Approximating Large Frequency Moments with Pick-and-Drop Sampling

Vladimir Braverman[*], Rafail Ostrovsky[†]

February 20, 2018

Abstract

Given data stream $D = \{p_1, p_2, \ldots, p_m\}$ of size m of numbers from $\{1, \ldots, n\}$, the frequency of i is defined as $f_i = |\{j : p_j = i\}|$. The k-th *frequency moment* of D is defined as $F_k = \sum_{i=1}^{n} f_i^k$. We consider the problem of approximating frequency moments in insertion-only streams for $k \geq 3$. For any constant c we show an $O(n^{1-2/k} \log(n) \log^{(c)}(n))$ upper bound on the space complexity of the problem. Here $\log^{(c)}(n)$ is the iterative $\log$ function. To simplify the presentation, we make the following assumptions: n and m are polynomially far; approximation error ϵ and parameter k are constants. We observe a natural bijection between streams and special matrices. Our main technical contribution is a non-uniform sampling method on matrices. We call our method a *pick-and-drop sampling*; it samples a heavy element (i.e., element i with frequency $\Omega(F_k)$) with probability $\Omega(1/n^{1-2/k})$ and gives approximation $\tilde{f}_i \geq (1-\epsilon) f_i$. In addition, the estimations never exceed the real values, that is $\tilde{f}_j \leq f_j$ for all j. As a result, we reduce the space complexity of finding a heavy element to $O(n^{1-2/k} \log(n))$ bits. We apply our method of recursive sketches and resolve the problem with $O(n^{1-2/k} \log(n) \log^{(c)}(n))$ bits.

[*]Johns Hopkins University, vova@cs.jhu.edu

[†]University of California Los Angeles, Department of Computer Science and Department of Mathematics, Email: rafail@cs.ucla.edu. Research supported in part by NSF grants CCF-0916574; IIS-1065276; CCF-1016540; CNS-1118126; CNS-1136174; US-Israel BSF grant 2008411, OKAWA Foundation Research Award, IBM Faculty Research Award, Xerox Faculty Research Award, B. John Garrick Foundation Award, Teradata Research Award, and Lockheed-Martin Corporation Research Award. This material is also based upon work supported by the Defense Advanced Research Projects Agency through the U.S. Office of Naval Research under Contract N00014-11-1-0392. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

1 Introduction

Given a sequence $D = \{p_1, p_2, \dots, p_m\}$ of size m of numbers from $\{1, \dots, n\}$, a frequency of i is defined as

$$f_i = |\{j : p_j = i\}|. \quad (1)$$

The k-th *frequency moment* of D is defined as

$$F_k = \sum_{i=1}^{n} f_i^k. \quad (2)$$

The problem of approximating frequency moments in one pass over D and using *sublinear* space has been introduced in the award-winning paper of Alon, Matias and Szegedy [1]. In particular, they observed a striking difference between "small" and "large" values of k: it is possible to approximate $F_k, k \leq 2$ in polylogarithmic space, but polynomial space is required when $k > 2$. Since 1996, approximating F_k has become one of the most inspiring problems in the theory of data streams. The incomplete list of papers on frequency moments include [18, 13, 3, 8, 4, 19, 10, 11, 12, 14, 17, 24, 6, 22, 23, 26, 28, 5, 7, 20, 2, 15, 16, 30, 21] and references therein. We omit the detailed history of the problem and refer a reader to [25, 29] for overviews.

In this paper we consider the case when $k \geq 3$. In their breakthrough paper Indyk and Woodruff [19] gave the first solution that is optimal up to a polylogarithmic factor. Numerous improvements were proposed in the later years (see the references above) and the latest bounds are due to Andoni, Krauthgamer and Onak [2] and Ganguly [15]. The latest bound by Ganguly [15] is

$$O(k^2 \epsilon^{-2} n^{1-2/k} E(p, n) \log(n) \log(nmM) / \min(\log(n), \epsilon^{4/k-2}))$$

where, $E(k, n) = (1-2/k)^{-1}(1-n^{-4(1-2/k)})$. This bound is roughly $O(n^{1-2/k} \log^2(n))$ for constant ϵ, k. The best known lower bound for insertion-only streams is $\Omega(n^{1-2/k})$, due to Chakrabarti, Khot and Sun [8].

We consider the problem of approximating frequency moments in insertion-only streams for $k \geq 3$. For any constant c we show an $O(n^{1-2/k} \log(n) \log^{(c)}(n))$ upper bound on the space complexity of the problem. Here $\log^{(c)}(n)$ is the iterative $\log$ function. To simplify the presentation, we make the following assumptions: n and m are polynomially far; approximation error ϵ and parameter k are constants. We observe a natural bijection between streams and special matrices. Our main technical contribution is a non-uniform sampling method on matrices. We call our method a *pick-and-drop sampling*; it samples a heavy element (i.e., element

i with frequency $\Omega(F_k)$) with probability $\Omega(1/n^{1-2/k})$ and gives approximation $\tilde{f}_i \geq (1-\epsilon)f_i$. In addition, the estimations never exceed the real values, that is $\tilde{f}_j \leq f_j$ for all j. As a result, we reduce the space complexity of finding a heavy element to $O(n^{1-2/k}\log(n))$ bits. We apply our method of recursive sketches [6] and resolve the problem with $O(n^{1-2/k}\log(n)\log^{(c)}(n))$ bits. We do not try to optimize the space complexity as a function of ϵ.

Overview of Main Ideas

Pick-and-drop sampling has been inspired by a very natural behavior of children. We observed the following pattern: a child picks a toy, briefly plays with it, then drops the toy and picks a new one. This pattern is repeated until the child picks the favorite toy and keeps it for a long time. Indeed, children develop algorithms for selectivity [27].

To illustrate the pick-and-drop method by example, assume that $m = r * t$ where $r = \lceil n^{1/k} \rceil$ and consider $r \times t$ matrix M with entries $m_{i,j} = p_{k(i-1)+j}$. For $m \leq n$ we aim to solve the following promise problem with probability $2/3$:

- Case 1: all frequencies are either zero or one.
- Case 2: z appears in every row of M exactly once (thus $f_z = r$). All other frequencies are either zero or one.

