Generalizing the Layering Method of Indyk and Woodruff: Recursive Sketches for Frequency-Based Vectors on Streams

Vladimir Braverman[1,⋆] and Rafail Ostrovsky[2,⋆⋆]

[1] Johns Hopkins University, Department of Computer Science
vova@cs.jhu.edu
[2] University of California Los Angeles,
Department of Computer Science and Department of Mathematics
rafail@cs.ucla.edu

Abstract. In their ground-breaking paper, Indyk and Woodruff (STOC 05) showed how to compute the k-th frequency moment F_k (for $k > 2$) in space $O(\textit{poly-log}(n,m) \cdot n^{1-\frac{2}{k}})$, giving the first optimal result up to poly-logarithmic factors in n and m (here m is the length of the stream and n is the size of the domain.) The method of Indyk and Woodruff reduces the problem of F_k to the problem of computing heavy hitters in the streaming manner. Their reduction only requires polylogarithmic overhead in term of the space complexity and is based on the fundamental idea of "layering." Since 2005 the method of Indyk and Woodruff has been used in numerous applications and has become a standard tool for streaming computations.

We propose a new recursive sketch that generalizes and improves the reduction of Indyk and Woodruff. Our method works for any non-negative frequency-based function in several models, including the insertion-only model, the turnstile model and the sliding window model. For frequency-based functions with sublinear polynomial space complexity our reduction only requires $\log^{(c)}(n)$ overhead, where $\log^{(c)}(n)$ is the iterative log function. Thus, we improve the reduction of Indyk and Woodruff by polylogarithmic factor. We illustrate the generality of our method by several applications: frequency moments, frequency based functions, spatial data streams and measuring independence of data sets.

Keywords: Data streams, frequencies, recursion, sketches.

⋆ This work was supported in part by DARPA grant N660001-1-2-4014. Its contents are solely the responsibility of the author and do not represent the official view of DARPA or the Department of Defense.

⋆⋆ 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.

P. Raghavendra et al. (Eds.): APPROX/RANDOM 2013, LNCS 8096, pp. 58–70, 2013.

1 Introduction

The celebrated paper of Alon, Matias and Szegedy [1] defined the following *streaming* model:

Definition 1. *Let m, n be positive integers. A* stream $D = D(n, m)$ *is a sequence of size m of integers $p_1, \ldots, p_m$, where $p_i \in \{1, \ldots, n\}$. A* frequency vector *is a vector of dimensionality n with* non-negative *entries $f_i, i \in [n]$ defined as:*

$$f_i = |\{j : 1 \leq j \leq m, p_j = i\}|.$$

Definition 2. *A k-th* frequency moment *of D is defined by $F_k(D) = \sum_{i \in [n]} f_i^k$. Also $F_\infty = \max_{i \in [n]} f_i$.*

Alon, Matias and Szegedy [1] initiated the study of approximating frequency moments with sublinear memory. Their surprising and fundamental results imply that for $k \leq 2$ it is possible to approximate F_k with polylogarithmic space; and that polynomial space is necessary for $k > 2$. Today, research on frequency moments is one of the central directions for streaming; many important discoveries have been made since [1]. The incomplete list of relevant work includes [22,19,3,14,4,16,17,18,20,21,27,25,7,9,26,28,30,5,13,24].

Indyk and Woodruff in their ground-breaking paper [23] gave the first optimal, up to polylogarithmic factor, algorithm for F_k. Their presented a two-pass algorithm with space complexity of $O\left(\frac{1}{\epsilon^{12}} \cdot (\log^2 n)(\log^6 m) \cdot n^{1-\frac{2}{k}}\right)$ and then shown how their two-pass algorithm can be converted to one-pass algorithm with additional poly-log multiplicative factors. Let us describe, very informally, the fundamental approach of Indyk and Woodruff [23]. They split the frequency vector into "layers," where each layer contains all entries with frequencies between, e.g., γ^i and γ^{i+1} for a carefully chosen $\gamma > 1$. Then they approximate the contribution of each layer by sampling the stream and by finding the heavy elements that contribute to the layer. Their elegant analysis shows that such a procedure ensures a good approximation with high probability. The method of Indyk and Woodruff reduces the problem of F_k to the problem of computing heavy hitters in the streaming manner. Their reduction requires polylogarithmic overhead in term of the space complexity. Since 2005 the method of Indyk and Woodruff has been used in numerous applications and has become a standard tool for streaming computations.

Our method has been inspired by the algorithm of Indyk and Woodruff. Specifically, we ask:

Question 1. For which functions $w : R \mapsto R$, it is possible to reduce the problem of computing $\sum_{i=1}^n w(f_i)$ to the problem of finding all $j \in [n]$ such that $w(j) = \Omega(\sum_{i=1}^n w(f_i))$? More generally, given an implicit vector V, when it is possible to reduce the problem of approximating $|V|$ (the L_1 norm) to the problem of finding heavy elements?

