i(k)$ . As this bound is valid for  $k = 0, \dots, w - 1$ , we find (25) which concludes the proof.*

In practical situations,  $t$  is large enough such that we can ignore the constants  $B_i(k)$  in (26) and the bound (25) depends solely on  $x$  and  $i$ .

We will illustrate the utility of Proposition 2 in the next section where we estimate a metric of interest in the heavy-hitters detection problem.

### 3.4 Heavy-Hitters Use Case: Lower Bound on the Precision

Detecting heavy-hitters in a stream can be done using a sketch. A heavy-hitter is an item that has request rate higher than a threshold  $\phi$ . However, when using a sketch (for instance CMS or CMS-CU), an item with a rate smaller than  $\phi$  can erroneously appear as a heavy-hitter because of the overestimation error; we call such an item a “false positive.” Let  $H$  be the set of heavy-hitters,  $H = \{i : n_i(t) \geq \phi \cdot t\}$ , and  $\hat{H}$  be the set of items classified as heavy-hitters by the sketch,  $\hat{H} = \{i : \hat{n}_i(t) \geq \phi \cdot t\}$ . The “precision” is one metric used for assessing the performance of the sketch, and is defined as follows:  $P = |H|/|\hat{H}|$ . For sake of simplicity, we assume that  $n_i(t) \approx p_i t$ , for  $t$  large enough, this is reasonable because of the law of large numbers. Under this approximation  $|H|$  is constant and we can write the expected value of the precision as:

$$\mathbb{E}[P] \approx \frac{|H|}{|H| + \sum_{i \notin H} \Pr\left(\frac{e_i(t)}{t} \geq \phi - p_i\right)}. \quad (27)$$

Combining (27) with Proposition 2 we obtain a lower bound on the expected precision when CMS-CU is used. This lower bound will be illustrated in Section 4.3 and compared to experimental values.

## 4 Experimental Evaluation and Numerical Analysis

### 4.1 Experimental Setting

To support our analysis, we have undertaken a series of experiments in which we simulated requests for items over time and used CMS-CU to count the requests for each item. We considered two settings in our experiments. In the first setting, we generated  $10^3$  synthetic streams from two different Zipf distributions, with shape parameter  $\alpha = 0.8$  and  $\alpha = 1.0$ . Each stream contains 1 million requests for items in the set  $I = \{1, \dots, 10^6\}$  ( $N = 10^6$ ). We selected different hash functions for each stream by choosing uniformly at random  $d$  different seeds in  $[10^6]$ . The experimental values reported for this setting are averaged over the  $10^3$  streams. We also computed the 95% confidence intervals but do not report them as they are very narrow and would hardly be visible in the figures.

In the second setting, we used a trace of accesses to Wikipedia pages in all languages during September 2007 [19]. The trace contains 10,628,125 requests. The number of distinct Wikipedia pages requested in this trace is 1,712,459. We extracted 10 non-overlapping stream from this trace, each containing  $10^6$  requests, and discarded the rest.

For each of the settings, we report the results obtained for two metrics: (i) the expected estimation error of the sketch for each item, and (ii) the precision in the heavy-hitters detection problem.

### 4.2 The Expected Estimation Error

In the synthetic setting, we computed for each item the average estimation error over the  $10^3$  streams, our upper bound as in Theorem 1 (but neglecting the constant  $B_i$ ), and the state-of-the-art bound (7) originally proposed in [7]. Results are shown in Fig. 1.

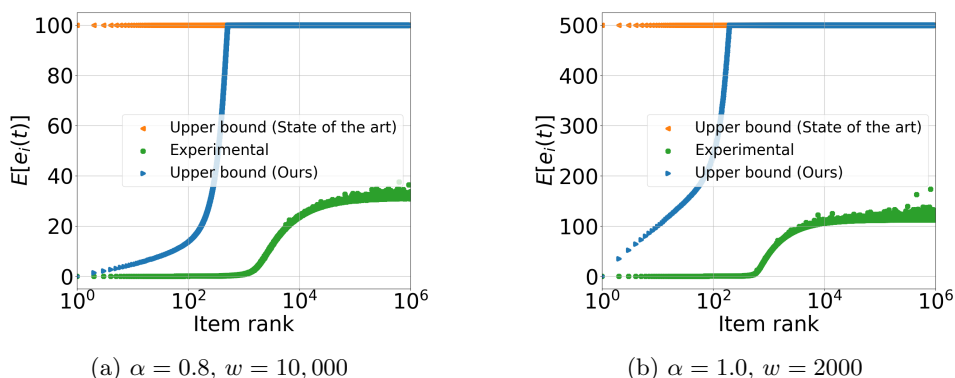


Figure 1: Synthetic traces: estimation error for each item,  $N = 10^6, d = 5, t = 10^6$ .



