A Note on Space Lower Bound for Samplers

January 3, 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 $c_1 = 0.9$ and $c_2 = 1.1$. 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 $\Omega(\log n \log \frac{1}{\delta} \log \frac{\log n}{\log \log \frac{1}{\delta}})$ bits, and finally $\Omega(\frac{\log^2 n \log \frac{1}{\delta}}{\log \log n + \log \log \frac{1}{\delta}})$ 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 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 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.*

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* $2^{-n^{0.9}} < \delta < \frac{1}{4}$*.*

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 2. *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 DEC'_1(SAMP)
    S ← ∅
    for i = 1, 2, ..., m do
        Obtain a sample s_i from SAMP
        Remove s_i from SAMP
        S ← S ∪ {s_i}
    end for
    return S
end procedure
```

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

In the previous section we have shown how to extract $\Theta(\log\frac{1}{\delta})$ words of information from a sampler. Our goal is to extract more words. New observation comes from the upper bound for constructing an ℓ_0 sampler. Let $\delta = \frac{1}{n}$, we have the following sampler algorithm that consumes $O(\log^3 n)$ bits. The sampler consists of $\log n$ layers, and on layer $i \in [\log n]$ it maintains a separate $\log n$-sparse recover system for the sub-stream generated by sub-sampling the items from the universe $[n]$ with probability 2^{-i}. Each sparse recovery system takes $O(\log^2 n)$ bits. Its correctness comes from the fact that

1. With probability at least $1 - n^{-c}$ there is some layer i that contains at least 1 and at most $\log n$ items whose coordinates are non-zero.

2. Conditioned on the event that on layer i the number of items whose coordinates are non-zero is between 1 and $\log n$, the sparse recovery system on layer i works with failure probability at most n^{-c}. In this context, we say the sparse recovery system works if it could recover at least one item.

Intuitively, the previous lower bound only extracts information from one single layer (i.e. layer 0). In this section, we build a framework to extract information from $\Theta(\log n)$ layers.

Alice wants to send random set A to Bob where $|A| = m = n^{1/4}$. Similar to ENC_1, Alice constructs SAMP by inserting all the items in A, and sends it to Bob. Moreover, Alice will send Bob a subset $B \subseteq A$ computed as follows. Initially $B = A$. Alice proceeds in $R = \log m - \log\log\frac{1}{\delta}$ rounds. Let $A_r = \{a \in A | a < \frac{n}{2^{r-1}}\}$. Furthermore, let $A_r^- = A_{r+1}$ and $A_r^+ = A_r \backslash A_r^-$. On round r (for $r = 1, \ldots, R$):

1. Alice makes a copy of SAMP, denoted by SAMP_r, and removes all items in $A\backslash A_r$ from SAMP_r.

2. Similar to ENC_1', obtain $n_r = \frac{1}{2} \cdot \frac{\log\frac{1}{\delta}}{(\log m)+2-r}$ samples from SAMP_r, denoted by S_r.

3. Remove items in $S_r \cap A_r^+$ from B.

Algorithm 4 Alice's Encoder.

procedure $\mathrm{ENC}_2(A)$
  $\mathsf{SAMP} \leftarrow \emptyset$
  Insert all elements in A into SAMP
  $B \leftarrow A$
  for $r = 1, \ldots, R$ **do**
    Let SAMP_r be a copy of SAMP
    Remove all elements in $A \backslash A_r$ from SAMP_r
    $S_r \leftarrow \emptyset$
    for $i = 1, \ldots, n_r$ **do**
      Obtain a sample s from SAMP_r
      Remove s from SAMP_r
      $S_r \leftarrow S_r \cup \{s\}$
    end for
    $B \leftarrow B \backslash (A_r^+ \cap S_r)$
  end for
  return (SAMP, B)
end procedure

Algorithm 5 Bob's Decoder.