In this paper we answer Question 1 for all non-negative functions w. For streaming applications, recursion can be helpful if it is possible to reduce computations to a *single*

instance of a smaller problem. We show that it is possible to reduce such a problem on a vector of size n to a single computation of a *random* vector of size approximately $\frac{1}{2}n$. In particular, $O(\log(n))$ overhead is sufficient for any frequency-based function. Our main technical result is shown in Theorem 1. For frequency-based functions that require polynomial space our reduction only requires $\log^{(c)}(n)$ overhead. This result is shown in Theorem 4. We illustrate the generality of our method by several applications: frequency moments, frequency based functions, spatial data streams and measuring independence of data sets.

The correctness of the basic step in our algorithm follows from elementary analysis. We then employ the basic step recursively and show that $\log(n)$ recursive calls can give an algorithm that reduces the problem of approximating the sum to the problem of finding heavy hitters. Further, it is possible to reduce the number of recursive calls $\log(n)$ to $\log\log(n)$ by applying the same argument, but stopping after $O(\log\log(n))$ steps. At the depth $O(\log\log(n))$ of the recursion, the number of positive frequencies in a corresponding vector is polylogarithmically smaller than n, with constant probability. Thus, any algorithm that works in $polylog(n,m)n^{\alpha}$ space (where $0 < \alpha < 1$ is a constant) will approximate such a vector with negligible cost. Employing such an algorithm at the bottom of $\log\log(n)$ recursion reduces the $\log(n)$ factor to a $poly(\log\log(n))$ factor. Further, the same idea may be repeated at least constant number of times; this is how we achieve our final bound. The simplest variant of the argument requires only pairwise independence, giving an algorithm that requires only 4-wise independence.

1.1 Roadmap

In Section 2 we introduce the basic argument and extend it to a special case, stuitable for streaming applications, case in Section 3. In Section 4 we describe a generic algorithm for recursive computations. In Section 5 we discuss our result and demonstrate its generality by explaining several applications.

2 Recursive Sketches

In this paper we denote by $|V|$ the L_1 norm of V, i.e., $|V| = \sum_{j\in[n]} v_j$.

Definition 3. *Heavy elements*
*Let V be a vector of dimensionality n with non-negative entries $v_i \geq 0$. Let $0 < \alpha \leq 1$. An element v_i is a α-*heavy *with respect to V if: $v_i \geq \alpha|V|$. A set $S \subseteq [n]$ is a α-*core *w.r.t. V if $i \in S$ for any α-heavy v_i.*

Lemma 1. *Let $V \in R^{[n]}$ be a fixed vector and let S be an α-core w.r.t. V. Let H be a random vector with uniform zero-one entries $h_i, i \in [n]$ that are pairwise-independent. Define*

$$X = \sum_{i\in S} v_i + 2\sum_{i\notin S} h_i v_i.$$

Then $P(|X - |V|| \geq \epsilon|V|) \leq \frac{\alpha}{\epsilon^2}$.

Proof. Clearly, $E(X) = |V|$. By the properties of variance, by pairwise independence of h_i and by the definition of α-core:

$$Var(X) = 4\sum_{i \notin S} v_i^2 Var(h_i) = \sum_{i \notin S} v_i^2 \leq \alpha |V|^2.$$

Thus, by Chebyshev inequality:

$$P(|X - |V|| \geq \epsilon |V|) \leq \frac{\alpha}{\epsilon^2}.$$

Corollary 1. *Let $V \in R^{[n]}$ be a random vector and let S be an α-core w.r.t. V. Let H be a random vector independent of V and S with uniform zero-one entries $h_i, i \in [n]$ that are pairwise-independent. Define*

$$X = \sum_{i \in S} v_i + 2\sum_{i \notin S} h_i v_i.$$

Then

$$P(|X - |V|| \geq \epsilon |V|) \leq \frac{\alpha}{\epsilon^2}.$$

Proof. For any fixed V and S the main claim is true since H is independent of V and S and by Lemma 1. Thus, the corollary follows.

Recursive Computations. Let ϕ be a parameter. Let $H_1, \ldots, H_\phi$ be i.i.d. random vectors with zero-one entries that are uniformly distributed and pairwise independent. For two vectors of dimensionality n define $Had(V, U)$ to be their Hadamard product; i.e., $Had(V, U)$ is a vector of dimensionality n with entries $v_i u_i$. Define:

$$V_0 = V, \text{ and } V_j = Had(V_{j-1}, H_j) \text{ for } j = 1, \ldots, \phi.$$

Denote by v_i^j and h_i^j the i-th entry of V_j and H_j respectfully. Let $S_0, \ldots, S_\phi$ be a sequence of subsets of $[n]$ such that S_j is an α-core of V_j. Define the sequence

$$X_j = \sum_{i \in S_j} v_i^j + 2\sum_{i \notin S_j} h_i^{j+1} v_i^j, \qquad j = 0, \ldots, \phi - 1,$$

and $X_\phi = |V_\phi|$.

