Estimating Frequency Moments of Data Streams using Random Linear Combinations

Sumit Ganguly

Indian Institute of Technology, Kanpur
e-mail: sganguly@iitk.ac.in

Abstract. The problem of estimating the k^{th} frequency moment F_k for any non-negative k, over a data stream by looking at the items exactly once as they arrive, was considered in a seminal paper by Alon, Matias and Szegedy [1, 2]. The space complexity of their algorithm is $\tilde{O}(n^{1-\frac{1}{k}})$. For $k > 2$, their technique does not apply to data streams with arbitrary insertions and deletions. In this paper, we present an algorithm for estimating F_k for $k > 2$, over general update streams whose space complexity is $\tilde{O}(n^{1-\frac{1}{k-1}})$ and time complexity of processing each stream update is $\tilde{O}(1)$.

Recently, an algorithm for estimating F_k over general update streams with similar space complexity has been published by Coppersmith and Kumar [7]. Our technique is, (a) basically different from the technique used by [7], (b) is simpler and symmetric, and, (c) is significantly more efficient in terms of the time required to process a stream update ($\tilde{O}(1)$ compared with $\tilde{O}(n^{1-\frac{1}{k-1}})$).

1 Introduction

A data stream can be viewed as a sequence of updates, that is, insertions and deletions of items. Each update is of the form $(l, \pm v)$, where, l is the identity of the item and v is the change in frequency of l such that $|v| \geq 1$. The items are assumed to draw their identities from the domain $[N] = \{0, 1, \ldots, N-1\}$. If v is positive, then the operation is an insertion operation, otherwise, the operation is a deletion operation. The frequency of an item with identity l, denoted by f_l, is the sum of the changes in frequencies of l from the start of the stream. In this paper, we are interested in computing the k^{th} frequency moment $F_k = \sum_l f_l^k$, for $k > 2$ and k integral, by looking at the items exactly once when they arrive.

The problem of estimating frequency moments over data streams using randomized algorithms was first studied in a seminal paper by Alon, Matias and Szegedy [1, 2]. They present an algorithm, based on sampling, for estimating F_k, for $k \geq 2$, to within any specified approximation factor ϵ and with confidence that is a constant greater than 1/2. The space complexity of this algorithm is $s = \tilde{O}(n^{1-\frac{1}{k}})$ (suppressing the term $\frac{1}{\epsilon^2}$) and time complexity per update is $\tilde{O}(n^{1-\frac{1}{k}})$, where, n is the number of distinct elements in the stream. This algorithm assumes that frequency updates are restricted to the form $(l, +1)$.

One problem with the sampling algorithm of [1, 2] is that it is not applicable to streams with arbitrary deletion operations. For some applications, the ability to handle deletions in a stream may be important. For example, a network monitoring application might be continuously maintaining aggregates over the number of currently open connections per source or destination.

In this paper, we present an algorithm for estimating F_k, for $k > 2$, to within an accuracy of $(1 \pm \epsilon)$ with confidence at least 2/3. (The method can be boosted using the median of averages technique to return high confidence estimates in the standard way $[1, 2]$.) The algorithm handles arbitrary insertions and legal deletions (i.e., net frequency of every item is non-negative) from the stream and generalizes the random linear combinations technique of $[1, 2]$ designed specifically for estimating F_2. The space complexity of our method is $\tilde{O}(n^{1-\frac{1}{k-1}})$ and the time complexity to process each update is $\tilde{O}(1)$, where, functions of k and ϵ that do not involve n are treated as constants.

In [7], Coppersmith and Kumar present an algorithm for estimating F_k over general update streams. Their algorithm has similar space complexity (i.e., $\tilde{O}(n^{1-\frac{1}{k-1}})$) as the one we design in this paper. The principal differences between our work and the work in [7] are as follows.

1. *Different Technique.* Our method constructs random linear combinations of the frequency vector using randomly chosen roots of unity, that is, we construct the sketch $Z = f_l x_l$, where, x_l is a randomly chosen k^{th} root of unity. Coppersmith and Kumar construct random linear combinations $C = f_l x_l$, where, for $l \in [N]$, $x_l = -1/n^{1-\frac{1}{k-1}}$ or $1 - 1/n^{1-\frac{1}{k-1}}$ with probability $1 - 1/n^{1-\frac{1}{k-1}}$ and $1/n^{1-\frac{1}{k-1}}$ respectively.
2. *Symmetric and Simpler Algorithm.* Our technique is a symmetric method for all $k \geq 2$, and is a direct generalization of the sketch technique of Alon, Matias and Szegedy $[1, 2]$. In particular, for every $k \geq 2$, $\mathbf{E}\big[\text{Re } Z^k\big] = F_k$. The method of Coppersmith and Kumar gives complicated expressions for estimating F_k, for $k \geq 4$. For $k = 4$, their estimator is $C^4 - B_n F_2^2$ (where, $B_n \approx n^{-4/3}(1 - n^{-2/3})^2$), and requires, in addition, an estimation of F_2 to within an accuracy factor of $(1 \pm n^{-1/3})$. The estimator expression for higher values of k (particularly, for powers of 2) are not shown in [7]. These expressions require auxiliary moment estimation and are quite complicated.
3. *Time efficient.* Our method is significantly more efficient in terms of the time taken to process an arrival over the stream. The time complexity to process a stream update in our method is $\tilde{O}(1)$, whereas, the time complexity of the Coppersmith Kumar technique is $\tilde{O}(n^{1-\frac{1}{k-1}})$.

The recent and unpublished work in [11] presents an algorithm for estimating F_k, for $k > 2$ and for the append only streaming model (used by $[1, 2]$), with space complexity $\tilde{O}(n^{1-\frac{2}{k+1}})$. Although, the algorithm in [11] improves on the asymptotic space complexity of the algorithm presented in this paper, it cannot handle deletion operations over the stream. Further, the method used by [11] is significantly different from the techniques used in this paper, or from the techniques used by Coppersmith and Kumar [7].

