CAS CS 591 B: Communication Complexity

Prof. Mark Bun — Fall 2019

Lecture Notes 1:

Introduction / Deterministic Protocols

Reading.

- Rao-Yehudayoff Introduction, Chapter 1

Communication is an essential resource for computing in distributed environments. It is also intimately connected to computation itself. In many situations, communication (or the inability to communicate) is the bottleneck for complex computation. In this class, we'll learn a number of techniques for understanding the optimal communication protocols for important problems. We'll also learn how to use this understanding to obtain a broad range of applications in theoretical computer science.

1 Some Examples

The Power of Randomness Suppose Alice holds a string $x \in \{0,1\}^n$ and Bob holds a string $y \in \{0,1\}^n$. How many bits do they need to exchange to determine whether $x = y$? Deterministically, this requires $n + 1$ bits of communication. (Why?) We'll actually see several different proofs of this. However, if Alice and Bob have access to a shared n-bit random string s, then Alice can send Bob $\langle x, s\rangle \bmod 2$, and Bob can check whether this is equal to $\langle y, s\rangle \bmod 2$. If $x = y$, these will be equal with probability 1, and if $x \neq y$, they will disagree with probability at least $1/2$. The failure probability of this protocol can be decreased to 2^{-k} by repeating it k times. We'll also see how to replace the shared random string s with "private" randomness in a generic fashion.

In the world of Turing machines, the question of whether $\mathbf{P} = \mathbf{RP}$ is fascinating, wide open, and believed to be true. But for communication, there's a maximal separation.

Surprising Protocols Let G be a graph on n vertices, and suppose Alice holds a clique C and Bob holds an independent set I. How many bits do they need to exchange to determine whether $C \cap I = \emptyset$? (Or whether they intersect in exactly one vertex.) Describing a clique or independent set requires about n bits, but there's an interactive protocol that does this using $\log^2 n$ bits. Each round of the protocol maintains a set V of "live" vertices. If $C \cap V$ contains a vertex v with degree $\leq |V|/2$, then Alice sends Bob the name of v using $\log n$ bits. Either $v \in I$, or all non-neighbors of v can be removed from V for the next round. On the other hand, if $I \cap V$ contains a vertex with degree $\geq |V|/2$, Bob can send Alice the name of the vertex and remove all of its neighbors. (If no such vertex exists, then $C \cap I = \emptyset$.) This protocol can last for at most $\log n$ rounds, giving $\log^2 n$ total communication.

Applications A major topic in circuit complexity, with applications to understanding neural networks, is to prove lower bounds on the expressive power of circuits comprised of threshold gates.

A linear threshold function computes $\text{sgn}(w_1 z_1 + \cdots + w_s z_s)$ for reals $w_1, \ldots, w_s$. Circuits of the form $\text{THR} \circ \text{MAJ}$ are essentially the most powerful threshold circuits against which we can prove lower bounds – using communication complexity!

The idea is to show that if $f(x, y)$ has a small circuit of this form, then it admits a randomized communication protocol that just barely beats random guessing. The protocol is as follows. Suppose the circuit has size s and the top threshold gate is computed by $T(z) = \text{sgn}(w_1 z_1 + \cdots + w_s z_s)$. Observe that if we sample an index i with probability $|w_i|/(|w_1| + \cdots + |w_s|$, then $\text{sgn}(w_i) \cdot z_i = T(z)$ with probability $> 1/2$. So Alice can simply send this random index, together with the sum of her inputs which feed into the MAJ gate at index i, using $\log s + \log n$ bits.

In 2001, Forster showed that the function $\text{IP}(x, y) = \langle x, y \rangle \bmod 2$ has nearly maximal communication complexity $\Omega(n)$ in this model. (We'll see this beautiful proof later in the course.) Hence it requires exponential $\text{THR} \circ \text{MAJ}$ size $2^{\Omega(n)}$.

2 The Deterministic Model

Let Alice's input come from a set X and let Bob's input come from a set Y. Their goal is to compute a function $f : X \times Y \to \{0, 1\}$. A *communication protocol* Π is a rooted binary tree which encodes all sequences of possible messages between Alice and Bob over the course of trying to compute f. Each internal vertex of the tree v is marked by a speaker (A or B) and a function $m_v : X \to \{0, 1\}$ if the speaker is Alice or a function $m_v : Y \to \{0, 1\}$ if the speaker is Bob. The leaves are marked by 0 or 1 indicating the result of the protocol.

To execute Π on an input (x, y), the parties start at the root r of the tree. If the speaker is Alice, she announces $m_r(x)$ — if the result is 0, the parties go on to the left child of r and if the result is 1, they go to the right child of r, and so forth, until they reach a leaf.