Fact 2

$$P(\bigcup_{j=0}^{\phi} (|X_j - |V_j|| \geq \epsilon |V_j|)) \leq \frac{(\phi + 1)\alpha}{\epsilon^2}.$$

Proof. Consider fixed $j < k$. It follows from the definitions that H_{j+1} is independent of V_j and S_j. Applying Corollary 1 and the union bound we obtain the proof.

Consider the following recursive definition:

$$Y_\phi = X_\phi, \quad Y_j = 2Y_{j+1} + \sum_{i \in S_j} (1 - 2h_i^{j+1}) v_i^j.$$

Lemma 3. *For any ϕ, γ, vector V and $\alpha = \Omega(\frac{\gamma^2}{\phi^3})$:*

$$P(|Y_0 - |V|| \geq \gamma |V|) \leq 0.2.$$

Proof. Denote $Err_j^1 = |V_j| - X_j$ and $Err_j^2 = |V_j| - Y_j$. We can rewrite

$$X_j = 2|V_{j+1}| + \sum_{i \in S_j} (1 - 2h_i^{j+1}) v_i^j.$$

Thus $X_j - Y_j = 2(|V_{j+1}| - Y_{j+1}) = 2Err_{j+1}^2$ and

$$|Err_j^2| = |Y_j - |V_j|| \leq |X_j - |V_j|| + |X_j - Y_j| = |Err_j^1| + 2|Err_{j+1}^2|.$$

By definition $Err_\phi^1 = Err_\phi^2 = 0$. Thus we can rewrite:

$$|Err_0^2| \leq |Err_0^1| + 2|Err_1^2| \leq \cdots \leq \sum_{j=0}^{\phi} 2^j |Err_j^1|.$$

Choose $\epsilon = \frac{\gamma}{10(\phi+1)}$; we have by Fact 2:

$$P(|Y_0 - |V|| \geq \gamma |V|) = P(|Err_0^2| \geq \gamma |V|) \leq P(\sum_{j=0}^{\phi} 2^j |Err_j^1| \geq \gamma |V|) \leq$$

$$P\left(\left(\sum_{j=0}^{\phi} 2^j |Err_j^1| \geq \gamma |V|\right) \cap \left(\bigcap_{j=0}^{\phi} \left(|Err_j^1| < \epsilon |V_j|\right)\right)\right) + P\left(\bigcup_{j=0}^{\phi} \left(|X_j - |V_j|| \geq \epsilon |V_j|\right)\right) \leq$$

$$P\left(\sum_{j=0}^{\phi} 2^j |V_j| \geq 10(\phi+1)|V|\right) + \frac{(\phi+1)\alpha}{\epsilon^2}.$$

For $j > 0$ we note that $|V_j|$ is a random variable defined as:

$$|V_j| = \sum_{i \in [n]} v_i \left(\prod_{t=1}^{j} h_i^t\right).$$

Since all H_j are mutually independent, we conclude that

$$E(\sum_{j=0}^{\phi} 2^j |V_j|) = \sum_{j=0}^{\phi} 2^j \left(\sum_{i \in [n]} v_i \left(\prod_{t=1}^{j} E(h_i^t)\right)\right) = \sum_{j=0}^{\phi} 2^j \left(\sum_{i \in [n]} v_i 2^{-j}\right) = (\phi+1)|V|.$$

Thus, and by Markov inequality, we have

$$P(\sum_{j=0}^{\phi} 2^j |V_j| \geq 10(\phi+1)|V|) \leq 0.1.$$

Also, $\frac{(\phi+1)\alpha}{\epsilon^2} \leq 0.1$ for sufficiently large $\alpha = \Omega(\frac{\gamma^2}{\phi^3})$. Thus,

$$P(|Y_0 - |V|| \geq \gamma |V|) \leq 0.2.$$

3 An Extension: Approximate and Random Cores

There are many ways to extend our basic result. We will explore one direction, when the cores are random and contain approximations of heavy hitters with high probability[1]. We consider vectors from a finite domain $[m]^n$.

Definition 4. *Let Ω be a finite set of real numbers. Define $Pairs_t$ to be a set of all sets of pairs of the form:*

$$\{(i_1, w_1), \ldots, (i_t, w_t)\}, \quad 1 \leq i_1 < i_2 < \ldots i_t \leq n, i_j \in N, w_j \in \Omega.$$

Further define

$$Pairs = \emptyset \cup \left(\bigcup_{t=1}^{n} Pairs_t \right).$$

Definition 5. *A non-empty set $Q \in Pairs_t$, i.e., $Q = \{(i_1, w_1), \ldots, (i_t, w_t)\}$ for some $t \in [n]$, is (α, ϵ)-cover w.r.t. vector $V \in [M]^n$ if the following is true:*

1. *$\forall j \in [t](1 - \epsilon)v_{i_j} \leq w_j \leq (1 + \epsilon)v_{i_j}$.*
2. *$\forall i \in [n]$ if v_i is α-heavy then $\exists j \in [t]$ such that $i_j = i$.*

