CAS CS 591 B: Communication Complexity

Prof. Mark Bun Fall 2019

Lecture Notes 3: Randomized Communication, Newman's Theorem

Reading.

- Rao-Yehudayoff Chapter 3, through pg. 71 + "Public Coins vs. Private Coins"

We'll begin our systematic study of randomized protocols, or protocols which are allowed to toss coins to make random choices. We will see that some functions can be computed much more efficiently with randomness, but also see lower bound techniques which will show that other functions remain hard.

Concentration inequalities will be used frequently throughout this lecture, so let's state Hoeffding's inequality (or an "additive Chernoff bound"') which will suffice for our purposes.

Theorem 1 (Hoeffding's Inequality)**.** *Let $Z_1, \dots, Z_k \in [0,1]$ be independent random variables, and let $Z = \frac{1}{k}\sum_{i=1}^n Z_i$. Then for every $\alpha > 0$, we have*

$$\Pr[Z - \mathbb{E}[Z] > \alpha] \leq e^{-2\alpha^2 k}.$$

Example 2. Let's design a protocol for estimating the fractional Hamming distance between two bit strings. Alice and Bob hold $x, y \in \{0,1\}^n$ and their goal is to determine $\Delta = |x - y|/n$ up to small additive error α. One strategy they can use is random sampling. Using shared randomness, Alice and Bob sample indices $i_1, \dots, i_k$ i.i.d. from $[n]$ and compute the empirical fraction

$$\hat{\Delta} = \frac{1}{k}\sum_{j=1}^{k} |x_{i_j} - y_{i_j}|.$$

We now analyze the approximation guarantees of this protocol. Define random variables $Z_1, \dots, Z_k \in \{0,1\}$ by $Z_j = 1$ if $x_{i_j} \neq y_{i_j}$. Then $\hat{\Delta} = \frac{1}{k}(Z_1 + \cdots + Z_k)$ and $\mathbb{E}[\hat{\Delta}] = \mathbb{E}[Z_j] = \Delta$. By Hoeffding's inequality,

$$\Pr[|\hat{\Delta} - \Delta| > \alpha] \leq 2\exp(-2\alpha^2 k).$$

So to get this probability below, say, 1/3 it suffices to take $k = \Omega(1/\alpha^2)$. In particular, this means we can estimate Δ to within constant error using constant communication, and to within $o(\sqrt{n})$ error using $o(n)$ communication.

On the other hand, deterministically, even estimating Δ to within constant error $\alpha < 1/4$ requires $\Omega(n)$ communication. To see this, let $E : \{0,1\}^n \to \{0,1\}^m$ be an error correcting code with relative distance $0 < \delta < 1/2$, meaning $x \neq y \in \{0,1\}^n$ implies $|E(x) - E(y)|/n > \delta$. Such codes exist with $m = O_\delta(1)$. Suppose we could deterministically estimate the Hamming distance between m-bit strings to within relative error $\delta/2$ using communication c. Then Alice and Bob could compute EQ_n with communication $c + 2$ by encoding their inputs, estimating the distance between their encodings, and accepting iff this estimate is $\leq \delta/2$. Hence $c = \Omega(n) = \Omega_\delta(m)$.

Example 3. In the first lecture, we saw a randomized protocol for computing EQ_n to error 2^{-k} using kn bits of *public* randomness. We can also design a protocol using private randomness. Alice selects a prime p at random among the first n^2 primes. Interpreting her input x as an n-bit number, she sends both p and $x \bmod p$ to Bob using $O(\log n)$ bits of communication. Bob then checks whether $x \bmod p = y \bmod p$.

If $x = y$, the protocol always succeeds. Otherwise, suppose $x \neq y$. Since $|x - y| \leq 2^n$, it has at most n distinct prime factors. So the protocol succeeds with probability at least $1 - 1/n$.

1 Private Coin Protocols

We'll begin by modeling private coin protocols. In a randomized protocol with private coins, Alice and Bob each sample independent bit strings r_A and r_B uniformly at random. (Usually the length of these strings will not be an important parameter for us, but one can also study tradeoffs between randomness and communication). These strings can be taken as additional parameters in their next message functions m_v, i.e., if a vertex in the protocol tree belongs to Alice, we can think of her next message as $m_v(x; r_A)$. Note that every fixing of the strings r_A and r_B gives nothing more than a deterministic protocol.

