A Note on Space Lower Bound for Samplers

Jelani Nelson, Jakub Pachocki, Zhengyu Wang

January 26, 2017

We study the space lower bound for maintaining a sampler over a turnstile stream. An ℓ_p-sampler with failure probability at most δ is a randomized data structure for maintaining vector $x \in \mathbb{R}^n$ (initially 0) under a stream of updates in the form of (i, Δ) (meaning that $x_i \leftarrow x_i + \Delta$); in the end, with probability at least $1 - \delta$, it gives an "ℓ_p-sample" according to x: namely, item i is sampled with probability $\frac{|x_i|^p}{\sum_{j\in[n]}|x_j|^p}$.

Note that updates are independent of the randomness used in the sampler. That is, for the purpose of proving a lower bound, we assume an oblivious adversary.

To the best of my knowledge, the best space upper bound for ℓ_0 sampler is $O(\log^2 n \log \frac{1}{\delta})$ bits, while the previous best lower bound is $\Omega(\log^2 n + \log \frac{1}{\delta})$ bits (where $\Omega(\log^2 n)$ is shown in [JST11]). The bound is tight for constant δ, while for example, when $\delta = \frac{1}{n}$, the gap is $\log n$.

We assume that

$$2^{-n^{c_1}} < \delta < (\log n)^{-c_2}, \tag{1}$$

where $c_1 = 0.01$ and $c_2 = 2$. For other range of δ we will study later.

In this note, we show space lower bounds for maintaining a sampler for a *binary* vector. That is, at any time, we are guaranteed that $x \in \{0, 1\}^n$. This makes our result strong in the sense that (1) the lower bound applies for any p; (2) the lower bound also works for strict turnstile streams.

In the following sections, we give sequentially improved lower bounds. First, we give a lower bound of $\Omega(\log n \log \frac{1}{\delta})$ bits. Then, we improve it to $\Omega(\frac{\log^2 n \log \frac{1}{\delta}}{(\log\log n + \log\log \frac{1}{\delta})^2})$ bits. The lower bounds are based on communication complexity in the public random coin model. Alice wants to send Bob a uniform random set $A \subseteq [n]$ of size m (Bob knows m, but the random source generating A is independent of the random source accessible to Bob). The one-way communication problem is: Alice sends some message to Bob, and Bob is required to recover A completely. Since the randomness in A contains $\log\binom{n}{m}$ bits of information, any randomized protocol that works with probability 1 requires at least $\log\binom{n}{m}$ expected bits.

Now Alice considers to attach (the memory of) a sampler SAMP in the message. The sampler uses public random coins as its random source, so that the sampler will behave the same at Alice's and Bob's as long as the updates are all the same. Alice will insert all the items in A into SAMP and send SAMP to Bob. In addition, Alice will send a subset $B \subseteq A$ to Bob, so that together with B and SAMP, Bob is able to recover A with good probability based on some protocol they have agreed on.

Now we turn the previous protocol into a new one without any failure. Let SUCC denote the event (or a subset of the event) that Bob successfully recovers A (note that Alice can simulate Bob, so she knows exactly when SUCC happens). If SUCC happens Alice will send Bob a message starting with a 1, followed by (the memory of) SAMP, then followed by the native encoding (explained later) of B; otherwise, Alice will send a message starting with a 0, followed by the native encoding of A. We say the native encoding of a set $S \subseteq [n]$ to be an integer (expressed in binary) in $[\binom{n}{|S|}]$ together with $|S|$ (taking $\log n$ bits). We drop the size of the set if it is known by the receiver.

Lemma 1. *Let s denote the space (in bits) used by a sampler with failure probability at most δ. Let s' denote the expected number of bits to represent B conditioned on SUCC (if we need to send some extra auxiliary information, we will also count it into s'). We have*

$$(1 + \mathsf{s} + \mathsf{s}') \cdot \mathbb{P}(\mathsf{SUCC}) + (1 + \log\tbinom{n}{m}) \cdot (1 - \mathbb{P}(\mathsf{SUCC})) \geq \log\tbinom{n}{m}.$$

If $\mathbb{P}(\mathsf{SUCC}) \geq 1/2$*, we have*

$$\mathsf{s} \geq \log\tbinom{n}{m} - \mathsf{s}' - 2. \tag{2}$$

Remark 1. *Because the space lower bound in this note is proven via communication complexity under public random coin model, it also applies to non-uniform models of computation such as circuits and branching programs.*

Remark 2. *The space lower bound in this note still applies if the sampler is required to output an arbitrary item whose coordinate is non-zero instead of a uniformly random one.*

1 $\Omega(\log n \log \frac{1}{\delta})$ Bits Lower Bound

Let $m = \frac{1}{2}\log\frac{1}{\delta}$, namely, Alice wants to send a uniform random set $A \subseteq [n]$ of size $\frac{1}{2}\log\frac{1}{\delta}$ to Bob. Let $A = \{a_1, \ldots, a_m\}$ and $a_1 < \ldots < a_m$.

