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

Lecture 14

October 28, 2020

Instructor: Soheil Behnezhad *Scribe: Ethan Mook*

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

In this lecture we continue our discussion of streaming lower bounds. In the last lecture, we saw the natural connection between space complexity lower bounds for streaming algorithms and communication complexity lower bounds. In this lecture we continue exploring that connection by using a communication lower bound for the indexing problem to prove a lower bound on single pass streaming algorithms for maximum matching.

1 Streaming Lower Bounds via Communication Complexity

Problem 1. In the indexing problem, denoted IND_n, Alice gets input $x \in \{0,1\}^n$ and Bob gets $i \in [n]$. The goal of the protocol is for Bob to output $x[i]$ the ith bit of x.

1.1 Communication Lower Bound for Indexing

Theorem 1. *The one-way communication complexity of* IND_n *is at least* $\vec{R}(\mathsf{IND}_n) = \Omega(n)$.

At a high level, we will prove Theorem 1 by arguing that in any one-way communication protocol that succeeds with good enough probability Alice's message must uniquely encode the entirety of her input, and thus must be at least as long as it is.

Definition 2. The *Hamming distance* of two strings $x, y \in \{0,1\}^n$ is given by

$$\Delta(x,y) := |\{i \in [n] : x_i \neq y_i\}|.$$

In words, $\Delta(x,y)$ counts the number of positions in which x and y differ.

Our proof will heavily rely on the following technical lemma.

Lemma 3. *For any parameter* $0 < \delta < 1/4$*, there exists a subset* $\mathcal{F} \subseteq \{0,1\}^n$ *of strings satisfying*

1. $|\mathcal{F}| \geq \exp(\frac{\delta^2 n}{4})$.
2. *Any pair of distinct* $x, y \in \mathcal{F}$ *are far apart in Hamming distance:* $\Delta(x,y) \geq \frac{n}{2} - \delta n$.

Proof. We proceed by the probabilistic method, that is, we will argue that a set satisfying the properties exists by showing that a randomly chosen set has a nonzero chance of satisfying them. To begin, consider two strings $x, y \in \{0,1\}^n$ sampled independently and uniformly at random. Observe that their expected Hamming distance is

$$\mathbb{E}[\Delta(x,y)] = \sum_{i=1}^{n} \Pr[x[i] \neq y[i]] = \frac{n}{2}.$$

Noting that each position in x and y is drawn independently, we can apply the Chernoff bound to bound the probability that $\Delta(x,y) < \frac{n}{2} - \delta n$.

$$\Pr\left[\Delta(x,y) < \frac{n}{2} - \delta n\right] \leq \exp\left(-\frac{\delta^2 n^2}{3(n/2)}\right) = \exp\left(-\frac{2}{3} \cdot \delta^2 n\right).$$

Now consider sampling an entire set $\{x_1, \ldots, x_t\}$ independently and uniformly at random, where we set $t = \exp\left(\frac{\delta^2 n}{4}\right)$. Since the Chernoff bound gives an exponentially small bound on the probability that any pair of the sampled points are close in Hamming distance, a simple application of the union bound suffices to complete the proof.

$$\begin{aligned}
\Pr\left[\exists i \neq j \in [t] : \Delta(x_i, x_j) \leq \frac{n}{2} - \delta n\right] &\leq \binom{t}{2} \Pr\left[\Delta(x_1, x_2) \leq \frac{n}{2} - \delta n\right] \\
&\leq t^2 \cdot \exp\left(-\frac{2}{3} \cdot \delta^2 n\right) \\
&= \exp\left(\frac{\delta^2 n}{2}\right) \cdot \exp\left(-\frac{2}{3} \cdot \delta^2 n\right) \\
&= \exp\left(-\frac{\delta^2 n}{6}\right) < 1.
\end{aligned}$$

Thus we have shown that $\{x_1, \ldots, x_t\}$ satisfies the properties of the lemma with nonzero probability, which means, in particular, that there exists a set $\mathcal{F}$ as desired. □

Now we are ready to prove Theorem 1.

Proof of Theorem 1. First note that it suffices to show that any (potentially randomized) protocol that solves IND_n with probability 0.9 (instead of 2/3) must have communication complexity $\Omega(n)$. This is because if a protocol succeeded with probability only 2/3 it could be repeated a constant number of times to amplify its success probability, while not affecting its asymptotic communication complexity.

