Deterministically estimating data stream frequencies

Sumit Ganguly

Indian Institute of Technology, Kanpur, India

Abstract. We consider updates to an n-dimensional frequency vector of a data stream, that is, the vector f is updated coordinate-wise by means of insertions or deletions in any arbitrary order. A fundamental problem in this model is to recall the vector approximately, that is to return an estimate $\hat{f}$ of f such that

$$|\hat{f}_i - f_i| < \epsilon \|f\|_p, \text{ for every } i = 1, 2, \ldots, n,$$

where ϵ is an accuracy parameter and p is the index of the ℓ_p norm used to calculate the norm $\|f\|_p$. This problem, denoted by $\text{ApproxFreq}_p(\epsilon)$, is fundamental in data stream processing and is used to solve a number of other problems, such as heavy hitters, approximating range queries and quantiles, approximate histograms, etc..

Suppressing poly-logarithmic factors in n and $\|f\|_1$, for $p = 1$ the problem is known to have $\tilde{\Theta}(1/\epsilon)$ randomized space complexity [2, 4] and $\tilde{\Theta}(1/\epsilon^2)$ deterministic space complexity[6, 7]. However, the deterministic space complexity of this problem for any value of $p > 1$ is not known. In this paper, we show that the deterministic space complexity of the problem $\text{ApproxFreq}_p(\epsilon)$ is $\tilde{\Theta}(n^{2-2/p}/\epsilon^2)$ for $1 < p < 2$, and $\Theta(n)$ for $p \geq 2$.

1 Introduction

In the data streaming model, computation is performed over a sequence of rapidly and continuously arriving data in an online fashion by maintaining a sub-linear space summary of the data. A data stream may be modeled as a sequence σ of updates of the form $(index, i, v)$, where, $index$ is the position of the update in the sequence, $i \in [n] = \{1, 2, \ldots, n\}$ and v is the update indicated by this record to the frequency f_i of i. The frequency vector $f(\sigma)$ of the stream σ is defined as:

$$f(\sigma) = \sum_{(index,i,v) \in \sigma} v \cdot e_i$$

where, $e_1, \ldots, e_n$ are the elementary n-dimensional unit vectors (i.e., e_i has 1 in position i and 0's elsewhere).

The problem of estimating the item frequencies of a data stream is to approximately recall the frequency vector of the stream. More precisely, the problem, denoted by $\text{ApproxFreq}_p(\epsilon)$, is to design a data stream processing algorithm that can return an n-dimensional vector f' satisfying $err_p(f', f(\sigma)) \le \epsilon$, for $p \ge 1$, where,

$$err_p(f', f) = \frac{\|f' - f\|_\infty}{\|f\|_p} \ .$$

This problem is fundamental in data stream processing. Solutions to this problem are used to find approximate frequent items (also called heavy hitters) [4, 5, 12, 13, 15], approximate range queries and quantiles [9, 4], and approximately v-optimal histograms [8, 10].

Review of algorithms for $\text{APPROXFREQ}_p(\epsilon)$. The problem $\text{APPROXFREQ}_p(\epsilon)$ is widely studied for $p = 1$ and for $p = 2$. For $p = 1$ and for insert-only streams, the algorithm of [15, 5, 12] uses space $\Theta((1/\epsilon) \log m)$, where $m = \max_i f_i$. The algorithm works only for insert-only streams (i.e., no decrement updates) and has optimal $O(1)$ time complexity for processing each stream update. Other algorithms presented for this problem include the sticky sampling technique [14] that uses space $O((1/\epsilon)(\log n)(\log m))$.

For general streams allowing arbitrary insertions and deletions, the randomized algorithms COUNT-MIN [4] and COUNTSKETCH [3] are applicable for solving the problems $\text{APPROXFREQ}_1(\epsilon)$ and $\text{APPROXFREQ}_2(\epsilon)$ respectively. These algorithms are randomized. The COUNT-MIN algorithm uses space $O((1/\epsilon)(\log mn)(\log 1/\delta))$, where $1 - \delta$ is the confidence parameter of the randomized algorithm. The COUNTSKETCH algorithm solves $\text{APPROXFREQ}_2(\epsilon)$ using space $O((1/\epsilon^2)(\log mn)(\log(1/\delta)))$. Both algorithms are space-optimal up to poly-logarithmic factors.

We now consider deterministic solutions to the problem $\text{APPROXFREQ}_p(\epsilon)$ for general streams. Deterministic algorithms have certain advantages, as is exemplified by the following scenario. Consider a service provider that wishes to give a discount to all its customers whose business with the company is a certain significant fraction (say 0.01%) of its revenue. The scheme is supposedly continuous, namely, that if a customer becomes a highly-valued customer then s/he gets the benefit immediately and vice-versa. For economy of space and time, the decision about whether a customer should be given a discount is done by a stream processing algorithm of the kind discussed earlier. If the algorithm is randomized, there is a chance, albeit small, that a highly valued customer is misclassified, resulting in an unhappy customer. Deterministic algorithms do not use random coin tosses and cannot lead to such grievances.