Consider the following sampling method. Pick r i.i.d. random numbers $I_1, \dots, I_r$, where I_i is uniformly distributed on $\{1, 2, \dots, t\}$. For each $i = 1 \dots r-1$ we check if there is a duplicate of m_{i,I_i} in the row $i+1$. If the duplicate is found then we output "Case 2" and stop; otherwise we repeat the test for $i+1$. That is, the i-th sample is "dropped," and the $(i+1)$-th sample is "picked". We repeat this experiment T times independently and output "Case 1" if no duplicate is found. Note that if the input represents Case 1 then our method will always output "Case 1." Consider Case 2 and observe that if $m_{i,I_i} = z$ then our method will output "Case 2". Indeed, since z appears in every row, the duplicate of z will be found. The probability to miss z entirely is

$$\left(1 - \frac{1}{t}\right)^{rT}. \tag{3}$$

Recall that $m \leq n, m = rt, r = \lceil n^{1/k} \rceil$. If $T = O(n^{1-2/k})$ with sufficiently large constant then the probability of error (3) is smaller than $1/3$. We conclude that our promise problem can be resolved with $O(n^{1-2/k}\log(n))$ space. Note how our solution depends on r. In general, the matrix should be carefully chosen.

Unfortunately the distribution of the frequent element in the stream can be arbitrary. Also our algorithm must recognize “noisy” frequencies that are large but negligible. Clearly, the sampling must be more intricate but, luckily, not by much. In particular, the following method works. We introduce a local counter for each sample that counts the number of times m_{i,I_i} appears in the suffix of the i-th row (this counting method is used in [1] for the entire stream). We maintain a global sample (and a global counter) as functions of the local samples and counters. Initially the global sample is the local sample of the first row. Under certain conditions, the global sample can be “dropped.” If this is the case then the local sample of the current row is “picked” and becomes the new global sample. The global sample is “dropped” when the local counter exceeds the global one. Also, the global sample is dropped if the global counter does not grow fast enough. We use function λq where λ is a parameter and q is the number of rows that the global counter survived. If the global counter is smaller than λq then the global sample is “dropped.”

In our analysis we concentrate on the case when 1 is the heavy element, but it is possible to repeat our arguments for any i. Our main technical contribution is Theorem 2.1 that claims that 1 will be outputted with probability $\Omega(\frac{f_1}{t})$ for sufficiently large f_1. Interestingly, Theorem 2.1 holds for arbitrary distributions of frequencies. In Theorem 3.6 we show that there exist r, t, λ such that a bound similar to (3) holds. We combine our new method with [6] and obtain our main result in Theorem 3.8.

2 Pick-and-Drop Sampling

Let M be a matrix with r rows and t columns and with entries $m_{i,j} \in [n]$. For $i \in [r], j \in [t], l \in [n]$ define:

$$d_{i,j} = |\{j' : j \leq j' \leq t, m_{i,j'} = m_{i,j}\}|, \tag{4}$$

$$f_{l,i} = |\{j \in [t] : m_{i,j} = l\}|, \tag{5}$$

$$f_l = |\{(i,j) : m_{i,j} = l\}|, \tag{6}$$

$$F_k = \sum_{l=1}^{n} f_l^k, G_k = F_k - f_1^k. \tag{7}$$

Note that there is a bijection between $r \times t$ matrices M and streams D of size $r \times t$ with elements $p_{it+j} = m_{i,j}$ where the definitions $(2), (1)$ and $(6), (7)$ define equivalent frequency vectors for a matrix and the corresponding stream. W.l.o.g,

we will consider streams of size $r \times t$ for some r, t and will interchange the notions of a stream and its corresponding matrix.
Let $\{I_j\}_{j=1}^r$ be i.i.d. random variables with uniform distribution on $[t]$. Define for $i = 1, \ldots, r$:

$$s_i = m_{i,I_i}, c_i = d_{i,I_i} \tag{8}$$

Let λ be a parameter. Define the following recurrent random variables:

$$S_1 = s_1, C_1 = c_1, q_1 = 1. \tag{9}$$

Also (for $i = 2, \ldots r$) if

$$(C_{i-1} < \max\{\lambda q_{i-1}, c_i\}) \tag{10}$$

then define

$$S_i = s_i, C_i = c_i, q_i = 1; \tag{11}$$

otherwise, define

$$S_i = S_{i-1}, C_i = C_{i-1} + f_{S_i,i}, q_i = q_{i-1} + 1 \tag{12}$$

Theorem 2.1. *Let M be a $r \times t$ matrix. There exist absolute constants α, β such that if*

$$\alpha(\lambda r + \frac{G_3}{\lambda t} + \frac{G_2}{t}) \leq f_1 \leq \beta t \tag{13}$$

then

$$P(S_r = 1) \geq \frac{f_1}{2t}. \tag{14}$$

Proof. Denote $Q = \{(i, j) : m_{i,j} = 1\}$. For $(i, j) \in Q$ define

$$T_{i,j} = \overline{(A_{i,j} \cup B_{i,j} \cup H_{i,j})}, \tag{15}$$

where for $i > 1$:

$$A_{i,j} = ((C_{i-1} \geq d_{i,j}) \cap (S_{i-1} \neq 1)), \tag{16}$$

for $i < r$:

$$B_{i,j} = \left(\bigcup_{h=i+1}^{r} \left(d_{i,j} + \sum_{u=i+1}^{h-1} f_{1,u} < c_h \right) \right), \tag{17}$$

$$H_{i,j} = \left(\bigcup_{h=i+1}^{r} \left(d_{i,j} + \sum_{u=i+1}^{h-1} f_{1,u} < (h-i)\lambda \right) \right), \tag{18}$$

and $A_{1,j} = B_{r,j} = H_{r,j} = \emptyset$. We have

$$((s_i = 1) \cap (S_{i-1} \neq 1) \cap \overline{A_{i,I_i}}) \subseteq ((s_i = 1) \cap (C_{i-1} < c_i)) \subseteq$$
$$((S_i = 1) \cap (q_i = 1)). \tag{19}$$

