CSE 531: Computational Complexity I **Winter 2016**

Lecture 16: Communication Complexity

Mar 2, 2016

Lecturer: Paul Beame *Scribe: Paul Beame*

1 Communication Complexity

In (two-party) communication complexity we begin with two players Alice and Bob who receive inputs $x \in X$ and $y \in Y$, respectively. Their goal is to compute some function f on $X \times Y$ as follows: They exchange messages until one of them sends the value $f(x, y)$.

Definition 1.1. *Formally, a protocol P is a rooted binary tree with each internal node v labelled by either "A" or "B" and two out-edges, one labelled 0 and the other labelled 1. Each leaf has an output label (typically in $\{0, 1\}$. There is a function f_v associated with node v; if v is labelled A then $f_v : X \to \{0, 1\}$ and if v is labelled B then $f_v : Y \to \{0, 1\}$. The bit sent at node v is sent by the corresponding player and the value is $f_v(x)$ if v has label A and $f_v(y)$ if v has label B. The output $P(x, y)$ of the protocol on input $(x, y) \in X \times Y$ is the label of the leaf reached on input x. The* cost *of the protocol is the height of P.*

Definition 1.2. *For any function f defined on $X \times Y$, define the* deterministic communication complexity *of f,*

$$D^{cc}(f) = \min\{cost(P) \mid P(x, y) = f(x, y) \text{ for all } (x, y) \in X \times Y\}.$$

Note that this definition allows the players to compute *any* function based on the information they have seen; i.e., there is no limit to the computational power of Alice and Bob.

Consider the case that $X = Y = \{0, 1\}^n$. If we have the $PARITY$ function $\oplus_{2n}(x, y)$ then there is a very simple protocol: Alice sends the parity $b = \oplus_n(x)$ of her inputs to Bob who then returns the output $b \oplus (\oplus_n(y))$. This is only 2 bits communication.

Now consider the $EQUALITY$ function which outputs 1 iff $x = y$. It seems that best one can do is have Alice to send x to Bob and then Bob output the answer. This has a total cost of $n + 1$ bits. Indeed, one can see by our definition that this is an upper bound for *any* Boolean function f with two n-bit inputs.

We now see how to show that this is the best possible.

Definition 1.3. *A (combinatorial) rectangle in $X \times Y$ is a set $A \times B$ for $A \subseteq X$, $B \subseteq Y$.*

Lemma 1.4. *If $Dcc(f) \leq c$ then $X \times Y$ can be partitioned into at most 2^c rectangles on which the value of f is constant.*

Proof. We prove by induction that the set of inputs that reach each node v of a protocol P on $X \times Y$ is a rectangle. This is certainly true of the root since that is the rectangle $X \times Y$. At an internal node v with associated rectangle $A_v \times B_v$, Alice or Bob, depending the label of v, computes function f_v of either x or y to determine the next bit to send. Therefore for the set of inputs that reach each of the two children of v, either only the set A_v has been restricted, or only the set B_v has been restricted, resulting in a rectangle in both cases. □

Fooling Sets We now give a method to show that a large number of rectangles are needed for a function f.

Definition 1.5. *For a function $f : X \times Y \to \{0,1\}$, and $b \in \{0,1\}$, a* b-fooling set for f, *is a set $\mathcal{F} = \{(x_1, y_1), \ldots, (x_k, y_k)\} \in X \times Y$ such that for all $(x_i, y_i) \in \mathcal{F}$, $f(x_i, y_i) = b$ but for every $i \neq j$, $f(x_i, y_j) \neq b$ or $f(x_j, y_i) \neq b$.*

It is immediate that no two elements of a b-fooling set for f can be in a rectangle on which the value of f is constant:

Lemma 1.6. *Any cover of $f^{-1}(b)$ by rectangles on which f is constant requires at least as many rectangles as the size of the largest b-fooling set for f.*

This immediately gives us a lower bound for $EQUALITY$. It is easiest to think of this by considering the matrix M_f for a function f, which is a $|X| \times |Y|$ matrix whose (x, y) entry has the value $f(x, y)$. For $EQUALITY$, this matrix is the $2^n \times 2^n$ identity matrix and the fooling set $\mathcal{F}$ consists of the 2^n diagonal elements (x, x) for $x \in \{0,1\}^n$. Therefore any partition of $X \times Y$ into rectangles on which $EQUALITY$ is constant requires at least 2^n rectangles on which $EQUALITY$ is 1. Since such a partition also requires at least one rectangle on which $EQUALITY$ has value 0, we get that there are at least $2^n + 1$ rectangles in any partition on which $EQUALITY$ is constant so the trivial upper bound is optimal:

Lemma 1.7. $D^{cc}(EQUALITY) = n + 1$.

