CS 7880: Algorithms for Big Data (Fall'22) **Northeastern University**

Lecture 13

October 25, 2022

Instructor: Soheil Behnezhad *Scribe: LaKyah Tyner*

Disclaimer: *These notes have not been edited by the instructor.*

1 Overview

In the last lecture, we proved the lower bound for a deterministic streaming algorithm solving the distinct elements problem. In this lecture, we will discuss the lower bound for randomized algorithms. In order to do this, we first discuss the Randomized Communication Complexity problem and demonstrate how to use communication lower bounds to define our streaming lower bound.

2 Randomized Communication Complexity

Recall the deterministic communication problem:

Problem 1. Two distributed parties, Alice and Bob, each receive an n-bit string $x \in \{0,1\}^n$ and $y \in \{0,1\}^n$ respectively. The goal is to compute a function $f(x,y) \to \{0,1\}$ with the least amount of communication between them.

We now consider the *randomized* communication problem in two models:

- Private Coin Model:
  Alice and Bob each have a private tape of random bits which they use for determining what to communicate.
- Public Coin Model:
  Alice and Bob have a shared tape of randomness between them. Note that this does not give us much additional power over the private coin model. The benefit of us this model is that it allows us to view the protocol as a distribution over deterministic protocols.

Definition 1. The *communication cost* of a protocol Π, denoted by $\|\Pi\|$, is the total number of communicated bits for worst-case inputs x, y and worst-case random tape s regardless of public or private randomness.

Note: For randomized protocols we want that the protocol solves the problem with probability $\geq 2/3$

Theorem 2. *Let Π be a protocol solving a function $f : \{0,1\}^n \times \{0,1\}^n$ with probability $1-\epsilon$ for any ϵ using public randomness. Then there is a private coin protocol σ solving f with probability $1-\epsilon$ such that $\|\sigma\| \leq \|\Pi\| + O(\log(n) + \log(1/\delta))$. In other words the communication of Π only differs from σ by $O(\log(n) + \log(1/\delta))$ bits.*

Proof. Recall by the Chernoff bound that for independent variables $Z_1, ..., Z_t \in \{0,1\}$ and any $0 < \alpha \leq 1$. $\bar{Z} = \frac{1}{t}\Sigma_i Z_i$. We have,

$$Pr[|\bar{Z} - \mathbb{E}[\bar{Z}]| \geq \alpha] \leq exp(-\frac{\alpha^2 - t^2}{3t}) = \frac{\alpha^2 t}{3}$$

Let $\Pi(x,y,r)$ be the output of protocol Π on inputs x, y and public random tape r. Take random tapes $r_1, ..., r_t$ for $t = \lceil \frac{12n}{\delta^2} \rceil$. Let $Z(x,y,r_i) := \mathbb{1}(\Pi(x,y,r_i) \neq f(x,y))$ then,

$$Pr[\sum_{i=1}^{t} \frac{1}{t} Z(x,y,r_i) \leq \epsilon + \delta] \leq exp(-\frac{\delta^2 \cdot 12n}{3\delta^2}) = exp(-4n)$$

We get this using the fact that $\mathbb{E}[\sum_{i=1}^{t} \frac{1}{t} Z(x,y,r_i)] \leq \epsilon$ since each $Z(x,y,r_i)$ is at most ϵ. Then by Union Bound,

$$Pr[\exists (x,y) s.t \sum_{i=1}^{t} \frac{1}{t} Z(x,y,r_i)] \geq exp(-4n) \cdot 2^{2n} < 1.$$

Thus, there exists a collection $\mathbb{T}$ of t random tapes $r_1, ..., r_t$ for all (x,y) such that,

$$\mathbb{E}[\sum_{i=1}^{t} \frac{1}{t} Z(x,y,r_i)] \leq \epsilon + \delta$$

Alice and Bob exhaustively goes over all sets of random tapes in lexicographical order to find the first $\mathbb{T}$ that satisfies the inequality. Once $\mathbb{T}$ is set the protocol can begin.

Protocol σ: Alice draws $i \in [t]$ uniformly at random, then she runs the protocol Π according to random tape r_i. Next she communicates what protocol Π does, along with i to Bob. He then continues the protocol using random take r_i.

We can see from the protocol σ, that $\|\sigma\| \leq \|\Pi\| + O(\log t) = \|\Pi\| + O(\log n + \log 1/\delta)$. This means that,

$$\begin{aligned} \Pr_{r_i \in \mathbb{T}}[\sigma \, errs \, on \, (x,y)] &= \sum_{i=1}^{t} Pr[r_i \, is \, selected] \times Pr[\sigma \, errs \, on \, (x,y) \mid r_i \, is \, selected] \\ &= \sum_{i=1}^{t} \frac{1}{t} Z(x,y,r_i) \leq \sigma + \delta \end{aligned}$$

□

Now that we have proven that the power difference between pubic and private coin randomness is not significant, we are able to use an analysis of the public coin model to generalize randomized algorithms.