Definition 4. A private-coin randomized protocol Π computes a function $f : X \times Y \to \{0, 1\}$ with two-sided error ε if for every $(x, y) \in X \times Y$,

$$\Pr_{r_A, r_B}[\Pi(x, y; r_A, r_B) = f(x, y)] \geq 1 - \varepsilon.$$

The $\mathbf{BPP}_{\varepsilon}^{priv}$ cost of such a protocol is the maximum depth of any of the induced deterministic protocols $\Pi(\cdot, \cdot; r_A, r_B)$, and $\mathbf{BPP}_{\varepsilon}^{priv}(f)$ is the least cost of any private coin protocol computing f to error ε.

As we've done before, we may also define the class $\mathbf{BPP^{cc}}$ to consist of the sequences of functions $\{f_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}\}$ for which $\mathbf{BPP}_{1/3}^{priv}(f_n) = \mathrm{polylog}(n)$. The rest of this lecture is devoted to understanding how robust this definition is. First, does the choice of the constant 1/3 matter? Second, does it matter that we are looking at private coin protocols? Or can public coins make a difference?

2 Error Reduction

Theorem 5. *Let* $0 < \varepsilon, \delta < 1/2$. *Then* $\mathbf{BPP}_{\varepsilon}^{priv}(f) \leq O(\log(1/\varepsilon)/\delta^2) \cdot \mathbf{BPP}_{1/2-\delta}^{priv}(f)$.

Proof. Suppose Π is a private-coin randomized protocol with error $\frac{1}{2} - \delta$. Let Π^k be the protocol obtained by running Π k times using independent coin tosses and taking the majority vote of the outcomes. Let us analyze the error of Π^k.

Fix an input $(x, y) \in X \times Y$. For $i = 1, \ldots, k$, let $Z_i \in \{0, 1\}$ be an indicator random variable for the event that the ith run of Π makes an error in computing f (over the random choices of r_A, r_B). Then $\mathbb{E}[Z_i] \leq \frac{1}{2} - \delta$. Note that an error of Π^k corresponds to the event $Z = \frac{1}{k}\sum_{i=1}^{k} Z_i \geq 1/2$. So by Hoeffding's inequality, the error probability of Π^k is

$$\Pr[Z \geq 1/2] \leq \Pr[Z - \mathbb{E}[Z] \geq \delta] \leq e^{2\delta^2 k}$$

which is at most ε by taking $k = \log(1/\varepsilon)/2\delta^2$. □

Can the error in a randomized protocol be completely removed? A different argument shows that this is possible with at most an exponential blowup in communication.

Theorem 6. *Let $0 < \delta < 1/2$, and let $c = \mathbf{BPP}^{priv}_{1/2-\delta}(f)$. Then $\mathbf{P^{cc}}(f) \leq 2^c \cdot (\log(1/\delta) + c)$. In particular, $\mathbf{BPP}^{priv}_{1/3}(f) \geq \Omega(\log \mathbf{P^{cc}}(f))$.*

Proof. Let Π be a randomized protocol tree of depth c. We will show how to simulate the execution of this tree using roughly 2^c bits of communication. For a fixed input (x, y), let p_ℓ be the probability of the parties reaching leaf ℓ during a random execution of the protocol. If the parties could compute all of the p_ℓ's, then they could simply accept iff the probability of reaching a 1 leaf is higher than the probability of reaching a 0 leaf.

Note that since Alice and Bob use independent coin tosses, we can decompose $p_\ell = p_\ell^A \cdot p_\ell^B$, where p_ℓ^A is the probability over Alice's coin tosses that her messages are consistent with those needed to reach leaf ℓ, and similarly for Bob. Alice can then send all 2^c probabilities p_ℓ^A. The problem is that each probability is a real number which may require many bits to send.

Alice can get around this by approximating each p_ℓ^A to a precision of $k = \log(1/\delta) + c$ bits. This guarantees that every estimate $\hat{p}_\ell^A$ satisfies $|p_\ell^A - \hat{p}_\ell^A| < 2^{-k} = 2^{-c}\delta$. Hence the total error over all the leaves is smaller than δ, which means that the weighted majority of the leaves still gives the correct answer. $\square$

Example 7. The private coin protocol we saw for Equality is optimal, since $\mathbf{BPP}^{priv}_{1/3}(\mathrm{EQ}_n) \geq \Omega(\log \mathbf{P^{cc}}(\mathrm{EQ}_n)) \geq \Omega(\log n)$. Since Equality has a constant communication public coin protocol, this implies that there can be arbitrarily large gaps between public and private coin communication costs.