Algorithm 1 Alice's Encoder.

```
procedure ENC_1(A)
    SAMP ← ∅
    for i = 1, 2, ..., m do
        Insert a_i into SAMP
    end for
    return SAMP
end procedure
```

Algorithm 2 Bob's Decoder.

```
procedure DEC_1(SAMP)
    S ← ∅
    for i = 1, 2, ..., m do
        Let SAMP_i be a copy of SAMP          ▷ So that SAMP_i can behave as if it is SAMP
        for s ∈ S do                          ▷ Enumerate the elements in S from smallest to biggest
            Remove s from SAMP_i
        end for
        Obtain a sample s_i from SAMP_i
        S ← S ∪ {s_i}
    end for
    return S
end procedure
```

Lemma 2. *For any* $A \subseteq [n]$*, where* $|A| = m = \frac{1}{2}\log\frac{1}{\delta}$*,* $\mathbb{P}(\mathsf{DEC}_1(\mathsf{ENC}_1(A)) = A) \geq 1/2$*.*

Proof. Let E_S denote the event that after removing all the items in S (in the order from smallest to biggest) from SAMP, it gives a valid sample when queried. We have

$$\mathbb{P}(\mathsf{DEC}_1(\mathsf{ENC}_1(A)) = A) \geq \mathbb{P}(\bigcap_{S \subseteq A, S \neq \emptyset} E_S) \geq 1 - \sum_{S \subseteq A, S \neq \emptyset} \mathbb{P}(\overline{E_S}) \geq 1 - \delta \cdot 2^{\frac{1}{2}\log\frac{1}{\delta}} \geq 1/2.$$

□

Lemma 3. *$\mathsf{s} = \Omega(\log n \log \frac{1}{\delta})$ for $2^{-n^{0.99}} < \delta < \frac{1}{4}$.*

Proof. It follows from Formula 2 in Lemma 1, where $\log \binom{n}{\frac{1}{2}\log\frac{1}{\delta}} = \Omega(\log n \log \frac{1}{\delta})$ and $\mathsf{s}' = 0$. □

Remark 3. *The following decoder DEC_1' is similar to DEC_1, but we will lose a factor of $\log\log\frac{1}{\delta}$ in the lower bound because by doing so we have to union-bound $O(m!)$ events instead of $O(2^m)$ events (so that in turn we have to set m to be $\frac{\log\frac{1}{\delta}}{\log\log\frac{1}{\delta}}$ in order to have good success probability).*

Algorithm 3 A Worse Decoder.

procedure $\mathrm{DEC}_1'(\mathsf{SAMP})$
  $S \leftarrow \emptyset$
  for $i = 1, 2, \ldots, m$ **do**
    Obtain a sample s_i from SAMP
    Remove s_i from SAMP
    $S \leftarrow S \cup \{s_i\}$
  end for
  return S
end procedure

2 $\Omega(\log^2 n \log \frac{1}{\delta})$ Bits Lower Bound

Algorithm 4 Variables Shared by Alice's ENC and Bob's DEC.

$m \leftarrow n^{0.99}$
$K \leftarrow \frac{1}{10}\log\frac{1}{\delta}$
$R \leftarrow \frac{1}{20}\log n \log\frac{1}{\delta}$
for $r = 0, \ldots, R$ **do**
  $n_r \leftarrow m \cdot 2^{-\frac{r}{K}}$
end for
for $a \in [n]$ **do**
  Let U_a be uniformly and independent sample from $[0, 1]$
end for

Algorithm 5 Alice's Encoder.

procedure ENC(A)
  $\mathsf{SAMP} \leftarrow \emptyset$
  Insert items in A into SAMP
  $A_0 \leftarrow A$
  $S \leftarrow \emptyset$
  for $r = 1, \ldots, R$ **do**
    Let SAMP_r be a copy of SAMP
    Remove items in $A \backslash A_{r-1}$ from SAMP_r ▷ So that SAMP_r contains the elements in A_{r-1}
    Let s_r be a sample from SAMP_r
    $A_r \leftarrow A_{r-1}$
    if $s_r \in A_r$ **then** ▷ i.e. if s_r is a valid sample
      $b_r \leftarrow 1$ ▷ b is a binary string of length R, indicating if sampler succeeds on round r
      $S \leftarrow S \cup \{s_r\}$
      $A_r \leftarrow A_r \backslash \{s_r\}$
    else
      $b_r \leftarrow 0$
    end if
    Remove $|A_r| - n_r$ elements from A_r with smallest U_a's among $a \in A_r$ ▷ So that $|A_r| = n_r$
  end for
  return $(A \backslash S, b, \mathsf{SAMP})$
end procedure