Definition 3. The *randomized communication complexity* $R(f)$ of a function f is $\min_{\Pi} \|\Pi\|$, where Π ranges over all public coin protocols solving f with probability $\geq 2/3$.

Definition 4. Given a distribution μ over inputs (x,y), the *distributional communication complexity* $D_\mu(f)$ of a function f is $\min_{\mu} \|\Pi\|$ where now Π ranges over deterministic protocols solving $f(x,y)$ with probability $\geq 2/3$.

To compare Definitions 3 and 4 we can apply Yao's Min-Max as follows.

Theorem 5. *(Yao's Min-Max)*
For any function f,

- $R(f) \geq D_\mu(f)$ *on any distribution* μ
- $R(f) = D_\mu(f)$ *on some distribution* μ

Theorem 5, tells us that in order to prove a lowerbound for the randomized communication complexity of a problem, we simply need to fix some input distribution μ and show that any deterministic protocol over μ requires a certain amount of communication. We will use this fact to prove the lowerbound for the randomized communication complexity of EQ_n.

Theorem 6. $R(EQ_n) = 1$

Proof. We can construct a simple algorithm solving this problem. Generate a public random string $a \in \{0,1\}^n$ where each a_i is chosen from $\{0,1\}$ independently and uniformly at random. Alice computes and sends $c_A = \sum_{i=1}^{n} y_i a_i \ mod\ 2$ to Bob. He computes $c_B = \sum_{i=1}^{n} x_i a_i \ mod\ 2$ and returns:

$$\begin{cases} 0 & \text{w.p } 1/3 \\ 1 & \text{w.p } 2/3 \text{ if } c_A = c_A \\ 0 & \text{if } c_B \neq c_B \end{cases}$$

Now, we need to show that Bob outputs the correct answer with probability $\geq 2/3$. Suppose $x = y$, then $c_A = c_B$ and Bob returns 1 with probability $2/3$. Now suppose $x \neq y$, then in this case $Pr[c_A = c_B] = 1/2$. We can show that the latter is true by considering the set a where each bit is fixed except the j-th bit (a_j). Let $c'_A := \sum_{i\in[n], i=j} x_i a_i \ mod\ 2$ and $c'_B := \sum_{i\in[n], i=j} y_i a_i \ mod\ 2$ then, we have the following two cases,

- Case 1: If $c'_A = c'_B$ then, $c_a = c_B$ if and only if $a_j = 0$, which happens with probability $1/2$
- Case 2: If $c'_A \neq c'_B$ then, $c_A = c_B$ if and only if $a_j = 1$, which happens with probability $1/2$

So, if $x \neq y$ the protocol returns 0 with probability $\frac{1}{3} + (\frac{2}{3} \times \frac{1}{2}) = \frac{2}{3}$.

□

As we can see, the protocol for EQ_n is very efficient and because of this, it will not give us a meaningful communication lower bound for proving the lower bound for the randomized streaming algorithms. We will consider a different problem for this purpose.

Problem 2. (The Index problem) The IND_n problem is defined as follows. Alice receives an n-bit string $x \in \{0,1\}^n$ and Bob receives an index $i \in [n]$. The goal is to return the i-th bit, $x[i]$, of Alice's string.

Claim 7. *Any randomized streaming algorithm for solving distinct elements requires $O(R(IND_n) - \log n)$ space.*

Proof. Alice defines a set $X = \{j : x_j = 1\}$ and feeds it to the DE algorithm. Then, Alice sends the memory state of the algorithm and $DE(X)$ to Bob. He then feeds i to the DE algorithm and computes the answer $DE(X \circ \{i\})$. Observe that the following are true:

- $X[i] = 0$ if and only if $DE(X \circ \{i\}) = DE(X) + 1$
- If $X[i] = 0$ then, $|X \cup \{i\}| = |X| + 1$ since $i \notin X$
- If $X[i] = 1$ then, $|X \cup \{i\}| = |X|$ since $i = x$

The communication cost of this protocol is the space complexity of DE ($S(DE)$) plus $O(\log n)$. This implies,

- $R(IND_n) \leq S(DE) + O(\log n)$
- $S(DE) \geq R(IND_n) - O(\log n)$

□

The problem here is that $R(IND_n) \leq O(\log n)$, because Bob sends his input to Alice who then return the bit at that index. So, it appear that this is another meaningless lowerbound. On the contrary, we are able to take advantage of a useful property of the reduction to IND_n which is that Bob does not have to respond to Alice's message. This is described in a variant communication model.

2.1 One Way Communication Model

Alice computes some function of her input and sends it to Bob. Instead of replying to Alice, Bob will simply returns the output. We can define communication complexity for this new model similarly to before:

- $\overrightarrow{D(f)}$: Deterministic one-way communication complexity of f
- $\overrightarrow{R(f)}$: Randomized one-way communication complexity of f
- $\overrightarrow{D_\mu(f)}$: Deterministic one-way distributional communication complexity of f