The algorithms in [5, 12, 15] are deterministic, however, these algorithms are applicable only for insert-only streams. The CR-precis algorithm [7] is a deterministic algorithm for $\text{APPROXFREQ}_1(\epsilon)$ for general streams with insertions and deletions and uses space $O(\epsilon^{-2}(\log m \log n)^2)$ bits, where, $m = \|f(\sigma)\|_\infty$. The work in [6] shows that any total, deterministic algorithm for solving the $\text{APPROXFREQ}_1(\epsilon)$ problem requires $\Omega\left((\log m)/\epsilon^2\right)$ bits. Thus, the deterministic space complexity of $\text{APPROXFREQ}_p(\epsilon)$ is resolved to $\tilde{\Theta}(\epsilon^{-2})$ for $p = 1$, where, $\tilde{\Theta}$ notation suppresses poly-logarithmic factors in n and m.

No results are known for space bounds for deterministic algorithms for $\text{APPROXFREQ}_p(\epsilon)$, for $p > 1$. The problem is fundamental, for instance, in the randomized case, the COUNTSKETCH algorithm solves $\text{APPROXFREQ}_2(\epsilon)$ using space $\tilde{\Theta}(\epsilon^{-2})$, and this important result is the basis for a number of space-optimal algorithms for estimating frequency moments [11, 1], approximate histograms [8], etc.. Therefore, understanding the space complexity of a deterministic solution to the problem $\textsc{ApproxFreq}_p(\epsilon)$ is of basic importance.

Contributions. We present space lower and upper bounds for deterministic algorithms for $\textsc{ApproxFreq}_p(\epsilon)$ for $p \geq 1$. We show that for $p \geq 2$, solving $\textsc{ApproxFreq}_p(\epsilon)$ requires $\Omega(n)$ space. For $p \in [1, 2)$, the space requirement is $\Omega(n^{2-2/p}(\log m)/\epsilon^2)$. Finally, we show that the upper bounds are matched by suitably modifying the CR-precis algorithm. The formal statement of our result is as follows.

Theorem 1. *For $\epsilon \leq 1/8$ and $p \geq 2$, any deterministic algorithm that solves $\textsc{ApproxFreq}_p(\epsilon)$ over general data streams requires space $\Omega(n \log m)$. For $1 \leq p < 2$ and $\epsilon \geq 0.5n^{1/p-1/2}$, any deterministic algorithm that solves $\textsc{ApproxFreq}_p(\epsilon)$ over general data streams requires space $\Omega(\epsilon^{-2}\ n^{2-2/p}\ \log m)$. Further, these lower bounds can be matched by algorithms up to poly-logarithmic factors.*

Organization. The remainder of the paper is organized as follows. Section 2 reviews work on stream automaton, which is is used to prove the lower bounds. Sections 3 and 4 presents lower and upper bounds respectively, for the space complexity of streaming algorithms for $\textsc{ApproxFreq}_p(\epsilon)$.

2 Review: Stream Automaton

We model a general stream over the domain $[n] = \{1, 2, \ldots, n\}$ as a sequence of individual records of the form $(index, a)$, where, *index* represents the position of this record in the sequence and a belongs to the set $\Sigma = \Sigma_n = \{e_1, -e_1, \ldots, e_n, -e_n\}$. Here, e_i refers to the n-dimensional elementary vector $(0, \ldots, 0, 1$ (ith position)$, 0 \ldots, 0)$. The *frequency* of a data stream σ, denoted by $f(\sigma)$ is defined as the sum of the elementary vectors in the sequence. That is,

$$f(\sigma) = \sum_{(index, v) \in \sigma} v \ .$$

The concatenation of two streams σ and τ is denoted by $\sigma \circ \tau$. The size of a data stream σ is defined as follows.

$$|\sigma| = \max_{\sigma' \text{ sub-sequence of } \sigma} \|f(\sigma')\|_\infty \ .$$

A deterministic **stream automaton** [6] is an abstraction for deterministic algorithms for processing data streams. It is defined as a two tape Turing machine, where the first tape is a one-way (unidirectional) input tape that contains the sequence σ of updates that constitutes the stream. Each update is a member of Σ, that is, it is an elementary vector or its inverse, e_i or $-e_i$. The second tape is a (bidirectional) two way work-tape. A configuration of a stream automaton is modeled as a triple (q, h, w), where, q is a state of the finite control, h is the current head position of the work-tape and w is the content of the work-tape. The set of configurations of a stream automaton A that are reachable from the initial configuration o on some input stream is denoted as $C(A)$. The set of configurations of an automaton A that is reachable from the origin o for some input stream σ with $|\sigma| \le m$ is denoted by $C_m(A)$. A stream automaton may be viewed as a tuple $(n, C, o, \oplus, \psi)$, where, $\oplus : C \times \Sigma \to C$ is the configuration transition function and $\psi : C \to O$ is the output function. The transition function, written as $s \oplus t$, where, $s \in C$ and t is a stream update, denotes the configuration of the algorithm after it starts from configuration s and processes the stream record t. We generally write the transition function in infix notation. The notation is generalized so that $a \oplus \sigma$ denotes the current configuration of the automaton starting from configuration and processing the records of the stream σ in a left to right sequence, that is,

$$s \oplus (\sigma \circ \tau) \stackrel{\text{def}}{=} (s \oplus \sigma) \oplus \tau \ .$$

