CS 229r: Algorithms for Big Data Fall 2015

Lecture Lecture 9 — October 1, 2015

Prof. Jelani Nelson *Scribe: Rachit Singh*

1 Overview

In the last lecture we covered the distance to monotonicity (DTM) and longest increasing subsequence (LIS) problems.

In this lecture we will talk about how to prove space lower bounds for a variety of problems using communication complexity.

2 Space lower bounds

We're going to see some sophisticated techniques to prove space lower bounds. These are all proved via something called **communication complexity**. The problems we're going to look at today are F_0 (distinct elements) - specifically any algorithm that solves F_0 within a factor of ϵ must use $\Omega(1/\epsilon^2 + \log n)$ bits. We're also going to discuss **median**, or randomized exact median, which requires $\Omega(n)$ space. Finally, we'll talk about F_p or $\|x\|_p$, which requires $\Omega(n^{1-2/p})$ space for a 2-approximation.

2.1 2 player communication complexity

Suppose we have Alice and Bob, and a function $f : X \times Y \to \{0, 1\}$. Alice gets $x \in X$, and Bob gets $y \in Y$. They want to compute $f(x, y)$. Suppose that Alice starts the conversation. Suppose she sends a message m_1 to Bob. Then Bob replies with m_2, and so on. After k iterations, someone can say that $f(x, y)$ is determined. The goal for us is to minimize the total amount of communcation, or $\sum_{i=1}^{k} |m_i|$, where the absolute value here refers to the length of the binary string.

A **communicaton protocol** is a way of conversing agreed upon ahead of time, where Alice and Bob both know f. There's obvious the two obvious protocols, where Alice sends $\log X$ bits to send x, or where Bob sends y via $\log Y$ bits to Alice. The goal is to either beat these trivial protocols or prove that none exists.

There's a natural connection between communicaton complexity and space lower bounds as follows: a communication complexity lower bound can yield a streaming lower bound. We'll restrict our attention to 1-way protocols, where Alice just sends messages to Bob. Suppose that we had a lower bound for a communication problem - Alice has $x \in X$, and Bob has $y \in Y$ and we know that the lower bound (LB) on the optimal communcation complexity is $\overrightarrow{D}(f)$. The D here refers to the fact that the communication protocol is deterministic. If there's a streaming problem, then Alice can run her streaming algorithm on x, the first half of the stream, and send the memory contents

across to Bob, who can then load it and pass y, the second half of the stream, and calculate $f(x, y)$, the final answer. So the minimal amount of space necessary is $\overrightarrow{D}(f)$.

2.2 F_0

Exact and deterministic F_0 requires $\Omega(n)$ space (we saw this in class via the compression argument, but we want to rephrase in the communication complexity argument). We'll use a reduction - if comm. complexity is hard, then the F_0 problem must also be hard, because otherwise we could use the above argument. We use the *equality problem* (EQ), which is where $f(x, y) = x == y$. We claim $D(EQ) = \omega(n)$. This is pretty simple to prove in the one-way protocol, by using the pigeonhole principle, as before.

We're going to reduce EQ to F_0. Suppose that there exists a streaming algorithm A for F_0 that uses S bits of space. Alice is going to run A on her stream x, and then send the memory contents to Bob. Bob then queries F_0, and then for each $i \in y$, he can append and query as before, and solve the equality problem. However, this solves EQ, which requires $\Omega(n)$ space, so S must be $\Omega(n)$. This is just a rephrasing of the earlier argument in terms of communication complexity.

Now, a few definitions:

- $D(f)$ is the optimal cost of a deterministic protocol
- $R_\delta^{\text{pub}}(f)$ is the optimal cost of the random protocol with failure probability δ such that there is a shared random string (written in the sky or something).
- $R_\delta^{\text{pri}}(f)$ is the same as above, but each of Alice/Bob have private random strings.
- $D_{\mu,s}(f)$ is the optimal cost of a deterministic protocol with failure probability δ where $(x, y) \sim \mu$.