Definition 6. *Let $\mathcal{D}$ be a probability distribution on $Pairs$. Let $V \in [m]^n$ be a fixed vector. We say that $\mathcal{D}$ is δ-good w.r.t. V if for a random element Q of Pairs with distribution D the following is true:*

$$P(Q \text{ is } (\alpha, \epsilon)\text{-cover of } V) \geq 1 - \delta.$$

Definition 7. *Let g be a mapping from $[M]^n$ to a set of all distributions on $Pairs$. We say that g is δ-good if for any fixed $V \in [M]^n$ the distribution $g(V)$ is δ-good w.r.t. V. Intuitively, g represents an output of an algorithm that finds heavy hitters (and their approximations) of input vector V w.p. $1 - \delta$.*

Definition 8. *For non-empty $Q \in Pairs$ define $Ind(Q)$ to be the set of indexes of Q. Formally, for $Q \in Pairs$, denote $Ind(Q) = \{i : \exists j < t$ such that for j-th pair (i_j, w_j) of Q it is true that $i_j = i\}$. For $i \in Ind(Q)$ denote by $w_Q(i)$ the corresponding approximation, i.e. if $i = i_j$ then $w_Q(i) = w_j$. (Note that since $i_j < i_{j+1}$ this is a valid definition.) For completeness, denote $w_Q(i) = 0$ for $i \notin Ind(Q)$ and $Ind(\emptyset) = \emptyset$.*

Now we are ready to repeat the arguments from the previous section.

Corollary 2. *Let $V \in R^{[n]}$ be a random vector. Let g be a δ-good mapping and let Q be a random element of $Pairs$ that is chosen according to a distribution $g(V)$. Let H be a random vector independent of V and Q with uniform zero-one entries $h_i, i \in [n]$ that are pairwise-independent. Define*

$$X' = \sum_{i \in Ind(Q)} v_i + 2 \sum_{i \notin Ind(Q)} h_i v_i.$$

[1] In this section we limit our discussion to finite sets and discrete distributions. This limitation is artificial but sufficient for our applications; on the other hand it simplifies the presentation.

Then

$$P(|X' - |V|| \geq \epsilon |V|) \leq \frac{\epsilon}{\alpha^2} + \delta.$$

Proof. Consider a fixed vector V_0 and an event that $V = V_0$. Conditioned on this event, the distribution $g(V)$ is fixed and δ-good w.r.t. V_0. Consider the event that $Q = Q_0$, where Q_0 is an (α, ϵ)-cover w.r.t. V_0. Conditioned on this event, $Ind(Q)$ is an α-cover w.r.t. V_0. Since H is independent of Q the claim is true for any such V_0 by Lemma 1 and by union bound. Thus, the corollary follows.

Recursive Computations Let ϕ be a parameter. Let $H_1, \ldots, H_\phi$ be i.i.d. random vectors with zero-one entries that are uniformly distributed and pairwise independent. Define:

$$V_0 = V, \text{ and } V_j = Had(V_{j-1}, H_j) \text{ for } j = 1, \ldots, \phi.$$

Denote by v_i^j and h_i^j the i-th entry of V_j and H_j respectfully. Let g be a δ-good mapping and let Q_i be a random element of Pairs with distribution $g(V_i)$. Define $w_j(i) = w_{Q_j}(i)$. Define the sequence:

$$X'_j = \sum_{i \in Ind(Q_j)} v_i^j + 2 \sum_{i \notin Ind(Q_j)} h_i^{j+1} v_i^j, \quad j = 0, \ldots, \phi - 1,$$

and $X'_\phi = |V_\phi|$. From Corollary 2 and by repeating the arguments from Fact 2 we obtain

Fact 4

$$P(\bigcup_{j=0}^{\phi} \left(|X'_j - |V_j|| \geq \epsilon |V_j|\right)) \leq (\phi + 1)(\frac{\alpha}{\epsilon^2} + \delta).$$

Consider the following recursive definition. Let $Y'_\phi = Y'_\phi(V_\phi)$ be a random variable that depends on random vector V_ϕ and such that for any fixed V_ϕ:

$$P(|Y'_\phi - |V_\phi|| \geq \epsilon |V_\phi|) \leq \delta.$$

Also, define for $j = 0, \ldots, \phi - 1$:

$$Y'_j = 2Y'_{j+1} + \sum_{i \in Ind(Q_j)} (1 - 2h_i^{j+1}) w_i^j.$$

Lemma 5. *For any ϕ, γ, vector V; for $\alpha = \Omega(\frac{\gamma^2}{\phi^3})$ and $\delta = \Omega(\frac{1}{\phi})$:*

$$P(|Y'_0 - |V|| \geq \gamma |V|) \leq 0.2.$$

Proof. Denote $Err_j^1 = |V_j| - X'_j$, $Err_j^2 = |V_j| - Y'_j$ and $Err_j^3 = \sum_{i \in Ind(Q_j)} |w_j(i) - v_i^j|$. We can rewrite

$$X'_j = 2|V_{j+1}| + \sum_{i \in Ind(Q_j)} (1 - 2h_i^{j+1}) v_i^j.$$

Thus $|X'_j - Y'_j| \leq 2|Err^2_{j+1}| + |Err^3_j|$ and

$$|Err^2_j| = |Y'_j - |V_j|| \leq |X'_j - |V_j|| + |X'_j - Y'_j| \leq |Err^1_j| + |Err^3_j| + 2|Err^2_{j+1}|.$$