As we observed above, our analysis correctly predicts that different items experience a different error and improve the current bound on the expected estimation error for the 500 and 190 most popular items for  $\alpha = 0.8$  and  $\alpha = 1$ , respectively.

For Wikipedia access trace, we used the first stream to estimate items' popularity distribution. Results in Fig. 3a are then averages over the remaining 9 streams computed as done for Fig. 1. The curves confirm qualitatively the observations on the synthetic traces: our approach leads to a smaller bound for the most popular items.

### 4.3 Precision in Detecting $\phi$ -Heavy-Hitters

As we discussed in Sec. 3.4, the CCDF upper bound in Proposition 2 allows us also to derive a lower bound on CMS-CU's expected precision using approximation (27). If the same approximation is combined with the CCDF upper bound (8) from [7], one can obtain an analogous lower bound on CMS-CU's expected precision. This lower bound is labelled "lower bound (State of the art)" in Figs. 2 and 3b.

In the experiments with the synthetic streams, we averaged the precision values obtained over the  $10^3$  runs. We have repeated these experiments for multiple width values. The experimental values are depicted in Fig. 2 together with our lower bound and the state of the art bound. Consistently with what observed for the CCDF, our approach improves also precision estimation. The bound becomes tighter for larger values of the width  $w$ .

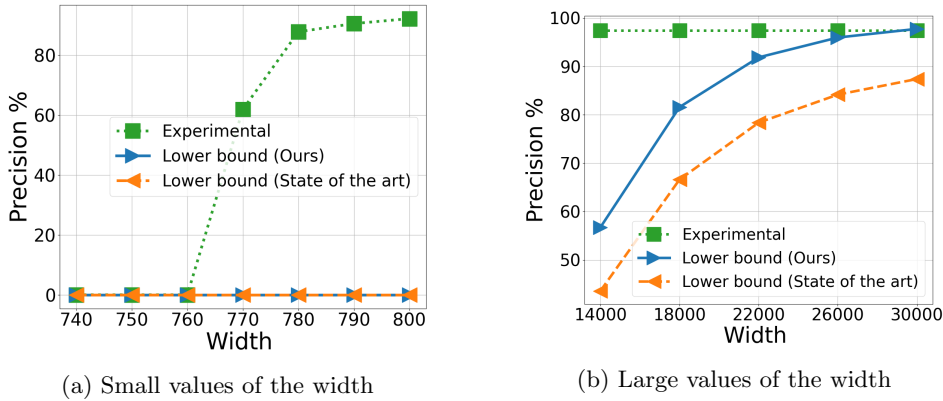


Figure 2: Synthetic trace: Precision as a function of the width  $w$ ,  $N = 10^6$ ,  $\alpha = 0.8$ ,  $d = 5$ ,  $t = 10^6$ ,  $\phi = 5 \times 10^{-4}$ .

In the experiments with Wikipedia trace, we slightly changed the popularity estimation procedure in comparison to Sec. 4.2. The empirical distribution over a stream was used as input to the analytical formulas to predict the precision in the following stream. Fig. 3b shows the corresponding results. The advantage of our approach is even more evident over this real trace. Note that the state-of-the-art CCDF bound in (8) depends on the count sketch parameters  $w$  and  $d$  but not on the popularity distribution. Nevertheless, the corresponding precision in Fig. 3b changes across streams: the approximated formula for the precision (27) depends on the specific stream because popularity distribution (and then also the number of heavy hitters  $|H|$ ) change from one stream to the other.