Lower bounds. The work in $[1, 2]$ shows space lower bounds for this problem to be $\Omega(n^{1-5/k})$, for any $k > 5$. Subsequently, the space lower bounds have been strengthened to $\Omega(\epsilon^2 n^{1-(2+\epsilon)/k})$, for $k > 2$, $\epsilon > 0$, by Bar-Yossef, Jayram, Kumar and Sivakumar [3], and further to $\Omega(n^{1-2/k})$ by Chakrabarti, Khot and Sun [5]. Saks and Sun [14] show that estimating the L_p distance d between two streaming vectors to within a factor of d^δ requires space $\Omega(n^{1-2/p-4\delta})$.

Other Related Work. For the special case of computing F_2, [1, 2] presents an $O(\log\ n + \log\ m)$ space and time complexity algorithm, where, m is the sum of the frequencies. Random linear combinations based on random variables drawn from stable distributions were considered by [13] to estimate F_p, for $0 < p \leq 2$. The work presented in [9] presents a sketch technique to estimate the difference between two streams based on the L_1 metric norm. There has been substantial work on the problem of estimating F_0 and related metrics (set expression cardinalities over streams) for the various models of data streams [10, 1, 4, 12].

The rest of the paper is organized as follows. Section 2 describes the method and Section 3 presents formal lemmas and their proofs. Finally we conclude in Section 4.

2 An overview of the method

In this section, we present a simple description of the algorithm and some of its properties. The lemmas and theorems stated in this section are proved formally in Section 3. Throughout the paper, we treat k as a fixed given value larger than 1.

2.1 Sketches using random linear combinations of k^{th} roots of unity

Let x be a randomly chosen root of the equation $x^k = 1$, such that each of the k roots is chosen with equal probability of $1/k$. Given a complex number z, its conjugate is denoted by $\bar{z}$. For any j, $1 \leq j \leq k$, the following basic property holds, as shown below.

$$\mathbf{E}\big[x^j\big] = \mathbf{E}\big[\bar{x}^j\big] = \begin{cases} 0 & \text{if } 1 \leq j < k \\ 1 & \text{if } j = k. \end{cases} \tag{1}$$

Proof. Let $j = k$. Then, $\mathbf{E}\big[x^j\big] = \mathbf{E}\big[x^k\big] = \mathbf{E}\big[1\big] = 1$, since, x is a root of unity.

Let $1 \leq j < k$ and let u be the elementary k^{th} root of unity, that is, $u = e^{2\pi\sqrt{-1}/k}$.

$$\mathbf{E}\big[x^j\big] = \frac{1}{k}\sum_{l=1}^{k}(u^l)^j = \frac{1}{k}\sum_{l=1}^{k}(u^j)^l = \frac{u^j}{k}\frac{(1-u^{jk})}{(1-u^j)}$$

where, the last equality follows from the sum of a geometric progression in the complex field. Since $u^k = 1$, it follows that $u^{jk} = 1$. Further, since u is the elementary k^{th} root of unity, $u^j = e^{2\pi j\sqrt{-1}/k} \neq 1$, for $1 \leq j < k$. Thus, the expression $(1 - u^{jk})/(1 - u^j) = 0$. Therefore, $\mathbf{E}\big[x^j\big] = 0$, for $1 \leq j < k$.

The conjugation operator is a 1-1 and onto operator in the field of complex numbers. Further, if x is a root of $x^k = 1$, then, $\bar{x}^k = \overline{x^k} = \bar{1} = 1$, and therefore, $\bar{x}$ is also a k^{th} root of unity. Thus, the conjugation operator, applied to the group of k^{th} roots of unity, results in a permutation of the elements in the group (actually, it is an isomorphism). It therefore follows that the sum of the j^{th} powers of the roots of unity is equal to the sum of the j^{th} powers of the conjugates of the roots of unity. Thus, $\mathbf{E}\big[\bar{x}^j\big] = \mathbf{E}\big[x^j\big]$. $\square$

Let Z be the random variable defined as $Z = \sum_{l \in [N]} f_l x_l$. The variable x_l, for each $l \in [N]$, is one of a randomly chosen root of $x^k = 1$. The family of variables $\{x_l\}$ is assumed to be $2k$-wise independent. The following lemma shows that Re Z^k is an unbiased estimator of F_k. Following [1, 2], we call Z as a *sketch*. The random variable Z can be efficiently maintained with respect to stream updates as follows. First, we choose a random hash function $\theta : [N] \rightarrow [k]$ drawn from a family of hash functions that is $2k$-wise independent. Further, we pre-compute the k^{th} roots of unity into an array $A[1..k]$ of size k (of complex numbers), that is, $A[r] = e^{2 \cdot \pi \cdot r \cdot \sqrt{-1}/k}$, for $r = 1, 2, \ldots, k$. For every stream update (l, v), we update the sketch as follows.

$$Z = Z + v \cdot A[\theta(l)]$$

The space required to maintain the hash function $\theta = \tilde{O}(k)$, and the time required for processing a stream update is also $\tilde{O}(k)$.

Lemma 1. $\mathbf{E}\big[\textit{Re } Z^k\big] = F_k$.

As the following lemma shows, the variance of this estimator is quite high.

Lemma 2. $\mathbf{Var}\big[\textit{Re } Z^k\big] = O(k^{2k} F_2^k)$.

This implies that $\mathbf{Var}\big[\text{Re } Z^k\big]/(\mathbf{E}\big[\text{Re } Z^k\big])^2 = O(F_2^k / F_k^2)$, which could be as large as n^{k-2}. To reduce the variance we organize the sketches in a hash table.

