CS369G: Algorithmic Techniques for Big Data Spring 2015-2016

Lecture 14: Intro to Communication Complexity

Prof. Moses Charikar *Scribes: Ali Shameli*

1 Communication problem

In this lecture we will talk about a specific set of techniques to prove lower bounds for different problems. The most common way of proving that a problem is hard is by showing that solving our problem would imply that we could solve a different problem (or a family of problems) that is believed to be hard. The prototypical example of this is NP-hardness. We are going to follow the same strategy and get lower bounds for algorithms by using reduction to a set of hard problems.

Communication Problem 1. *The general problem that we need to solve is as follows:*
Alice has a binary string $x \in \{0,1\}^a$ and Bob has a binary string $y \in \{0,1\}^b$. They communicate with each other by sending bits and in the end, they need to compute some function $f(x,y)$.

Some remarks:

1. There are different ways of communication. Here we mostly consider (in streaming algorithms) one-way communcation. This means that Alice just sends the messages and Bob should eventually compute $f(x,y)$.
2. We can approach this problem in many different contexts (Deterministic / Randomized (private vs. public)) but we only consider the public randomized scenario.
3. We also need the error probability to be $\epsilon < 1/2$. Note that by having this error probability, we can arbitrarily reduce the error probability by repeating the algorithm. In randomized algorithms, we usually fix the error probability and then prove lowerbound for that particular probability.

Given any streaming algorithm $\mathcal{A}$ that solves a problem P, the strategy to get lower bounds on its space requirement of is :

- to show that any algorithm for P with space s can be turned into a (one-way) communication protocol to compute a specific function $f(x,y)$ with communication complexity s (number of bits sent).
- then we utilize some known lower bound ℓ_f on the communication complexity of $f(x,y)$ to show that $s \geq \ell_f$ for any such algorithm $\mathcal{A}$.

Thus, communication complexity is the study of a collection of communication problems f, for which we can show interesting lower bounds ℓ_f on their communication complexity under different types of protocols (one-way, multi-way, multi-party), that can be used to emulate different kind of algorithmic primitives.

2 Disjointness problem

We start by introducing a very basic communication problem, that of disjointness, and proving lower bounds on its communication complexity.

Disjointness. Given two binary strings $x, y \in \{0,1\}^n$, we define the function

$$DISJ(x,y) = \begin{cases} 0 & \textit{iff } \langle x, y \rangle = 0 \\ 1 & \textit{otherwise} \end{cases}$$

Using the pigeonhole principle we can get the following lower bound on deterministic communication protocols.

Lemma 2. *Every deterministic protocol for DISJ(x,y), $x, y \in \{0,1\}^n$ needs n bits of communication.*

Proof. The proof is simple. We can argue that using less than n bits of communication, there will be two different strings that will get mapped to the same output which in turn will allow us to construct a string y for Bob such that the output for those two different strings will be different. □

As in the previous example many communication complexity lower bounds are proven using counting/entropic arguments. Communication complexity packages such arguments in a clean interface through a few basic problems, such that we can invoke those arguments by simply showing that our problem contains as a special case one of the basic problems. In the next section, we will see a more general version of a counting argument that will allow us to prove the following theorem.

Theorem 3. *Randomized protocol for DISJ(x, y) with probability at least $2/3$ must use $\Omega(n)$ bits of communication.*

Before proceeding with the proof, let us see an example of how we can use this theorem to give a lower bound on streaming algorithm.

Corollary 4. *Any randomized algorithm to estimate F_∞ within $(1 \pm .2)$ must use $\Omega(n)$ bits.*

Proof. We will show that if we can approximate F_∞ within the given accuracy we would be able to compute disjointness. Given Alice's string $x \in \{0,1\}^n$ and Bob's string $y \in \{0,1\}^n$ define respectively the streams $S_x = \{i : x_i = 1\}$ and $S_y = \{i : y_i = 1\}$. We have two cases:

1. $S_x \cap S_y \neq \emptyset$, then $F_\infty = \max_i |f_i| = 2$
2. otherwise $F_\infty \in \{0, 1\}$

Since, $(1 - 0.2) \cdot 2 = 1.6 > (1 + 0.2) \cdot 1 = 1.2 > 0$, we may distinguish between the two cases and thus we can compute disjointness.

□

But how can we prove a lower bound on a randomized algorithm? The idea is to use the minimax theorem to show that we can lower bound the worst-case performance of any randomized algorithm on deterministic inputs by exhibiting a distribution over inputs that is "bad" for all deterministic algorithms.

Lemma 5 (Yao's Lemma [1])**.** *If there exists a distribution D over all possible input strings $(x, y) \in \{0,1\}^a \times \{0,1\}^b$ such that for any deterministic one-way communication protocol P with,*

$$Pr_{(x,y)\in D}[P \textit{ returning the wrong answer on } (x,\ y)] \leq \epsilon$$

the communication cost is bigger or equal to k (in the case of public random bits), then we can infer that any randomize one-way protocol with error less or equal to ϵ on every input, also has communication cost bigger or equal to k.