procedure $\mathrm{DEC}_2(\mathsf{SAMP}, B)$
  $A \leftarrow B$
  for $r = 1, 2, \ldots, R$ **do**
    Let SAMP_r be a copy of SAMP
    Remove all the items in $\{a \in A | a \geq \frac{n}{2^{r-1}}\}$ from SAMP_r
    $S_r \leftarrow \emptyset$
    for $i = 1, \ldots, n_r$ **do**
      Obtain a sample s from SAMP_r
      Remove s from SAMP_r
      $S_r \leftarrow S_r \cup \{s\}$
    end for
    $A \leftarrow A \cup \{a \in S_r | a \geq \frac{n}{2^r}\}$
  end for
  return A
end procedure

3 Lower Bound for a Promised Problem

Peeling off by chunks (instead of one by one).

4 A Fake $\Omega(\frac{\log^2 n \log \frac{1}{\delta}}{\log\log n + \log\log \frac{1}{\delta}})$ Bits Lower Bound. Oh no!!!

Let $m = n^{0.99}$, $K = \log n \cdot \log \frac{1}{\delta}$ and $R = \frac{1}{50} \log n$. Let $A_r = \{a \in A | a < \frac{n}{2^{r-1}}\}$. Let $A_r^- = A_{r+1}$ and $A_r^+ = A_r \backslash A_r^-$. We partition each A_r into K chunks, where chunk k ($k = 1, \ldots, K$) $C_{r,k} = \{a \in A | \frac{k-1}{K} \leq \frac{a}{n/2^{r-1}} < \frac{k}{K}\}$. Assume that K is dividable by 2 so that A_r^- contains the first $\frac{K}{2}$ chunks and A_r^+ contains the last $\frac{K}{2}$ chunks. Let $n_r = \frac{1}{20} \cdot \frac{\log \frac{1}{\delta}}{\log K}$. All these parameters are shared by Alice's ENC_4 and Bob's DEC_4.

Algorithm 6 Alice's Encoder.

procedure $\mathrm{ENC}_4(A)$
  $\mathsf{SAMP} \leftarrow \emptyset$
  Insert all elements in A into SAMP
  $B \leftarrow A$
  $C \leftarrow \emptyset$
  for $r = 1, \ldots, R$ **do**
    Let SAMP_r be a copy of SAMP
    Remove all elements in $A \backslash A_r$ from SAMP_r
    $S_r \leftarrow \emptyset$
    for $i = 1, \ldots, n_r$ **do**
      Obtain a sample s from SAMP_r
      Find k such that $s \in C_{r,k}$
      Remove all elements in $C_{r,k}$ from SAMP_r
      $S_r \leftarrow S_r \cup \{s\}$
      if $k \leq \frac{K}{2}$ **then**
        $C \leftarrow C \cup C_{r,k}$
      end if
    end for
    $B \leftarrow B \backslash (A_r^+ \cap S_r)$
  end for
  $B \leftarrow B \cup C$
  return (SAMP, B)
end procedure

Algorithm 7 Bob's Decoder.

procedure $\mathrm{DEC}_4(\mathsf{SAMP}, B)$
  $A \leftarrow B$
  for $r = 1, 2, \ldots, R$ **do**
    Let SAMP_r be a copy of SAMP
    Remove all the items in $\{a \in A | a \geq \frac{n}{2^{r-1}}\}$ from SAMP_r
    for $i = 1, \ldots, n_r$ **do**
      Obtain a sample s from SAMP_r
      $A \leftarrow A \cup \{s\}$
      Find k such that $\frac{k-1}{K} \leq \frac{s}{n/2^{r-1}} < \frac{k}{K}$
      Remove items in $\{a \in A | \frac{k-1}{K} \leq \frac{a}{n/2^{r-1}} < \frac{k}{K}\}$ from SAMP_r
    end for
  end for
  return A
end procedure

4.1 Analysis

Note that $\mathbb{E}(|C_{r,k}|) = m \cdot 2^{1-r} \cdot \frac{1}{K}$. By the setting of parameters we have $m = n^{0.99}$, $2^{1-r} \geq 2^{-R} = n^{-0.02}$ and $\frac{1}{K} = \frac{1}{\log n \log \frac{1}{\delta}} \geq n^{-0.91}$ (note that the range of δ is specified by Formula 1). Therefore $\mathbb{E}(|C_{r,k}|) \geq n^{0.07}$, for $r = 1, \ldots, R$ and $k = 1, \ldots, K$. Let $\varepsilon = 0.01$. For any pair of (r, k), because of the randomness in A, we can prove by Chernoff bound (and negative correlation) that the probability that $||C_{r,k}| - \mathbb{E}(|C_{r,k}|)| > \varepsilon \, \mathbb{E}(|C_{r,k}|)$ is exponentially small. By union bound, with high probability, for all $C_{r,k}$, its size deviates its expectation by a factor of at most ε. In the following, we will discuss conditioned on that.