One very important problem that comes up frequently in applications of communication complexity is the set disjointness problem (which probably would have been better termed set intersection). Define $DISJ : \{0,1\}^n \times \{0,1\}^n$ by

$$DISJ(x, y) = \begin{cases} 1 & \exists i \; x_i = y_i = 1 \\ 0 & \text{otherwise.} \end{cases}$$

The reason for the terminology is that we can view x and y as characteristic vectors of subsets of $\{1, \ldots, n\}$ and then $DISJ(x, y) = 0$ iff x and y represent disjoint sets. (Some people flip the values 0 and 1 in defining the function. I prefer this format because then $DISJ(x, y) = \bigvee_{i=1}^{n}(x_i \wedge y_i)$ is a simple monotone function.)

Lemma 1.8. $D^{cc}(DISJ) = n + 1$.

Proof. We define a 0-fooling set for $DISJ$ as follows: For $x \in \{0,1\}^n$, let $\overline{(x)}$ denote the bitwise complement of x; that is $\overline{x}_i = 1 - x_i$. Let $\mathcal{F} = \{(x, \overline{x}) \mid x \in \{0,1\}^n\}$. Clearly $DISJ(x, \overline{x}) = 0$ for $x \in \{0,1\}^n$. Now consider $(x, \overline{x}), (y, \overline{y})$ for $x \neq y \in \{0,1\}^n$. Since $x \neq y$ there is some i such that $x_i = 0$ and $y_i = 1$, or vice versa. Without loss of generality suppose that $x_i = 0$ and $y_i = 1$. Then $\overline{x}_i = 1$ and so $DISJ(\overline{x}, y) = 1 \neq 0$; if we had $x_i = 1$ and $y_i = 0$, then $DISJ(x, \overline{y}) = 1 \neq 0$. Therefore $\mathcal{F}$ is a 0-fooling set. Since it has 2^n elements and $DISJ$ has at least one rectangle on which it has value 1, at least 2^n+1 rectangles are required. Taking logarithms yields the bound. □

There is another method that is useful for proving communication complexity lower bounds.

Theorem 1.9. $D^{cc}(f) \geq \log_2 \textit{rank}(M_f)$ *where the rank is computed over the rational numbers.*

Proof. Given a rectangle $R = A \times B \subseteq X \times Y$, define M_R to be the 01-matrix such that $M_R(x, y) = 1$ iff $(x, y) \in R$. We can write the matrix $M_R = u_A v_B^T$ where u_A is the characteristic vector of the set A and v_B is the characteristic vector of the set B. Therefore rank$(M_R) = 1$.

Let P_f be a protocol for f with cost at most $c = D^{cc}(f)$ and $\mathcal{R}_1(P_f)$ be the set of rectangles given by the protocol that partition $f^{-1}(1)$, the 1's of f. We have $|\mathcal{R}_1(P_f)| \leq 2^c$. Now write

$$M_f = \sum_{R \in \mathcal{R}_1(P_f)} M_R.$$

Since for any matrices M and M', rank$(M + M') \leq$ rank$(M) +$ rank(M'), we have

$$\begin{aligned} \text{rank}(M_f) &\leq \sum_{R \in \mathcal{R}_1(P_f)} \text{rank}(M_R) \\ &= \sum_{R \in \mathcal{R}_1(P_f)} 1 \\ &= |\mathcal{R}_1(P_f)| \leq 2^c = 2^{D^{cc}(f)}. \end{aligned}$$

Taking logarithms base 2 yields the theorem. □

This gives an alternative proof of a lower bound of n for $EQUALITY$ since the rank of the $2^n \times 2^n$ identity matrix is clearly 2^n.

Log-rank conjecture It is conjectured that $D^{cc}(f)$ is at most $\log_2 \text{rank}^{O(1)}(M_f)$. This is a major open problem in communication complexity.

Examples are known for which an exponent of $\log_3 6 \approx 1.63$ is required. The best upper bound is a recent quadratic improvement due to Lovett who showed that $D^{cc}(f)$ is $O(\sqrt{\text{rank}}(M_f))$; Thomas Rothvoss has a nice short proof of the bound. Note this is still exponentially far from the conjectured upper bound.