Consider the case when $S_i = 1$ and $q_i = 1$ and

$$d_{i,I_i} + \sum_{u=i+1}^{h-1} f_{1,u} \geq \max(\lambda(h-i), c_h)$$

for all $h > i$. In this case S_h will be defined by (12) and not by (11); in particular, $S_h = S_i = 1$. Therefore,

$$((S_i = 1) \cap (q_i = 1) \cap \overline{B_{i,I_i}} \cap \overline{H_{i,I_i}}) \subseteq (\bigcap_{h=i}^{r}(S_h = 1)). \tag{20}$$

Define $V_1 = ((s_1 = 1) \cap T_{1,I_1})$ and, for $i > 1$, $V_i = ((s_i = 1) \cap (S_{i-1} \neq 1) \cap T_{i,I_i})$. If follows from $(2), (20)$ that, for any $i \in [r]$:

$$V_i \subseteq (S_r = 1), \tag{21}$$

$$V_i \cap V_j = \emptyset. \tag{22}$$

Thus,

$$\sum_{i=1}^{r} P(V_i) = P\left(\cup_{i=1}^{r} V_i\right) \leq P(S_r = 1). \tag{23}$$

For any $i > 1$:

$$P(V_i) \geq P((s_i = 1) \cap T_{i,I_i}) - P(s_i = S_{i-1} = 1).$$

Also,

$$\sum_{i=2}^{r} P(s_i = S_{i-1} = 1) \leq \sum_{i=2}^{r} P((s_i = 1) \cap (\cup_{h \neq i}(s_h = 1))) \leq$$
$$(\sum_{i=1}^{r} P(s_i = 1))^2 = \left(\frac{f_1}{t}\right)^2.$$

For any fixed $(i,j) \in Q$ events $I_i = j$ and $T_{i,j}$ are independent. Indeed, $A_{i,j}$ is defined by $\{S_{i-1}, C_{i-1}\}$ that, in turn, is defined by $\{I_1, \ldots, I_{i-1}\}$. Similarly, $B_{i,j}$ is defined by $\{I_{i+1}, \ldots, I_r\}$. Note that $H_{i,j}$ is a deterministic event. By definition, $\overline{\{I_1, \ldots, I_{i-1}, I_{i+1}, \ldots, I_r\}}$ are independent of I_i; thus event $I_i = j$ and $T_{i,j} = \overline{(A_{i,j} \cup B_{i,j} \cup H_{i,j})}$ are independent. Thus,

$$\sum_{i=1}^{r} P((s_i = 1) \cap T_{i,I_i}) = \sum_{(i,j)\in Q} P((I_i = j) \cap T_{i,j}) =$$

$$\sum_{(i,j)\in Q} P(I_i = j) P(T_{i,j}) = \frac{1}{t} \sum_{(i,j)\in Q} P(T_{i,j}). \tag{24}$$

Thus,

$$P(S_r = 1) \geq \frac{1}{t} \sum_{(i,j)\in Q} P(T_{i,j}) - \left(\frac{f_1}{t}\right)^2.$$

Lemma 2.2 implies that $\sum_{(i,j)\in Q} P(T_{i,j}) \geq 0.8 f_1$. Thus if $\beta < 0.3$ then:

$$P(S_r = 1) \geq \frac{f_1}{t}(0.8 - \frac{f_1}{t}) \geq \frac{f_1}{2t}.$$

Here we only use the second part of (13). The first part is used in the proof of Lemma 2.2. ☐

Lemma 2.2. *There exist absolute constants* α, β *such that* (13) *implies*

$$\sum_{(i,j)\in Q} P(T_{i,j}) > 0.8 f_1.$$

It follows from Lemmas 2.9, 2.17, 2.14 and the union bound that there exists at least $0.97 f_1$ pairs $(i,j) \in Q$ such that $P(A_{i,j} \cup B_{i,j} \cup H_{i,j}) \leq 0.02$. Recall that $T_{i,j} = \overline{(A_{i,j} \cup B_{i,j} \cup H_{i,j})}$; the lemma follows.

2.1 Events of type A

For $(i,j) \in Q$ s.t. $i > 1$ and for $l > 1$ define:

$$Y_{l,(i,j)} = \mathbf{1}_{A_{i,j}} \mathbf{1}_{(S_{i-1}=l)},$$

$$Y_{l,i} = \sum_{j\in[t],(i,j)\in Q} Y_{l,(i,j)},$$

$$Y_l = \sum_{i=2}^{r} Y_{l,i},$$

$$Y = \sum_{l=2}^{n} Y_l,$$

Fact 2.3. *$C_i \leq f_{S_i}$. Also, if $q_i = 1$ then $C_i \leq f_{S_i,i}$.*

Proof. Follows directly from $(11), (12)$. It is sufficient to prove that, for any i, there exists a set Q_i such that $C_i = |Q_i|$ and simultaneously Q_i is a subset of $\{(i', j) : m_{i',j} = S_i, i' \leq i\}$. We prove the above claim by induction on i. For $i = 1$ the claim is true since we can define $Q_1 = \{(1, j) : j \geq I_1\}$. For $i > 2$ the description of the algorithm implies the following. If $q_i = 1$ then we can put $Q_i = \{(i, j) : j \geq I_i\}$. If $q_i > 1$ then define $Q_i = Q_{i-1} \cup \{(i, j) : m_{i,j} = S_i\}$. Note that in this case $S_i = S_{i-1}$. The second part follows from the description of the algorithms: if $p_i = 1$ then $C_i = c_i, S_i = s_i$ and $c_i = d_{i,I_i}(s_i) \leq f_{s_i,i}$. □

Fact 2.4.

