# On Estimating Frequency Moments of Data Streams

Sumit Ganguly and[1] Graham Cormode[2]

[1] Indian Institute of Technology, Kanpur, sganguly@iitk.ac.in
[2] AT&T Labs–Research, graham@research.att.com

**Abstract.** Space-economical estimation of the $p$th frequency moments, defined as $F_p = \sum_{i=1}^{n} |f_i|^p$, for $p > 0$, are of interest in estimating all-pairs distances in a large data matrix [14], machine learning, and in data stream computation. Random sketches formed by the inner product of the frequency vector $f_1, \ldots, f_n$ with a suitably chosen random vector were pioneered by Alon, Matias and Szegedy [1], and have since played a central role in estimating $F_p$ and for data stream computations in general. The concept of $p$-stable sketches formed by the inner product of the frequency vector with a random vector whose components are drawn from a $p$-stable distribution, was proposed by Indyk [11] for estimating $F_p$, for $0 < p < 2$, and has been further studied in Li [13].
In this paper, we consider the problem of estimating $F_p$, for $0 < p < 2$. A disadvantage of the stable sketches technique and its variants is that they require $O(\frac{1}{\epsilon^2})$ inner-products of the frequency vector with dense vectors of stable (or nearly stable [14, 13]) random variables to be maintained. This means that each stream update can be quite time-consuming. We present algorithms for estimating $F_p$, for $0 < p < 2$, that does not require the use of stable sketches or its approximations. Our technique is elementary in nature, in that, it uses simple randomization in conjunction with well-known summary structures for data streams, such as the COUNT-MIN sketch [7] and the COUNTSKETCH structure [5]. Our algorithms require space $\tilde{O}(\frac{1}{\epsilon^{2+p}})$ [3] to estimate $F_p$ to within $1 \pm \epsilon$ factors and requires expected time $O(\log F_1 \log \frac{1}{\delta})$ to process each update. Thus, our technique trades an $O(\frac{1}{\epsilon^p})$ factor in space for much more efficient processing of stream updates. We also present a stand-alone iterative estimator for $F_1$.

## 1 Introduction

Recently, there has been an emergence of *monitoring applications* in diverse areas including network traffic monitoring, network topology monitoring, sensor networks, financial market monitoring, and web-log monitoring. In these applications, data is generated rapidly and continuously, and must be analyzed efficiently, in real-time and in a single-pass over the data to identify large trends, anomalies, user-defined exception conditions, and so on. In many of these applications, it is often required to continuously track the "big picture", or an aggregate view of the data, as opposed to a detailed view of the data. In such scenarios, efficient approximate computation is often acceptable. The data streaming model has gained popularity as a computational model for

[3] Following standard convention, we say that $f(n)$ is $\tilde{O}(g(n))$ if $f(n) = O\left(\left(\frac{1}{\epsilon}\right)^{O(1)} (\log m)^{O(1)} (\log n)^{O(1)} g(n)\right)$.

such applications—where incoming data (or updates) are processed very efficiently and in an online fashion using space, much less than that needed to store the data in its entirety.

A data stream $\mathcal{S}$ is viewed as a sequence of arrivals of the form $(i, v)$, where $i$ is the identity of an item that is a member of the domain $[n] = \{1, \ldots, n\}$ and $v$ is the *update* to the frequency of the item. $v > 0$ indicates an insertion of multiplicity $v$, while $v < 0$ indicates a corresponding deletion. The frequency of an item $i$, denoted by $f_i$, is the sum of the updates to $i$ since the inception of the stream, that is, $f_i = \sum_{(i,v) \text{ appears in } \mathcal{S}} v$.

In this paper, we consider the problem of estimating the $p$th frequency moment of a data stream, defined as $F_p = \sum_{i=1}^{n} |f_i|^p$, for $0 < p < 2$. Equivalently, this can be interpreted as the $p$th power of the $L_p$ norm of a vector defined by the stream. The techniques used to design algorithms and lower bounds for the frequency moment problem have been influential in the design of algorithmic and lower bound techniques for data stream computation. We briefly review the current state of the art in estimating $F_p$, with particular emphasis to the range $0 < p < 2$.

### 1.1 Review

Alon, Matias and Szegedy [1] present the seminal technique of AMS sketches for estimating $F_2$. An (atomic) AMS sketch is a random integer $X = \sum_{i=1}^{n} f_i \xi_i$, where $\{\xi_i\}_{i=1,2,\ldots,n}$ is a family of four-wise independent random variables assuming the values 1 or $-1$ with equal probability. An AMS sketch is easily maintained over a stream: for each update $(i, v)$, the sketch $X$ is updated as $X := X + v\xi_i$. Moreover, since the family $\{\xi_i\}$ is only 4-wise random, for each $i$, $\xi_i$ can be obtained from a randomly chosen cubic polynomial $h$ over a field $F$ that contains the domain of items $[n]$, ($\xi_i = 1$ if the last bit of $h(i)$ is 1 and $\xi_i = -1$ otherwise). It then follows that $\mathsf{E}\left[X^2\right] = F_2$ and $\mathsf{Var}\left[X^2\right] \leq 2F_2^2$ [1]. An estimate of $F_2$ that is accurate to within $1 \pm \epsilon$ with confidence $\frac{7}{8}$ can therefore be obtained as the average of the squares of $O(\frac{1}{\epsilon^2})$ independent sketch values.

There has also been significant study of the case $p = 0$, also known as the distinct elements problem. Alon, Matias and Szegedy [1] gave a constant factor approximation in small space. Gibbons and Tirthapura [10] showed a $(1 \pm \epsilon)$ factor approximation space $\tilde{O}(\frac{1}{\epsilon^2})$; subsequent work has improved the (hidden) logarithmic factors [2].

*$p$-stable sketches.* The use of $p$-stable sketches was pioneered by Indyk [11] for estimating $F_p$, with $0 < p < 2$. A stable sketch $S$ is defined as $Y = \sum_{i=1}^{n} f_i s_i$, where $s_i$ is drawn at random from a $p$-stable distribution, denoted $S(p, 1)$ (the second parameter of $S(\cdot, \cdot)$ is the scale factor). By the defining property of $p$-stable distribution, $Y$ is distributed as $S(p, (\sum_{i=1}^{n} |f_i|^p)^{1/p})$. In other words, $Y$ is $p$-stable distributed, with scale factor $F_p^{1/p}$. Indyk gives a technique to estimate $F_p$ by keeping $O(\frac{1}{\epsilon^2})$ independent $p$-stable sketches and returning the median of the these observations [11]. Woodruff [18] presents an $\Omega(\frac{1}{\epsilon^2})$ space lower bound for the problem of estimating $F_p$, for all $p \geq 0$, implying that the stable sketches technique is space optimal up to logarithmic factors.