Using the above lemma, it suffices to find a distribution D on inputs for which every deterministic algorithm does badly.

3 INDEX problem

To provide a lower bound on randomized protocols for disjointness we are going to use a slightly easier problem that of INDEX.

INDEX. Given a string $x \in \{0,1\}^n$ and an index $i \in [n]$ compute $INDEX(x,i) = x_i \in \{0,1\}$.

To see, why index is easier, we are going to show how we can use Disjointness to solve it. Given input (x,i) for INDEX, construct the input (x, e_i) for DISJ. Solving (x, e_i) for DISJ yields the answer to INDEX. So to find a lower bound for DISJ, it suffices to find a lower bound for INDEX.

Theorem 6. *The randomized Communication Complexity of INDEX is $\Omega(n)$.*

Proof. With Yao's lemma in mind, we are going to construct a distribution D over (x,i) such that any deterministic protocol that answers correctly constant probability $> 1/2$ for index uses at least $\Omega(n)$. D is going to be the uniform measure over $\{0,1\}^n \times [n]$, i.e., we pick $x \in_R \{0,1\}^n$ and $i \in_R [n]$ uniformly at random. We need to prove that any deterministic protocol with less or equal to $0.1n$ bits of communication must have error bigger or equal to $\dfrac{1}{8}$.

Since, the protocol is deterministic and Alice transmits $c \cdot n$ bits of information, it means that Alice has a function $f : \{0,1\}^n \to \{0,1\}^{cn}$ that maps input x to output $z \in \{0,1\}^{cn}$. Suppose the vector reconstructed by Bob from z is the vector $a(z) \in \{0,1\}^n$. We have:

$$Pr[error] = Pr_{(x,i)\sim D}[x_i \neq a(f(x))_i].$$

Since, all the inputs have the same probability, and the maps f and a are deterministic, it suffices to prove a lower bound on the *number of "bad" inputs*, that is we will employ a counting argument!

Observe that since the index i is picked at random, given two vectors $x, y \in \{0,1\}^n$ the event that

$$Pr_{i\sim[n]}[x_i \neq y_i] = \frac{d_H(x,y)}{n}$$

Set $y = y(x) := a(f(x))$ then we have the following consequence:

$$Pr[error] = \mathbb{E}_{x\sim\{0,1\}^n}\left[\mathbb{E}_{i\sim[n]}[\mathbb{I}(x_i \neq y_i)|x]\right] = \mathbb{E}_{x\sim\{0,1\}^n}\left[\frac{d_H(x, a(f(x)))}{n}\right]$$

The function $a \circ f$ maps the 2^n binary strings to 2^{cn} other binary strings. Thus, for each string y in the image of $a \circ f$, if there can not to many strings are close to y, then the error (average distance) will be large. The following lemma quantifies this.

Lemma 7. *At most half of $x \in \{0,1\}^n$ satisfy $d_H(x, a(f(x))) \leq \frac{n}{4}$.*

Proof. Fix any $y \in (a \circ f)(\{0,1\}^n)$, we have that the number of elements that are within distance $n/4$ are at most

$$|\text{Ball}(y, \frac{n}{4})| = \sum_{j=0}^{\frac{n}{4}} \binom{n}{j} \leq n2^{0.861n}$$

Therefore most points $x \in \{0,1\}^n$ are more that $\frac{n}{4}$ away from $a(f(x))$. □

□

4 Estimating number of distinct elements(F_0)

We have seen in previous lectures that a $1-\epsilon$ approximation for estimating F_0 this problem $\Omega(\frac{1}{\epsilon^2})$ space. We are going to use communication complexity to first show that.

Theorem 8. *A $\left(1 + \frac{c}{\sqrt{n}}\right)$ approximation of F_0s needs $\Omega(n)$ space.*

To prove this, first consider the Gap Hamming Problem:

Gap-Hamming. Given two inputs $x \in \{0,1\}^n$ and $y \in \{0,1\}^n$ we define the function

$$\text{GapHam}(x,y) = \begin{cases} 1 & \text{if } d_H(x,y) \leq \frac{n}{2} - c\sqrt{n} \\ \text{undefined} & \text{otherwise} \\ 0 & \text{if } d_H(x,y) \geq \frac{n}{2} + c\sqrt{n} \end{cases}$$

In the next lecture we will prove the following theorem.

Theorem 9. *Any randomized algorithm for Gap Hamming need $\Omega(n)$ bits of communication*

References

[1] Andrew C. Yao, "Lower bounds by probabilistic arguments", *Foundations of Computer Science (FOCS)*, 1983.