1. *$Y_{l,i} \leq f_l$,*
2. *If $q_{i-1} = 1$ then $Y_{l,i} \leq f_{l,i-1}$.*

Proof. Let $(i, j) \in Q$ be such that $d_{i,j} > f_l$; then:

$$Y_{l,(i,j)} = \mathbf{1}_{(C_{i-1} \geq d_{i,j})} \mathbf{1}_{(S_{i-1}=l)} = \mathbf{1}_{(f_l \geq C_{i-1})} \mathbf{1}_{(C_{i-1} \geq d_{i,j})} \mathbf{1}_{(S_{i-1}=l)}.$$

We use Fact 2.3 for the last equality. Thus, $Y_{l,(i,j)} = 0$. Definition of $d_{i,j}$ implies $|\{j : (i, j) \in Q, d_{i,j} \leq f_l\}| \leq f_l$ for any fixed i and l. Thus,

$$Y_{l,i} = \sum_{j \in [t], (i,j) \in Q} Y_{l,(i,j)} \leq f_l.$$

Part 2 following by repeating the above arguments and using the second statement of Fact 2.3. □

Definition 2.5. *Let $1 \leq r_1 \leq r_2 \leq r$ and $l \in [n]$. Call a pair $[r_1, r_2]$ an l-epoch if*

$$\forall i = r_1, \ldots, r_2 : \quad S_i = l,$$

and

$$q_{r_1} = q_{r_2+1} = 1,$$

and

$$\forall i = r_1 + 1, \ldots, r_2 : \quad q_i = q_{i-1} + 1.$$

Lemma 2.6. *Let $[r_1, r_2]$ be an l-epoch. If $r_2 > r_1$ then*

$$r_2 - r_1 \leq \frac{1}{\lambda} \sum_{i=r_1}^{r_2-1} f_{l,i}.$$

Proof. First, observe that $q_{r_2-1} = r_2 - r_1$. Second, $q_i > 1$ implies that S_i is defined by (12) and not by (11) for all $r_1 < i \leq r_2$. In particular, $C_{r_1} \leq f_{l,r_1}$ and for $r_1 < i \leq r_2$ we have $C_i = C_{i-1} + f_{l,i}$. Thus,

$$C_{r_2-1} \leq \sum_{i=r_1}^{r_2-1} f_{l,i}.$$

Third, $C_{r_2-1} \geq \lambda q_{r_2-1}$ since (10) must be false for $i = r_2$. Therefore,

$$r_2 - r_1 = q_{r_2-1} \leq \frac{1}{\lambda} C_{r_2-1} \leq \frac{1}{\lambda} \sum_{i=r_1}^{r_2-1} f_{l,i}.$$

□

Lemma 2.7. $Y_l \leq \frac{f_l^2}{\lambda} + f_l$.

Proof. Observe that the set $\{i : S_i = l\}$ is a collection of disjoint l-epochs. Recall that $Y_l = \sum_{i=2}^{r} Y_{l,i}$ and $Y_{l,i}$ is non-zero only if S_{i-1} is equal to l. Thus we can rewrite Y_l as:

$$Y_l = \sum_{(r_1,r_2) \text{is an } l\text{-epoch}} \left(\sum_{i=r_1+1}^{r_2+1} Y_{l,i} \right).$$

For any epoch such that $r_2 > r_1$ we have by Lemmas 2.4 and 2.6:

$$\sum_{i=r_1+1}^{r_2} Y_{l,i} \leq (r_2 - r_1) f_l \leq \frac{f_l}{\lambda} \sum_{i=r_1}^{r_2-1} f_{l,i}.$$

Since all epochs are disjoint we have

$$Y_l = \sum_{(r_1<r_2) \text{is an } l\text{-epoch}} \left(\sum_{i=r_1+1}^{r_2+1} Y_{l,i} \right) + \sum_{(r_1=r_2) \text{is an } l\text{-epoch}} Y_{l,r_2+1} =$$

$$\sum_{(r_1<r_2) \text{is an } l\text{-epoch}} \left(\sum_{i=r_1+1}^{r_2} Y_{l,i} \right) + \sum_{(r_1,r_2) \text{is an } l\text{-epoch}} Y_{l,r_2+1} \leq$$

$$\frac{f_l}{\lambda} \sum_{(r_1<r_2) \text{is an } l\text{-epoch}} \left(\sum_{i=r_1}^{r_2-1} f_{l,i} \right) + \sum_{(r_1,r_2) \text{is an } l\text{-epoch}} f_{l,r_2+1} \leq$$

$$\frac{f_l^2}{\lambda} + f_l.$$

□

Lemma 2.8. $P(Y_l > 0) \leq \frac{f_l}{t}$.

Proof. Since I_i are independent and $0 \leq \frac{f_{l,i}}{t} \leq 1$ we can apply Fact 2.10:

$$P(\cap_{i=1}^r (m_{i,I_i} \neq l)) = \prod_{i=1}^r (1 - \frac{f_{l,i}}{t}) \geq (1 - \frac{f_l}{t}).$$

Thus,

$$P(Y_l > 0) \leq P(\cup_{i=1}^r (m_{i,I_i} = l)) \leq \frac{f_l}{t}. \tag{25}$$

□

Lemma 2.9. *There exists an absolute constant* α *such that* (13) *implies that* $P(A_{i,j}) \leq 0.01$ *for at least* $0.99 f_1$ *pairs* $(i,j) \in Q$.

Proof. ¿From Lemmas 2.7, 2.8:

$$E(Y_l) \leq \frac{f_l}{t}(\frac{f_l^2}{\lambda} + f_l),$$

$$E(Y) = \sum_{l=2}^n E(Y_l) \leq \frac{G_3}{\lambda t} + \frac{G_2}{t}.$$

If follows that $\sum_{(i,j)\in Q} \mathbf{1}_{A_{i,j}} = Y$. Recall that by (13):

$$|Q| = f_1 \geq \alpha(\frac{G_3}{\lambda t} + \frac{G_2}{t}) \geq \alpha E(\sum_{(i,j)\in Q} \mathbf{1}_{A_{i,j}}).$$

Fact 2.11 implies that there exists an absolute constant α such that the lemma is true. □

The following fact is a well known. For completeness we present the proof.