After processing the input stream σ, the stream automaton prints the output

$$\text{output}_A(\sigma) = \psi(o \oplus \sigma) \ .$$

The automaton A is said to have space function $\text{Space}(A, m)$, provided, for all input streams σ such that $|\sigma| \le m$, the number of cells used on the work-tape during the processing of input is bounded above by $\text{Space}(A, m)$. It is said to have communication function $\text{Comm}(A, m) = \log|C_m(A)|$. The communication function can be viewed as a lower bound of the *effective* space usage of an automaton. The space or communication function does not include the space used by the automaton A to print its output. This allows the automaton to print outputs of size $\Omega(\text{Space}(A, m))$.

The approximate computation of a function $g : \mathbb{Z}^n \to O$ of the frequency vector $g(f(\sigma))$ is specified by a binary approximation predicate $\textsc{Approx} : E \times E \to \{\textsc{true}, \textsc{false}\}$ such that an estimate $\hat{a} \in O$ is considered an acceptable approximation to the true value $a \in O$ provided $\textsc{Approx}(\hat{a}, a) = \textsc{true}$ and is not considered to be an acceptable approximation if $\textsc{Approx}(\hat{a}, a) = \textsc{false}$. A stream automaton A is said to compute a function $g : \mathbb{Z}^n \to O$ of the frequency vector $f(\sigma)$ of its input stream σ with respect to the approximation predicate $\textsc{Approx}$, provided

$$\textsc{Approx}(\psi(\sigma), g(f(\sigma))) = \textsc{true}$$

for all feasible input streams σ. A stream automaton is said to be *total* if the feasible input set is the set of all input streams over the domain $[n]$ and is said to be *partial* otherwise. The class $\text{STR}\mathsf{freq}$ represents data streaming algorithms for computing approximation of (partial or total) functions of the frequency vector of the input stream. The notation $\mathbb{Z}_{2m+1}$ denotes the set of integers $\{-m, \ldots, 0, \ldots, m\}$.

A stream automaton is said to be *path independent* if for any reachable configuration $a \in C(A)$, the configuration obtained by starting from a and processing any input stream σ is dependent only on a and $f(\sigma)$. That is, $a \oplus \sigma$ depends only on a and $f(\sigma)$. The *kernel* of a path independent automaton is defined as

$$K(A) = \{a \in C(A) \mid \exists \sigma \text{ s.t. } o \oplus \sigma = o \text{ and } f(\sigma) = 0\} \ .$$

It is shown in [6] that the kernel of a path independent automaton is a sub-module of $\mathbb{Z}^n$. A stream automaton is said to be *free* if it is path-independent and its kernel is a free module. We present the basic theorem of stream automaton.

Theorem 2 ([6]). *For every stream automaton $A = (n, C_A, o_A, \oplus_A, \psi_A)$, there exists a path-independent stream automaton $B = (n, C_B, o_B, \oplus_B, \psi_B)$ such that the following holds.*

(1.) *For any* Approx *predicate and any total function $g : \mathbb{Z}^n \to O$,* Approx$(\psi_B(\sigma), g(\sigma))$ *holds if*
Approx$(\psi_A(\sigma), g(\sigma))$ *holds.*

(2.) *$Comm(B, m) \leq Comm(A, m)$.*

(3.) *There exists a sub-module $M \subset \mathbb{Z}^n$ and an isomorphic map $\varphi : C_B \to \mathbb{Z}^n/M$ where, $(\mathbb{Z}^n/M, \bigoplus)$ is viewed as a module with binary addition operation $\bigoplus$, such that for any stream σ,*

$$\varphi(a \oplus \sigma) = \varphi(a) \bigoplus \ [f(\sigma)]$$

where, $x \mapsto [x]$ is the canonical homomorphism from $\mathbb{Z}^n$ to $\mathbb{Z}^n/M$ (that is, $[x]$ is the unique coset of M to which x belongs).

(4.) *$Comm(B, m) = O((n - \dim M) \log m)$, where, $\dim M$ is the dimension of M.*

Conversely, given any sub-module $M \subset \mathbb{Z}^n$, a stream automaton $A = (n, C_A, o_a, \oplus_A, \psi_A)$ can be constructed such that there is an isomorphic map $\varphi : C_A \to \mathbb{Z}^n/M$ such that for any stream σ,

$$\varphi(a \oplus \sigma) = \varphi(a) \bigoplus \ [f(\sigma)] \ .$$

where, $\bigoplus$ is the addition operation of $\mathbb{Z}^n/M$, and

$$\begin{aligned} Comm(A, m) &= \log \left\lceil \left| \{[x] : x \in \mathbb{Z}^n_{2m+1}\} \right| \right\rceil \\ &= \Theta((n - \dim M) \log m) \quad \square \end{aligned}$$

3 Lower bounds for ApproxFreq$_p$

In this section, we establish deterministic space lower bounds for ApproxFreq$_p(\epsilon)$

Theorem 2 enables us to restrict attention to path independent automata in general, for all frequency-dependent computation. Lemma 1 further allows us to restrict our attention to free automata, for the problem of ApproxFreq$_p(\epsilon)$, while incurring a factor of 4 relaxation.