Claim 1. $D(f) \geq R_\delta^{\text{pri}}(f) \geq R_\delta^{\text{pub}}(f) \geq D_{\mu,s}(f)$

Proof. The first inequality is clear, since we can just simulate the problem. The second inequality follows from the following scheme: Alice just uses the odd bits, and Bob just uses the even bits in the sky. The final inequality follows from an indexing argument: suppose that P is a public random protocol with a random string s, $\forall (x, y)\, \mathbb{P}(P_s\text{correct}) \geq 1 - \delta$. Then there exists an s^* such that the probability of P_s succeeding is large. Note that s^* depends on μ. □

If we want to do a lower bound on deterministic algorithms, we want to lower bound $D(f)$. If we want to do the lower bound of a randomized algorithm, we want to lower bound $R_\delta^{\text{pri}}(f)$. We need Alice to communicate the random bits over to Bob so that he can continue running the algorithm, and we need to *include* these bits in the cost since we store the bits in memory. So, to lower bound randomized algorithms, we lower bound $D_{\mu,s}(f)$.

If you want to learn more, you can read a book called *Communication Complexity* by Kushilevitz and Nisan **?**. Fun fact: you can solve EQ using public randomness with constant number of bits. If you want to solve it using private randomness for EQ, you need $\log n$ bits. Alice picks a random

prime, and she sends $x \mod p$ and sends across $x \mod p$ and the prime. Neumann's theorem says that you can reverse the middle inequality in the above at a cost of $\log n$ (i.e. the LHS is smaller than $\log n$ times the RHS).

We're going to show that *INDEX*, the problem of finding the jth element of a streamed vector, is hard. Then, we'll show that this reduces to *GAPHAM*, or Gap Hamming which'll reduce to F_0. Also, INDEX reduces to *MEDIAN*. Finally, $DISJ_t$ reduces (with $t = (2n)^{1/p}$) to F_p, $p > 2$.

2.3 Index

INDEX is a two-player problem. Alice gets $x \in \{0,1\}^n$, and Bob gets $j \in [n]$, and $INDEX(x, j) = x_j$.

Claim 2. $R_\delta^{\text{pub}\rightarrow}(INDEX) \geq (1 - \mathfrak{H}_2(\delta))n$, *where* $\mathfrak{H}_2(\delta) = \delta \log(\delta) + (1-\delta)\log(1-\delta)$, *the entropy function. If* $\delta \approx 1/3$.

In fact, it's true that the distributional complexity has the same lower bound. The reason this is hard is because it's one way - Alice doesn't know which bit to send to Bob.

2.4 Information Theory crash course

Mostly definitions, but you can check out *Essentials of Information Theory* by Cover and Thomas for more details.

Definitions: if we have a random variable X, then

- $H(X) = \sum_x p_x \log(p_x)$ (*entropy*)
- $H(X, Y) = \sum_{(x,y)} p_{x,y} \log p_{x,y}$ (*joint entropy*)
- $H(X|Y) = \mathbb{E}_y(H(X|Y = y))$ (*conditional entropy*)
- $I(X, Y) = H(X) - H(X|Y)$ (*mutual information*) (note that this is a symmetric quantity)

The *entropy* is the amount of information or bits we need to send to communicate $x \in X$ in expectation. This can be achieved via Huffman coding (in the limit). The mutual information is how much of X we get by communicating Y.

Here are some basic lemmas involving these equalities

Lemma 3.

- *Chain rule:* $H(X, Y) = H(X) + H(Y|X)$
- *Chain rule for mutual information:* $I(X, Y|Z) = I(X, Z) + I(Y, Z|X)$
- *Subadditivity:* $H(X, Y) \leq H(X) + H(Y)$
- *Chain rule + subadditivity:* $H(X|Y) \leq H(X)$.

- *Basic* $H(X) \leq \log |\operatorname{supp}(X)|$.
- $H(f(X)) \leq H(X) \quad \forall f$ *(no free lunch)*

Theorem 4. ***Fano's Inequality***

Formally, if there exist two random variables X, Y *and a predictor* g *such that* $\mathbb{P}(g(Y) \neq X) \leq \delta$*, then* $H(X|Y) \leq H_2(\delta) + \delta \cdot \log_2(|\operatorname{supp}(X)| - 1)$.

Note that if X is a binary random variable then the second term vanishes. Intuitively, if all you have is Y, and based on Y you make a guess of X. Then if you're able to guess well, then they must be correlated in some way. Note that for small δ, $H_2(\delta) \approx \delta$. Now we'll go back to INDEX, and our earlier claim.