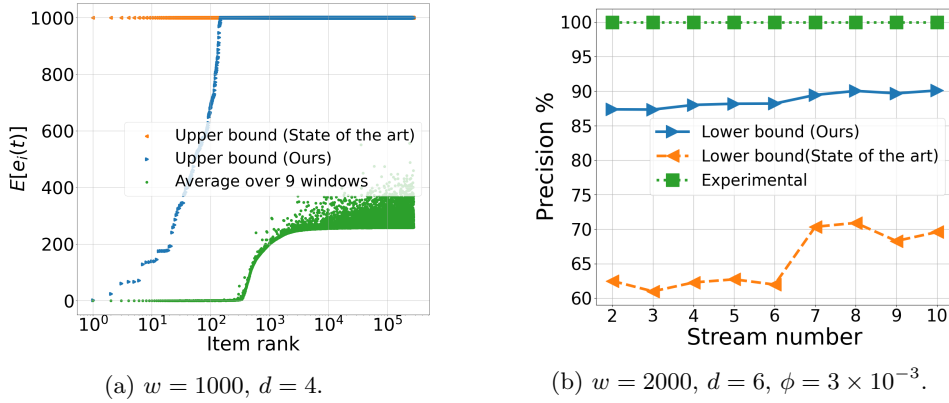


Figure 3: Real trace: estimation error for each item (left) and precision (right),  $t = 10^6$ .

#### 4.4 Configuring CMS-CU with QoS Guarantees

The bounds we derived can also be used to configure the width  $w$  and the depth  $d$  of CMS-CU in order to achieve the desired precision with the minimum amount of memory. If each counter uses 4 bytes, the memory cost of a CMS-CU is  $M = 4wd$  bytes. We compared numerically the memory requirements determined by our approach and by the state-of-the-art one. In particular, for target precision values in the range 0.8–0.975, we performed a search for memory values between 20 Kbytes and 3.2 Mbytes (with a step of 20 Kbytes) and depth values between 2 and 8 (the width is then determined as  $w = M/4d$ ) to find the smallest memory which guarantees the target precision. Figure 4 shows the corresponding curves obtained using our approach and the state-of-the-art-one for the synthetic and Wikipedia trace.

Our approach leads to configuring CMS-CU using a reduced amount of memory, e.g., for 97.5% target precision, the improvement factors are 6.63 and 7.12 for the synthetic and the Wikipedia trace, respectively.

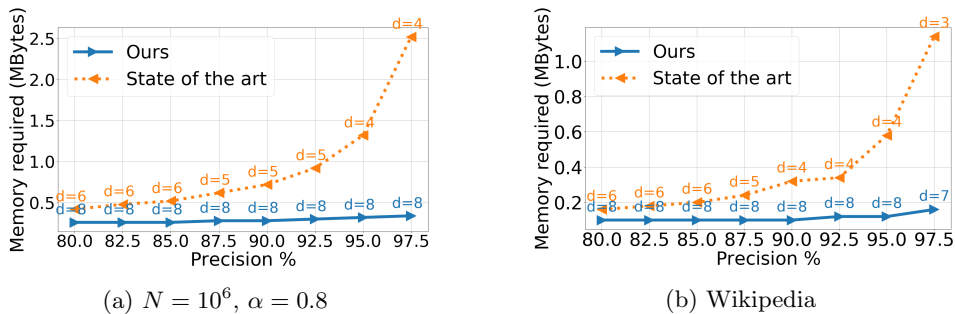


Figure 4: Memory required for a given precision: (a)  $\phi = 5 \cdot 10^{-4}$ , (b)  $\phi = 10^{-3}$ .

## 5 Conclusion and Perspectives

While it is a common belief that CMS-CU leads to smaller estimation errors for the most popular items [3], our paper is the first to provide quantitative support for such property, thanks to a per-item study of the estimation error. We showed that our analysis significantly improves existing bounds for the most popular items and leads, in comparison to the state of the art, to more accurate estimations for the precision in heavy-hitter detection problems as well as to improved configuration rules, which avoid to oversize the counting data structure.

For less popular items, our bounds are not tighter than existing ones. In the future, we want then to focus on improving the bounds for the tail of the popularity distribution. A possible approach is to combine our analysis with existing methods to estimate the CMS-CU error floor when items have similar popularities like those in [3].