Fact 2.10. *Let* $\alpha_1, \ldots, \alpha_r$ *be real numbers in* $[0,1]$. *Then*

$$\prod_{i=1}^r (1 - \alpha_i) \geq 1 - (\sum_{i=1}^r \alpha_i).$$

Proof. If $\sum_{i=1}^r \alpha_i \geq 1$ then

$$\prod_{i=1}^r (1 - \alpha_i) \geq 0 \geq 1 - (\sum_{i=1}^r \alpha_i).$$

Thus we can assume that $\sum_{i=1}^{r} \alpha_i < 1$. We will prove the claim by induction on r. For $r = 2$ we obtain $(1-\alpha_1)(1-\alpha_2) = (1-\alpha_1-\alpha_2 x+\alpha_1\alpha_2) \geq (1-\alpha_1-\alpha_2)$. For $r > 2$, we have, by induction,

$$\prod_{i=1}^{r}(1-\alpha_i) \geq (1-(\sum_{i=1}^{r-1}\alpha_i))(1-\alpha_r) \geq 1-(\sum_{i=1}^{r}\alpha_i).$$

□

Fact 2.11. *Let $X_1, \ldots, X_u$ be a sequence of indicator random variables. Let $S = \{i : P(X_i = 1) \leq \nu\}$. If $E(\sum_{i=1}^{u} X_i) \leq \mu u$ then $|S| \geq (1 - \frac{\mu}{\nu})u$.*

Proof. Indeed,

$$\mu u \geq \sum_{i \notin S} P(X_i = 1) \geq \nu(u - |S|).$$

□

2.2 Events of type B

For $(i, j) \in Q$ let $Z_{(i,j)} = \mathbf{1}_{B_{i,j}}$. Let $Z = \sum_{(i,j)\in Q} Z_{(i,j)}$. We use arguments that are similar to the ones from the previous section. To stress the similarity we abuse the notation and denote by $Y_{l,h,(i,j)}$ the indicator of the event that $h > i+1$, $s_h = l$ and

$$\left(d_{i,j} + \sum_{u=i+1}^{h-1} f_{1,u}\right) < c_h.$$

Define $Y_{l,h} = \sum_{(i,j)\in Q} Y_{l,h,(i,j)}$, $Y_l = \sum_{h=1}^{r} Y_{l,h}$.

Fact 2.12. $Y_l \leq f_l$.

Proof. Repeating the arguments from Fact 2.4 we have $c_h \mathbf{1}_{s_h=l} \leq f_{l,h}$ and thus $Y_{l,h} \leq f_{l,h}$. □

Fact 2.13. $P(Y_l > 0) \leq \frac{f_l}{t}$.

Proof. The proof is identical to the proof of Lemma 2.8. □

Lemma 2.14. *There exist absolute constants α, β such that* (13) *implies that $P(B_{i,j}) \leq 0.01$ for at least $0.99 f_1$ pairs $(i, j) \in Q$.*

Proof. Denote $Y = \sum_{l=1}^{n} Y_l$. If follows that $Z \leq Y$ and $E(Z) \leq E(Y)$. By Facts 2.13 and 2.12 if follows that $E(Y_l) \leq \frac{f_l^2}{t}$. Thus by (13):

$$E(Z) \leq E(Y) \leq \frac{F_2}{t} = \frac{G_2}{t} + f_1\frac{f_1}{t} \leq (\alpha+\beta)f_1.$$

We repeat the arguments from Lemma 2.9. □

2.3 Events of type H

Definition 2.15. *Let $U = \{u_1, \dots, u_t\}$ and $W = \{w_1, \dots, w_t\}$ be two sequences of non-negative integers. Let (i, j) be a pair such that $1 \le i \le t$ and $1 \le j \le u_i$. Denote (i, j) as a* loosing *pair (w.r.t. sequences U, W) if there exists $h, i \le h \le t$ such that:*

$$-j + \sum_{s=i}^{h} (u_s - w_s) < 0.$$

Denote any pair that is not a loosing pair as a a winning *pair.*

In this section we consider the following pair (U, W) of sequences. For $i = 1, \dots, r$ let $u_i = f_{1,i}$ and $w_i = \lambda$.

Fact 2.16. *If (i, j) is a winning pair w.r.t. (U, W) then $H_{i,j'}$ does not occur where j' is such that $m_{i,j'} = 1$ and $d_{i,j'} = f_{1,i} - j + 1$.*

Proof. By Definition 2.15, for every $i \le h \le r$:

$$-j + \sum_{l=i}^{h} u_l \ge \sum_{l=i}^{h} w_l. \tag{26}$$

Since $\sum_{l=i}^{h} w_i = (h - i + 1)\lambda$ and $d_{i,j'} = f_{1,i} - j + 1$ we have for every $i \le h \le r$:

$$d_{i,j'} + \sum_{l=i+1}^{h} d_{l,1} = f_{i,1} - j + 1 + \sum_{l=i+1}^{h} f_{l,1} =$$

$$-j + 1 + \sum_{l=i}^{h} u_l \ge -j + \sum_{l=i}^{h} u_l \ge \sum_{l=i}^{h} w_l = (h - i + 1)\lambda.$$

Substitute h by $h - 1$ (for $h > i$):

$$d_{i,j'} + \sum_{l=i+1}^{h-1} d_{l,1} \ge (h - i)\lambda.$$

Thus $H_{i,j'}$ does not occur, by (18). □

Lemma 2.17. *There exists an absolute constant α such that (13) implies that $H_{i,j}$ does not occur for at least $0.99 f_1$ pairs $(i, j) \in W$.*

Proof. By Lemma 2.20 there exist at least

$$\sum_{i=1}^{r}(u_i - w_i)$$

winning pairs (i,j) w.r.t. the (U,W). Also, $\sum_{i=1}^{r} u_i = \sum_{i=1}^{r} f_{1,i} = f_1$ and $\sum_{i=1}^{r} w_i = \lambda r$. Thus there exist at least $f_1 - \lambda r$ winning pairs (i,j) w.r.t. the (U,W). In the statement of Fact 2.16 the mapping from j to j' is a bijection; thus there exist at least $f_1 - \lambda r$ pairs (i,j') s.t. $m_{i,j'} = 1$ and $H_{i,j'}$ does not occur. By (13) we have $f_1 \geq \alpha\lambda r$ and the lemma follows. □