Thus we can rewrite:

$$|Err^2_0| \leq |Err^1_0| + |Err^3_0| + 2|Err^2_1| \leq \cdots \leq 2^k Err^2_\phi + \sum_{j=0}^{\phi} 2^j |Err^1_j| + \sum_{j=0}^{\phi} 2^j |Err^3_j|.$$

Choose $\epsilon = \frac{\gamma}{30(\phi+1)}$ and denote $Z = 2^k Err^2_\phi + \sum_{j=0}^{\phi} 2^j |Err^1_j| + \sum_{j=0}^{\phi} 2^j |Err^3_j|$. Then

$$P(|Y'_0 - |V|| \geq \gamma|V|) = P(|Err^2_0| \geq \gamma|V|) \leq P(Z \geq \gamma|V|) \leq$$

$$P\left((Z \geq \gamma|V|) \cap \left(\bigcap_{j=0}^{\phi} \left(|Err^1_j| < \epsilon|V_j|\right)\right) \cap \left(\bigcap_{j=0}^{\phi} \left(|Err^3_j| < \epsilon|V_j|\right)\right) \cap \left(|Err^2_\phi| < \epsilon|V_\phi|\right)\right) +$$

$$P\left(|Err^2_\phi| \geq \epsilon|V_\phi|\right) + P\left(\bigcup_{j=0}^{\phi} \left(|Err^1_j| \geq \epsilon|V_j|\right)\right) + P\left(\bigcup_{j=0}^{\phi} \left(|Err^3_j| \geq \epsilon|V_j|\right)\right).$$

Note that by the definition of Y'_ϕ, we have $P(|Err^2_\phi| \geq \epsilon|V_\phi|) \leq \delta$. Also, by the definition of Q_j and union bound,

$$P(\bigcup_{j=0}^{\phi} (|Err^3_j| \geq \epsilon|V_j|)) \leq (\phi+1)\delta.$$

Thus and by Fact 4:

$$P(|Y'_0 - |V|| \geq \gamma|V|) \leq P\left(\sum_{j=0}^{\phi} 2^j |V_j| \geq 10(\phi+1)|V|\right) + (\phi+2)(\frac{\alpha}{\epsilon^2} + 2\delta).$$

The lemma follows by repeating the concluding arguments from Lemma 3.

4 A Generic Algorithm

Let D be a stream as in Definition 1. For a function $H : [n] \mapsto \{0,1\}$, define D_H to be a sub-stream of D that contains only elements $p \in D$ such that $H(p) = 1$. Let $V = V(D)$ be an implicit vector of dimensionality n defined by a stream, e.g., a frequency moment vector from Definition 1. We say that a vector V is *separable* if for any H, we have $Had(V(D), H) = V(D_H)$. Let $HH(D, \alpha, \epsilon, \delta)$ be an algorithm that produces (α, ϵ)-cover w.r.t. $V(D)$ w.p. $1 - \delta$, i.e., produces δ-good distribution w.r.t. $V(D)$ for some suitable finite set of Pairs, as defined in Definition 4.

Algorithm 6 . *Recursive Sum[0]*(D, ϵ)

1. *Generate* $\phi = O(\log(n))$ *pairwise independent zero-one vectors* $H_1, \ldots, H_\phi$. *Denote* D_j *to be a stream* $D_{H_1 H_2 \ldots H_j}$.
2. *Compute, in parallel, random cores* $Q_j = HH(D_j, \frac{\phi^3}{\epsilon^2}, \epsilon, \frac{1}{\phi})$
3. *If* $F_0(V_\phi) > 10^{10}$ *then output* 0 *and stop. Otherwise compute precisely* $Y_\phi = |V_\phi|$.
4. *For each* $j = \phi - 1, \ldots, 0$, *compute*

$$Y_j = 2Y_{j+1} - \sum_{i \in Ind(Q_j)} (1 - 2h_i^j) w_{Q_j}(i).$$

5. *Output* Y_0.

Theorem 1. *Algorithm 6 computes* $(1 \pm \epsilon)$*-approximation of* $|V|$ *and errs w.p. at most* 0.3. *The algorithm uses* $O(\log(n)\mu(n, \frac{1}{\epsilon^2 \log^3(n)}, \epsilon, \frac{1}{\log(n)}))$ *memory bits, where* μ *is the space required by the above algorithm* HH.

Proof. The correctness follows directly from the description of the algorithm and Lemma 5 and Markov inequality. The memory bounds follows from the direct computations.