Li [13] further analyses of stable sketches and suggests the use of the geometric mean estimator, that is,

$$\hat{F}_p = c \cdot |Y_1|^{1/k} |Y_2|^{1/k} \cdots |Y_k|^{1/k}$$

where $Y_1, Y_2, \ldots, Y_k$ are $k$ independent $p$-stable sketches of the data stream. Li shows the above estimator is unbiased, that is, $\mathsf{E}\big[\hat{F}_p\big] = F_p$ and $\mathsf{Var}\big[\hat{F}_p\big] \approx \frac{\pi^2 F_p^2}{6kp^2}$. It follows (by Chebychev's inequality) that returning the geometric mean of $O(\frac{1}{\epsilon^2 p^2})$ sketches returns an estimate for $F_p$ that is accurate to within factors of $(1 \pm \epsilon)$ with probability $\frac{7}{8}$. Li also shows that the geometric means estimator has lower variance than the median estimator proposed originally by Indyk [11].

*Very sparse sketches.* The "very sparse sketch" method due to Li *et al.* aims to maintain the same space and accuracy bounds, but reduce the time cost to process each update [14, 13]. Note that this technique applies only when the data satisfies some uniformity properties, whereas the preceeding techniques need no such assumptions. A very sparse (nearly) $p$-stable sketch is a linear combination of the form $W = \sum_{i=1}^{n} f_i w_i$, where $w_i$ is $P_p$ with probability $\beta/2$, $-P_p$ with probability $\beta/2$, and 0 otherwise. Here, $P_p$ is the $p$-Pareto distribution with probability tail function $\mathsf{Pr}\{P_p > t\} = \frac{1}{t^p}$, $t \geq 1$. Pareto distributions are proposed since they are much simpler to sample from as compared to stable distributions. Further, Li shows that $W$ is asymptotically $p$-stable provided $\frac{F_\infty}{F_p^{1/p}} \to 0$. Thus, very sparse sketches provide for a principled way of reducing the data stream processing time provided the data satisfies certain uniformity properties.

*Drawbacks of stable-based methods.* A drawback of the original technique of stable sketches is that, in general, for each stream update all of the $O(\frac{1}{\epsilon^2})$ stable sketches must be updated. Each sketch update requires a pseudo-random generation of a random variable drawn from a $p$-stable distribution, making it time-consuming to process each stream update. The very sparse stable sketches somewhat alleviates this problem by speeding up the processing time by a factor of approximately $1/\beta$, although the data must now satisfy certain uniformity conditions. In general, it is not possible to a-priori guarantee or verify whether the data stream satisfies the desired properties. We therefore advocate that in the absence of knowledge of the data distribution, the geometric means estimator over $p$-stable sketches is the most reliable of the known estimators—which is quite expensive.

*Contributions.* In this paper, we present a technique for estimating $F_p$, for $0 < p < 2$. Our technique requires space $O(\frac{1}{\epsilon^{2+p}} \log^2 F_1 \log n)$ to estimate $F_p$ to within relative error $(1 \pm \epsilon)$ with probability $7/8$. Further, it requires $O(\log n \log F_1)$ expected time (and $O(\log F_1 \log^2 n)$ worst-case time) to process each stream update. Thus, our technique trades a factor of $O(\frac{1}{\epsilon^p})$ space for improved processing time per stream update. From an algorithm design viewpoint, perhaps the most salient feature of the technique is that it does not recourse to stable distributions. Our technique is elementary in nature and uses simple randomization in conjunction with well-known summary structures for data streams, such as the COUNT-MIN sketch [7] and the COUNTSKETCH structure [5]. It is based on making some crucial and subtle modifications to the HSS technique [3].

*Organization.* The remainder of the paper is organized as follows. In Section 2, we review the HSS technique for estimating a class of data stream metrics. Sections 3 and 4 respectively, present a family of algorithms for estimating $F_p$ and a recursive estimator for $F_1$, respectively. Finally, we conclude in Section 5.

## 2 Review of HSS technique

In this section, we briefly review the HSS (for "Hierarchical Sampling from Sketches") procedure [3] for estimating $F_p$, $p > 2$ over data streams. Appendix A reviews the COUNTSKETCH and the COUNT-MIN algorithms for finding frequent items in a data stream and algorithms to estimate the residual first and second moments respectively of a data stream [9].

The HSS method is a general technique for estimating a class of metrics over data streams of the form:

$$\Psi(\mathcal{S}) = \sum_{i: f_i > 0} \psi(f_i). \quad (1)$$

From the input stream $\mathcal{S}$, sub-streams $\mathcal{S}_0 \ldots \mathcal{S}_L$ are created by successive sub-sampling, that is, $\mathcal{S}_0 = \mathcal{S}$ and for $1 \le l \le L$, $\mathcal{S}_l$ is obtained from $\mathcal{S}_{l-1}$ by sub-sampling each distinct item appearing in $\mathcal{S}_{l-1}$ independently with probability $\frac{1}{2}$ (hence $L = O(\log n)$). Let $k$ be a space parameter. At each level $l$, we keep a frequent items data-structure, denoted by $\mathcal{D}_l(k)$, that takes as input the sub-stream $\mathcal{S}_l$, and returns an approximation to the *top*$(k)$ items of its input stream and their frequencies. $\mathcal{D}_l(k)$ is instantiated by either the COUNT-MIN or COUNTSKETCH data structures. At level $l$, suppose that the frequent items structure at this level has an additive error of $\Delta_l(k)$ (with high probability), that is, $|\hat{f}_i - f_i| \le \Delta_l(k)$ with probability $1 - 2^{-t}$ where $t$ is a parameter. Define $F_1^{res}(k, l)$ to be (the random variable denoting) the value of $F_1$ of the sub-stream $S_l$ after removing the $k$ largest absolute frequencies; and $F_2^{res}(k, l)$ to be the corresponding value of $F_2$. The (non-random value) $F_1^{res}(k, 0)$ (respectively, $F_2^{res}(k, 0)$) is written as $F_1^{res}(k)$ (resp. $F_2^{res}(k)$).

**Lemma 1 (Lemma 1 from [3]).**

1. *For $l \ge 1$ and $k \ge 48$, $F_1^{res}(k, l) \le \frac{F_1^{res}(k)}{2^l}$ with probability $\ge 1 - 2^{-\frac{k}{24}+1}$.*
2. *For $l \ge 1$ and $k \ge 48$, $F_2^{res}(k, l) \le \frac{F_2^{res}(k)}{2^l}$ with probability $\ge 1 - 2^{-\frac{k}{24}+1}$.*

By the above lemma, let $\Delta_0 = \frac{F_1^{res}(k)}{k}$ or $\Delta_0 = \left(\frac{F_2^{res}(k)}{k}\right)^{1/2}$, depending on whether the COUNT-MIN or the COUNTSKETCH structure is being used as the frequent items structure at each level. Let $\bar{\epsilon} = \frac{\epsilon}{16}$, $T_0 = \frac{2\Delta_0}{\bar{\epsilon}}$ and $T_l = \frac{T_0}{2^l}$, $l = 0, 1, 2 \ldots, \log T_0$. The items are grouped into groups $G_0, G_1, \ldots, G_L$ as follows: $G_0 = \{i \in \mathcal{S} : \ f_i \ge T_0\}$ and $G_l = \{i \in \mathcal{S} : \ T_l \le f_i < T_{l-1}\}, \ \ 1 \le l \le L$. It follows that, with high probability, for all items of $G_l$ that are present in the random sub-stream $\mathcal{S}_l$, $\hat{f}_i \ge \frac{\Delta_l}{\bar{\epsilon}}$ and $|\hat{f}_i - f_i| \le \epsilon f_i$.