2.2 Organizing sketches in a hash table

Let $\phi : \{0, 1, \ldots, N-1\} \rightarrow [B]$ be a hash function that maps the domain $\{0, 1, \ldots, N-1\}$ into a hash table consisting of B buckets. The hash function ϕ is drawn from a family of hash functions $\mathcal{H}$ that is $2k$-wise independent. The random bits used by the hash family is independent of the random bits used by the family $\{x_l\}_{l \in \{0,1,\ldots,N-1\}}$, or, equivalently, the random bits used to generate ϕ and θ are independent. The indicator variable $y_{l,b}$, for any domain element $l \in \{0, 1, \ldots, N-1\}]$ and bucket $b \in [B]$, is defined as $y_{l,b} = 1$ if $\phi(l) = b$ and $y_{l,b} = 0$ otherwise. Associated with each bucket b is a sketch Z_b of the elements that have hashed to that bucket. The random variables, Y_b and Z_b are defined as follows.

$$Z_b = \sum_l f_l \cdot x_l \cdot y_{l,b}, \qquad Y_b = \text{Re } Z_b^k, \quad \text{and} \quad Y = \sum_{b \in [B]} Y_b$$

Maintaining the hash table of sketches in the presence of stream updates is analogous to maintaining Z. As discussed previously, let $\theta : \{0, 1, \ldots, N-1\} \rightarrow [k]$ denote a random hash function that is chosen from a $2k$-wise independent family of hash functions (and independently of the bits used by ϕ), and let $A[1 \ldots k]$ be an array whose j^{th} entry is $e^{2 \cdot \pi \cdot j \cdot \sqrt{-1}/k}$, for $j = 1, \ldots, k$. For every stream update (l, v), we perform the following operation.

$$Z_{\phi(l)} = Z_{\phi(l)} + v \cdot A[\theta(l)]$$

The time complexity of the update operation is $\tilde{O}(k)$. The sketches in the buckets except the bucket numbered $\phi(l)$ are left unchanged.

The main observation of the paper is that the hash partitioning of the sketch Y into $\{Y_b\}_{b\in[B]}$ reduces the variance of Y significantly, while maintaining that $\mathbf{E}\big[Y\big] = F_k$. This is stated in the lemma below.

Lemma 3. *Let $B \le 2n^{1-\frac{1}{k}}$. Then, $\mathbf{Var}\big[Y\big] = O(F_k^2 n^{k-2}/B^{k-1})$.*

A hash table organization of the sketches is normally used to reduce the time complexity of processing each stream update [6, 8]. However, for $k > 2$, the hash table organization of the sketches has the additional effect of reducing the variance.

Finally, we keep s_1 independent copies $Y[0], \ldots, Y[s_1 - 1]$ of the variable Y. The average of these variables is denoted by $\bar{Y}$; thus $\mathbf{Var}\big[\bar{Y}\big] = (1/s_1)\mathbf{Var}\big[Y\big]$. The result of the paper is summarized below, which states that $\bar{Y}$ estimates F_k to within an accuracy factor of $(1 \pm \epsilon)$ with constant probability greater than 1/2 (at least 2/3).

Theorem 4. *Let $n^{1-\frac{1}{k-1}} \le B \le 2 \cdot n^{1-\frac{1}{k-1}}$ and $s_1 = 6 \cdot 2^k \cdot k^{3k}/\epsilon^2$. Then, $\mathbf{Pr}\left\{|\bar{Y} - F_k| > \epsilon F_k\right\} \le 1/3$.*

The space usage of the algorithm is therefore $\tilde{O}(B \cdot s_1) = O(n^{1-\frac{1}{k-1}})$ bits, since a logarithmic overhead is required to store each sketch Z_b. To boost the confidence of the answer to at least $1 - 2^{-\Omega(s_2)}$, a standard technique of returning the median value among s_2 such average estimates can be used, as shown in [1, 2].

The algorithm assumes that the number of buckets in the hash table is B, where, $n^{1-\frac{1}{k-1}} \le B \le 2 \cdot n^{1-\frac{1}{k-1}}$. Since, in general, the number of distinct items in the stream is not known in advance, one possible method that can be used is as follows. First estimate n to within a factor of $(1 \pm \frac{1}{8})$ using an algorithm for estimating F_0, such as [10, 1, 2, 4]. This can be done with high probability, in space $O(\log N)$. Keep $2\log N + 4$ group of (independent) hash tables, such that the i^{th} group uses $B_i = \lceil 2^{i/2} \rceil$ buckets. Each group of the hash tables uses the data structure described earlier. At the time of inference, first n is estimated as $\hat{n}$, and, then, we choose a hash table group indexed by i such that $i = 2 \cdot \lceil (1 - \frac{1}{k-1}) \log (8 \cdot \hat{n}/7) \rceil$. This ensures that the hash table size B_i satisfies $n^{1-\frac{1}{k-1}} \le B_i \le 2 \cdot n^{1-\frac{1}{k-1}}$, with high probability. Since, the number of hash table groups is $2 \cdot \log N$, this construction adds an overhead in terms of both space complexity and update time complexity by a factor of $2 \cdot \log N$. In the remainder of the paper, we assume that n is known exactly, with the understanding that this assumption can be alleviated as described.

3 Analysis

The j^{th} frequency moment of the set of elements that map to bucket b under the hash function ϕ, is a random variable denoted by $F_{j,b}$. Thus, $F_{j,b} = \sum_l f_l^j y_{l,b}$. Further, since every element in the stream hashes to exactly one bucket, $\sum_b F_{j,b} = F_j$. We define $h_{l,b}$, for $l \in \{0, 1, \ldots, N-1\}$ and $b \in [B]$ to be $h_{l,b} = f_l \cdot y_{l,b}$. Thus, $F_{j,b} = \sum_l h_{l,b}^j$, for $j \ge 1$.