Lemma 4. *With probability at least $\frac{9}{10}$, all the queries to the samplers in ENC_4 are answered successfully.*

Proof. By union bound, the failure probability is at most $\sum_{i=1}^{R} K^{n_i} \cdot \delta = \frac{\delta^{19/20} \log n}{50} < \frac{1}{10}$. □

Lemma 5. *Conditioned on the samplers answer all the queries successfully, we have $\mathsf{DEC}_4(\mathsf{ENC}_4(A)) = A$.*

Lemma 6. *Conditioned on the samplers answer all the queries successfully, we have $\mathbb{E}(|A| - |B|) = \Omega(\frac{\log n \log \frac{1}{\delta}}{\log\log n + \log\log \frac{1}{\delta}})$, where B is the set ENC_4 outputs.*

Proof. Let $Y_{r,k}$ ($r = 1, \ldots, R$ and $k = \frac{K}{2} + 1, \ldots, K$) denote the indicator random variable that on round r the sampler returns a sample s so that $s \in C_{r,k}$ and on that round $s \notin C$ (or equivalently, $s \in A \backslash B$).

The probability that on round r the sampler return a sample s so that $s \in C_{r,k}$ is at least $\frac{1-\varepsilon}{1+\varepsilon} \cdot \frac{n_r}{K}$. The probability that $s \in C$ is at most $\sum_{i=1}^{r-1} \frac{1+\varepsilon}{1-\varepsilon} \cdot \frac{n_i}{K}$. **This two events are independent**, so

$$\mathbb{E}(Y_{r,k}) \geq \frac{1-\varepsilon}{1+\varepsilon} \cdot \frac{n_r}{K} \cdot (1 - \sum_{i=1}^{r-1} \frac{1+\varepsilon}{1-\varepsilon} \cdot \frac{n_i}{K}) \geq \frac{n_r}{2K} = \frac{\log \frac{1}{\delta}}{40K \log K} = \frac{\log \frac{1}{\delta}}{40K(\log\log n + \log\log \frac{1}{\delta})}.$$

We have $|A| - |B| = \sum_{r=1}^{R} \sum_{k=\frac{K}{2}+1}^{K} Y_{r,k}$. Taking the expectation on both sides we get the desired bound. □

Lemma 7. *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 $\mathbb{E}(\log \binom{n}{m} - \log \binom{n}{X}) \geq \frac{d}{200} \log n$. □

Theorem 1. *$\mathsf{s} = \Omega(\frac{\log^2 n \log \frac{1}{\delta}}{\log\log n + \log\log \frac{1}{\delta}})$ for $2^{-n^{0.9}} < \delta < (\log n)^{-1.1}$.*

Proof. Let SUCC be the conjunction of the following events:

1. For all $C_{r,k}$ ($r = 1, \ldots, R, k = 1, \ldots, K$), $||C_{r,k}| - \mathbb{E}(|C_{r,k}|)| \leq \varepsilon \, \mathbb{E}(|C_{r,k}|)$.
2. The samplers in ENC_4 answer all the queries successfully on input A.

By Lemma 4, we have $\mathbb{P}(\mathsf{SUCC}) \geq \frac{1}{2}$. By Lemma 1, we have $\mathsf{s} \geq \log\binom{n}{m} - s' - 2$. By definition, $s' = \log n + \mathbb{E}(\log\binom{n}{|B|} | \mathsf{SUCC})$. By Lemma 6 and Lemma 7, we get $\mathbb{E}(\log\binom{n}{|B|} | \mathsf{SUCC}) = \log\binom{n}{m} - \Omega(\frac{\log^2 n \log \frac{1}{\delta}}{\log\log n + \log\log \frac{1}{\delta}})$. □

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.