Corresponding to every stream update $(i, v)$, we use a hash-function $h : [n] \to [n]$ to map the item $i$ to a level $u = lsb(h(i))$, where, $lsb(x)$ is the position of the least significant "1" in binary representation of $x$. The stream update $(i, v)$ is then propagated to the frequent items data structures $\mathcal{D}_l$ for $0 \le l \le u$, so in effect, $i$ is included in the sub-streams from level 0 to level $u$. The hash function is assumed to be chosen randomly from a fully independent family; later we reduce the number of random bits required by a standard data streaming argument.

At the time of inference, the algorithm collects samples as follows. From each level $l$, the set of items whose estimated frequency crosses the threshold $\frac{\Delta_0}{\epsilon 2^l}$ are identified, using the frequent items structure $\mathcal{D}_l$. It is possible for the estimate $\hat{f}_{i,l}$ of an item $i$ obtained from the sub-stream $\mathcal{S}_l$ to exceed this threshold for multiple levels $l$. We therefore apply the "disambiguation-rule" of using the estimate obtained from the *lowest level* at which it crosses the threshold for that level. The estimated frequency after disambiguation is denoted as $\hat{f}_i$. Based on their disambiguated frequencies, the sampled items are sorted into their respective groups, $\bar{G}_0, \bar{G}_1, \ldots, \bar{G}_L$, as follows:

$$\bar{G}_0 = \{i | \hat{f}_i \geq T_0\} \text{ and } \bar{G}_l = \{i | T_{l-1} < \hat{f}_i \leq T_l \text{ and } i \in \mathcal{S}_l\}, 1 \leq l \leq L \ .$$

We define the estimator $\hat{\bar{\Psi}}$ and a second idealized estimator $\bar{\Psi}$ which is used for analysis only.

$$\hat{\Psi} = \sum_{l=0}^{L} \sum_{i \in \bar{G}_l} \psi(\hat{f}_i) \cdot 2^l \qquad\qquad \bar{\Psi} = \sum_{l=0}^{L} \sum_{i \in \bar{G}_l} \psi(f_i) \cdot 2^l \tag{2}$$

We now briefly review the salient points in the error analysis. Lemma 2 shows that the expected value of $\bar{\Psi}$ is close to $\Psi$.

**Lemma 2 (Lemma 2 from [3]).** *Suppose that for* $0 \leq i \leq N-1$ *and* $0 \leq l \leq L$, $|\hat{f}_{i,l} - f_i| \leq \epsilon f_i$ *with probability* $\geq 1 - 2^{-t}$. *Then* $|\mathsf{E}[\bar{\Psi}] - \Psi| \leq \Psi \cdot 2^{-t+\log L}$.

We now present a bound on the variance of the idealized estimator. The frequency group $G_l$ is partitioned into three sub-groups, namely, $\mathrm{lmargin}(G_l) = [T_l, T_l(1+\bar{\epsilon}/2)]$, $\mathrm{rmargin}(G_l) = [T_{l-1}(1-\bar{\epsilon}), T_{l-1}]$ and $\mathrm{midregion}(G_l) = [T_l(1+\bar{\epsilon}/2), T_{l-1}(1-\bar{\epsilon})]$, that respectively denote the $\mathrm{lmargin}$ (left-margin), $\mathrm{rmargin}$ (right-margin) and $\mathrm{midregion}$ of the group $G_l$. An item $i$ is said to belong to one of these regions if its true frequency lies in that region. For any item $i$ with non-zero frequency, we denote by $l(i)$ the group index $l$ such that $i \in G_l$. For any subset $T \subset [n]$, denote by $\psi(T)$ the expression $\sum_{i \in T} \psi(f_i)$. Let $\Psi^2 = \Psi^2(\mathcal{S})$ denote $\sum_{i=1}^{n} \psi^2(f_i)$.

**Lemma 3 (Lemma 3 from [3]).** *Suppose that for all* $0 \leq i \leq N-1$ *and* $0 \leq l \leq L$, $|\hat{f}_{i,l} - f_i| \leq \epsilon f_i$ *with probability* $\geq 1 - 2^{-t}$. *Then,*

$$\mathsf{Var}[\bar{\Psi}] \leq 2^{-t+L+2} \cdot \Psi^2 + \sum_{i \notin (G_0 - \mathrm{lmargin}(G_0))} \psi^2(f_i) \cdot 2^{l(i)+1} \ .$$

**Corollary 4.** *If the function* $\psi(\cdot)$ *is non-decreasing in the interval* $[0 \ldots T_0 + \Delta_0]$, *then, choosing* $t = L + \log \frac{1}{\epsilon^2} + 2$, *we get*

$$\mathsf{Var}[\bar{\Psi}] \leq \epsilon^2 \Psi^2 + \sum_{l=1}^{L} \psi(T_{l-1}) \psi(G_l) 2^{l+1} + 2\psi(T_0 + \Delta_0)\psi(\mathrm{lmargin}(G_0)) \tag{3}$$

The error incurred by the estimate $\hat{\Psi}$ is $|\hat{\Psi} - \Psi|$, and can be written as the sum of two error components using the triangle inequality.

$$|\hat{\Psi} - \Psi| \leq |\bar{\Psi} - \Psi| + |\hat{\Psi} - \bar{\Psi}| = \mathcal{E}_1 + \mathcal{E}_2$$

Here, $\mathcal{E}_1 = |\Psi - \bar{\Psi}|$ is the error due to sampling and $\mathcal{E}_2 = |\hat{\Psi} - \bar{\Psi}|$ is the error due to the approximate estimation of the frequencies. By Chebychev's inequality, $\mathcal{E}_1 = |\Psi - \bar{\Psi}| \leq |E[\bar{\Psi}] - \Psi| + 3\sqrt{\mathsf{Var}[\bar{\Psi}]}$ with probability $\frac{8}{9}$. Using Lemma 2 and Corollary 4, and choosing $t = L + \log \frac{1}{\epsilon^2} + 2$, the expression for $\mathcal{E}_1$ can be simplified as follows:

$$\mathcal{E}_1 \leq \frac{\epsilon^2 L \Psi}{2^L} + 3\big(\epsilon^2 \Psi^2 + \sum_{l=1}^{L} \psi(T_{l-1})\psi(G_l)2^{l+1} + 2\psi(T_0 + \Delta_0)\psi(\text{lmargin}(G_0))\big)^{1/2} \tag{4}$$

with probability $\frac{8}{9}$. We now present an upper bound on $\mathcal{E}_2$.

**Lemma 5.** *Suppose that for $1 \leq i \leq n$ and $0 \leq l \leq L$, $|\hat{f}_{i,l} - f_i| \leq \epsilon f_i$ with probability $\geq 1 - 2^{-t}$. Then $\mathcal{E}_2 \leq \Delta_0 \sum_{l=0}^{L} \sum_{i \in G_l} \frac{|\psi'(\xi_i)|}{2^l}$ with probability $\geq \frac{9}{10} - 2^{-t}$, where for $i \in G_l$, $\xi_i$ lies between $f_i$ and $\hat{f}_i$, and maximizes $\psi'()$.*

The analysis assumes that the hash function mapping items to levels is completely independent. We adopt a standard technique of reducing the required randomness by using a pseudo-random generator (*PRG*) of Nisan [15] along the lines of Indyk in [11] and Indyk and Woodruff in [12]. More details are provided in Appendix B.