Notation: Marginal expectations. The random variables, $Y, \{Y_b\}_{b \in B}$ are functions of two families of random variables, namely, $\mathbf{x} = \{x_l\}_{l \in \{0,1,\ldots,N-1\}}$, used to generate the random roots of unity, and $\mathbf{y} = \{y_{l,b}\}$, $l \in \{0,1,\ldots,N-1\}$ and $b \in [B]$, used to map elements to buckets in the hash table. Our independence assumptions imply that these two families are mutually independent (i.e., their seeds use independent random bits), that is, $\mathbf{Pr}\{\mathbf{x} = \mathbf{u} \text{ and } \mathbf{y} = \mathbf{v}\} = \mathbf{Pr}\{\mathbf{x} = \mathbf{u}\} \cdot \mathbf{Pr}\{\mathbf{y} = \mathbf{v}\}$ Let $W = W(\mathbf{x}, \mathbf{y})$ be a random variable that is a function of the random variables in $\mathbf{x}$ and $\mathbf{y}$. For a fixed random choice of $\mathbf{y} = \mathbf{y_0}$, $\mathbf{E}_{\mathbf{x}}\big[W\big]$ denotes the marginal expectation of W as a function of y. That is, $\mathbf{E}_{\mathbf{x}}\big[W\big] = \sum_{\mathbf{u}} W(\mathbf{u}, \mathbf{y_0})\mathbf{Pr}\{\mathbf{x} = \mathbf{u}\}$. It follows that $\mathbf{E}\big[W\big] = \mathbf{E}_{\mathbf{y}}\big[\mathbf{E}_{\mathbf{x}}\big[W\big]\big]$.

Overview of the analysis. The main steps in the proof of Theorem 4 are as follows. In Section 3.1, we show that $\mathbf{E}_{\mathbf{x}}\big[Y\big] = F_k$. In Section 3.2, we show that $\mathbf{E}\big[\text{Re } Z^k\big] \leq k^{2k} F_2^k$. In Section 3.3, using the above result, we show that $\mathbf{E}_{\mathbf{x}}\big[Y^2\big] \leq k^{2k} \sum_b F_{2,b}^k$. Section 3.4 shows that $\mathbf{E}_{\mathbf{y}}\big[F_{2,b}^k\big] \leq (2/B + 2^k \cdot n^{k-2}/B^k) F_k^2$ and also concludes the proof of Theorem 4. Finally, we conclude in Section 4.

Notation: Multinomial Expansion. Let X be defined as $X = \sum_{l \in \{0,1,\ldots,N-1\}} a_l$, where, $a_l \geq 0$, for $l \in \{0,1,\ldots,N-1\}$. Then, X^k can be written as

$$X^k = \sum_{s=1}^{k} \sum_{e_1 + \cdots e_s = k, e_1 > 0, \cdots, e_s > 0} \binom{k}{e_1 e_2 \cdots e_s} \sum_{l_1 < l_2 < \cdots < l_s} a_{l_1}^{e_1} a_{l_2}^{e_j} \cdots a_{l_s}^{e_s}$$

where, s is the number of distinct terms in the product and e_i is the exponent of the i^{th} product term. The indices l_i are therefore necessarily distinct, $l_i \in \{0,1,\ldots,N-1\}, i = 1,2,\ldots,s$. For easy reference, the above equation is written and used in the following form.

$$X^k = \sum_{s, \mathbf{e}: Q(\mathbf{e}, s)} C(\mathbf{e}) \sum_{\mathbf{l}: R(\mathbf{e}, \mathbf{l}, s)} \Big(\prod_{j=1}^{s} a_{l_j}^{e_j}\Big) \quad . \tag{2}$$

where, $Q(\mathbf{e}, s) \equiv 1 \leq s \leq k$ and $\mathbf{e} = (e_1, e_2, \ldots, e_s)$ is s-dimensional and $\sum_{j=1}^{s} e_j = k$; $R(\mathbf{e}, \mathbf{l}, s) \equiv \mathbf{l} = (l_1, l_2, \ldots, l_s)$ is s-dimensional and $0 \leq l_1 < l_2 < \cdots < l_s \leq N-1$; and the multinomial coefficient $C(\mathbf{e}) = \binom{k}{e_1, \ldots, e_s}$. In this notation, the following inequality holds .

$$\sum_{\mathbf{l}: R(\mathbf{e}, \mathbf{l}, s)} \prod_{j=1}^{s} a_{l_j}^{e_j} \leq \prod_{j=1}^{s} \Big(\sum_{l} a_l^{e_j}\Big) \quad . \tag{3}$$

By setting $n = k$, and $a_1 = a_2 = \cdots = a_k = 1$, we obtain,

$$k^k = \sum_{\mathbf{e}, s} C(\mathbf{e}) \binom{k}{s} > \sum_{\mathbf{e}, s} C(\mathbf{e}).$$

By squaring the above equation on both sides, we obtain that $k^{2k} = (\sum_{\mathbf{e}, s} C(\mathbf{e}) \binom{k}{s})^2 > \sum_{\mathbf{e}, s} C^2(\mathbf{e})$. We therefore have the following inequalities.

$$\sum_{\mathbf{e}, s} C(\mathbf{e}) < k^k, \quad \sum_{\mathbf{e}, s} C^2(\mathbf{e}) < k^{2k} \quad . \tag{4}$$

3.1 Expectation

In this section, we show that $\mathbf{E}\big[\text{Re } Z^k\big] = F_k$, thereby proving Lemma 1, and that $\mathbf{E}_\mathbf{x}\big[Y\big] = F_k$.

Proof (of Lemma 1). Since the family of variables x_l's is k-wise independent, therefore

$$\mathbf{E}\big[\prod_{j=1}^{s} x_{l_j}^{e_j}\big] = \prod_{j=1}^{s} \mathbf{E}\big[x_{l_j}^{e_j}\big] \ .$$

Applying equation (2) to $Z^k = (\sum_l f_l x_l)^k$ and using linearity of expectation and k-wise independence property of x_l's, we obtain

$$\mathbf{E}\big[Z^k\big] = \sum_{s,\mathbf{e}:Q(\mathbf{e},s)} C(\mathbf{e}) \sum_{\mathbf{l}:R(\mathbf{e},\mathbf{l},s)} \big(\prod_{j=1}^{s} f_{l_j}^{e_j}\big)\big(\prod_{j=1}^{s} \mathbf{E}\big[x_{l_j}^{e_j}\big]\big) \ .$$