As we've seen before, we will choose an input distribution μ for which we can prove $\vec{D}_\mu(\mathsf{IND}_n) = \Omega(n)$, then Yao's minimax principle will imply $\vec{R}(\mathsf{IND}_n) = \Omega(n)$ as well. Let μ be the distribution where x is sampled uniformly from the set $\mathcal{F}$ from Lemma 3 (for some fixed constant δ to be defined later) and i is sampled independently and uniformly from $[n]$. Now let $\mathcal{A}$ be a deterministic protocol that solves IND_n for input distribution μ with success probability at least 0.9. Then

$$\Pr_{(x,i)\sim\mu}[\mathcal{A} \text{ errors on } (x, i)] \leq 0.1.$$

Let $\mathcal{F}' \subseteq \mathcal{F}$ be the set of all $x \in \mathcal{F}$ for which $\mathcal{A}$ makes an error on at most a 0.2 fraction of indices, that is,

$$\mathcal{F}' := \left\{x \in \mathcal{F} : \Pr_i[\mathcal{A} \text{ errors on } (x, i)] \leq 0.2\right\}.$$

Our result will follow from the following two claims.

Claim 4. $|\mathcal{F}'| \geq \frac{1}{2}|\mathcal{F}|$.

Proof. Observe that any $x \in \mathcal{F} \setminus \mathcal{F}'$ has the property that $\mathcal{A}$ makes an error on strictly more than a 0.2 fraction of indices i. Then if $\mathcal{F} \setminus \mathcal{F}'$ is too big, then the total probability of $\mathcal{A}$ making an error would exceed 0.1.

$$\Pr_{(x,i)\sim\mu}[\mathcal{A} \text{ errors on } (x, i)] \geq \Pr_{(x,i)\sim\mu}[\mathcal{A} \text{ errors on } (x, i) \mid x \in \mathcal{F} \setminus \mathcal{F}'] \cdot \frac{|\mathcal{F} \setminus \mathcal{F}'|}{|\mathcal{F}|} > 0.2 \cdot \frac{|\mathcal{F} \setminus \mathcal{F}'|}{|\mathcal{F}|}.$$

The left-hand side is at most 0.1 by the assumption of $\mathcal{A}$'s success probability, therefore $|\mathcal{F} \setminus \mathcal{F}'|/|\mathcal{F}| < 1/2$ and hence $|\mathcal{F}'| \geq |\mathcal{F}|/2$ as desired. □

Claim 5. *For any $x \in \mathcal{F}'$ let $M(x)$ be the message Alice sends to Bob when she executes $\mathcal{A}$ on input x (recall this doesn't depend on Bob's input because we're considering one-way communication protocols). Given only $M(x)$ we can uniquely recover $x \in \mathcal{F}'$. Equivalently, the map $M : \mathcal{F}' \to \{0,1\}^*$ is injective.*

Proof. For $x \in \mathcal{F}'$ define the string $y(x)$ by setting, for each $i \in [n]$,

$$y(x)_i = \begin{cases} 1 & \text{if Bob returns 1 on index } i \text{ for message } M(x) \\ 0 & \text{otherwise.} \end{cases}$$

By construction of $\mathcal{F}'$, the protocol $\mathcal{A}$ makes an error on at most a 0.2 fraction of indices when Alice's input is x so it must be the case that $\Delta(x, y(x)) \leq 0.2n$. For any other string $x' \in \mathcal{F}'$ with $x' \neq x$ we have

$$\Delta(x', y(x)) \leq \Delta(x', x) - \Delta(x, y) \geq \frac{n}{2} - \delta n - 0.2n,$$

where the first inequality follows by the triangle inequality. Setting $\delta < 0.1$, we get

$$\Delta(x', y(x)) > (0.5 - 0.1 - 0.2)n = 0.2n.$$

Thus given only $M(x)$, we can compute $y(x)$ and find the unique $x \in \mathcal{F}'$ with $\Delta(x, y(x)) \leq 0.2n$, and thus fully recover x. □

Now to complete the proof of Theorem 1, observe there are $|\mathcal{F}'| \geq |\mathcal{F}|/2 = \Omega(\exp \delta^2 n/4)$ possible inputs from $\mathcal{F}'$ that Alice could receive. Therefore the worst-case message length $\max_x |M(x)|$ must be at least $\log_2 |\mathcal{F}'| = \Omega(n)$ which completes the proof. □

2 Streaming Lower Bound for Maximum Matching

Now we will show how to use the indexing lower bound we just proved to get a lower bound on the space complexity of computing the maximum matching of a graph.