5 Discussion and Applications

We propose a new recursive sketch that generalizes and improves the reduction of Indyk and Woodruff. Our method works for any non-negative frequency-based function in several models, including the insertion-only model, the turnstile model and the sliding window model. For frequency-based functions with sublinear polynomial space complexity our reduction requires $O(\log^{(c)}(n))$ overhead. We believe that there are many other potential applications for our method, e.g., the algorithms that currently employ the method of Indyk and Woodruff. Improving the bounds for these problems is an interesting direction for the future work. Reducing the factor to $o(\log^{(c)}(n))$ is another important open question.

5.1 Approximating Large Frequency Moments on Streams with CountSketch

We apply our technique to the problem of frequency moments.

Fact 7 *Let* V *be a vector of dimensionality* n *with non-negative entries and let* n_0 *be a number of non-zero entries in* V. *Let* $0 < \alpha < 1$ *and let* v_i *be such that* $v_i^k \geq \alpha \sum_{j \in [n]} v_j^k$. *Then* $v_i^2 \geq 0.5 \alpha^{\frac{2}{k}} n_0^{\frac{2}{k}-1} \sum_{j \neq i} v_j^2$.

Proof. If $n_0 = 0$ the fact is trivial. Otherwise, by Hölder's inequality, $\sum_{j \neq i} v_j^2 \leq n_0^{1-\frac{2}{k}} \left(\sum_{j \neq i} v_j^k \right)^{\frac{2}{k}} \leq n_0^{1-\frac{2}{k}} \alpha^{-\frac{2}{k}} v_i^2$.

The famous Count-Sketch [15] algorithm finds all α-heavy elements. In particular, the following is a corollary from [15].

Theorem 2. *(from [15]) Let a_t be the frequency of the t-th most frequent element. There exists an algorithm that w.p. $1-\delta$ outputs t pairs (i, f_i') such that $(1-\epsilon)f_i \leq f_i' \leq (1+\epsilon)f_i$ and such that all elements with $f_i \geq (1-\epsilon)a_t$ appear in the list. The algorithm uses $O((t + \frac{\sum_{i\in[n], f_i<a_t} f_i^2}{(\epsilon a_t)^2})\log(m/\delta)\log(m))$ memory bits.*

Combining with Fact 7 we obtain

Corollary 3. *There exists an algorithm that w.p. $1-\delta$ outputs $O(\alpha^{-1})$ pairs $(i, f_i'^k)$ such that $(1-\epsilon)f_i^k \leq f_i'^k \leq (1+\epsilon)f_i^k$ and such that all elements with $f_i^k \geq \alpha\sum_{j\in[n]} f_j^k$ appear in the list. The algorithm uses $O((\alpha^{-1} + \frac{k^2}{\epsilon^2}\alpha^{-2/k}n^{1-2/k})\log(m/\delta)\log(m))$ memory bits.*

The algorithm from Corollary 3 defines a δ-good distribution w.r.t. to the input vector $V(D)$ over some finite set[2] from Definition 4. Denote the algorithm from Corollary 3 by $CS(D, \alpha, \epsilon, \delta)$. Thus, combining with Algorithm 6 if gives an algorithm errs w.p. δ, outputs $(1\pm\epsilon)$-approximation of F_k and uses $O(\frac{k^2}{\epsilon^{2+4/k}}n_0^{1-2/k}\log(mn)\log(m)\log^{1+6/k}(n)\log(1/\delta))$ memory bits, nearly matching the bound in [6]. Denote this algorithm by $\mathcal{A}_0(D, \epsilon, \delta)$. We can improve the bound further recursively:

Algorithm 8 . *Recursive F_k[1]*(D, ϵ)

1. *Generate $\phi = O(\log\log(n))$ pairwise independent zero-one vectors $H_1, \ldots, H_\phi$. Denote D_j to be a stream $D_{H_1H_2\ldots H_\phi}$.*
2. *Compute, in parallel, $Q_j = CS(D_j, \frac{\epsilon^2}{\phi^3}, \epsilon, \frac{1}{100\phi})$*
3. *Compute $Y_\phi = \mathcal{A}_0(D_\phi, \epsilon, 0.1)$.*
4. *For each $j = \phi-1, \ldots, 0$, compute*

$$Y_j = 2Y_{j+1} - \sum_{i\in Ind(Q_j)} (1-2h_i^j)w_{Q_j}(i).$$

5. *Output Y_0.*

There exists a constant c such that for $\phi = c\log\log(n)$, except with a small constant probability, $F_0(D_\phi) \leq \frac{n}{\log^{10}(n)}$. Thus, executing $\mathcal{A}_0$ for $n' = \frac{n}{\log^{10}(n)}$ we obtain an approximation of $F_k(D_\phi)$ using $O(\frac{k^2}{\epsilon^{2+4/k}}n^{1-2/k}\log(mn)\log(m))$ memory bits. Since $\phi = O(\log\log(n))$, the complexity of the new algorithm becomes $O(\frac{k^2}{\epsilon^{2+4/k}}n^{1-2/k}\log(m\log(n))\log(m)(\log\log(n))^4)$. Repeating this argument a constant number of times we arrive at[3]:

[2] Indeed, we can define the finite set Ω from Definition 4 as a set of all possible outputs of Count-Sketch executed over all vectors on $[m]^n$. This is a finite set (for finite n, m) and thus we can define Pairs accordingly.