Definition 2.18. *Let $U = \{u_1,\ldots,u_t\}$ and $W = \{w_1,\ldots,w_t\}$ be two sequences of non-negative integers. Let $1 \leq h < t$. Let U', W' be two sequences of size $t-h$ defined by $p'_i = u_{i+h}$, $q'_i = w_{i+h}$ for $i = 1,\ldots,t-h$. Denote U', W' as h-tail of the sequences U, W.*

Fact 2.19. *If (i,j) is a winning pair w.r.t. h-tail of U,W then $(i+h,j)$ is a winning pair w.r.t. U,W. If (i,j) is a winning pair w.r.t. h-tail of U,W then (i,j) is a winning pair w.r.t. U,W.*

Proof. Follows directly from Definitions 2.15 and 2.18. □

Lemma 2.20. *If $\sum_{s=1}^{t}(u_s - w_s) > 0$ then there exist at least $\sum_{s=1}^{t}(u_s - w_s)$ winning pairs.*

Proof. We use induction on t. For $t = 1$, any pair $(1,j)$ is winning if $1 \leq j \leq u_1 - w_1$. Consider $t > 1$ and apply the following case analysis.

1. Assume that there exist $1 \leq h < t$ such that $\sum_{s=1}^{h}(u_s - w_s) \leq 0$. Consider the h-tail of U,W. By induction and by Fact 2.19, there exist at least $\sum_{s=h+1}^{t}(u_s - w_s) \geq \sum_{s=1}^{t}(u_s - w_s)$ winning pairs w.r.t. U,W.

2. Assume that $(1,u_1)$ is a winning pair; it follows that $(1,j)$, $j < u_1$ is a winning pair as well. If $\sum_{s=2}^{t}(u_s - w_s) > 0$ then, by induction and by Fact 2.19, there exist at least $\sum_{s=2}^{t}(u_s - w_s)$ winning pairs of the form (i,j) where $i > 1$. In total there are $u_1 + \sum_{s=2}^{t}(u_s - w_s) \geq \sum_{s=1}^{t}(u_s - w_s)$ winning pairs w.r.t. U,W. The case when $\sum_{s=2}^{t}(u_s - w_s) < 0$ is trivial.

3. Assume that $(1),(2)$ do not hold. Then $u_1 > 0$. Indeed otherwise $u_1 - w_1 \leq 0$ and thus (1) is true. Also $(1,1)$ is a winning pair. Indeed, otherwise there exists $1 \leq h < t$ such that $-1 + \sum_{i=1}^{h}(u_i - w_i) < 0$. All numbers are integers thus $\sum_{i=1}^{h}(u_i - w_i) \leq 0$ and (1) is true. Thus $(1,1)$ is a winning pair and

$(1, u_1)$ is not a winning pair (by (2)). Therefore there exist $1 < u \leq u_1$ such that $(1, u-1)$ is a winning pair and $(1, u)$ is not a winning pair. In particular, there exists $1 \leq h < t$ such that

$$-u + \sum_{s=1}^{h} (u_s - w_s) < 0.$$

On the other hand $(1, u-1)$ is a winning pair thus

$$0 \leq 1 - u + \sum_{s=1}^{h} (u_s - w_s).$$

All numbers are integers and thus we conclude that

$$\sum_{s=1}^{h} (u_s - w_s) = u - 1.$$

Consider the h-tail of U, W. By induction, there exists at least

$$\sum_{i=h+1}^{t} (u_i - w_i) = \sum_{i=1}^{t} (u_i - w_i) - (u-1)$$

winning pairs w.r.t. the h-tail of U, W. By Fact 2.19 there exist at least as many winning pairs w.r.t. U, W of the form (i, j) where $i > 1$. By properties of u there exist additional $(u-1)$ winning pairs of the form $(1, j), j \leq u-1$. Summing up we obtain the fact.

□

3 The Streaming Algorithm

Fact 3.1. *Let $v_1, \ldots, v_n$ be a sequence of non-negative numbers and let $k > 2$. Then*

$$\left(\sum_{i=1}^{n} v_i^2\right)^{(k-1)} \leq \left(\sum_{i=1}^{n} v_i^k\right)\left(\sum_{i=1}^{n} v_i\right)^{(k-2)}$$

Proof. Define $\lambda_i = \frac{v_i}{\sum_{j=1}^{n} v_j}$. Since $g(x) = x^{k-1}$ is convex on the interval $[0, \infty)$ we can apply Jensen's inequality and obtain:

$$\left(\frac{\sum_{i=1}^{n} v_i^2}{\sum_{i=1}^{n} v_i}\right)^{(k-1)} = (\sum_{i=1}^{n} \lambda_i v_i)^{(k-1)} \leq (\sum_{i=1}^{n} \lambda_i v_i^{(k-1)}) = \frac{\sum_{i=1}^{n} v_i^k}{\sum_{i=1}^{n} v_i}.$$

□

Let D be a stream. Define

$$\psi = \frac{n^{1-(1/k)}G_k^{1/k}}{F_1}, \delta = 2^{\lceil 0.5\log_2(\psi)\rceil}, t = \left\lceil \frac{\delta F_1}{n^{1/k}} \right\rceil, \lambda = \left\lceil \frac{F_1\delta^3}{n} \right\rceil, \quad (27)$$

where we use (2) to define F_k. We will make the following assumptions:

$$f_1 \leq 0.1F_1, \quad t \leq F_1, \quad F_1(\mod t) = 0. \quad (28)$$

Then it is possible to define a matrix a $r \times t$ matrix M, where $r = F_1/t$ and with entries $m_{i,j} = p_{ir+j}$.