The log-rank conjecture would have other important implications also. It would yield good bounds on the chromatic number of any graph in terms of the rank of its adjacency matrix.

Communication complexity has a vast number of applications. One can think of the immediately obvious connections to distributed computation, but there are also applications in data structures, where we can think of Alice holding a query and Bob holding the data structure and reading a location of the data structure corresponds to sending an address to Bob who returns the memory contents. It comes up in time-space tradeoffs, where the bits of communication are the contents of the storage between different points in time. The set disjointness problem has come up in the context of algorithmic game theory and mechanism design as the basis for showing lower bounds on implementations of combinatorial auctions. Communication complexity also has applications in circuit complexity also. The earliest motivation came for proving lower bounds in communication complexity was in VLSI.

In VLSI, circuit elements and input locations are laid out in rectangular grids. For any rectangular grid one can partition the input locations precisely in half by making a cut along the short side of the grid which has length at most $\sqrt{A}$ and simulating one side by Alice and the other by Bob. In each time step, each wire across that grid can only send one bit so the total communication in T steps is only $\sqrt{A}T$. Using communication lower bounds of $\Omega(n)$ yields lower bounds of $\sqrt{A}T = \Omega(n)$, or $AT^2 = \Omega(n^2)$ which shows that fast chips require a lot of area. For these problems, our current lower bounds only work if we know which input bits are on each side of the partition. For $EQUALITY$ and $DISJ$, partitions that put x_i and y_i on the same side yield easy functions.

Instead for these VLSI bounds one looks at *best partition* communication complexity in which the protocol gets to choose the partition depending on the function (but not the input values). Nonetheless, there are many natural functions that are hard under the best partition model, for example, $SHIFTED\text{-}EQUALITY(x, y, k)$ for $0 \leq k < n = |x| = |y|$ which has value 1 iff $x_i = y_{(i+k) \bmod n}$ for all i.

We can also insist on 1-way communication complexity, in which Alice sends one message to Bob, who computes the answer. This kind of communication is useful in analyzing streaming algorithms or certain representations of Boolean functions called BDDs. For 1-way communication complexity, it is clear that Alice cannot send the same message on different rows of the matrix M_f so the number of different rows of M_f is a lower bound. A canonical example of a hard problem for 1-way communication complexity that is easy in general is the $INDEX$ function, where Alice

get $x \in \{0,1\}^n$ and Bob gets $i \in \{1,\ldots,n\}$ and the requirement is to output x_i.

2 Nondeterministic Communication Complexity

We can define the two-party nondeterministic communication complexity $N^{cc}(f)$ in simple form by having Alice nondeterministically send a message m_A consistent with her input x to Bob, who simply outputs a value depending on m_A and his input y. A nondeterministic protocol correctly computes $f : X \times Y \to \{0,1\}$ iff

$$f(x,y) = 1 \Leftarrow \exists \text{ consistent message } m_A \text{ s.t. Bob outputs } 1.$$

The complexity of the protocol is the maximum number of bits of m_A and $N^{cc}(f)$ is the minimum complexity over all choices of protocol. Similarly, we can define $coN^{cc}(f) = N^{cc}(\overline{f})$.

Note that though we only had a 1-way model in this case, it is without loss of generality: If we had a nondeterministic protocol that allowed more complicated interaction, Alice could simply nondeterministically guess the entire transcript of the protocol and send it to Bob. Bob would then simply verify that the transcript is consistent with what he would have done on input y and that he would have output 1.

Observe that $N^{cc}(\overline{EQUALITY}) = \log_2 n + 1$ using the following simple non-deterministic protocol: Alice guesses an index i on which inputs x and y differ and sends (i, x_i) to Bob. Bob outputs 1 if and only if $y_i \neq x_i$.

For each choice of Alice's message m_A on which a nondeterministic protocol outputs 1, there is a set $A \subseteq X$ of inputs x that are consistent with message m_A and a set $B \subseteq Y$ of inputs y that are consistent with Bob's output of 1 given m_A. Therefore, if $N^{cc}(f) = c$ then there is a cover of $f^{-1}(1)$ by at most 2^c rectangles on which f has value 1 (1-rectangles). Moreover, given a cover of $f^{-1}(1)$ by 1-rectangles of size at most 2^c, one can obtain a nondeterministic protocol of complexity at most c by having Alice send the name of a rectangle consistent with her input x to Bob who checks whether y is also consistent with that rectangle.