[3] We note that this algorithm was proposed in the previous version of the paper [11].

Theorem 3. *Define $g_1(n) = \log(n)$ and $g_t(n) = \log(g_{t-1}(n))$. For any constant t there exist an algorithm computes a $(1 \pm \epsilon)$-approximation of $F_k(D)$, errs w.p. at most $\frac{1}{3}$ and uses $O(c_t k^2 \epsilon^{-2-(4/k)} n^{1-\frac{2}{k}} g_t(n) \log^2(m))$ memory bits, where c_t is a constant that depends on t.*

5.2 Recursive Sketches for Frequency-Based Functions with Sublinear Polynomial Space Complexity

Theorem 3 can be generalized to any non-negative frequency based functions with sublinear polynomial space complexity. In particular, we show that the problem of approximating $|V|$ and finding heavy hitters are almost equivalent.

Theorem 4. *Define $g_1(n) = \log(n)$ and $g_t(n) = \log(g_{t-1}(n))$. Let V a vector such that it is possible to find heavy elements using space $S(n) = \Omega(n^\alpha)$ for some constant α such that $0 < \alpha < 1$. Then for any constant ϵ and for any constant t there exist an algorithm that computes $(1 \pm \epsilon)$-approximation of $|V|$, errs w.p. at most $\frac{1}{3}$ and uses $O(c_t n^\alpha g_t(n))$ memory bits, where c_t is a constant that depends on t and ϵ.*

5.3 Other Models of Streaming Computations

Our method can be directly translated to any model that allows separability and preserves non-negative values of the implicit vector. Specifically, we need the non-negativity to apply the Markov inequality. For example, we can apply the reduction to the turnstile model where we observe two streams D_1 and D_2 and need to compute the $\sum_{i=1}^{n} w(|u_i - z_i|)$ where u_i and z_i are frequencies in D_1 and D_2. Similarly, our reduction will work for the sliding window model. For example, it should improve (by polylog factor) the results for the frequency moments from [9].

5.4 Approximating Large Frequency Moments on Streams with Pick-and-Drop Sampling

In [8] we combine our method of recursive sketches with the pick-and-drop sampling and compute F_k with $O(n^{1-2/k} \log(n) \log^{(c)}(n))$ bits. We reduce the ratio between the upper and lower bounds from $O(\log^2(n))$ to $O(\log(n) \log^{(c)}(n))$. Thus, we provide a (roughly) quadratic improvement of the result of Andoni, Krauthgamer and Onak [2]. To the best of our knowledge, this is the best currently known result for constant ϵ and for insertion-only streams.

5.5 Spatial Data Streams

Recursive sketching is not limited to the frequency moments or to the insertion-only streams. For example, the method has found applications in the work of Tirthapura and Woodruff [29] on spatial data streams. Specifically, Tirthapura and Woodruff say "We choose to follow [11] since it provides a simpler exposition and has several properties we will exploit."

5.6 Frequency-Based Functions and Measuring Independence of Datasets

In [12] we consider zero-one frequency laws for the frequency-based functions. As one of the key steps we employ general reduction from sums to heavy hitters. Our method follows the ideas of Indyk and Woodruff and involves large polylogarithmic factors. Replacing our method in [12] we should be able to achieve polylogarithmic improvements in space. Similar polylogarithmic improvement should be possible for measuring independence (in terms of total variation distanced) of datasets [10].