Using equation (1), we note that the term $\big(\prod_{j=1}^{s} \mathbf{E}\big[x_{l_j}^{e_j}\big]\big) = 0$, if $s > 1$, since in this case, $e_j < k$, for each $j = 1, \ldots, s$. Thus, the above summation reduces to

$$\mathbf{E}\big[Z^k\big] = \sum_l f_l^k = F_k \ .$$

Since F_k is real, $\mathbf{E}\big[\text{Re } Z^k\big]$ is also F_k, proving Lemma 1. □

Lemma 5. *Suppose that the family of random variables $\{x_l\}$ is k-wise independent. Then, $\mathbf{E}_\mathbf{x}\big[Y_b\big] = F_{k,b}$ and $\mathbf{E}_\mathbf{x}\big[Y\big] = \mathbf{E}\big[Y\big] = F_k$.*

Proof. We first show that $\mathbf{E}_\mathbf{x}\big[Y_b\big] = F_{k,b}$. $\mathbf{E}_\mathbf{x}\big[Z_b^k\big] = \mathbf{E}_\mathbf{x}\big[(\sum_l f_l y_{l,b} x_l)^k\big] = \mathbf{E}_\mathbf{x}\big[(\sum_l h_{l,b} x_l)^k\big]$, by letting $h_{l,b} = f_l \cdot y_{l,b}$. By an argument analogous to the proof of Lemma 1, we obtain $\mathbf{E}_\mathbf{x}\big[(\sum_l h_{l,b} x_l)^k\big] = \sum_l h_{l,b}^k = \sum_l f_l^k y_{l,b}^k = \sum_l f_{l,k}^k y_{l,b} = F_{k,b}$, (since $y_{l,b}$'s are binary variables). Since $F_{k,b}$ is always real, $\mathbf{E}_\mathbf{x}\big[Y_b\big] = \mathbf{E}_\mathbf{x}\big[\text{Re } Z_b^k\big] = F_{k,b}$. Finally, $\mathbf{E}_\mathbf{x}\big[Y\big] = \mathbf{E}_\mathbf{x}\big[\sum_b Y_b\big] = \sum_b \mathbf{E}_\mathbf{x}\big[Y_b\big] = \sum_b F_{k,b} = F_k$, since each element is hashed to exactly one bucket. Further, $\mathbf{E}\big[Y\big] = \mathbf{E}_\mathbf{y}\big[\mathbf{E}_\mathbf{x}\big[Y\big]\big] = \mathbf{E}_\mathbf{y}\big[F_k\big] = F_k$. □

3.2 Variance of Re Z^k

In this section, we estimate the variance of Re Z^k and derive some simple corollaries.

Lemma 6. *Let $W = Re \ (\sum_l a_l x_l)^k$. Then, $\mathbf{Var}\big[W\big] \le k^{2k}(\sum_l a_l^2)^k$.*

Proof. Let $X = (\sum_l a_l x_l)^k$. Then, $\mathbf{Var}\big[W\big] = \mathbf{E}\big[W^2\big] - (\mathbf{E}\big[W\big])^2 \le \mathbf{E}\big[X\bar{X}\big] - (\mathbf{E}\big[W\big])^2$. Using equation (2), for $X, \bar{X}$, we obtain the following.

$$X = \sum_{s,\mathbf{e}:Q(\mathbf{e},s)} C(\mathbf{e}) \sum_{\mathbf{l}:R(\mathbf{e},\mathbf{l},s)} \big(\prod_{j=1}^{s} a_{l_j}^{e_j}\big) \cdot \big(\prod_{j=1}^{s} x_{l_j}^{e_j}\big)$$

$$\bar{X} = \sum_{t,\mathbf{g}:Q(\mathbf{g},t)} C(\mathbf{g}) \sum_{\mathbf{l}:R(\mathbf{g},\mathbf{m},t)} \big(\prod_{j'=1}^{t} a_{m_{j'}}^{g_{j'}}\big) \cdot \big(\prod_{j'=1}^{t} \bar{x}_{m_{j'}}^{g_{j'}}\big)$$

Multiplying the above two equations, we obtain

$$X \cdot \bar{X} = \sum_{s,\mathbf{e}:Q(\mathbf{e},s)} \sum_{t,\mathbf{g}:Q(\mathbf{g},t)} C(\mathbf{e}) \cdot C(\mathbf{g}) \sum_{\mathbf{l}:R(\mathbf{e},\mathbf{l},s)} \sum_{\mathbf{l}:R(\mathbf{g},\mathbf{m},t)} \left(\prod_{j=1}^{s} a_{l_j}^{e_j}\right) \cdot \left(\prod_{j'=1}^{t} a_{m_{j'}}^{g_{j'}}\right) \cdot \left(\prod_{j=1}^{s} x_{l_j}^{e_j}\right) \cdot \left(\prod_{j'=1}^{t} \bar{x}_{m_{j'}}^{g_{j'}}\right).$$

The general form of the product of random variables that arises in the multinomial expansion of $X\bar{X}$ is $(\prod_{j=1}^{s} x_{l_j}^{e_j})(\prod_{j'=1}^{t} \bar{x}_{m_{j'}}^{g_{j'}})$. Since the random variables x_l's are $2k$-wise independent, using equation (1), it follows that,

$$\mathbf{E}\Big[\prod_{j=1}^{s} x_{l_j}^{e_j} \prod_{j'=1}^{t} \bar{x}_{m_{j'}}^{g_j}\Big] = \begin{cases} 1 & \text{if } s = t = 1, e_1 = g_1 = k \\ 1 & \text{if } s = t, t > 1, \mathbf{e} = \mathbf{g} \text{ and } \mathbf{l} = \mathbf{m}, \\ 0 & \text{otherwise.} \end{cases}$$

This directly yields the following.