Fact 3.2. $1 \leq \delta \leq 2n^{(k-1)/2k.}$

Proof. Indeed, $G_1 \leq G_k^{1/k} n^{1-1/k}$ by Hölder inequality and since $f_1 \leq 0.1F_1$ by (28) we have $\psi \geq 0.5$; thus, $\lceil 0.5\log_2(\psi)\rceil \geq 0$ and the lower bound follows. Also, $F_k^{1/k}$ is the L_k norm for the frequency vector since since all frequencies are non-negative. Since $L_k \leq L_1$ we conclude that $\psi \leq n^{1-1/k}$ and the fact follows. □

Observe that there exists a frequency vector with $\delta = O(1)$: put $f_j = 1$ for all $i \in [n]$. At the same time there exists a vector with $\delta = \Omega(n^{(k-1)/2k})$: put $f_1 = n$ and $f_j = 1$ for $j > 2$. It is not hard to see that if δ is sufficiently large then a naïve sampling method will find a heavy element. For example, in the latter case, the heavy element occupies half of the stream.

Fact 3.3. $\lambda r \leq 4G_k^{1/k}$.

Proof. Recall that $F_1 = rt$. The fact follows from the definitions of λ and t. □

Fact 3.4.

$$\frac{G_2}{t} \leq G_k^{1/k}.$$

Proof. Define $\alpha = \frac{k-3}{2(k-2)}$. We have by Hölder inequality:

$$G_2^{\alpha} \leq G_k^{\frac{2\alpha}{k}} n^{\alpha(1-\frac{2}{k})} = G_k^{\frac{k-3}{k(k-2)}} n^{\frac{k-3}{2k}}. \quad (29)$$

Also, by Fact 3.1

$$G_2^{1-\alpha} = G_2^{\frac{k-1}{2(k-2)}} \leq G_k^{\frac{1}{2(k-2)}} G_1^{\frac{1}{2}}. \quad (30)$$

Thus,

$$G_2 \leq G_k^{\frac{k-3}{k(k-2)}} n^{\frac{k-3}{2k}} G_k^{\frac{1}{2(k-2)}} F_1^{\frac{1}{2}} =$$

$$G_k^{\frac{1}{k}} \frac{F_1}{n^{1/k}} \left(\frac{G_k^{\frac{1}{k}} n^{\frac{k-1}{k}}}{F_1} \right)^{1/2} = tG_k^{\frac{1}{k}}.$$

□

Fact 3.5. $\frac{G_3}{\lambda t} \leq G_k^{1/k}$.

Proof. By Hölder inequality,

$$G_3 \leq G_k^{3/k} n^{1-(3/k)}. \tag{31}$$

Thus

$$\frac{G_3}{\lambda t} = \frac{n^{1+(1/k)} G_3}{F_1^2 \delta^4} \leq \frac{n^{2-(2/k)} G_k^{3/k}}{F_1^2 \delta^4} \leq G_k^{1/k}.$$

□

Theorem 3.6. *Let M be a $r \times t$ matrix such that (27) is true. Then there exist absolute constants α, β such that*

$$\alpha G_k^{1/k} \leq f_1 \leq \beta t \tag{32}$$

imply

$$P(S_r = 1) \geq \frac{\delta}{2n^{1-(2/k)}}. \tag{33}$$

Proof. By (32) and Facts 3.5, 3.4, 3.3:

$$6\alpha(\lambda r + \frac{G_3}{\lambda t} + \frac{G_2}{t}) \leq f_1 \leq \beta t.$$

Also, (27) implies $f_1/t \geq \frac{\delta}{n^{1-(2/k)}}$. Thus, (33) follows from Theorem 2.1. □

Algorithm 1 describes our implementation of the pick-and-drop sampling.

Theorem 3.7. *Denote $f_i^k > 100 \sum_{j \neq i} f_j^k$ as a* heavy *element. There exist a (constructive) algorithm that makes one pass over the stream and uses $O(n^{1-2/k} \log(n))$ bits. The algorithm outputs a pair $(i, \tilde{f}_i)$ such that $\tilde{f}_i \leq f_i$ with probability 1. If there exists a heavy element f_i then also with constant probability the algorithm will output $(i, \tilde{f}_i)$ such that $(1 - \epsilon) f_i \leq \tilde{f}_i$.*

Proof. Define t as in (27). W.l.o.g., we can assume that F_1 is divisible by t. Note that if $t > F_1$ or $f_1 \geq 0.1 F_1$ then it is possible to find a heavy element with $O(n^{1-2/k})$ bits by existing methods such as [9]. Otherwise, a stream D defines a matrix M for which we compute $O(n^{1-2/k}/\epsilon\delta)$ independent pick-and-drop samples. Since we do not know the value of δ we should repeat the experiment for all possible values of δ. Output the element with the maximum frequency. With constant probability the output of the pick-and-drop sampling will include a $(1-\epsilon)$ approximation of the frequency f_i. Thus, there will be no other f_j that can give

Algorithm 1 P&D(M, r, t, λ)

Generate i.i.d. r.v. $\{I_j\}_{j=1}^r$ with uniform distribution on $[t]$.
$S_1 = m_{1,I_1}$,
$C_1 = d_{1,I_1}$,
$q_1 = 1$.
for $i = 2 \to r$ **do**
  compute $s_i = m_{i,I_i}, c_i = d_{i,I_i}$
  if $(C_{i-1} < \max\{\lambda q_{i-1}, c_i\})$ **then**
    $S_i = s_i$,
    $C_i = c_i$,
    $q_i = 1$
  else
    $S_i = S_{i-1}$,
    $C_i = C_i + f_{S_i,l}$,
    $q_i = q_{i-1} + 1$
  end if
end for
Output (S_r, C_r).

a larger approximation and replace a heavy element. The total space will define geometric series that sums to $O(n^{1-2/k}\log(n))$.