Theorem 6. *Any (potentially randomized) single pass streaming algorithm that exactly computes the size of the maximum matching of a graph requires $\Omega(n^2)$ space.*

Proof. Given an instance of IND_{n^2} the indexing problem over n^2 bits, index Alice's input by pairs from $[n]$. That is write $x = x_{11}x_{12} \cdots x_{21} \cdots x_{nn}$. Define the input graph G for the streaming problem as follows (see also Figure 2): G contains a bipartite graph H over $2n$ vertices. We add an edge between vertex i in the left half of H and vertex j in the right half if and only if $x_{ij} = 1$ in Alice's input. For each vertex in H we add an additional corresponding outer vertex in G. Given Bob's input index ij, we add an edge between every vertex in H and its corresponding outer vertex except for the vertices i in the left half of H and j in the right half of H. Finally we define the stream so that we send all of the edges in H over first, and then send all the outer edges.

Observe that every vertex in H can be matched with its corresponding outer vertex except for the vertices defined by Bob's input, and those vertices can be matched if and only if they are connected within H. Therefore the size of the maximum matching is either $2n - 1$ if $x_{ij} = 1$ or $2n - 2$ if $x_{ij} = 0$.

Intuitively, any streaming algorithm must "remember" all of the edges of H in order to answer correctly later when it is able to learn Bob's input. To make this formal, suppose there exists a single-pass streaming algorithm that solves maximum matching. Now consider the communication protocol where Alice uses her input to compute H and streams it into the algorithm. Then she sends the state of the algorithm to Bob who then uses his input to compute the set of outer edges and streams that into the algorithm as the rest of the input. Finally Bob outputs 1 if the algorithm outputs $2n - 1$ and 1 otherwise.

It is clear that Alice and Bob can perfectly recreate the stream from their input, so their protocol is correct exactly when the streaming algorithm is. Thus we conclude that the streaming algorithm must use $\Omega(n^2)$ space because otherwise that would induce a communication protocol solving the indexing problem in sub-linear communication complexity, contradicting Theorem 1. □

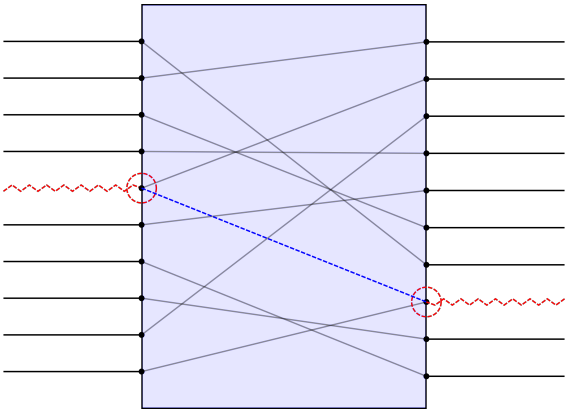

Figure 1: The graph G used in the reduction to indexing. The blue region denotes the subgraph H and the vertices circled in red are defined by Bob's input.

Open Problem. What is the best approximation of maximum matching that we can achieve in sub-quadratic in n space?

There's a simple $O(n \log n)$ space algorithm for finding a *maximal* matching and thus a $(1/2)$-approximation for maximum matching.

Theorem 7. *Any single-pass streaming algorithm for obtaining a $\frac{1}{1+\ln 2}$-approximation of maximum matching size requires space*

$$\Omega\left(n^{1+\frac{1}{\log\log n}}\right) \gg n \cdot \operatorname{poly}\log(n).$$

Theorem 8. *There is a $o(n^2)$ space single-pass algorithm that obtains a $(1-o(1))$-approximation for maximum matching.*

Exercise. Prove that finding the maximal independent set of a graph in a single pass requires $\Omega(n^2)$ space even with a randomized algorithm.

Open Problem. Is there a better than $\log\log n$-pass streaming algorithm solving maximal independent set with $\tilde{O}(n)$ space?