The protocol Π *computes* f if $\Pi(x, y) = f(x, y)$ for every $(x, y) \in X \times Y$. The length of the protocol is the depth of the protocol tree, and the number of rounds is the maximum number of alternations between Alice and Bob. Let $\mathbf{P^{cc}}(f)$ denote the minimum length of a protocol which computes f.

2.1 Counting Lower Bounds

Every function $f : \{0, 1\}^n \times \{0, 1\}^n \to \{0, 1\}$ can be computed using n bits of communication. (Alice sends her input to Bob). It turns out that almost all functions require essentially maximal communication. This can be proved by a counting argument. There are $2^{2^{2n}}$ functions $f : \{0, 1\}^n \times \{0, 1\}^n \to \{0, 1\}$. How many functions can be computed by communication protocols of length c?

A protocol tree of depth c has at most 2^c leaves and 2^c internal vertices. Each internal vertex has a speaker (2 choices) and a next-message function (2^{2^n} choices). This gives at most $(2 \cdot 2^{2^n})^{2^c} \leq 2^{2^{n+c+1}}$ protocols. Together with the 2^{2^c} labelings of the leaves, this is at most $2^{2^c + 2^{n+c+1}}$ functions. Even taking $c = n - 2$, the fraction of all possible functions this covers is only $2^{2^{n-2} + 2^{2n-1} - 2^{2n}} \leq 2^{-2^{2n-2}}$.

This lower bound shows that hard functions are ubiquitous in communication complexity. But we are still confronted with the the usual problem in complexity theory of "finding hay in a haystack": Can we identify explicit functions which are hard? Or better, prove that functions we care about require high communication?

2.2 Rectangles

To answer this question, we take a closer look at the combinatorial structure of communication protocols. Every protocol can be thought of as being built from combinatorial rectangles. A *rectangle* is a set of the form $A \times B \subseteq X \times Y$ where $A \subseteq X$ and $B \subseteq Y$. Equivalently, a rectangle is a set $R \subseteq X \times Y$ such that for every $(x, y), (x', y') \in R$ we have $(x, y') \in R$ and $(x', y) \in R$.

Lemma 1 *Let v be a vertex in a protocol tree Π. Let R_v be the set of inputs (x, y) which cause Π to pass through v. Then R_v is a rectangle.*

Proof: We prove this by induction on the structure of the tree. As the base case, every input passes through the root r, and $R_r = X \times Y$ is a rectangle. Now let $R_v = A \times B$ be a rectangle of inputs passing through vertex v, and let u and w be the children of v. Suppose Alice is the speaker at vertex v, and let $A_0 = \{x \in A : m_v(x) = 0\}$ and $A_1 = \{x \in A : m_v(x) = 1$. Then the sets of inputs which pass through u and w are the rectangles $A_0 \times B$ and $A_1 \times B$, respectively. ■

This proof actually shows a bit more: Namely, the leaves of the protocol tree Π *partition* the input space $X \times Y$ into disjoint rectangles. Moreover, every such rectangle R is monochromatic, in the sense that $\Pi(x, y) = 0$ for all $(x, y) \in R$ or $\Pi(x, y) = 1$ for all $(x, y) \in R$. The following fact gives us our first lower bound technique for explicit functions:

Theorem 2 *If $f : X \times Y \to \{0, 1\}$ is computed by a protocol of cost c, then $X \times Y$ can be partitioned into at most 2^c rectangles which are monochromatic with respect to f.*

Intuitively, if a function f only has "small" monochromatic rectangles, then we need many of these rectangles (and hence a long protocol) to cover the input space. Let's try applying this observation to prove a lower bound for equality:

Example 3 *Let $\mathrm{EQ}_n : \{0, 1\}^n \times \{0, 1\}^n \to \{0, 1\}$ denote the equality function $\mathrm{EQ}(x, y) = 1$ iff $x = y$. We claim that $\mathbf{P^{cc}}(\mathrm{EQ}_n) = n + 1$. To see this, we observe that every 1-monochromatic rectangle with respect to EQ_n has size 1. Hence, we need at least 2^n rectangles to cover the 1-inputs of g, plus at least one more rectangle to cover the 0-inputs. So the communication complexity is exactly $n + 1$.*

Unfortunately, arguments based on rectangle size alone aren't enough for some hard problems. Consider, for instance, the Greater-Than function $\mathrm{GT}_n : \{0, 1\}^n \times \{0, 1\}^n \to \{0, 1\}$ defined by $\mathrm{GT}_n(x, y) = 1$ iff $x > y$ in lexicographic order. Then GT_n has both large 0-monochromatic and 1-monochromatic rectangles.

Next time, we'll see additional techniques which will let us prove lower bounds for this function.