## 3 Estimating $F_p$

In this section, we use the HSS technique with some subtle but vital modifications to estimate $F_p$ for $0 < p < 2$. We use the COUNTSKETCH structure as the frequent items structure at each level $l$.

We observe that a direct application of the HSS technique does not present an $\tilde{O}(1)$ space procedure, and so we need some novel analysis. To see this, suppose that $k$ is the space parameter of the COUNTSKETCH procedure at each level. Firstly, observe that

$$\mathcal{E}_2 \leq \Delta_0 \sum_{l=0}^{L} \sum_{i \in G_l} \frac{|\psi'(\xi_i)|}{2^l} = \sum_{l=0}^{L} \sum_{i \in G_l} \frac{\Delta_0}{2^l} \cdot p \cdot |f_i|^{p-1} \leq 2\epsilon \sum_{l=0}^{L} \sum_{i \in G_l} |f_i|^p \leq 2\epsilon^{1+p/2} F_p \leq 2\epsilon F_p$$

as required, since $p < 2$ and $\Delta_0 2^{-l} \leq \epsilon |f_i|$ for $i \in G_l$. Now, by equation (4) and using $t = L + 2\log \frac{1}{\epsilon} + 2$, we have

$$\mathcal{E}_1 \leq \epsilon^2 F_{2p} + 3\left(\epsilon^2 F_{2p} + \sum_{l=1}^{L} \left(\frac{F_2^{res}(k)}{2^l k}\right)^{\frac{p}{2}} F_p(G_l) 2^{l+1} + 2(1+\epsilon)\left(\frac{F_2^{res}(k)}{k}\right)^{\frac{p}{2}} F_p(\text{lmargin}(G_0))\right)^{\frac{1}{2}}.$$

Further, if we write $f_{rank(r)}$ to denote the $r$th largest frequency (in absolute value)

$$F_2^{res}(k) = \sum_{r>k} f_{rank(r)}^2 \leq \sum_{r>k} f_{rank(k+1)}^{2-p} f_{rank(r)}^p \leq \left(\frac{F_p}{k}\right)^{2/p-1} \cdot F_p \leq k \cdot \left(\frac{F_p}{k}\right)^{2/p}, \text{ for } p > 0$$

and hence the expression for $\mathcal{E}_1$ simplifies to

$$\mathcal{E}_1 \le \epsilon^2 F_{2p} + 3\left(\epsilon^2 F_{2p} + F_p \sum_{l=1}^{L} 2^{l+1-lp/2} F_p(G_l) + \frac{2(1+\epsilon)F_p}{k} \cdot F_p(\mathrm{lmargin}(G_0))\right)^{1/2}$$

The main problem arises with the the middle term in the above expression, namely, $F_p \sum_{l=1}^{L} 2^{l+1-lp/2} F_p(G_l)$, which can be quite large. Our approach relies on altering the group definitions to make them depend on the (randomized) quantities $F_2^{res}(k,l)$ so that the resulting expression for $\mathcal{E}_1$ becomes bounded by $O(\epsilon F_p)$. We also note that the expression for $\mathcal{E}_2$ derived above remains bounded by $\epsilon F_p$.

*Altered group definitions and estimator.* We use the COUNTSKETCH data structure as the frequent items structure at each level $l = 0, 1, \ldots, L$, with $k = O(\frac{1}{\bar{\epsilon}^{2+p}})$ buckets per hash function and $s = O(\log F_1)$ hash functions per sketch. We first observe that Lemma 1 can be slightly strengthened as follows:

**Lemma 6 (A slightly stronger version of Lemma 1).** *For $l \ge 1$ and $k \ge 48$*

1. $$F_1^{res}(k,l) \le \frac{F_1^{res}(2^{l-1}k)}{2^l} \text{ with probability } \ge 1 - 2^{-\frac{k}{24}+1}.$$
2. $$F_2^{res}(k,l) \le \frac{F_2^{res}(2^{l-1}k)}{2^l} \text{ with probability } \ge 1 - 2^{-\frac{k}{24}+1}$$

*Proof.* The result follows from the proof given for Lemma 1 [3]. □

At each level $l$, we use the COUNTSKETCH structure to estimate $F_2^{res}(k,l)$ to within an accuracy of $(1 \pm \frac{1}{4})$ with probability $1 - \frac{1}{16L}$, where, $\bar{\epsilon} = \frac{\epsilon}{32}$. Let $\bar{F}_2^{res}(k,l)$ denote $\frac{5}{4} \cdot F_2^{res}(k,l)$. We redefine the thresholds $T_0, T_1, \ldots$, as follows:

$$\Delta_l = \left(\frac{\bar{F}_2^{res}(k,l)}{k \cdot 2^l}\right)^{1/2}, \quad T_l = \frac{\Delta_l}{\epsilon}, \quad l = 0, 1, 2, \ldots, L \enspace .$$

The groups $G_0, G_1, G_2, \ldots$, are set in the usual way, using the new thresholds:

$$G_0 = \{i \mid f_i \ge T_0\} \text{ and } G_l = \{i \mid T_l \le f_i \le T_{l-1}\}$$

The estimator is defined by (1) as before.

**Lemma 7.** *Suppose $k \ge \frac{16}{\epsilon^{2+p}}$. Then, $\mathcal{E} \le 8\epsilon F_p$ with probability at least $\frac{3}{4}$.*

*Proof.* We use the property of $F_2^{res}(t)$ derived above, that for any $1 \le t \le F_0$.

$$F_2^{res}(t) \le t\left(\frac{F_p}{t}\right)^{2/p} = \frac{F_p^{2/p}}{t^{2/p-1}}, \text{ for } p > 0. \tag{5}$$

We therefore have,

$$
\begin{aligned}
T_l &= 2\left(\frac{\bar{F}_2^{res}(k,l)}{\epsilon^2 k \cdot 2^l}\right)^{1/2} \le 2\left(\frac{5F_2^{res}(k\cdot 2^{l-1})}{4\epsilon^2 k\cdot 2^l}\right)^{1/2}, \quad \text{by Lemma 6}\\
&\le 2\left(\frac{5F_p^{2/p}}{4(k\cdot 2^{l-1})^{2/p-1}\epsilon^2 k\cdot 2^l}\right)^{1/2}, \quad \text{by (5)}\\
&= \frac{1}{\epsilon}\sqrt{\frac{5}{2}}\left(\frac{2F_p}{k2^l}\right)^{1/p} \qquad (6)
\end{aligned}
$$

