COS597D: Information Theory in Computer Science November 7, 2011

Lecture 13

Lecturer: Mark Braverman *Scribe: Gabriel Cadamuro**

1 Introduction to communication complexity

Communication complexity roughly asks how much two information two parties need to exchange between one another in order to compute a function on their inputs.

This has applications in areas like circuit design, where we can model a board as being partitioned in two halves and we wish to find how how information has to pass from one partition to another. Of course, there are also many applications in the design of computer networks.

Definition 1. *Let X and Y be the respective set of inputs for Alice and Bob (the two parties) Then Alice and Bob wish to compute some function f: $X \times Y \mapsto Z$*

Note that for the reminder of these notes X and Y are assumed to be n-length binary inputs and Z is assumed to be the set (0,1). A picture of f being evaluated on $X \times Y$ is shown below for 3 bit inputs

	000	001	010	011	100	101	110	111
000	0	1	1	0	1	0	0	0
001	1	0	0	0	0	0	0	1
010	1	0	0	0	1	0	0	0
011	0	0	1	0	0	0	0	1
100	1	0	0	0	1	0	0	1
101	1	1	1	0	0	0	1	1
110	0	0	0	0	1	0	0	0
111	0	1	1	0	1	1	0	1

Definition 2. *A protocol on the function f is a set of instructions which given a party's input and the bits exchanged between the parties so far determines which bit to exchange next or whether to return a 1 or 0 for the inputs.*

A protocol can be thought of more concretely as a binary tree. Each edge corresponds to a 0 or 1 and each node v will contain a function a_v or b_v whose output represents which edges to take. a_v is the function that Alice computes and it depends on X (Alice's input) and all the bits sent between Alice and Bob so far. b_v is defined equivalently for Bob. So each party can do some (possibly very long) computation, send a bit to the other party and then waits for a bit reply before running the next step in the protocol.

In this model the leaves of the tree would represent when the solution for f on the output is given.

Definition 3. *The cost of a protocol is the largest number of bits which are exchanged on a given input pair x and y. Thinking of a protocol as a tree, this would be the depth of the tree. The cost $D(f)$ of a function is the cost of the least expensive protocol which correctly computes it.*

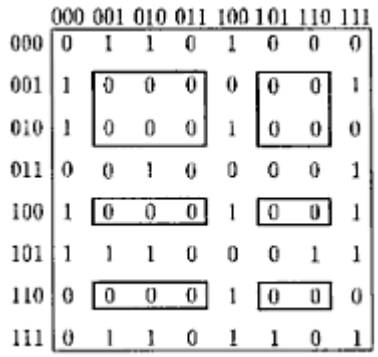
Example 4. *Let f be any function on $X \times Y$. Then note that $D(f) \leq min(\lceil \log |X| \rceil + \lceil \log |Y| \rceil)) + \lceil \log |Z| \rceil$. This is as we could simply transfer the smaller input to the other party and let them compute the function on their own. This bound is tight on some functions such as the equality function, henceforth referred to as EQ.*

*Some use was made of "Communication Complexity" by Kushilevitz and Nisan, the diagrams are taken from that text. All logarithms are taken with base 2.

2 Rectangles

Definition 5. *$R \subseteq X \times Y$ is a rectangle if $R = S \times T$ where $S \subseteq X$ and $R \subseteq Y$.*

As indicated by the notation in the definition, the rectangles of interest will be taken over the product of input sets. Recalling the picture in section 1 we now show a it with the rectangle $\{1, 2, 4, 6\} \times \{1, 2, 3, 4, 6\}$



Now let l be a leaf in the protocol, and R_l be the set of inputs which end up in this leaf. Then we obtain the following theorem.