$$\begin{aligned}
\mathbf{E}\big[X\bar{X}\big] &= \sum_{\mathbf{e},s:Q(\mathbf{e},s)} C^2(\mathbf{e}) \sum_{\mathbf{l}:R(\mathbf{e},\mathbf{l},s)} \prod_{j=1}^{s} a_{l_j}^{2e_j} \\
&\le \sum_{\mathbf{e},s} C^2(\mathbf{e}) \prod_{j=1}^{s} \Big(\sum_{l} a_l^{2e_j}\Big), \text{ by equation (3)} \\
&\le \sum_{\mathbf{e},s} C^2(\mathbf{e}) \prod_{j=1}^{s} \Big(\sum_{l} a_l^{2}\Big)^{e_j}, \text{ since } \sum_{l} a_l^{2e_j} \le \Big(\sum_{l} a_l^{2}\Big)^{e_j} \\
&= \Big(\sum_{l} a_l^{2}\Big)^{k} \Big(\sum_{\mathbf{e},s} C^2(\mathbf{e})\Big), \quad \text{since } \sum_{j=1}^{s} e_j = k \\
&\le \Big(\sum_{l} a_l^{2}\Big)^{k} \cdot k^{2k}, \quad \text{by equation (4).} \quad \Box
\end{aligned}$$

By letting $a_l = f_l$, $l \in \{0, 1, 2 \ldots, N-1\}$, Lemma 6 yields

$$\mathbf{Var}\big[\text{Re } Z^k\big] = \mathbf{Var}\big[\text{Re } \big(\sum_{l} f_l x_l\big)^k\big] \le k^{2k} F_2^k, \tag{5}$$

which is the statement of Lemma 2. By letting $a_l = h_{l,b} = f_l \cdot y_{l,b}$, where, b is a fixed bucket index, and $l \in \{0, 1, 2 \ldots, N-1\}$, yields the following equation.

$$\mathbf{E}_{\mathbf{x}}\big[Y_b^2\big] \le k^{2k} F_{2,b}^k, \quad \text{for } b \in [B]. \tag{6}$$

3.3 $\mathbf{Var}\big[Y\big]$: Vanishing of cross-bucket terms

We now consider the problem of obtaining an upper bound on $\mathbf{Var}\big[Y\big]$. Note that $\mathbf{Var}\big[Y\big] = \mathbf{E}_{\mathbf{y}}\big[\mathbf{E}_{\mathbf{x}}\big[Y^2\big]\big] - (\mathbf{E}_{\mathbf{y}}\big[\mathbf{E}_{\mathbf{x}}\big[Y\big]\big])^2$. From Lemma 5, $\mathbf{E}_{\mathbf{y}}\big[\mathbf{E}_{\mathbf{x}}\big[Y\big]\big] = F_k$. Thus,

$$\mathbf{Var}\big[Y\big] = \mathbf{E}_{\mathbf{y}}\big[\mathbf{E}_{\mathbf{x}}\big[Y^2\big]\big] - F_k^2 \ . \tag{7}$$

Lemma 7. $\mathbf{Var}[Y] \le k^{2k} \sum_b \mathbf{E_y}[F_{2,b}^k]$, *assuming independence assumption* I.

Proof. $\mathbf{E_x}[Y^2] = \mathbf{E_x}[(\sum_b Y_b)^2] = \mathbf{E_x}[\sum_b Y_b^2 + \sum_{a\neq b} Y_a Y_b] = \sum_b \mathbf{E_x}[Y_b^2] + \sum_{a\neq b} \mathbf{E_x}[Y_a Y_b]$.

We now consider $\mathbf{E_x}[Y_a Y_b]$, for $a \neq b$. Recall that $Y_a = \text{Re } Z_a^k$ (and analogously, Y_b is defined). For any two complex numbers z, w, (Re z)(Re w) $= (1/2)$Re $(z(w + \bar{w}))$. Thus, $Y_a Y_b = (\text{Re } Z_a^k)(\text{Re } Z_b^k) = (1/2)\text{Re } (Z_a^k Z_b^k + Z_a^k \overline{Z_b^k})$.

Let us first consider $\mathbf{E_x}[Z_a^k Z_b^k]$. The general term involving product of random variables is $(\prod_{j=1}^s f_{l_j}^{e_j}) \cdot (\prod_{j'=1}^t f_{m_j}^{g_{j'}}) \cdot (\prod_{j=1}^s y_{l_j,a} \cdot x_{l_j}^{e_j}) \cdot (\prod_{j'=1}^t y_{m_{j'},b} \cdot x_{m_{j'}}^{g_j})$. Consider the last two product terms in the above expression, that is, $(\prod_{j=1}^s y_{l_j,a} \cdot x_{l_j}^{e_j}) \cdot (\prod_{j'=1}^t y_{m_{j'},b} \cdot x_{m_{j'}}^{g_j})$. For any $1 \le j \le s$ and $1 \le j' \le t$, it is not possible that $l_j = m_{j'}$, that is, the same element whose index is given by $l_j = m_{j'}$ cannot simultaneously hash to two distinct buckets, a and b (recall that $a \neq b$). By $2k$-wise independence, we therefore obtain that the only way the above product term can be non zero (i.e., 1) on expectation, is that $s = t = 1$ and therefore, $e_1 = k$ and $g_1 = k$. Thus, we have $\mathbf{E}[Z_a^k Z_b^k] = \sum_{l,m} h_{l,a}^k h_{m,b}^k = F_{k,a} F_{k,b}$.

Using the same observation, it can be argued that $\mathbf{E_x}[Z_a^k \overline{Z_b^k}] = F_{k,a} F_{k,b}$. It follows that $\mathbf{E_x}[(1/2)(Z_a^k Z_b^k + Z_a^k \overline{Z_b^k}))] = F_{k,a} F_{k,b}$, which is a real number. Therefore $\mathbf{E_x}[\text{Re } (1/2)(Z_a^k Z_b^k + Z_a^k \overline{Z_b^k})] = F_{k,a} F_{k,b} = \mathbf{E_x}[Y_a Y_b]$.