By equation (4), one component of $\mathcal{E}_1$ can be simplified as follows:

$$
\begin{aligned}
\mathcal{E}_{1,1} &\le \sum_{l=1}^{L} T_l^p F_p(G_l)\cdot 2^{l+1}\\
&\le \epsilon^{-p}\left(\frac{5}{2}\right)^{p/2}\sum_{l=1}^{L}\frac{2F_p}{k\cdot 2^l}F_p(G_l)2^{l+1} \quad \text{substituting (6)}\\
&\le \frac{10}{k\epsilon^p}\cdot F_p\sum_{l=1}^{L}F_p(G_l) \quad \text{since } p<2\\
&= \frac{10}{k\epsilon^p}\cdot F_p(F_p - F_p(G_0))\\
&\le \frac{5}{8}\epsilon^2 F_p^2 \quad \text{since } k \ge \frac{16}{\epsilon^{2+p}}
\end{aligned}
$$

The other component of $\mathcal{E}_1$ is

$$
\mathcal{E}_{1,2} = 2T_0^p(1+\epsilon)F_p(\mathrm{lmargin}(G_0)) \le 2\epsilon^{-p}(5/2)^{p/2}2\frac{F_p}{k}(1+\epsilon)F_p \le \frac{10}{k\epsilon^p}(1+\epsilon)F_p^2 \le \epsilon^2 F_p^2,
$$

also using $k \ge \frac{16}{\epsilon^{2+p}}$ and $\epsilon < \frac{1}{2}$. Substituting in (4), we have,

$$
\mathcal{E}_1 \le \epsilon F_p + 3(\epsilon^2 F_p^2 + \mathcal{E}_{1,1} + \mathcal{E}_{1,2})^{1/2} < 6\epsilon F_p \ . \qquad (7)
$$

Adding, the total error is bounded by

$$
\mathcal{E} \le \mathcal{E}_1 + \mathcal{E}_2 \le 8\epsilon F_p \ \Box
$$

We summarize this section in the following theorem.

**Theorem 8.** *There exists an algorithm that returns $\hat{F}_p$ satisfying $|\hat{F}_p - F_p| \le \epsilon F_p$ with probability $\frac{3}{4}$ using space $O(\frac{1}{\epsilon^{2+p}}(\log^2 n)(\log F_1))$ and processes each stream update in expected time $O(\log n \log F_1)$ and worst case time $O(\log^2 n \log F_1)$ standard arithmetic operations on words of size $\log F_1$ bits.* $\Box$

*Remarks.* 1. We note that for $0 < p < 1$, an estimator for $F_p$ with similar properties may be designed in an exactly analogous fashion by using COUNT-MIN instead of COUNTSKETCH as the frequent items structure at each level. Such an estimator would require an $\epsilon$-accurate estimation of $F_1$ (which would imply estimation of $F_1^{res}$ using standard techniques), which could either be done using Cauchy-sketches [11, 13] or using the stand-alone technique presented in Section 4. However, using Cauchy-sketches means that, in general, $O(\frac{1}{\epsilon^2})$ time is required to process each stream update. In order to maintain poly-logarithmic processing time per stream update, the technique of Section 4 may be used.

2. The space requirement of the stable sketches estimator grows as $\tilde{O}(\frac{1}{\epsilon^2 p^2})$ as a function of $p$ [13], whereas, the HSS-based technique requires space $\tilde{O}(\frac{1}{\epsilon^{2+p}})$. For small values of $p$, i.e. $p = O\left(\frac{1}{\log \epsilon^{-1}(1+\log\log \epsilon^{-1})}\right)$, the HSS technique can be asymptotically more space-efficient.

## 4 An iterative estimator for $F_1$

In this section, we use the HSS technique to present a stand-alone, iterative estimator for $F_1 = \sum_{i=1}^{n} |f_i|$. The previous section presents an estimator for $F_p$ that uses, as a sub-routine, an estimator for $F_2^{res}(k)$ at each level of the structure. In this section, we present a stand-alone estimator that uses only COUNT-MIN sketch to estimate $F_1$. The technique may be of independent interest.

The estimator uses two separate instantiations of the HSS structure. The first instantiation uses COUNT-MIN sketch structure with $k = \frac{8}{\bar{\epsilon}^3}$ buckets per hash function, and $s = O(\log G)$ hash functions, where, $G = O(F_2)$ and $\bar{\epsilon} = \frac{\epsilon}{8}$. A collection of $s_2 = O(\log \frac{1}{\delta})$ independent copies of the structure are kept for the purpose of boosting the confidence of an estimate by taking the median. The second instantiation of the HSS structure uses $k' = \frac{128}{\bar{\epsilon}^3}$ buckets per hash function (so $k' = 16k$) and $s = O(\log G)$ hash functions. For estimating $F_1$, we use a two-step procedure, namely, (1) first, we obtain an estimate of $F_1$ that is correct to within a factor of 16 using the first HSS instantiation and (2) then, we use the second instantiation to obtain an $\epsilon$-accurate estimation of $F_1$.

The first step of the estimation is done using the first instantiation of the HSS structure as follows. We set the threshold $T_0$ to a parameter $t$, $T_l = \frac{T_0}{2^l}$ and the threshold frequency for finding in group $l$ to be $\frac{T_l}{2}$. The group definitions are as defined earlier: $G_0 = [t, F_1]$, $G_l = [\frac{t}{2^l}, \frac{t}{2^{l-1}})$, $1 \le l \le L$. The disambiguation rule for the estimated frequency is as follows: if $\hat{f}_{i,l} > T_l$, then, $\hat{f}_i$ is set to the estimate obtained from the lowest of these levels. The sampled groups $\bar{G}_l$ are defined as follows.

$$\bar{G}_0 = \{i \mid \hat{f}_i \ge T_0\}, \qquad \bar{G}_l = \left\{ i \mid \frac{T_0}{2^l} \le \hat{f}_i < \frac{T_0}{2^{l-1}} \text{ and } i \in \mathcal{S}_l \right\}, 1 \le i \le L.$$

The estimators $\hat{F}_1$ and $\bar{F}_1$ are defined as before—these are now functions of $t$.

$$\hat{F}_1(t) = \sum_{l=0}^{L} \sum_{i \in \bar{G}_l} |\hat{f}_i| 2^l \qquad\qquad \bar{F}_1(t) = \sum_{l=0}^{L} \sum_{i \in \bar{G}_l} |f_i| 2^l \ .$$

*Estimator.* Let $t$ iterate over values $1, 2, 2^2, \ldots, G$ and for each value of $t$ let $\hat{F}_1^{\text{med}}(t)$ denote the median of the estimates $\hat{F}_1$ returned from the $s_1 = O(\log \frac{1}{\delta})$ copies of the HSS structure, each using independent random bits and the same value of $t$. Let $t_{\max}$ denote the largest value of $t$ satisfying

$$\hat{F}_1^{\text{med}}(t) \geq \frac{16t}{1.01\epsilon^2} \ .$$

The final estimate returned is $\hat{F}_1^{\text{med}}(t_{\max})$ using the second HSS instantiation.

*Analysis.* We first note that Lemmas 2 and 3 hold for all choices of $t$. Lemma 5 gets modified as follows.