2.5 INDEX revisited

Let Π be the transcript of the optimal communication protocol. It's a one-way protocol here, so it's just what Alice said. So, we know that $R_\delta^{\text{pub}}(INDEX) \geq H(\Pi) \geq I(\Pi, \text{input}) = I(\Pi, \text{input})$

We know that for all x and for all j, $\mathbb{P}_s(\text{Bob is correct}) \geq 1 - \delta$, which implies that for all j, $\mathbb{P}_{X \sim \text{Unif}}\, \mathbb{P}_s(\text{Bob is correct}) \geq 1 - \delta$, which them implies that by Fano,

$$H(X_j|\Pi) \geq H_2(\delta)$$

Note that Π is a random variable because of the random string in the sky, and also because it is dependent on X.

Note that we have

$$\begin{aligned}
|\Pi| &\geq I(X; \Pi) \\
&= \sum_{i=1}^{n} I(X_i; \Pi | X_1, \dots X_{i-1}) \text{ (chain rule n times)} \\
&= \sum_{i} H(X_i|X^{<i}) - H(X_i|\Pi, X^{<i}) \\
&\geq \sum_{i} 1 - H_2(\delta) = n(1 - H_2(\delta))
\end{aligned}$$

Now that we have INDEX, let's use it to prove another lower bound, namely MEDIAN. We want a randomized, exact median of $x_1, \dots x_n$ with probability $1 - \delta$. We'll use a reduction (see **?**).

Claim: INDEX on $\{0,1\}^n$ reduces to MEDIAN with $m = 2n + 2$, with string length $2n - 1$. To solve INDEX, Alice inserts $2 + x_1, 4 + x_2, 6 + x_3 \dots$ into the stream, and Bob inserts $n - j$ copies of 0, and another $j - 1$ copies of $2n + 2$.

Suppose that $n = 3$ and $x = 101_2$. Then Alice will choose $3, 4, 7$ out of $2, 3, 4, 5, 6, 7$. Bob cares about a particular index, suppose the first index. Bob is going to make this stream length 5, such that the median of the stream is exactly the index he wants. Basically, we can insert 0 or $2n + 2$ exactly where we want, moving around the j index to be the middle, which then we can then output.

2.6 INDEX $\rightarrow$ GAPHAM $\rightarrow$ F_0

GAPHAM (Gap Hamming): Alice gets $x \in \{0,1\}^n$ and Bob gets $y \in \{0,1\}^n$. They're promised that the Hamming distance $\Delta(x,y) > n/2 + c\sqrt{n}$ or $\Delta(x,y) < n/2 - c\sqrt{n}$ for some constant c, and we need to decide which.

The reduction INDEX $\rightarrow$ GAPHAM was shown by **?**, implying an $\Omega(n)$ lower bound for GAPHAM. The lower bound $R_{1/3}^{\text{pub}\rightarrow}(\text{GAPHAM}) = \Omega(n)$ was also shown earlier, without this reduction, by **??**. It was later shown that even if you allow yourself an arbitrary number of rounds, you still need $\Omega(n)$ communication **?**.

An F_0 algorithm that fails with probability $1/3$ and gives a $(1+\epsilon)$ approximation requires $\Omega(1/\epsilon^2)$ space (assume that $1/\epsilon^2 < n$).

Proof: Reduce from GAPHAM. Alice and Bob get t bit vectors, where $t = \Theta(1/\epsilon^2)$. Note that $c\sqrt{t} \leq \epsilon t/3$. Now, note that $2F_0 = |\operatorname{supp}(x)| + |\operatorname{supp}(y)| + \Delta(x,y)$. Alice sends the streaming memory and $|\operatorname{supp}(x)|$ which is $S + \log t$ bits (where S is the space complexity of the streaming algorithm for F_0 approximation). Bob knows $2(1 \pm \epsilon)F_0 = 2F_0 \pm \epsilon t/2$. Then he can estimate $\tilde{\Delta} = 2F_0 \pm \epsilon t/2 - \|\operatorname{supp}(x)| + |\operatorname{supp}(y)| = \Delta \pm \epsilon t/2$ and can decide GAPHAM. Thus $S + \log t = \Omega(t)$, implying $S = \Omega(t)$.