Lemma 1. *Suppose A is a path independent stream automaton for solving* $\textsc{ApproxFreq}_p(\epsilon)$ *over domain $[n]$ and has kernel M. Then, there exists a free automaton B with kernel M' such that $M' \supset M$, $\mathbb{Z}^n/M'$ is free, and* $err_p(\min_p(x+M'), x) \le 4\epsilon$.

The proof is similar in spirit to a corresponding Lemma in [6] and is given in the Appendix for completeness.

Consider a free automaton A over domain $[n]$ with kernel M that is a free module and let M^e denote the unique smallest dimension subspace of $\mathbb{R}^n$ that contains M. Let V be a $n \times k$ matrix whose columns are orthonormal and form a basis of $\mathbb{R}^n/M^e$. Let U denote an orthonormal basis of M^e, so that $[V\ U]$ forms an orthonormal basis of $\mathbb{R}^n$. For $x \in \mathbb{R}^n$, the coset $x + M^e = \{y : V^T y = V^T x\}$. For a given coset $x + M^e$, let $\bar{x}$ denote the element $y \in x + M^e$ with the smallest value of $\|y\|_2$. Clearly, $\bar{x}$ is the element in $x + M^e$ whose coordinates along U are all 0. Therefore,

$$\bar{x} = [V\ U]\begin{bmatrix} V^T x \\ 0 \end{bmatrix} = VV^T x \ . \tag{1}$$

Lemma 2. *If* $err_2(\bar{x}, x) \le \epsilon$ *for all x, then,* $\mathrm{rank}(V) \ge n(1-\epsilon)$.

Proof. Let $\mathrm{rank}(V) = k$. The condition $err_2(\bar{x}, x) \le \epsilon$ is equivalent to

$$\|(VV^T - I)x\|_\infty \le \epsilon \|x\|_2 \ .$$

In particular, this condition holds for the standard unit vectors $x = e_1, e_2, \ldots, e_n$ respectively. Thus, $\|VV^T e_i - e_i\|_\infty \le \epsilon$, for $i = 1, 2, \ldots, n$. This implies that $|(VV^T)_{ii} - 1| \le \epsilon$. Thus,

$$\mathrm{trace}(VV^T) \ge n(1-\epsilon) \ .$$

Since V has rank k and has k orthonormal columns, the eigenvalues of VV^T are 1 with multiplicity k and 0 with multiplicity $n-k$. Thus, $\mathrm{trace}(VV^T) = k$. Therefore, $n(1-\epsilon) \le \mathrm{trace}(VV^T) = k$. □

The lower bound proof for $1 \le p < 2$ is slightly more complicated. We first prove the following lemma.

Lemma 3. *For any orthonormal basis $[VU]$ of $\mathbb{R}^n$ such that* $\mathrm{rank}(V) = k$ *and for any $1 < p < 2$, there exists $i \in [n]$ such that $\|VV^T e_i\|_2 \le 2k/n$ and $\|VV^T e_i\|_p \le 2n^{1/p-1}\sqrt{k}$.*

Proof. Since, V has orthonormal columns

$$\|VV^T e_i\|_2^2 = \|V^T e_i\|_2^2 = (VV^T e_i)_i \ . \tag{2}$$

Therefore,

$$\mathrm{trace}(VV^T) = \sum_{i=1}^{n} (VV^T e_i)_i = \sum_{i=1}^{n} \|VV^T e_i\|_2^2 \tag{3}$$

The trace of VV^T is the sum of the eigenvalues of VV^T. Suppose $\mathrm{rank}(V) = k$. Since, V has orthonormal columns and has rank k, VV^T has eigenvalue 1 with multiplicity k and eigenvalue 0 with multiplicity $n-k$. Thus, $\mathrm{trace}(VV^T) = k$. By (3)

$$k = \mathrm{trace}(VV^T) = \sum_{i=1}^{n} \|VV^T e_i\|_2^2 \ . \tag{4}$$

Further, since, $\|x\|_p \le \|x\|_2 \cdot n^{1/p-1/2}$

$$\begin{aligned}\sum_{i=1}^{n} \|VV^T e_i\|_p &\le \sum_{i=1}^{n} \|VV^T e_i\|_2 (n^{1/p-1/2}) \\ &\le \sqrt{n} \left(\sum_{i=1}^{n} \|VV^T e_i\|_2^2 \right)^{1/2} n^{1/p-1/2} \\ &\quad \{\text{by Cauchy-Schwartz inequality }\} \\ &= n^{1/p}\sqrt{k} \qquad \text{by (4)} \ .\end{aligned} \tag{5}$$