**Lemma 9.** *Suppose that for* $1 \leq i \leq n$ *and* $0 \leq l \leq L$, $|\hat{f}_{i,l} - f_i| \leq \frac{\Delta_0}{2^l}$ *with probability* $\geq 1 - 2^{-t}$, *where,* $\Delta_0 = \frac{F_1^{res}(k)}{k}$. *Then,* $\mathcal{E}_2 \leq 16 \cdot \Delta_0 \sum_{l=0}^{L} \sum_{i \in G_l} \frac{|\psi'(\xi_i)|}{2^l}$ *with probability* $\geq \frac{9}{10} - 2^{-t}$, *where for an* $i \in G_l$, $\xi_i$ *lies between* $f_i$ *and* $\hat{f}_i$, *and maximizes* $\psi'()$. □

**Lemma 10.** *Let* $\bar{\epsilon} = \frac{\epsilon}{8}$, $k = \frac{8}{\bar{\epsilon}^3}$ *and* $\epsilon \leq \frac{1}{8}$. *Then, with probability* $1 - \delta$ *each,*

1. *For* $\frac{4F_1}{\bar{\epsilon}k} \leq t \leq \frac{8F_1}{\bar{\epsilon}k}$

$$|\hat{F}_1^{med}(t) - F_1| \leq \frac{1.01\epsilon F_1}{2} \qquad \textit{and} \qquad \hat{F}_1^{med}(t) \geq \frac{16t}{1.02\epsilon^2}$$

2. *For any* $t \geq \frac{64F_1}{\bar{\epsilon}k}$, $\hat{F}_1^{med}(t) < \frac{16t}{1.02\epsilon^2}$ *with probability* $1 - \delta$.

*Proof.* We consider the two summation terms in error term $\mathcal{E}_1$ given by equation (4) separately.

$$\mathcal{E}_{1,1} = \sum_{l=1}^{L} \frac{t}{2^l} F_1(G_l) 2^{l+1} \leq 2t(F_1 - F_0), \quad \text{and } \mathcal{E}_{1,2} = 2tF_1(\text{lmargin}(G_0)).$$

Adding, $\mathcal{E}_1 \leq (2(t + \Delta_0)F_1)^{1/2}$ .

We ignore the term $2^{-t+L+2}F_2$ in $\mathcal{E}_1$, since, by choosing $t = O(L)$, this term can be made arbitrarily small in comparison with the other two terms. Since, $|G_l| \leq \frac{F_1 \cdot 2^l}{t}$, our bound on $\mathcal{E}_2$ becomes

$$\mathcal{E}_2 \leq 16\Delta_0 \sum_{l=0}^{L} \frac{|G_l|}{2^l} \leq \frac{16F_1^2}{kt} \ .$$

Therefore, the expression for total error is $\mathcal{E}(t) = \mathcal{E}_1 + \mathcal{E}_2$

$$\mathcal{E}(t) \leq 2F_1 \left( \frac{t}{F_1} + \frac{1}{k} \right)^{1/2} + \frac{16F_1^2}{kt} \ . \tag{8}$$

Suppose $k = \frac{128}{\bar{\epsilon}^3}$ and $\frac{4F_1}{\bar{\epsilon}k} \le t \le \frac{8F_1}{\bar{\epsilon}k}$. Using $\epsilon \le \frac{1}{8}$ and $\bar{\epsilon} = \frac{\epsilon}{8}$, we have

$$\mathcal{E}(t) \le 2F_1 \left( \frac{t}{F_1} + \frac{1}{k} \right)^{1/2} + \frac{16F_1^2}{kt} \le \frac{1.01\epsilon F_1}{2} \quad . \tag{9}$$

We therefore have,

$$|\hat{F}_1^{\text{med}}(t) - F_1| \le \mathcal{E}(t) \le \frac{1.01\epsilon F_1}{2}, \quad \text{for } \frac{4F_1}{\bar{\epsilon}k} \le t \le \frac{8F_1}{\bar{\epsilon}k} \text{ with probability } 1-\delta.$$

Therefore, for $\frac{4F_1}{\bar{\epsilon}k} \le t \le \frac{8F_1}{\bar{\epsilon}k}$, with probability $1-\delta$, we have from (9) that

$$\frac{1.01\epsilon F_1}{2} \ge 2\sqrt{tF_1} \text{ and so } t \le \frac{(1.01)^2\epsilon^2 F_1}{16} \le \frac{1.01\epsilon^2 \hat{F}_1^{\text{med}}}{16}(1+0.505\epsilon) \text{ so } \hat{F}_1^{\text{med}}(t) \ge \frac{16t}{1.02\epsilon^2}. \tag{10}$$

Let $t = \frac{2^{j+2}F_1}{\bar{\epsilon}k}$ for some $j \ge 0$, and suppose that (10) is satisfied. Then by (8)

$$\mathcal{E}(t) \le 2^{(j-1)/2-2}\epsilon(1.01)F_1 + 2^{-j-2}\epsilon F_1 \le 2^{j/2}\epsilon F_1 \quad \text{with probability } 1-\delta.$$

With probability $1-\delta$, $|\hat{F}_1^{\text{med}}(t) - F_1| \le \mathcal{E}$ and so, using $\epsilon \ge \frac{1}{8}$,

$$\begin{aligned} 2^{j/2-3}F_1 \ge 2^{j/2}\epsilon F_1 \ge \mathcal{E}(t) \ge |\hat{F}_1^{\text{med}}(t) - F_1| &\ge \frac{16t}{1.02\epsilon^2} - F_1 \qquad \text{by (10)} \\ &\ge \frac{2^{j-3}F_1}{1.02} - F_1 \qquad \text{using } \bar{\epsilon} = \frac{\epsilon}{8} \text{ and } k = \frac{8}{\bar{\epsilon}^3} \end{aligned}$$

which is a contradiction for $j \ge 4$, proving claim 2. □

The correctness of the algorithm follows from the above Lemma. The space requirement of the algorithm is $O(\frac{1}{\epsilon^3}(\log^3 n)(\log^2 F_1))$ bits and the expected time taken to process each stream update is $O(\log F_1 \log \frac{1}{\delta})$ standard arithmetic operations on words of size $O(\log n)$.

## 5 Conclusions

We present a family of algorithms for the randomized estimation of $F_p$ for $0 < p < 2$ and another family of algorithms for estimating $F_p$ for $0 < p < 1$. The first algorithm family estimates $F_p$ by using the COUNTSKETCH structure and $F_2$ estimation as sub-routines. The second algorithm family estimates $F_p$ by using the COUNT-MIN sketch structure and $F_1$ estimation as a sub-routines. The space required by these algorithms are $O(\frac{1}{\epsilon^{2+p}}(\log^2 n)(\log^2 F_1)(\log \frac{1}{\delta})$ and the expected time required to process each stream update is $O(\log n \log F_1 \log \frac{1}{\delta})$. Finally, we also present a stand-alone iterative estimator for $F_1$ that only uses the COUNT-MIN sketch structure as a sub-routine.

Prior approaches to the problem of estimating $F_p$ [11, 13] used sketches of the frequency vector with random variables drawn from a symmetric $p$-stable distribution. An interesting feature of the above algorithms is that they do not require the use of stable distributions. The proposed algorithms trade an extra factor of $O(\epsilon^{-p})$ factor of space for improved procesing time (with no polynomial dependency on $\epsilon$) per stream update.