By equation (7), $\mathbf{Var}[Y] = \mathbf{E_y}[\mathbf{E_x}[Y^2]] - F_k^2$. Further, from Lemma 5, $F_k = \mathbf{E_y}[\sum_b F_{k,b}]$. We therefore have,

$$\begin{aligned}
\mathbf{Var}[Y] &= \mathbf{E_y}\big[\mathbf{E_x}[Y^2] - \big(\sum_b F_{k,b}\big)^2\big] \\
&= \mathbf{E_y}\big[\sum_b \mathbf{E_x}[Y_b^2] + \sum_{a\neq b} \mathbf{E_x}[Y_a Y_b] - \big(\sum_b F_{k,b}\big)^2\big] \\
&= \mathbf{E_y}\big[\sum_b \mathbf{E_x}[Y_b^2] + \sum_{a\neq b} F_{k,a} F_{k,b} - \big(\sum_b F_{k,b}\big)^2\big], \quad \text{by above argument} \\
&= \mathbf{E_y}\big[\sum_b \mathbf{E_x}[Y_b^2] - \sum_b F_{k,b}^2\big] \\
&\le \mathbf{E_y}\big[\sum_b \mathbf{E_x}[Y_b^2]\big] \\
&\le \mathbf{E_y}\big[\sum_b k^{2k} F_{2,b}^k\big], \quad \text{by equation (6)}
\end{aligned}$$

□

3.4 Calculation of $\mathbf{E}[F_{2,b}^k]$

Given a t-dimensional vector $\mathbf{e} = (e_1, \ldots, e_t)$ such that $e_i > 0$, for $1 \le i \le t$ and $\sum_{j=1}^t e_j = k$, we define the function $\psi(\mathbf{e})$ as follows. Without loss of generality, let the indices e_j be arranged in non-decreasing order. Let $r = r(\mathbf{e})$ denote the largest index such that $e_r < k/2$. Then, we define the function $\phi(e)$ as follows.

$$\psi(\mathbf{e}) = n^{\sum_{j=1}^r (1 - 2e_j/k)} / B^t$$

The motivation of this definition stems from its use in the following lemma.

Lemma 8. *Suppose* $\sum_{j=1}^{t} e_j = k$ *and* $e_j > 0$*, for* $j = 1, \ldots, t$*. Then,* $\prod_{j=1}^{t} F_{2e_j} \leq \psi(\mathbf{e}) \cdot F_k^2 \cdot B^t$.

Proof. From [1, 2], $F_j \leq n^{1-j/k} F_k^{j/k}$, if $j < k$ and $F_j \leq F_k^{j/k}$, if $j > k$. Thus,

$$\prod_{j=1}^{t} F_{2e_j} = \left(\prod_{j=1}^{r} F_{2e_j}\right)\left(\prod_{j=r+1}^{t} F_{2e_j}\right) = \left(\prod_{j=1}^{r} n^{1-2e_j/k} F_k^{2e_j/k}\right)\left(\prod_{j=r+1}^{t} F_k^{2e_j/k}\right)$$
$$= n^{\sum_{j=1}^{r}(1-2e_j/k)} F_k^{\sum_{j=1}^{t} 2e_j/k} = \psi(\mathbf{e}) \cdot B^t \cdot F_k^2, \quad \text{since} \sum_j e_j = k. \square$$

The function ψ satisfies the following property that we use later.

Lemma 9. *If* $B < 2 \cdot n^{1-\frac{1}{k}}$*, then,* $\psi(\mathbf{e}) \leq$ *max* $(2/B, 2^k \cdot n^{k-2}/B^k)$.

Proof. Let $\mathbf{e}$ be a t-dimensional vector. If $t = 1$, $\psi(\mathbf{e}) = 1/B$. If $t = r$, then $\psi(\mathbf{e}) = n^{t-2}/B^t \leq 2^k \cdot n^{k-2}/B^k$. If $t \geq r+2$, then $\psi(\mathbf{e}) = (2^t/B^t) \cdot n^{t-((t-r)+\sum 2e_j/k)} < 2^t \cdot n^{t-2}/B^t \leq 2^k \cdot n^{k-2}/B^k$. Finally, let $t = r+1$. Then,

$$\psi(\mathbf{e}) = 2^t \cdot n^{t-1-2\sum e_j/k}/B^t \leq 2^t \cdot n^{t-1-2(t-1)/k}/B^t,$$

since $\sum_{j=1}^{r} e_j \geq r = t-1$. Thus,

$$\psi(\mathbf{e}) \leq \psi(\mathbf{e})(2 \cdot n^{1-\frac{1}{k}}/B)^{k-t} \leq (2^t \cdot n^{t-1-2(t-1)/k}/B^t)(2 \cdot n^{1-\frac{1}{k}}/B)^{k-t} =$$
$$2^k \cdot n^{k-2-(t-2)/k}/B^k \leq 2^k \cdot n^{k-2}/B^k \ .$$

where, the first inequality follows from the assumption that $B < n^{1-\frac{1}{k}}$ and the second inequality follows because $t \geq 2$. $\square$

Lemma 10. *Let* $B < 2 \cdot n^{1-\frac{1}{k}}$*. Then,* $\mathbf{E}\big[F_{2,b}^k\big] < k^k F_k^2(2/B + 2^k \cdot n^{k-2}/B^k)$.

Proof. For a fixed b, the variables $y_{l,b}$ are k-wise independent. $F_{2,b}$ is a linear function of $y_{l,b}$. Thus, $F_{2,b}^k$ is a symmetric multinomial of degree k, as follows.

$$F_{2,b}^k = (\sum_l f_l^2 y_{l,b})^k$$
$$= \sum_{s,\mathbf{e}} C(\mathbf{e}) \sum_{l_1 < l_2 < \cdots l_s} f_{l_1}^{2e_1} \cdots f_{l_s}^{2e_s} y_{l_1,b} \cdot y_{l_2,b} \cdot y_{l_s,b} \ .$$

