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

Lecture 12

October 21, 2022

Instructor: Soheil Behnezhad *Scribe: LaKyah Tyner*

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

1 Overview

In this lecture, we present a lower bound for the streaming algorithm to compute the number of distinct elements. In order to do this, we first discuss Communication Complexity and demonstrate how to use communication lower bounds to define our streaming lower bound.

2 Communication Complexity

We consider the following basic problem from Communication Complexity:

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. Lets take an example where the two are trying to determine if their inputs are equal.

2.1 Equality Testing

Alice and Bob wish to compute the function $EQ_N(x, y)$ and participate in a protocol Π to determine the result. The equality function is defined as follows:

$$EQ_N(x, y) = \begin{cases} 1 & \text{if } x = y \\ 0 & otherwise \end{cases} \tag{1}$$

Definition 1. The *communication cost* of a protocol Π, denoted by $\|\Pi\|$, is the worst-case number of bits communicated by Π.

Definition 2. The *deterministic communication complexity* of a function f is defined $D(f) = min_{\Pi}\|\Pi\|$ where Π ranges over all "deterministic" protocols for solving f in the worst-case.

3 Protocol Tree

A protocol tree allows us to visualize the communication between Alice and Bob as they interact in a protocol Π. Let the set $S_0 = \{0,1\}^n \times \{0,1\}^n$. Each set S_i is a subset of the one represented by its parent resulting from Alice or Bob communicating an additional bit. In any leaf node, all inputs x, y have the same $f(x, y)$ value. The tree continues to grow until the protocol ends. The communication cost then is the depth of the protocol tree.

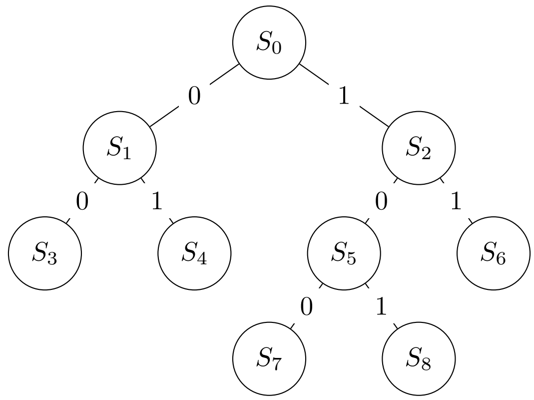

Figure: A protocol tree in which Alice sends a bit first, then Bob sends a bit ending with a bit from Alice.

3.1 Combinatorial Rectangle

Theorem 3. *For a node N in the protocol tree, the set $S \subseteq \{0,1\}^n \times \{0,1\}^n$ forms a combinatorial rectangle. That is $S = X \times Y$ for some $X \subseteq \{0,1\}^n$ and $Y \subseteq \{0,1\}^n$. In other words,*

- *If, $(x, y) \in S$ and $(a, b) \in S$*
- *Then, $(x, b) \in S$ and $(a, y) \in S$*

Proof. We can prove this using induction. It is easy to see that the base case holds i.e that the root node represents a combinatorial rectangle. Lets assume we have also shown that some node N' is a combinatorial rectangle. The subset represented by N' is defined by $S_{N'} = X_{N'} \times Y_{N'}$. Without loss of generality, suppose that at N' Alice communicates the bit 0 and we arrive at a node N. Let S^- denote the subset of strings pruned from $S_{N'}$ by Alice's choice. Since it is Alice who speaks, $X_{N'}$ is pruned to X_N, but $Y_{N'}$ remains unchanged and we have,

$$
\begin{aligned}
S_N &= S_{N'} \setminus S^- \\
&= (X_{N'} \times Y_{N'}) \setminus (X_N \times Y_{N'}) \\
&= (X_{N'} \setminus X_N) \times Y_{N'}
\end{aligned}
\tag{2}
$$

The last equality follows using the definition of Cartesian Product, Set Difference, Union, and DeMorgan's Law.

Claim 4. $D(EQ_N) \geq n$. *We can define a matrix displaying the output of EQ_N on all possible strings communicated by Alice and Bob.*

EQ_N	X_1	X_2	X_3	X_4	X_5	X_6	...	X_n
Y_1	1	0	0	0	0	0	...	0
Y_2	0	1	0	0	0	0	...	0
Y_3	0	0	1	0	0	0	...	0
Y_4	0	0	0	1	0	0	...	0
Y_5	0	0	0	0	1	0	...	0
Y_6	0	0	0	0	0	1	...	0
...	...	...	...	...	...	...	...	0
Y_n	0	0	0	0	0	0	...	1

Figure: This matrix displays the outputs of the equality function on all inputs.

We know that for each $(x_i, y_i) \in S_N$ the value for EQ_N is the same. It is easy to see then that any leaf node that contains a 1 must be a singleton; we are unable to form a larger rectangle without also including 0's. This implies that the number of leaves is at least the number of 1's in the matrix which is $\geq 2^n$. It follows that the depth is equal to $\log_2(2^n) \geq n$.

Theorem 5. *Any deterministic streaming algorithm for exactly solving distinct elements requires* $\Omega(\min(n, m))$ *bits of space where* m *is the size of the alphabet and* n *is the length of the string.*

Proof. Suppose there is a deterministic streaming algorithm $\mathcal{A}$ solving the distinct elements problem in less than $O(\min(n, m))$ space. Let $S_x = \{i | x_i = 1\}$ and $S_y = \{i | y_i = 1\}$. Alice feeds S_x to $\mathcal{A}$, she then sends the memory state of $\mathcal{A}$, denoted my $M_{\mathcal{A}}$, and $DE(S_x)$ to Bob. After receiving communication from Alice, Bob continues to run the streaming algorithm with his string S_y.

- If $S_x = S_y$, then $DE(S_x \circ S_y) = DE(S_x) = DE(S_y)$.
- If $S_x \neq S_y$, then $DE(S_x \circ S_y) > \min(DE(S_x), DE(S_y))$.

The above describes a communication protocol for solving $EQ(X, Y)$ using $|M_{\mathcal{A}}| + |DE(S_x)|$ bits. Claim 4 tells us that $D(EQ(X, Y)) \geq n$ which implies that, $|M_A| + O(\log n) \geq n$ and $|M_{\mathcal{A}}| = \Omega(n)$.