Let

$$\begin{aligned} J &= \{i : \|VV^T e_i\|_2^2 \le 2k/n\}, \text{ and} \\ K &= \{i : \|VV^T e_i\|_p \le 2n^{1/p-1}\sqrt{k}\} \ . \end{aligned}$$

Therefore, by (4) and (5), $|J| > \frac{n}{2}$ and $|K| > \frac{n}{2}$. Hence, $J \cap K \ne \phi$, that is, there exists i such that

$$\|VV^T e_i\|_2 \le (2k/n)^{1/2} \text{ and } \|VV^T e_i\|_p \le 2n^{1/p-1}\sqrt{k} \ .$$

□

Lemma 4. *Let A be a free automaton that solves the problem* $\textsc{ApproxFreq}_p(\epsilon)$ *over the domain $[n]$ for some $1 \le p < 2$ and has kernel M. Let M^e be the smallest dimension subspace of $\mathbb{R}^n$ containing M. Let V, U be a collection of vectors that forms an orthonormal basis for $\mathbb{R}^n$ such that U spans M^e and V spans $\mathbb{R}^n / M^e$. Then, for $\epsilon \ge 2n^{1/2-1/p}$, $\mathrm{rank}(V) \ge \frac{n^{2-2/p}}{16\epsilon^2}$.*

Proof. By Lemma 3, there exists i such that

$$\begin{aligned} \|VV^T e_i\|_2^2 &\le \frac{2k}{n} \text{ and} \\ \|VV^T e_i\|_p &\le 2n^{1/p-1}\sqrt{k} \ . \end{aligned} \tag{6}$$

Since, $e_i - VV^T e_i = UU^T e_i \in M^e$, therefore,

$$\begin{aligned} \epsilon &\ge err_p(e_i - VV^T e_i, 0) \\ &= \frac{\|e_i - VV^T e_i\|_\infty}{\|e_i - VV^T e_i\|_p} \ . \end{aligned}$$

Therefore,

$$\|e_i - VV^T e_i\|_\infty \le \epsilon \|VV^T e_i - e_i\|_p \ . \tag{7}$$

By (2),

$$(VV^T e_i)_i = \|VV^T e_i\|_2^2 \le \frac{2k}{n} \ .$$

Therefore,

$$\begin{aligned}\|e_i - VV^T e_i\|_\infty &\ge |(e_i - VV^T e_i)_i| = 1 - \|VV^T e_i\|_2^2 \\ &\ge 1 - \frac{2k}{n}, \quad \text{by (6).}\end{aligned}$$

Substituting in (7),

$$\begin{aligned}1 - \frac{2k}{n} &\le \|e_i - VV^T e_i\|_\infty \\ &\le \epsilon \|VV^T e_i - e_i\|_p \\ &\le \epsilon\big(\|VV^T e_i\|_p + 1\big) \\ &\le \epsilon(2n^{1/p-1}\sqrt{k} + 1)\end{aligned}$$

where, the second to last inequality follows from using triangle inequality over pth norms and the last inequality follows from (6). Simplifying, we obtain that

$$k \ge \frac{n^{2-2/p}}{16\epsilon^2}, \text{ provided, } \epsilon \ge 2n^{1/2-1/p} \ .$$

□

We recall that as shown in [6], $\text{Comm}(A, m) \ge \text{rank}(V) \log(2m + 1)$.

Proof (Of Theorem 1). We first consider the case $p = 2$ and $p > 2$. By Theorem 2, it follows that corresponding to any stream automaton A_n, there exists a path independent stream automaton B_n that is an output restriction of A_n and such that $\text{Comm}(B_n, m) \le \text{Comm}(A_n, m)$. By Lemma 1, it follows that if B_n solves $\textsc{ApproxFreq}_p(\epsilon)$, then, there exists a free automaton C_n that solves $\textsc{ApproxFreq}_p(4\epsilon)$. Thus, by Theorem 2, it follows that if B_n solves $\textsc{ApproxFreq}_2(\epsilon)$ for $4\epsilon \le 1$, then,

$$\text{Comm}(A_n, m) \ge \text{Comm}(B_n, m) \ge \text{Comm}(C_n, m) \ge \text{rank}(V_{C_n} \log m$$

and

$$\text{rank}(V_{C_n}) \ge n(1 - 4\epsilon)\log(2m + 1), \text{ by Lemma 2 } .$$

Here V_{C_n} is the vector space $\mathbb{R}^n / M^e(C_n)$, where, $M^e(C_n)$ is the kernel of C_n.

Further, for $p > 2$, $\|f\|_p \le \|f\|_2$, for any $f \in \mathbb{R}^n$. Therefore, $err_p(\hat{f}, f) \le \epsilon$ implies that $err_2(\hat{f}, f) \le \epsilon$. Thus, the space lower bound for err_2 as given by Lemma 2 holds for err_p, for any $p > 2$.