Taking expectations, and using k-wise independence of the $y_{l,b}$'s, we have,

$$\begin{aligned}
\mathbf{E}\big[F_{2,b}^k\big] &= \sum_{s,\mathbf{e}} C(\mathbf{e}) \sum_{l_1<l_2<\cdots<l_s} f_{l_1}^{2e_1}\cdots f_{l_j}^{2e_j}\,\mathbf{E}\big[y_{l_1,b}\cdot y_{l_2,b}\cdot y_{l_s,b}\big] \\
&= \sum_{s,\mathbf{e}} C(\mathbf{e}) \sum_{l_1<l_2<\cdots<l_s} f_{l_1}^{2e_1}\cdots f_{l_j}^{2e_j}\,\mathbf{E}\big[y_{l_1,b}\big]\cdot\mathbf{E}\big[y_{l_2,b}\big]\cdots\mathbf{E}\big[y_{l_s},b\big] \\
&= \sum_{s,\mathbf{e}} C(\mathbf{e}) \sum_{l_1<l_2<\cdots<l_s} f_{l_1}^{2e_1}\cdots f_{l_j}^{2e_j}\,\frac{1}{B^s}, \quad \text{since, } \mathbf{E}\big[y_{l_j,b}\big] = \frac{1}{B} \\
&\le \sum_{s,\mathbf{e}} C(\mathbf{e})\cdot(1/B^s)\cdot\prod_{j=1}^{s} F_{2e_j} \\
&\le \sum_{s,\mathbf{e}} C(\mathbf{e})\cdot\psi(\mathbf{e})\cdot F_k^2, \quad \text{by Lemma 8} \\
&\le \sum_{s,\mathbf{e}} C(\mathbf{e})\cdot F_k^2\cdot(2/B + 2^k\cdot n^{k-2}/B^k), \quad \text{by Lemma 9} \\
&\le k^k\cdot F_k^2\cdot(2/B + 2^k\cdot n^{k-2}/B^k), \quad \text{since, } \sum_{s,\mathbf{e}} C(\mathbf{e}) < k^k \qquad \square
\end{aligned}$$

Combining the result of Lemma 7 with Lemma 10, we obtain the following bound on $\mathbf{Var}\big[Y\big]$.

$$\mathbf{Var}\big[Y\big] \le k^{3k}\cdot F_k^2\cdot(2 + 2^k\cdot n^{k-2}/B^{k-1}) \qquad (8)$$

Recall that $\bar{Y}$ is the average of s_1 independent estimators, each calculating Y. The main theorem of the paper now follows simply.

Proof (of Theorem 4). By Chebychev's inequality, $\mathbf{Pr}\left\{|\bar{Y} - F_k| > \epsilon F_k\right\} < \mathbf{Var}\big[\bar{Y}\big]/(\epsilon^2 F_k^2)$. Substituting Equation (8), we have $\mathbf{Var}\big[\bar{Y}\big]/(\epsilon^2\cdot F_k^2) \le 1/3$. $\square$

4 Conclusions

The paper presents a method for estimating the k^{th} frequency moment, for $k > 2$, of data streams with general update operations. The algorithm has space complexity $\tilde{O}(n^{1-\frac{1}{k-1}}))$ and is based on constructing random linear combinations using randomly chosen k^{th} roots of unity. A gap remains between the lower bound for this problem, namely, $O(n^{1-2/k})$, for $k > 2$, as proved in [3, 5] and the complexity of a known algorithm for this problem.

References

1. Noga Alon, Yossi Matias, and Mario Szegedy. "The Space Complexity of Approximating the Frequency Moments". In *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing STOC, 1996*, pages 20–29, Philadelphia, Pennsylvania, May 1996.
2. Noga Alon, Yossi Matias, and Mario Szegedy. "The space complexity of approximating frequency moments". *Journal of Computer Systems and Sciences*, 58(1):137–147, 1998.

3. Ziv Bar-Yossef, T.S. Jayram, Ravi Kumar, and D. Sivakumar. “An information statistics approach to data stream and communication complexity”. In *Proceedings of the 34th ACM Symposium on Theory of Computing (STOC), 2002*, pages 209–218, Princeton, NJ, 2002.
4. Ziv Bar-Yossef, T.S. Jayram, Ravi Kumar, D. Sivakumar, and Luca Trevisan. “Counting distinct elements in a data stream”. In *Proceedings of the 6th International Workshop on Randomization and Approximation Techniques in Computer Science, RANDOM 2002*, Cambridge, MA, 2002.
5. Amit Chakrabarti, Subhash Khot, and Xiaodong Sun. “Near-Optimal Lower Bounds on the Multi-Party Communication Complexity of Set Disjointness”. In *Proceedings of the 18th Annual IEEE Conference on Computational Complexity, CCC 2003*, Aarhus, Denmark, 2003.
6. Moses Charikar, Kevin Chen, and Martin Farach-Colton. “Finding frequent items in data streams”. In *Proceedings of the 29th International Colloquium on Automata Languages and Programming*, 2002.
7. Don Coppersmith and Ravi Kumar. “An improved data stream algorithm for estimating frequency moments”. In *Proceedings of the Fifteenth ACM SIAM Symposium on Discrete Algorithms*, New Orleans, LA, 2004.
8. G. Cormode and S. Muthukrishnan. “What’s Hot and What’s Not: Tracking Most Frequent Items Dynamically”. In *Proceedings of the Twentysecond ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, San Diego, California, May 2003.
9. Joan Feigenbaum, Sampath Kannan, Martin Strauss, and Mahesh Viswanathan. “An Approximate L^1-Difference Algorithm for Massive Data Streams”. In *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*, New York, NY, October 1999.
10. Philippe Flajolet and G.N. Martin. “Probabilistic Counting Algorithms for Database Applications”. *Journal of Computer Systems and Sciences*, 31(2):182–209, 1985.
11. Sumit Ganguly. “A bifocal technique for estimating frequency moments over data streams”. *Manuscript*, April 2004.
12. Sumit Ganguly, Minos Garofalakis, and Rajeev Rastogi. “Processing Set Expressions over Continuous Update Streams”. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, San Diego, CA, 2003.
13. Piotr Indyk. “Stable Distributions, Pseudo Random Generators, Embeddings and Data Stream Computation”. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science*, pages 189–197, Redondo Beach, CA, November 2000.
14. M. Saks and X. Sun. “Space lower bounds for distance approximation in the data stream model”. In *Proceedings of the 34th ACM Symposium on Theory of Computing (STOC), 2002*, 2002.