References

1. Alon, N., Matias, Y., Szegedy, M.: The space complexity of approximating the frequency moments. J. Comput. Syst. Sci. 58(1), 137–147 (1999)
2. Andoni, A., Krauthgamer, R., Onak, K.: Streaming algorithms via precision sampling. In: FOCS, pp. 363–372 (2011)
3. Bar-Yossef, Z., Jayram, T.S., Kumar, R., Sivakumar, D.: An information statistics approach to data stream and communication complexity. J. Comput. Syst. Sci. 68(4), 702–732 (2004)
4. Bar-Yossef, Z., Jayram, T.S., Kumar, R., Sivakumar, D., Trevisan, L.: Counting distinct elements in a data stream. In: Rolim, J.D.P., Vadhan, S.P. (eds.) RANDOM 2002. LNCS, vol. 2483, pp. 1–10. Springer, Heidelberg (2002)
5. Beame, P., Jayram, T.S., Rudra, A.: Lower bounds for randomized read/write stream algorithms. In: STOC 2007: Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing, pp. 689–698. ACM, New York (2007)
6. Bhuvanagiri, L., Ganguly, S., Kesh, D., Saha, C.: Simpler algorithm for estimating frequency moments of data streams. In: SODA, pp. 708–713 (2006)
7. Braverman, V., Gelles, R., Ostrovsky, R.: How to catch l_2-heavy-hitters on sliding windows. In: Du, D.-Z., Zhang, G. (eds.) COCOON 2013. LNCS, vol. 7936, pp. 638–650. Springer, Heidelberg (2013)
8. Braverman, V., Ostrovsky, R.: Generalizing the layering method of indyk and woodruff: Recursive sketches for frequency-based vectors on streams. In: Raghavendra, P., Raskhodnikova, S., Jansen, K., Rolim, J.D.P. (eds.) APPROX/RANDOM 2013. LNCS, vol. 8096, pp. 58–70. Springer, Heidelberg (2013)
9. Braverman, V., Ostrovsky, R.: Smooth histograms for sliding windows. In: Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2007, pp. 283–293. IEEE Computer Society, Washington, DC (2007)
10. Braverman, V., Ostrovsky, R.: Measuring independence of datasets. In: Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, pp. 271–280. ACM, New York (2010)
11. Braverman, V., Ostrovsky, R.: Recursive sketching for frequency moments. CoRR, abs/1011.2571 (2010)
12. Braverman, V., Ostrovsky, R.: Zero-one frequency laws. In: Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, pp. 281–290. ACM, New York (2010)
13. Chakrabarti, A., Cormode, G., McGregor, A.: Robust lower bounds for communication and stream computation. In: STOC 2008: Proceedings of the 40th Annual ACM Symposium on Theory of Computing, pp. 641–650. ACM, New York (2008)
14. Chakrabarti, A., Khot, S., Sun, X.: Near-optimal lower bounds on the multi-party communication complexity of set disjointness. In: IEEE Conference on Computational Complexity, pp. 107–117 (2003)

15. Charikar, M., Chen, K., Farach-Colton, M.: Finding frequent items in data streams. In: Widmayer, P., Triguero, F., Morales, R., Hennessy, M., Eidenbenz, S., Conejo, R. (eds.) ICALP 2002. LNCS, vol. 2380, pp. 693–703. Springer, Heidelberg (2002)
16. Coppersmith, D., Kumar, R.: An improved data stream algorithm for frequency moments. In: SODA, pp. 151–156 (2004)
17. Cormode, G., Datar, M., Indyk, P., Muthukrishnan, S.: Comparing data streams using hamming norms (how to zero in). IEEE Trans. on Knowl. and Data Eng. 15(3), 529–540 (2003)
18. Feigenbaum, J., Kannan, S., Strauss, M., Viswanathan, M.: An approximate l1-difference algorithm for massive data streams. In: FOCS 1999: Proceedings of the 40th Annual Symposium on Foundations of Computer Science, p. 501. IEEE Computer Society, Washington, DC (1999)
19. Flajolet, P., Nigel Martin, G.: Probabilistic counting algorithms for data base applications. J. Comput. Syst. Sci. 31(2), 182–209 (1985)
20. Ganguly, S.: Estimating frequency moments of data streams using random linear combinations. In: Jansen, K., Khanna, S., Rolim, J.D.P., Ron, D. (eds.) APPROX and RANDOM 2004. LNCS, vol. 3122, pp. 369–380. Springer, Heidelberg (2004)
21. Ganguly, S., Cormode, G.: On estimating frequency moments of data streams. In: Charikar, M., Jansen, K., Reingold, O., Rolim, J.D.P. (eds.) APPROX and RANDOM 2007. LNCS, vol. 4627, pp. 479–493. Springer, Heidelberg (2007)
22. Indyk, P.: Stable distributions, pseudorandom generators, embeddings, and data stream computation. J. ACM 53(3), 307–323 (2006)
23. Indyk, P., Woodruff, D.L.P.: Optimal approximations of the frequency moments of data streams. In: STOC 2005: Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing, pp. 202–208. ACM, New York (2005)
24. Jayram, T.S., McGregor, A., Muthukrishnan, S., Vee, E.: Estimating statistical aggregates on probabilistic data streams. In: PODS 2007: Proceedings of the Twenty-sixth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, pp. 243–252. ACM, New York (2007)
25. Kane, D.M., Nelson, J., Woodruff, D.P.: 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)
26. Kane, D.M., Nelson, J., Woodruff, D.P.: An optimal algorithm for the distinct elements problem. In: PODS 2010: Proceedings of the Twenty-ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems of Data, pp. 41–52. ACM, New York (2010)
27. Li, P.: Compressed counting. In: SODA 2009: Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 412–421. Society for Industrial and Applied Mathematics, Philadelphia (2009)
28. Nelson, J., Woodruff, D.P.: Fast manhattan sketches in data streams. In: PODS 2010: Proceedings of the Twenty-ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems of Data, pp. 99–110. ACM, New York (2010)
29. Tirthapura, S., Woodruff, D.P.: Rectangle-efficient aggregation in spatial data streams. In: Proceedings of the 31st Symposium on Principles of Database Systems, PODS 2012, pp. 283–294. ACM, New York (2012)
30. Woodruff, D.P.: Optimal space lower bounds for all frequency moments. In: SODA 2004: Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 167–175 (2004)