Lemma 2.1. $N^{cc}(f) = \log_2$ *of the minimum number of 1-rectangles in a cover of* $f^{-1}(1)$.

Together with Lemma 1.6, this immediately implies:

Corollary 2.2. *If f has a 1-fooling set of size k then $N^{cc}(f) \geq \log_2 k$.*

Corollary 2.3. $N^{cc}(EQUALITY) = n$ *and* $coN^{cc}(DISJ) = N^{cc}(\overline{DISJ}) = n$.

On the other hand, we also have

Lemma 2.4. $N^{cc}(DISJ) = \log_2 n$.

Proof. Alice guesses an i such that $x_i \wedge y_i = 1$ and sends i to Bob who outputs y_i. □

Part of the importance of $DISJ$ is that it in some sense characterizes nondeterministic communication complexity: If $N^{cc}(f) = c$ then there is a cover of $f^{-1}(1)$ by 2^c 1-rectanges. We can reduce f to a disjointness problem on vectors (x', y') of 2^c input bits each by having $x'_i = 1$ iff x is consistent with the i-th rectangle and $y'_i = 1$ iff y is consistent with the i-th rectangle. Alice and Bob can calculate x' and y' without interaction.

3 Randomized Communication Complexity

In this extension, Alice and Bob can each make random choices. We can define analogues $BP^{cc}_{\varepsilon}(f)$ and $R^{cc}_{\varepsilon}(f)$ of ordinary randomized acceptance probabilities.

We can see that this can sometimes yield much more efficient protocols even than nondeterministic ones. For example, consider the following protocol for $EQUALITY$:

Lemma 3.1. *$BP^{cc}(EQUALITY)$ is $O(\log n)$.*

Proof. Alice chooses a random prime p with $n < p < n^2$ (at most $2\log_2 n$ bits to represent) and sends $(p, x \bmod p)$ to Bob where we interpret x as an integer between 0 and $2^n - 1$. Bob outputs 1 iff $y \bmod p = x \bmod p$. Clearly this is correct if $x = y$. An error occurs when $x \neq y$ only if $x \equiv y \pmod p$, but this can only occur if p divides $x - y$. Since $|x - y| < 2^n$, there are fewer than n prime factors of $x - y$ and there are vastly more primes between n and n^2 so this has a tiny failure probability. □

Nonetheless, one can show that

Theorem 3.2. *$BP^{cc}(DISJ)$ is $\Omega(n)$.*

The proof is too complicated for us to cover here. The original proof is due to Kalyanasundaram and Schnitger, but there have been other useful proofs by Razborov and others that simplify the argument somewhat. This result probably has the most applications of any theorem in communication complexity and is worth making note of for future reference.

We conclude with an idea of how such results are proved. An important piece to remember is Yao's Lemma, which has many applications outside of communication complexity.

As we have described things, if Alice or Bob wants to let the other player know about their random choice, it costs bits to send them to each other. One can also define a variant in which the random choices are publically viewable by both players. It turns out that this public variant yields precisely

same complexity up to an additive $O(\log n)$ bound. With public randomness, we can think of randomized protocols as simply distributions on deterministic protocols.

Definition 3.3. *Let μ be a probability distribution on $X \times Y$. Write $D^{cc}_{\mu,\varepsilon}(f)$ to be the minimum number of bits sent by any protocol P such that $\mathbb{P}_\mu[P(x,y) \neq f(x,y)] \leq \varepsilon$.*

Lemma 3.4 (Yao's Lemma, easy part)**.** *For any distribution μ on $X \times Y$, $BP^{cc}_\varepsilon(f) \geq D^{cc}_{\mu,\varepsilon}(f)$.*

Proof. We can view every c-bit randomized protocol for f with error at most ε as a distribution on c-bit deterministic protocols, each one given by a fixing of the public randomness. For each fixed $(x,y) \in X \times Y$, the average over the errors of these deterministic protocols is at most ε. Therefore, if we average over these choices of (x,y) according to μ as well as over the choices of the deterministic protocols then the average error is at most ε. We can think of this as a big matrix (not M_f) with rows indexed by (x,y) and columns indexed by protocols. The average error over the whole matrix is at most ε, so there is some column whose error is at most ε with respect to f. Fix this deterministic protocol. It is a protocol witness the fact that $D^{cc}_{\mu,\varepsilon}(f) \leq c$, which proves what we needed to show. □