3 Newman's Theorem

In the public coin model, Alice and Bob are both given access to a shared random string r. Public coin protocols are, of course, more powerful than private coin protocols since Alice and Bob could simply partition r into r_A, r_B to simulate a private coin protocol. Moreover, there is a strict separation. Private coin protocols for Equality require $\Omega(\log n)$ communication, but we saw a public coin protocol with $O(1)$ communication. How much more powerful can public coin protocols be?

It turns out that the answer is "not so much." The following theorem, due to Ilan Newman, shows that any public coin protocol can be simulated by a private coin protocol with a small penalty in the error and a small additive penalty in communication.

Theorem 8. *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. For every $\varepsilon, \delta > 0$, we have $\mathbf{BPP}^{priv}_{\varepsilon+\delta}(f) \leq \mathbf{BPP}^{pub}_{\varepsilon}(f) + O(\log n + \log(1/\delta))$.*

Proof. Suppose we have a public coin protocol Π that achieves error ε using a public random string r. We will compile this protocol into a private coin protocol with similar error and communication. The protocol will be based on the following Lemma:

Lemma 9. *For some $t = O(n/\delta^2)$ there exist strings $r_1, \ldots, r_t$ such that, for every $(x, y) \in \{0,1\}^n \times \{0,1\}^n$, we have*

$$\Pr_{i \leftarrow [t]}[\Pi(x, y; r_i) \neq f(x, y)] \leq \varepsilon + \delta.$$

Assuming the Lemma, let's see how to design a good protocol. Hardcode the set of strings $r_1, \dots, r_t$ into the design of our new protocol. Alice samples (using private randomness) an index $i \in [t]$ and sends it to Bob using $O(\log n) + O(\log(1/\delta))$ bits. The parties then simulate the protocol Π as if the public random string were r_i. The Lemma shows that this succeeds with error at most $\varepsilon + \delta$, and incurs communication overhead $O(\log n) + O(\log(1/\delta))$ over Π itself. $\square$

Let us now see how to prove the Lemma.

Proof of Lemma 9. We construct the set of strings $r_1, \dots, r_t$ using the probabilistic method. That is, we show that a random choice of $r_1, \dots, r_t$ satisfy the conditions of the Lemma with positive probability.

For a pair of inputs (x, y) and a randomness string r, let $Z(x, y, r) = 1$ if $\Pi(x, y; r) \neq f(x, y)$ and $Z(x, y, r) = 0$ otherwise. We can rephrase the Lemma as asserting that the set of strings $r_1, \dots, r_t$ satisfies

$$\frac{1}{t}\sum_{i=1}^{t} Z(x, y, r_i) \leq \varepsilon + \delta$$

for every pair (x, y). Pick a sequence $r_1, \dots, r_t$ independently at random, and fix a pair of inputs (x, y). Then $\mathbb{E}[Z(x, y, r_i)] \leq \varepsilon$, so by Hoeffding's inequality,

$$\Pr_{r_1,\dots,r_t}\left[\frac{1}{t}\sum_{i=1}^{t} Z(x, y, r_i) > \varepsilon + \delta\right] \leq e^{-2\delta^2 t}.$$

For $t = O(n/\delta^2)$, this probability is smaller than 2^{-2n}. By union bounding over all pairs (x, y), we have

$$\Pr_{r_1,\dots,r_t}\left[\exists (x, y) \in \{0,1\}^n \times \{0,1\}^n : \frac{1}{t}\sum_{i=1}^{t} Z(x, y, r_i) > \varepsilon + \delta\right] < 1,$$

and hence some good set of strings $r_1, \dots, r_t$ exists. $\square$

Newman's Theorem allows us to focus our study to public coin protocols, which can be convenient for several reasons. We've already seen a few examples of how it can be easier to design public coin protocols. Another nice feature of public coin protocols is that they can be thought of as arbitrary distributions over deterministic protocols. (I.e., Alice and Bob use public randomness to jointly sample a protocol from a collection of deterministic protocols, and execute it over their inputs.) We'll see in the next lecture how this perspective is useful for proving lower bounds.