By Lemma 4, it follows that if B_n solves $\textsc{ApproxFreq}_p(\epsilon)$, for $4\epsilon \geq 2n^{1/2-1/p}$, then,

$$\begin{aligned}\mathrm{Comm}(A_n, m) \geq \mathrm{Comm}(B_n, m) \geq \mathrm{Comm}(C_n, m) &\geq \mathrm{rank}(V_{C_n}) \log m \\ &\geq \frac{n^{2-2/p}}{64\epsilon^2} \log m \ .\end{aligned}$$

Finally, we note that for any stream automaton A_n, $\mathrm{Comm}(A_n, m)$ is a lower bound on the effective space usage $\mathrm{Space}(A_n, m)$.

This proves the lower bound assertion of Theorem 1. □

4 Upper Bound

Lemma 5 presents a (nearly) matching upper bound for the $\textsc{ApproxFreq}_p(\epsilon)$ problem, for $1 \leq p < 2$.

Lemma 5. *For any $1 < p < 2$ and $1 > \epsilon > \frac{1}{\sqrt{n}}$, there exists a total stream algorithm for solving* $\textsc{ApproxFreq}_p(\epsilon)$ *using space* $O(\epsilon^{-2} n^{2-2/p} (\log \| f(\sigma)_1 \|)$ $(p/(p-1+p(\log(1/\epsilon)/\log n)))^2)$.

Proof. By a standard identity between norms, for any vector $f \in \mathbb{R}^n$, $\|f\|_1 \leq n^{1-1/p}\|f\|_p$. Therefore,

$$err_1(\hat{f}, f) \leq \frac{\epsilon}{n^{1-1/p}} \text{ implies } err_p(\hat{f}, f) \leq \epsilon \ .$$

So let $\epsilon' = \epsilon / n^{1-1/p}$, and use the CR-precis algorithm with accuracy parameter ϵ'. This requires space

$$O((\epsilon')^{-2} (\log \| f(\sigma)\|_1)(\log^2 n)/(\log^2 (1/\epsilon')) \ .$$

Substituting the value of ϵ', we obtain the statement of the lemma. □

The statement of the lemma is equivalent to the assertion of Theorem 1 for upper bounds. This completes the proof of Theorem 1.

References

1. L. Bhuvanagiri, S. Ganguly, D. Kesh, and C. Saha. "Simpler algorithm for estimating frequency moments of data streams". In *Proceedings of ACM Symposium on Discrete Algorithms (SODA)*, pages 708–713, 2006.
2. P. Bose, E. Kranakis, P. Morin, and Y. Tang. "Bounds for Frequency Estimation of Packet Streams". In *Jop F. Sibeyn (Ed.) Proceedings of the 10th Internaltional Colloquium on Structural Information Complexity, June 18-20, 2003, Ume Sweden. Informatics 17 Carleton Scientific 2003, ISBN 1-894145-16-X*, pages 33–42, 2003.
3. Moses Charikar, Kevin Chen, and Martin Farach-Colton. "Finding frequent items in data streams". In *Widmayer, P. Ruiz, F.T., Bueno, R.M., Hennessy, M., Eidenbenz, S. Conejo, R. (Eds.): ICALP 2002, LNCS, Vol. 2380 Springer 2002*, pages 693–703.

4. Graham Cormode and S. Muthukrishnan. “An Improved Data Stream Summary: The Count-Min Sketch and its Applications”. *J. Algorithms*, 55(1):58–75, April 2005.
5. E. D. Demaine, A. López-Ortiz, and J. I Munro. “Frequency estimation of internet packet streams with limited space”. In *Möhring, R.H., Raman, R. (Eds.): ESA 2002, LNCS Vol. 2461 Springer 2002* , pages 348–360.
6. S. Ganguly. “Lower bounds for frequency estimation over data streams”. In *Hirsch, E.A, Razborov, A. A., Semenov, A.L., Slissenko, A (Eds.): CSR 2008, June 7-12, 2008, LNCS Vol. 5010 Springer 2008,* , pages 204–215,
7. S. Ganguly and A. Majumder. “CR-precis: A Deterministic Summary Structure for Update Streams”. In *Chen, B, Paterson, M., Zhang, G. (Eds.): ESCAPE 2007, LNCS Vol. 4614 Springer 2007*, pages 48–59.
8. Anna Gilbert, Sudipto Guha, Piotr Indyk, Y. Kotidis, S. Muthukrishnan, and Martin Strauss. “Fast Small-space Algorithms for Approximate Histogram Maintenance”. In *Proceedings of ACM STOC*, pages 152–161, 2002.
9. Anna C. Gilbert, Y. Kotidis, S. Muthukrishnan, and Martin J. Strauss. “Surfing Wavelets on Streams: One-pass Summaries for Approximate Aggregate Queries”. In *Proceedings of VLDB*, pages 79–88, Roma, Italy, September 2001.
10. Sudipto Guha, Piotr Indyk, S. Muthukrishnan, and Martin Strauss. “Histogramming Data Streams with Fast Per-Item Processing”. In *Widmayer, P. Ruiz, F.T., Bueno, R.M., Hennessy, M., Eidenbenz, S. Conejo, R. (Eds.): ICALP 2002, LNCS, Vol. 2380 Springer 2002*, pages 681–692, 2002.
11. Piotr Indyk and David Woodruff. “Optimal Approximations of the Frequency Moments”. In *Proceedings of ACM Symposium on Theory of Computing STOC*, pages 202–298, Baltimore, Maryland, USA, June 2005.
12. R.M. Karp, S. Shenker, and C.H. Papadimitriou. “A Simple Algorithm for Finding Frequent Elements in Streams and Bags”. *ACM Trans. Data. Syst.*, 28(1):51–55, 2003.
13. L.K. Lee and H.F. Ting. “A simpler and more efficient deterministic scheme for finding frequent items over sliding windows”. In *Proceedings of ACM International Symposium on Principles of Database Systems (PODS)*, pages 263–272, 2006.
14. G. Manku and R. Motwani. “Approximate Frequency Counts over Data Streams”. In *Proceedings of VLDB*, pages 346–357, August 2002.
15. J. Misra and D. Gries. “Finding repeated elements”. *Sci. Comput. Programm.*, 2:143–152, 1982.