## References


1. N. Alon, Y. Matias, and M. Szegedy. "The space complexity of approximating frequency moments". *Journal of Computer and System Sciences*, 58(1):137–147, 1998.
2. Z. Bar-Yossef, T. S. Jayram, R. Kumar, D. Sivakumar, and L. Trevisan. Counting distinct elements in a data stream. In *RANDOM*, 2002.
3. L. Bhuvanagiri and S. Ganguly. "Estimating Entropy over Data Streams". In *Proc. ESA*, 2006.
4. D. Carney, U. Cetintemel, M. Cherniack, C. Convey, S. Lee, G. Seidman, M. Stonebraker, N. Tatbul, and S. Zdonik. "Monitoring streams – a new class of data management applications". In *Proc. VLDB*, 2002.
5. M. Charikar, K. Chen, and M. Farach-Colton. "Finding frequent items in data streams". In *Proc. ICALP, 2002*, pages 693–703.
6. G. Cormode and S. Muthukrishnan. "What's New: Finding Significant Differences in Network Data Streams". In *IEEE INFOCOM*, 2004.
7. G. Cormode and S. Muthukrishnan. "An Improved Data Stream Summary: The Count-Min Sketch and its Applications". *J. Algorithms*, 55(1):58–75, April 2005.
8. P. Flajolet and G.N. Martin. "Probabilistic Counting Algorithms for Database Applications". *Journal of Computer and System Sciences*, 31(2):182–209, 1985.
9. S. Ganguly, D. Kesh, and C. Saha. "Practical Algorithms for Tracking Database Join Sizes". In *Proc. FSTTCS*, 2005.
10. P. Gibbons and S. Tirthapura. Estimating simple functions on the union of data streams. In *Proc. SPAA*, 2001.
11. P. Indyk. "Stable Distributions, Pseudo Random Generators, Embeddings and Data Stream Computation". In *Proc. IEEE FOCS*, 2000.
12. P. Indyk and D. Woodruff. "Optimal Approximations of the Frequency Moments". In *Proc. ACM STOC*, 2005.
13. P. Li. "Very Sparse Stable Random Projections, Estimators and Tail Bounds for Stable Random Projections". *Manuscript*, 2006.
14. P. Li, T. J. Hastie and K. W. Church. "Very Sparse Random Projections". In *Proc. ACM SIGKDD*, 2006.
15. N. Nisan. "Pseudo-Random Generators for Space Bounded Computation". In *Proc. ACM STOC*, 1990.
16. M. Thorup and Y. Zhang. "Tabulation based 4-universal hashing with applications to second moment estimation". In *Proc. ACM SODA*, pages 615–624, January 2004.
17. M.N. Wegman and Carter J. L. "New Hash Functions and their Use in Authentication and Set Equality". *Journal of Computer and System Sciences*, 22:265–279, 1981.
18. D.P. Woodruff. "Optimal approximations of all frequency moments". In *Proc. ACM SODA* , pages 167–175, January 2004.


## A COUNT-MIN and COUNTSKETCH summaries

Given a data stream defining a set of item frequencies, *rank*$(r)$ returns an item with the $r^{th}$ largest absolute value of the frequency (ties are broken arbitrarily). We say that an item $i$ has rank $r$ if *rank*$(r) = i$. For a given value of $k$, $1 \leq k \leq n$, the set *top*$(k)$ is the set of items with rank $\leq k$. The residual second moment [5] of a data stream, denoted by $F_2^{res}(k)$, is defined as the second moment of the stream after the top-$k$ frequencies have been removed. Then, $F_2^{res}(k) = \sum_{r>k} f^2_{rank(r)}$. The residual first moment [7] of a data stream, denoted by $F_1^{res}$, is analogously defined as the $F_1$ norm of the data stream after the top-$k$ frequencies have been removed, that is, $F_1^{res} = \sum_{r>k} |f_{rank(r)}|$.

A *sketch* [1] is a random integer $X = \sum_i f_i \cdot x_i$, where, $x_i \in \{-1, +1\}$, for $i \in \mathcal{D}$ and the family of variables $\{x_i\}_{i \in \mathcal{D}}$ with certain independence properties. The family of random variables $\{x_i\}_{i \in \mathcal{D}}$ is referred to as the *sketch basis*. For any $d \geq 2$, a $d$-wise independent sketch basis can be constructed in a pseudo-random manner from a truly random seed of size $O(d \log n)$ bits as follows. Let $F$ be field of characteristic 2 and of size at least $n+1$. Choose a degree $d-1$ polynomial $g : F \to F$ with coefficients that are randomly chosen from $F$ [17]. Define $x_i$ to be 1 if the first bit (i.e., the least significant position) of $g(i)$ is 1, and define $x_i$ to be $-1$ otherwise. The $d$-wise independence of the $x_i$'s follows from an application of Wegman and Carter's universal hash functions [17].

Pair-wise independent sketches are used in [5] to design the COUNTSKETCH algorithm for finding the top-$k$ frequent items in an insert-only stream. The data structure consists of a collection of $s = O(\log \frac{1}{\delta})$ independent hash tables $U_1, U_2, \ldots, U_s$ each consisting of $8k$ buckets. A pair-wise independent hash function $h_j : [n] \to \{1, 2, \ldots, 8k\}$ is associated with each hash table that maps items randomly to one of the $8k$ buckets, where, $k$ is a space parameter. Additionally, for each table index $j = 1, 2, \ldots, s$, we keep a pair-wise independent family of random variables $\{x_{ij}\}_{i \in [n]}$, where, each $x_{ij} \in \{-1, +1\}$ with equal probability. Each bucket keeps a sketch of the sub-stream that maps to it, that is, $U_j[r] = \sum_{i:h_j(i)=r} f_i x_{ij}$, $1 \leq j \leq s, 1 \leq r \leq 8k$. An estimate $\hat{f}_i$ is returned as follows: $\hat{f}_i = \text{median}_{j=1}^{s} U_j[h_j(i)] x_{ij}$. The accuracy of estimation is stated as a function $\Delta$ of the residual second moment given parameters $k$ and $b$ is defined as [5]

$$\Delta(b, k) = 8 \left( \frac{F_2^{res}(k)}{b} \right)^{1/2} .$$

The space versus accuracy guarantees of the COUNTSKETCH algorithm is presented in Theorem 11.