If we know F_1 ahead of time then we can compute the value of t for any possible δ and thus solve the problem in one pass. However, one can show that the well-known doubling technique (when we double our parameter t each time the size of the stream doubles) will work in our case and thus one pass is sufficient even without knowing F_1. □

Recall that in [6] we developed a method of recursive sketches with the following property: given an algorithm that finds a heavy element and uses memory $\mu(n)$, it is possible to solve the frequency moment problem in space $O(\mu(n)\log^{(c)}(n))$. In [6] we applied recursive sketches with the method of Charikar et.al. [9]. Thus, we can replace the method from [9] with Theorem 3.7 and obtain:

Theorem 3.8. *Let ϵ and k be constants. There exists a (constructive) algorithm that computes $(1 \pm \epsilon)$-approximation of F_k, uses $O(n^{1-2/k}\log(n)\log^{(c)}(n))$ memory bits, makes one pass and errs with probability at most $1/3$.*

References

[1] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *J. Comput. Syst. Sci.*, 58(1):137–147, 1999.

[2] Alexandr Andoni, Robert Krauthgamer, and Krzysztof Onak. Streaming algorithms via precision sampling. In *FOCS*, pages 363–372, 2011.

[3] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *J. Comput. Syst. Sci.*, 68(4):702–732, 2004.

[4] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, D. Sivakumar, and Luca Trevisan. Counting distinct elements in a data stream. In *RANDOM '02: Proceedings of the 6th International Workshop on Randomization and Approximation Techniques*, pages 1–10, London, UK, 2002. Springer-Verlag.

[5] Paul Beame, T. S. Jayram, and Atri Rudra. Lower bounds for randomized read/write stream algorithms. In *STOC '07: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 689–698, New York, NY, USA, 2007. ACM.

[6] Vladimir Braverman and Rafail Ostrovsky. Recursive sketching for frequency moments. *CoRR*, abs/1011.2571, 2010.

[7] Amit Chakrabarti, Graham Cormode, and Andrew McGregor. Robust lower bounds for communication and stream computation. In *STOC '08: Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 641–650, New York, NY, USA, 2008. ACM.

[8] Amit Chakrabarti, Subhash Khot, and Xiaodong Sun. Near-optimal lower bounds on the multi-party communication complexity of set disjointness. In *IEEE Conference on Computational Complexity*, pages 107–117, 2003.

[9] Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. In *ICALP '02: Proceedings of the 29th International Colloquium on Automata, Languages and Programming*, pages 693–703, London, UK, 2002. Springer-Verlag.

[10] Don Coppersmith and Ravi Kumar. An improved data stream algorithm for frequency moments. In *SODA*, pages 151–156, 2004.

[11] Graham Cormode, Mayur Datar, Piotr Indyk, and S. Muthukrishnan. Comparing data streams using hamming norms (how to zero in). *IEEE Trans. on Knowl. and Data Eng.*, 15(3):529–540, 2003.

[12] Joan Feigenbaum, Sampath Kannan, Martin Strauss, and Mahesh Viswanathan. An approximate l1-difference algorithm for massive data streams. In *FOCS '99: Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, page 501, Washington, DC, USA, 1999. IEEE Computer Society.

[13] Philippe Flajolet and G. Nigel Martin. Probabilistic counting algorithms for data base applications. *J. Comput. Syst. Sci.*, 31(2):182–209, 1985.

[14] Sumit Ganguly. Estimating frequency moments of data streams using random linear combinations. In *APPROX-RANDOM*, pages 369–380, 2004.

[15] Sumit Ganguly. Polynomial estimators for high frequency moments. *CoRR*, abs/1104.4552, 2011.

[16] Sumit Ganguly. A lower bound for estimating high moments of a data stream. *CoRR*, abs/1201.0253, 2012.

[17] Sumit Ganguly and Graham Cormode. On estimating frequency moments of data streams. In *APPROX '07/RANDOM '07: Proceedings of the 10th International Workshop on Approximation and the 11th International Workshop on Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 479–493, Berlin, Heidelberg, 2007. Springer-Verlag.

[18] Piotr Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *J. ACM*, 53(3):307–323, 2006.

[19] Piotr Indyk and David Woodruff. Optimal approximations of the frequency moments of data streams. In *STOC '05: Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 202–208, New York, NY, USA, 2005. ACM.

[20] T. S. Jayram, Andrew McGregor, S. Muthukrishnan, and Erik Vee. Estimating statistical aggregates on probabilistic data streams. In *PODS '07: Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 243–252, New York, NY, USA, 2007. ACM.

[21] T. S. Jayram and David Woodruff. Optimal bounds for johnson-lindenstrauss transforms and streaming problems with sub-constant error. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '11, pages 1–10. SIAM, 2011.

[22] Daniel M. Kane, Jelani Nelson, and David P. Woodruff. On the exact space complexity of sketching and streaming small norms. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2010)*, 2010.

[23] Daniel M. Kane, Jelani Nelson, and David P. Woodruff. An optimal algorithm for the distinct elements problem. In *PODS '10: Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems of data*, pages 41–52, New York, NY, USA, 2010. ACM.

[24] Ping Li. Compressed counting. In *SODA '09: Proceedings of the Nineteenth Annual ACM -SIAM Symposium on Discrete Algorithms*, pages 412–421, Philadelphia, PA, USA, 2009. Society for Industrial and Applied Mathematics.

[25] S. Muthukrishnan. Data streams: algorithms and applications. *Found. Trends Theor. Comput. Sci.*, 1(2):117–236, 2005.

[26] Jelani Nelson and David P. Woodruff. Fast manhattan sketches in data streams. In *PODS '10: Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems of data*, pages 99–110, New York, NY, USA, 2010. ACM.

[27] Anne D. Pick and Gusti W. Frankel. A developmental study of strategies of visual selectivity. *Child Development*, 45(4):pp. 1162–1165, 1974.

[28] David Woodruff. Optimal space lower bounds for all frequency moments. In *SODA '04: Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 167–175, 2004.

[29] David P. Woodruff. Frequency moments. In *Encyclopedia of Database Systems*, pages 1169–1170. 2009.

[30] David P. Woodruff and Qin Zhang. Tight bounds for distributed functional monitoring. In *Proceedings of the 44th symposium on Theory of Computing*, STOC '12, pages 941–960, New York, NY, USA, 2012. ACM.