## References

- [1] Ran Ben Basat, Gil Einziger, Roy Friedman, and Yaron Kassner. Optimal elephant flow detection. In *INFOCOM*, pages 1–9. IEEE, 2017.

- [2] Giuseppe Bianchi, Nico d’Heureuse, and Saverio Niccolini. On-demand time-decaying Bloom filters for telemarketer detection. *ACM SIGCOMM Comput. Communi. Rev.*, 41(5):5–12, 2011.
- [3] Giuseppe Bianchi, Ken Duffy, Douglas Leith, and Vsevolod Shneer. Modeling conservative updates in multi-hash approximate count sketches. In *24th ITC*, pages 1–8. IEEE, 2012.
- [4] Peiqing Chen, Yuhan Wu, Tong Yang, Junchen Jiang, and Zaoxing Liu. Precise error estimation for sketch-based flow measurement. In *Proc. 21st ACM Internet Measurement Conf.*, pages 113–121, 2021.
- [5] Saar Cohen and Yossi Matias. Spectral Bloom filters. In *Proc. ACM SIGMOD Int. Conf. Management Data*, pages 241–252, 2003.
- [6] Graham Cormode and Marios Hadjieleftheriou. Methods for finding frequent items in data streams. *The VLDB Journal*, 19(1):3–20, 2010.
- [7] Graham Cormode and Shan Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithms*, 55(1):58–75, 2005.
- [8] Graham Cormode and Shan Muthukrishnan. Summarizing and mining skewed data streams. In *Proc. SIAM Int. Conf. Data Mining*, pages 44–55. SIAM, 2005.
- [9] Gil Einziger and Roy Friedman. A formal analysis of conservative update based approximate counting. In *ICNC*, pages 255–259. IEEE, 2015.
- [10] Gil Einziger, Roy Friedman, and Ben Manes. TinyLFU: A highly efficient cache admission policy. *ACM Trans. Storage*, 13(4):1–31, 2017.
- [11] Cristian Estan and George Varghese. New directions in traffic measurement and accounting: Focusing on the elephants, ignoring the mice. *ACM Trans. Comput. Syst.*, 21(3):270–313, 2003.
- [12] Ronald Fagin. Asymptotic miss ratios over independent references. *Journal of Computer and System Sciences*, 14(2):222–250, 1977.
- [13] Amit Goyal, Hal Daumé III, and Graham Cormode. Sketch algorithms for estimating point queries in NLP. In *Proc. 2012 joint Conf. on empirical methods in natural language processing and computational natural language learning*, pages 1093–1103, 2012.
- [14] Chen-Yu Hsu, Piotr Indyk, Dina Katabi, and Ali Vakilian. Learning-based frequency estimation algorithms. In *ICLR*, 2019.
- [15] Kumar Joag-Dev and Frank Proschan. Negative association of random variables with applications. *The Annals of Statistics*, pages 286–295, 1983.
- [16] Rajeev Motwani and Prabhakar Raghavan. *Randomized algorithms*. Cambridge university press, 1995.
- [17] Shanmugavelayutham Muthukrishnan. *Data streams: Algorithms and applications*. Now Publishers Inc, 2005.
- [18] Alan Siegel. On universal classes of extremely random constant-time hash functions. *SIAM Journal on Computing*, 33(3):505–543, 2004.
- [19] Guido Urdaneta, Guillaume Pierre, and Maarten van Steen. Wikipedia workload analysis for decentralized hosting. *Elsevier Computer Networks*, 53(11):1830–1845, July 2009.
- [20] Federica Ventruto, Marco Pulimeno, Massimo Cafaro, and Italo Epicoco. On frequency estimation and detection of heavy hitters in data streams. *Future Internet*, 12(9):158, 2020.
- [21] Rui Wang, Hongchao Du, Zhaoyan Shen, and Zhiping Jia. DAP-Sketch: An accurate and effective network measurement sketch with Deterministic Admission Policy. *Computer Networks*, 194:108155, 2021.
- [22] Tong Yang et al. Elastic sketch: Adaptive and fast network-wide measurements. In *Proc. 2018 Conf. ACM Special Interest Group Data Communication*, pages 561–575, 2018.