**Theorem 11 ([5]).** *Let $\Delta = \Delta(k, 8k)$. Then, for any given $i \in [n]$, $\Pr\{|\hat{f}_i - f_i| \leq \Delta\} \geq 1 - \delta$. The space used is $O(k \cdot \log \frac{1}{\delta} \cdot (\log F_1))$ bits, and the time taken to process a stream update is $O(\log \frac{1}{\delta})$.*

The COUNTSKETCH algorithm can be adapted to return approximate frequent items and their frequencies. The original algorithm [5] uses a heap for maintaining the current top-$k$ items in terms of their estimated frequencies. After processing each arriving stream record of the form $(i, v)$, where, $v$ is assumed to be non-negative, an estimate for $\hat{f}_i$ is calculated using the scheme outline above. If $i$ is already in the current estimated top-$k$ heap then its frequency is correspondingly increased. If $i$ is not in the heap but $\hat{f}_i$ is larger than the current smallest frequency in the heap, then it replaces that element in the heap. This scheme is applicable to insert-only streams. A generalization of this method for strict update streams is presented in [6] and returns, with probability $1 - \delta$, (a) all items with frequency at least $(\frac{F_2^{res}(k)}{k})^{1/2}$ and, (b) does not return any item with frequency less than $(1-\epsilon)(\frac{F_2^{res}(k)}{k})^{1/2}$ using space $O\left(k\epsilon^{-2} \log n \log(k\epsilon^{-1} \log(k\epsilon^{-1})) \log F_1\right)$ bits. For general update streams, a variation of this technique can be used for retrieving items satisfying properties (a) and (b) above using space $O(\epsilon^{-2} k \log(\delta^{-1} n) \log F_1)$ bits.

The COUNT-MIN algorithm [7] for finding approximate frequent items keeps a collection of $s = O(\log \frac{1}{\delta})$ independent hash tables $T_1, T_2, \ldots, T_s$, where each hash table $T_j$ is of size $b = 2k$ buckets and uses a pair-wise independent hash function $h_j : [n] \to \{1, \ldots, 2k\}$, for $j = 1, 2, \ldots, s$. The bucket $T_j[r]$ is an integer counter that maintains the following sum $T_j[r] = \sum_{i: h_j(i) = r} f_i$. The estimated frequency $\hat{f}_i$ is obtained as $\hat{f}_i = \text{median}_{r=1}^{s} T_j[h_j(i)]$. The space versus accuracy guarantees for the COUNT-MIN algorithm is given in terms of the quantity $F_1^{res}(k) = \sum_{r > k} |f_{\mathit{rank}(r)}|$.

**Theorem 12 ([7]).** $\Pr\{|\hat{f}_i - f_i| \leq \frac{F_1^{res}(k)}{k}\} \geq 1 - \delta$ *with probability using space* $O(k \log \frac{1}{\delta} \log F_1)$ *bits and time* $O(\log \frac{1}{\delta})$ *to process each stream update.*

*Estimating* $F_1^{res}$*and* $F_2^{res}$*.* [9] presents an algorithm to estimate $F_2^{res}(k)$ to within an accuracy of $(1 \pm \epsilon)$ with confidence $1 - \delta$ using space $O(\frac{k}{\epsilon^2} \log(F_1) \log(\frac{n}{\delta}))$ bits. The data structure used is identical to the COUNTSKETCH structure. The algorithm basically removes the top-$k$ estimated frequencies from the COUNTSKETCH structure and then estimates $F_2$. Let $\hat{f}_{\tau_1}, \ldots, \hat{f}_{\tau_k}$ denote the top-$k$ estimated frequencies from the COUNTSKETCH structure. Next, the contributions of these estimates are removed from the structure, that is, $U_j[r]$**:=**$U_j[r] - \sum_{t: h_j(\tau_t) = r} f_{\tau_t} x_{j \tau_t}$. Subsequently, the *Fast-AMS* algorithm [16], a variant of the original sketch algorithm [1], is used to estimate the second moment as $\hat{F}_2^{res} = \text{median}_{j=1}^{s} \sum_{r=1}^{8k} (U_j[r])^2$. Formally, we can state:

**Lemma 13 ([9]).** *For a given integer* $k \geq 1$ *and* $0 < \epsilon < 1$*, there exists an algorithm for update streams that returns an estimate* $\hat{F}_2^{res}(k)$ *satisfying* $|\hat{F}_2^{res}(k) - F_2^{res}(k)| \leq \epsilon F_2^{res}(k)$ *with probability* $1 - \delta$ *using space* $O(\frac{k}{\epsilon^2} (\log \frac{F_1}{\delta})(\log F_1))$ *bits.*

A similar argument can be applied to estimate $F_1^{res}(s)$, where, instead of using the COUNTSKETCH algorithm, we use the COUNT-MIN algorithm for retrieving the top-$k$ estimated absolute frequencies. In parallel, a set of $s = O(\frac{1}{\epsilon^2})$ sketches based on a 1-stable distribution [11] (i.e., $Y_j = \sum_i f_i z_{ji}$, where $z_{ji}$ is drawn from a 1-stable distribution). After retrieving the top-$k$ frequencies $f_{\tau_1}, \ldots, f_{\tau_k}$ with respect to their absolute values, we reduce the sketches $Y_j$**:=**$Y_j - \sum_{r=1}^{k} f_{\tau_r} z_{j \tau_r}$ and estimate $F_1^{res}(k)$ as $\text{median}_{j=1}^{s} |Y_j|$. We summarize this in Lemma 14.

**Lemma 14.** *For a given integer* $k \geq 1$ *and* $0 < \epsilon < 1$*, there exists an algorithm for update streams that returns an estimate* $\hat{F}_1^{res}(k)$ *satisfying* $|\hat{F}_1^{res}(k) - F_1^{res}(k)| \leq \epsilon F_1^{res}(k)$ *with probability* $1 - \delta$ *using* $O(\frac{1}{\epsilon}(k + \frac{1}{\epsilon})(\log \frac{k}{\delta})(\log F_1))$ *bits.*

## B Reducing random bits by using a PRG

We use a standard technique of reducing the randomness by using a pseudo-random generator (*PRG*) of Nisan [15] along the lines of Indyk in [11] and Indyk and Woodruff in [12].

*Notation.* Let $M$ be a finite state machine that uses $S$ bits and has running time $R$. Assume that $M$ uses the random bits in $k$ segments, each segment consisting of $kb$ bits. Let $U^r$ be a uniform distribution over $\{0,1\}^r$ and for a discrete random variable $X$, let $\mathcal{F}[X]$ denote the probability distribution of $X$, treated as a vector. Let $M(x)$ denote the state of $M$ after using the random bits in $x$. The generator $G : \{0,1\}^u \rightarrow \{0,1\}^{kb}$ expands a "small" number of $u$ bits that are truly random to a sequence of $kb$ bits that "appear" random to $M$. $G$ is said to be a pseudo-random generator for a class $\mathcal{C}$ of finite state machines with parameter $\epsilon$, provided, for every $M \in \mathcal{C}$

$$\left|\mathcal{F}[M_{x \in U^{kb}}(x)] - \mathcal{F}[M_{x \in U^m}(G(x))]\right|_1 \leq \epsilon$$

where, $|y|_1$ denotes the $F_1$ norm of the vector $y$. Nisan [15] shows the following property (this version is from [11]).

**Theorem 15 ([15]).** *There exists a* PRG $G$ *for Space*$(S)$ *and Time*$(R)$ *with parameter* $\epsilon = 2^{-O(S)}$ *that requires* $O(S)$ *bits such that* $G$ *expands* $O(S \log R)$ *bits into* $O(R)$ *bits.*

This is sufficient due to the fact that we can compute the frequency moments by considering each (aggregate) frequency $f_i$ in turn and use only segments of $O(\log F_1)$ bits to store and process it.