Algorithm 6 Bob's Decoder.

procedure DEC(B, b, SAMP)
  $S \leftarrow \emptyset$
  $C_0 \leftarrow \emptyset$
  for $r = 1, \ldots, R$ **do**
    $C_r \leftarrow C_{r-1}$
    if $b_r = 1$ **then**
      Let SAMP_r be a copy of SAMP
      Remove items in C_{r-1} from SAMP_r ▷ Invariant: $C_r = A \backslash A_r$ (A_r is defined in ENC)
      Let s_r be a sample from SAMP_r
      $S \leftarrow S \cup \{s_r\}$
      $C_r \leftarrow C_r \cup \{s_r\}$
    end if
    Insert $m - n_r - |C_r|$ elements into C_r with smallest U_a's among $a \in B \backslash C_r$
  end for
  return $B \cup S$
end procedure

2.1 Analysis

Lemma 4. *Let function f: $\{0,1\}^n \times \{0,1\}^m \to \{0,1\}$. Let X be a uniformly random string in $\{0,1\}^n$. If for any $y \in \{0,1\}^m$ we have $\mathbb{P}(f(X,y) = 1) \leq \delta$ where $0 < \delta < 1$, then for any random variable Y in $\{0,1\}^m$, we have*

$$\mathbb{P}(f(X,Y) = 1) \leq \frac{I(X;Y) + 1}{\log \frac{1}{\delta}},$$

where $I(X;Y)$ is the mutual information (in bits) between X and Y.

Proof. It is equivalent to prove $I(X;Y) \geq \mathbb{E}(f(X,Y)) \cdot \log \frac{1}{\delta} - 1$. By definition of mutual entropy, $I(X;Y) = H(X) - H(X|Y)$ where $H(X) = n$ and $H(X|Y) \leq 1 + (1 - \mathbb{E}(f(X,Y))) \cdot n + \mathbb{E}(f(X,Y)) \cdot (n - \log \frac{1}{\delta}) = n + 1 - \mathbb{E}(f(X,Y)) \cdot \log \frac{1}{\delta}$. The upper bound for $H(X|Y)$ is obtained by considering the following one-way communication problem: Alice obtains both X and Y while Bob only gets Y, what is the (minimum) expected number of bits that Alice sends to Bob so that Bob can recover X? Any protocol gives an upper bound for $H(X|Y)$, and we simply take the following protocol: first Alice sends Bob $f(X,Y)$ (taking 1 bit); and then if $f(X,Y) = 0$ Alice sends X directly (taking n bits), otherwise, $f(X,Y) = 1$, Alice sends the index of X in $\{x | f(x,Y) = 1\}$ (taking $\log(\delta 2^n) = n - \log \frac{1}{\delta}$ bits). □

We do the analysis conditioned on A. Let X denote the random source used by the sampler. By Lemma 4, the probability that the sampler fails on round r is upper bounded by $\frac{I(X;A_r)+1}{\log \frac{1}{\delta}}$. Namely $\mathbb{E}(b_r) \geq 1 - \frac{I(X;A_r)+1}{\log \frac{1}{\delta}}$.

$I(X;A_r) = H(A_r) - H(A_r|X)$. Since $|A_r| = n_r$ and $A_r \subseteq A$ where $|A| = m$, $H(A_r) \leq \log \binom{m}{n_r}$. Now we want to lower bound $H(A_r|X)$. By definition of conditional entropy, $H(A_r|X) = \sum_x p_x \cdot H(A_r|X = x)$. We fix an arbitrary x. If we can prove that for any $T \subseteq A$ where $|T| = n_r$, $\mathbb{P}(A_r = T|X = x) \leq p$, then by definition of entropy we have $H(A_r|X = x) \geq \log \frac{1}{p}$. In fact, for any fixed T, we have

$$H(A_r = T|X = x) \leq \prod_{i=1}^{r} \frac{\binom{n_{i-1}-n_r-1}{n_{i-1}-n_i-1}}{\binom{n_{i-1}-1}{n_{i-1}-n_i-1}},$$

because on round i ($1 \leq i \leq r$), Alice removes $n_{i-1} - n_i$ elements from A_{i-1} to get A_i. Conditioned on the event that $A_{i-1} \supseteq T$, the probability that $A_i \supseteq T$ is at most $\binom{n_{i-1}-n_r-1}{n_{i-1}-n_i-1} / \binom{n_{i-1}-1}{n_{i-1}-n_i-1}$, where the equation achieves when $s_i \in A_{i-1} \backslash T$, and we take a uniformly random subset of $A_{i-1} \backslash \{s_i\}$ of size $n_{i-1} - n_i - 1$, so that the subset does not intersect with T.