A Proofs

Let M be the kernel of A_n and let M' be defined as follows.

$$M' = \{x \mid \exists a \in \mathbb{Z}, ax \in M\} \tag{8}$$

It follows that M' is torsion-free.

Fact 3 *Let $b_1, b_2, \ldots, b_r$ be a basis of M'. Then, $\exists\ \alpha_1, \ldots, \alpha_r \in \mathbb{Z} - \{0\}$ such that $\alpha_1 b_1, \ldots, \alpha_r b_r$ is a basis for M. Hence, $M^e = (M')^e$.*

Proof (Of Fact 3). It follows from standard algebra that the basis of M is of the form $\alpha_1 b_1, \ldots, \alpha_r b_r$. It remains to be shown that the α_i's are non-zero. Suppose that $\alpha_1 = 0$. For any $a \in \mathbb{Z}$, $a \neq 0$, suppose $ax \in M$ and $x \in M'$. Then, x has a unique representation as $x = \sum_{j=1}^{r} x_j b_j$. Thus, $ax = \sum_{j=1}^{r}(ax_j)b_j \in M$ and has the same representation in the basis $\{\alpha_j b_j\}_{j=1,\ldots,n}$. Therefore, $ax_1 = 0$ or $x_1 = 0$ for all $x \in M'$, which is a contradiction.

Let $\{b_1, b_2, \ldots, b_r\}$ be a basis for M'. Then, by the above paragraph, there exist non-zero elements $\alpha_1, \ldots, \alpha_r$ such that $\{\alpha_1 b_1, \alpha_2 b_2, \ldots, \alpha_r b_r\}$ is a basis for M. Therefore, over reals, $(b_1, \ldots, b_r) = (\alpha_1 b_1, \ldots, \alpha_r b_r)$. Thus, $M^e = (M')^e$. $\square$

Lemma 6. *Let M be a sub-module of $\mathbb{Z}^n$. (1) if there exists h_p such that $err_p(h_p, M) \leq \epsilon$, then, $err_p(0, M) \leq \epsilon$, and, (2) if $err_p(0, M) \leq \epsilon$ then $err_p(0, M^e) \leq \epsilon$.*

Proof. Part (1). For any $y_i \in \mathbb{Z}$,

$$\max(|(h_p)_i - y_i|, |(h_p)_i + y_i|) \geq |y_i|.$$

Therefore,

$$\max(\|h_p - y\|_\infty, \|h_p + y\|_\infty) \geq \|y\|_\infty \ .$$

Let $y \in M$. Since, M is a module, $-y \in M$. Thus,

$$\begin{aligned}
err_p(0, y) &= err_p(0, -y) \\
&= \frac{\|y\|_\infty}{\|y\|_p} \\
&\leq \frac{1}{\|y\|_p} \max(\|h_p - y\|_\infty, \|h_p + y\|_\infty) \\
&= \max(err_p(h_p, y), err_p(h_p, -y)) \\
&\leq \epsilon
\end{aligned}$$

$\square$

Part 2. Let $z \in M^e$. Let $b_1, b_2, \ldots, b_r$ be a basis of the free module M. For $t > 0$, let tz be expressed uniquely as $tz = \alpha_1 b_1 + \ldots + \alpha_r b_r$, where, α_i's belong to $\mathbb{R}$. Consider the vertices of the parallelopiped P_{tz} whose sides are $b_1, b_2, \ldots, b_r$ and that encloses tz.

$$\begin{aligned}
P_{tz} = [\alpha_1]b_1 + [\alpha_2]b_2 + \ldots + [\alpha_n]b_n \\
+ \{\beta_1 b_1 + \beta_2 b_2 + \ldots + \beta_r b_r \mid \beta_j \in \{0,1\}, j = 1, 2, \ldots, r\}
\end{aligned}$$

where, $[\alpha]$ denotes the largest integer smaller than or equal to α. Since, ℓ_∞ is a convex function $\|tz\|_\infty \leq \|y\|_\infty$ for some $y \in P_{tz}$. Let $y = \sum_{j=1}^{r} \beta_j b_j$, for $\beta_j \in \{0,1\}$, $j = 1, 2, \ldots, r$.