For notation simplicity, let $n^{\underline{k}}$ denote $n \cdot (n-1) \dots (n-k+1)$. We have

$$\prod_{i=1}^{r} \frac{\binom{n_{i-1}-n_r-1}{n_{i-1}-n_i-1}}{\binom{n_{i-1}-1}{n_{i-1}-n_i-1}} = \prod_{i=1}^{r} \frac{(n_{i-1}-n_r-1)! n_i!}{(n_{i-1}-1)!(n_i-n_r)!} = \prod_{i=1}^{r} \frac{n_i^{\underline{n_r}}}{(n_{i-1}-1)^{\underline{n_r}}} = \prod_{i=1}^{r} \left(\frac{n_i^{\underline{n_r}}}{n_{i-1}^{\underline{n_r}}} \cdot \frac{n_{i-1}}{n_{i-1}-n_r} \right)$$

By telescoping

$$\prod_{i=1}^{r} \frac{n_i^{\underline{n_r}}}{n_{i-1}^{\underline{n_r}}} = \frac{n_r^{\underline{n_r}}}{n_0^{\underline{n_r}}} = \frac{n_r!(n_0-n_r)!}{n_0!} = \frac{1}{\binom{n_0}{n_r}}$$

And

$$\prod_{i=1}^{r} \frac{n_{i-1}}{n_{i-1}-n_r} = \prod_{i=1}^{r} \frac{1}{1-2^{(i-1-r)/K}} = \prod_{j=1}^{r} \frac{1}{1-2^{-j/K}} \leq \prod_{j=1}^{\infty} \frac{1}{1-2^{-j/K}}$$

Lemma 5. *Let $K \in \mathbb{N}$ and $K \geq 1$. We have $\prod_{j=1}^{\infty} \frac{1}{1-2^{-j/K}} \leq 2^{5K}$.*

Proof. First, we bound the product of first $2K$ terms. Note that $\frac{1}{1-2^{-x}} \leq \frac{8}{3x}$ for $0 < x \leq 2$. Therefore, $\prod_{j=1}^{2K} \frac{1}{1-2^{-j/K}} \leq (8/3)^{2K} \cdot \frac{K^{2K}}{(2K)!} \leq (8/3)^{2K} \cdot \frac{K^{2K}}{(2K/e)^{2K}} = (4e/3)^{2K} < 2^{4K}$.

On the other hand, the product of the rest terms $\prod_{j=2K+1}^{\infty} \frac{1}{1-2^{-j/K}} \leq \prod_{j=2K+1}^{\infty} \frac{1}{1-2^{-\lfloor j/K \rfloor}} \leq \prod_{i=2}^{\infty} \left(\frac{1}{1-2^{-i}} \right)^K \leq \left(\frac{1}{1-\sum_{i=2}^{\infty} 2^{-i}} \right)^K = 2^K$.

Multiplying two parts proves the lemma. □

Combine together we get $\mathbb{E}(b_r) \geq 1 - \frac{5K+1}{\log \frac{1}{\delta}} \geq \frac{2}{5}$.

Lemma 6. *Let $m = n^{0.99}$. Let $X \in \mathbb{N}$ be a random variable, and $X \leq m$. Moreover, $\mathbb{E}(X) \leq m - d$. We have $\mathbb{E}(\log \binom{n}{m} - \log \binom{n}{X}) = \Omega(d \log n)$.*

Proof.

$$\begin{aligned}
\log \binom{n}{m} - \log \binom{n}{X} &= \log \frac{n!/(m!(n-m)!)}{n!/(X!(n-X)!)} \\
&= \sum_{i=1}^{m-X} \log \frac{n-X-i+1}{m-i+1} \\
&\geq (m-X) \cdot \log \frac{n-X}{m} \\
&\geq (m-X) \cdot \log n^{1/200}
\end{aligned}$$

Taking expectation on both sides, we get $\mathbb{E}(\log \binom{n}{m} - \log \binom{n}{X}) \geq \frac{d}{200} \log n$. □

References

[JST11] Hossein Jowhari, Mert Sağlam, and Gábor Tardos. Tight bounds for lp samplers, finding duplicates in streams, and related problems. In *Proceedings of the thirtieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 49–58. ACM, 2011.