$$\|y - tz\|_1 = \|\sum_{j=1}^{r}(\beta_j - [\alpha_j])b_j\|_1 \leq \sum_{j=1}^{r}\|(\beta_j - [\alpha_j])b_j\|_1 \leq \sum_{j=1}^{r}\|b_j\|_1$$

$$\text{or,} \quad \|tz\|_1 \geq \|y\|_1 - \sum_{j=1}^{r}\|b_j\|_1$$

Therefore,

$$
\begin{aligned}
err_p(0, tz) = \frac{\|tz\|_\infty}{\|tz\|_1} &\le \frac{\|y\|_\infty}{\|y\|_1 - \sum_{j=1}^{r}\|b_j\|_1} \\
&\le \left(\frac{\|y\|_1}{\|y\|_\infty} - \frac{\sum_{j=1}^{r}\|b_j\|_1}{\|y\|_\infty}\right)^{-1} \le \left(\frac{1}{\epsilon} - \frac{\sum_{j=1}^{r}\|b_j\|_1}{\|y\|_\infty}\right)^{-1}
\end{aligned}
$$

where, the last step follows from the assumption that $y \in M$ and therefore, $err_p(0, y) = \frac{\|y\|_\infty}{\|y\|_1} \le \epsilon$. The ratio $\frac{\sum_{j=1}^{r}\|b_j\|_1}{\|y\|_\infty}$ can be made arbitrarily small by choosing t to be arbitrarily large. Thus, $\lim_{t\to\infty} err_p(0, tz) \le \epsilon$. Since, $err_p(0, tz) = \frac{\|tz\|_\infty}{\|tz\|_1} = \frac{\|z\|_\infty}{\|z\|_1} = err_p(0, z)$, for all t, we have, $err_p(0, z) \le \epsilon$. □

Proof (Of Lemma 1.). By construction, M' is the smallest module that contains M as a sub-module and M' is free. This also implies that $\mathbb{Z}^n/M'$ is free. For $x \in \mathbb{Z}^n$, define

$$
h_p(x + M') = \min_{\ell_p}(x + M') \ .
$$

That is, $h_p(x + M')$ is the element with the smallest ℓ_p norm among all vectors in $x + M'$.

Let $y \in x + M'$. Then, $y \in x_p + M$ for some x_p. Let $\hat{y} = \text{output}_A(x_p + M)$ denote the output of A for an input stream with frequency in $x_p + M$ (they all return the same value, since, A is path independent and has kernel M) and let $y'_p = \min_{\ell_p}(x_p + M)$. Let h_p denote $h_p(x + M')$ and let $\hat{h} = \text{output}_A(h_p + M)$. Therefore,

$$
\begin{aligned}
err(h_p, y) &= \frac{\|y - h_p\|_\infty}{\|y\|_p} \\
&\le \frac{\|y - \hat{y}\|_\infty}{\|y\|_p} + \frac{\|\hat{y} - y'_p\|_\infty}{\|y\|_p} + \frac{\|y'_p - h_p\|_\infty}{\|y\|_p} \qquad (9)
\end{aligned}
$$

The first and the second terms above are bounded by ϵ as follows. The first term $\frac{\|y-\hat{y}\|_\infty}{\|y\|_p} = err_p(\hat{y}, y) \le \epsilon$, since, $y \in x_p + M$ and $\hat{y}$ is the estimate returned by A_n for this coset. The second term

$$
\frac{\|\hat{y} - y'_p\|_\infty}{\|y\|_p} \le \frac{\|\hat{y} - y'_p\|_\infty}{\|y'_p\|_p} = err(\hat{y}, y'_p) \le \epsilon
$$

since, $\|y'_p\|_p \le \|y\|_p$ and y'_p lies in the coset $x_p + M$. The third term in (9) can be rewritten as follows. Since, M' is a free module, $y'_p - h_p \in M'$ and $M' \subset M^e$.

Therefore,

$$
\begin{aligned}
&\frac{\|y'_p - h_p\|_\infty}{\|y\|_p} \\
&\leq \frac{\|y'_p - h_p\|_\infty}{\|y'_p - h\|_p} \cdot \frac{\|y'_p - h_p\|_p}{\|y'_p\|_p}, \qquad \text{since, } \|y'_p\|_p \leq \|y\|_p \\
&\leq \epsilon \cdot \frac{\|y'_p\|_p + \|h_p\|_p}{\|y'_p\|_p} \quad \text{by Lemma 6 and by triangle inequality} \\
&\leq 2\epsilon, \qquad \text{since, } \|h_p\|_p \leq \|y'_p\|_p
\end{aligned}
$$

By (9), $err(h, y) \leq \epsilon + \epsilon + 2\epsilon = 4\epsilon$. The automaton B_n with kernel M' is constructed as in Theorem 2. □