Theorem 6. *R_l is a rectangle.*

Proof

Claim 7. *If (x_1, y_1) and $(x_2, y_2) \in R_l$ then (x_2, y_1) and $(x_1, y_2) \in R_l$*

Let us consider how input (x_2, y_1) progresses through the protocol tree. The first step is taken by Alice, which outputs the same result as the first step on (x_2, y_2), as no information from Bob is available yet. Now note that (x_1, y_1) and (x_2, y_2) end up in the same leaf and so this output is the same as for (x_2, y_2). Hence now Bob will produce the same output as for his first step for (x_1, y_1) as he got the same input and received the same first bit from Alice. Continuing this logic for all the levels of the tree show us that (x_2, y_1) follows the same path as (x_1, y_1), the same can be said of (x_1, y_2) by symmetry of the argument.

Claim 8. *If $R \subseteq X \times Y$ is such that $(x_1, y_1), (x_2, y_2) \in R \implies (x_1, y_2) \in R$ then this implies R is a rectangle*

This follows by letting $S = \{x \text{ s.t } \exists (x, y) \in R\}$ and $T = \{y \text{ s.t } \exists (x, y) \in R\}$. Then we note that for any $(x, y) \in S \times T$ that $\exists x', y'$ such that $(x, y') \in R$ and $(x', y) \in R$ But by the assumption of the claim this tells us $(x, y) \in R$ for any x, y we pick. This implies $R = S \times T$.

Combining the second claim, generally true about rectangles, and our first claim about sets corresponding to leaves proves the theorem. ■

Now note that any protocol that computes f will induce a partition on $X \times Y$, where one partition holds all inputs that return a value of 1 and the other holding all the inputs which return 0. Moreover note that a protocol will also split the inputs by which leaves they end in, which we know from above is now a partition into monochromatic rectangles. We take monochromatic to mean that all the values in the rectangle either all take 0 or all take 1 on f. This gives us our first method to bound $D(f)$

Corollary 9. *If a partition of $X \times Y$ into monochromatic rectangles takes $\geq 2^t$ rectangles, then $D(f) \geq \lceil t \rceil$*

This is as you need at least 2^t leaves, which can only be achieved with a binary tree of depth at least t. This allows us to revisit the simple example involving the equality function:

Example 10. *The equality function's partition on the matrix $X \times Y$ can be though of as an 2^n by 2^n identity matrix if you consider the inputs that work as 1's and the other inputs as 0.*
Note that any rectangle containing a 1 with side length > 1 will also contain 0's and hence not be monochromatic. Hence we need at least 1 new rectangle for each 1, and several more for the 0s. As such the number of rectangles $> 2^n$ hence $D(f) \geq n + 1$ using the corollary.

3 Fooling sets

Definition 11. *A Fooling set $S \subseteq X \times Y$ is a set such that given a fixed z, $\forall s \in S, f(s) = z$ and for all distinct pairs in S, no two can be in the same monochromatic rectangle.*

Clearly, if you have a fooling set S, then we note from the section about rectangles that you get $D(f) > log(|S|)$
We now illustrate the use of Fooling sets by giving an example where they are easier to use than rectangles.

Example 12. *Let $DISJ(A, B)$ be the disjointness function on subsets of $\{1, \ldots, n\}$ that returns 1 when A and B are disjoint and 0 otherwise. Consider the set of inputs $S = \{(A, A^c)\}$. Note that all input are such that $DISJ$ returns 1 on them. However, note that for any two $(A, A^c), (B, B^c)$ we have that either $DISJ(A, B^c) = 0$ or $DISJ(A^c, B) = 0$ (or both). As such, no two $s_1, s_2 \in S$ are in a rectangle and we hence have a fooling set of size 2^n.*

A reformulation of this is as follows. Let μ be a probability distribution on $X \times Y$ such that each monochromatic rectangle has size $< \delta$ when weighted by μ. This means there are at least $\frac{1}{\delta}$ rectangles and hence $D(f) > log(\frac{1}{\delta})$

This reformulation can be used to solve another problem easily:

Example 13. *The function in question is the inner product, where $f(x, y) = x \cdot y \mod 2$. In this case let us take μ to be uniform over all entries that equal 0 and let $A \times B$ be a monochromatic rectangle containing inputs that evaluate to 0. Letting A' and B' equal span(A) and span(B) respectively, we note that the inner product of any $a' \in A', b' \in B'$ will be the sum of the inner products of $a \in A, b \in B$ and hence equal to 0. Therefore $A' \times B'$ is also a monochromatic rectangle. Noting that $dim(A') + dim(A') \leq n$ implies that $A \times B \leq 2^n$ Taking μ to be the distribution that is uniform over all values with $f = 0$, we note that the weighted probability is less than : $\frac{2^n}{2^{2n-1}} = 2^{-(n-1)}$ We then see than this implies $D(f) > log(2^{n-1})$ hence proving the linear communication complexity of this function.*

4 Rank method

An interesting note is that in the equality function example and the disjointness example we obtained the same result, linear communication complexity, is a similar way. In the equality function there was a diagonal distribution of the inputs which had $f(x, y) = 1$, and the disjoint set we found a fooling set of size 2^n which was also row and column disjoint. The intuition is then that D(f) is related to the rank of the input matrix evaluated on f.

Lemma 14. *Let u_f be the matrix resulting from applying f to each element of the matrix $X \times Y$ taken over the real numbers. Then $D(f) > log(rank(u_f))$.*

Proof For each monochromatic rectangle R which has f equal to 1 on it, denote u_R as the $X \times Y$ matrix with all entries in R equal to 1, and all the other entries equal to 0. Note than $u_f = \sum_{R \text{ is a 1-rectangle}} u_R$. A matrix with only a single rectangle only has one linearly independent row. As such

$$\sum_{R \text{ is a 1-rectangle}} rank(u_R) = |\{R \text{ is a 1-rectangle}\}|,$$

and so $rank(u_f) \leq \sum_{R \text{ is a 1-rectangle}} rank(u_R) = |\{R \text{ is a 1-rectangle}\}| \leq 2^{(D(f))}$ ■

An interesting open problem mentioned in class is the problem of proving whether the converse $D(f) \leq \log(rank(u_f))$ is true or false. Disproving it could be done by constructing a low rank matrix which is nevertheless complicated enough to have high communication complexity. These types of constructions are hard to do unfortunately.

5 Randomized communication

In the previous sections we have always been attempting to find a protocol that solves f perfectly. However we can also define randomized communication, where some errors are allowed and random bits are used.

Definition 15. *Let us say Alice and Bob both have access to the result of the same set of random coin flips and can use these results in their protocol. If this protocol correctly computes f with error at most ϵ on any input then we say this is a protocol for f in the "public randomness" model with error ϵ.*

Definition 16. *$R_\epsilon^{pub}(f)$ is the size of the smallest protocol that solves f in the public randomness model with error ϵ.*

It is very important to note that the error here is being taken with respect to the results of the random coin flips not over any distribution on the input.
There is also an equivalent description when we do not allow the two parties to share their random bits.

Definition 17. *Let the conditions be set up as for the public randomness model, except now Alice and Bob receive separate random coin flips and have no starting information about the other's results. This gives us the equivalent notions of "private randomness" and $R_\epsilon^{pri}(f)$*

It is easy enough to see that $R_\epsilon^{pri}(f) \geq R_\epsilon^{pub}(f)$. This is as we can always make a publicly random protocol that simulates a private random protocol by simply dividing the random coin flips into two partitions and having the rules be such that one party never looks at the other side.
However, it will also be shown in the next lecture that publicly random results are also no more than a log factor better than private results.
We now give an example where the two give an asymptotically equal result.

Example 18. *Consider again the equality function EQ. We would like to show a logarithmic* private coins *protocol for EQ. Private randomness accomplishes this in a different way. Let p be a prime such that p is order of n^2 and then define $p_x(z) = \sum_{i=0}^{n} x_i z^i$ on the finite field of size p. Note that x_i is the i-th bit in the input x. Define $p_y(z)$ in the same manner. Note that these polynomials will only be equal on n different values of z if the polynomials are not the same. As such, Alice can evaluate her polynomial on a p-bit random r_1 from her private randomness and then send both r_1 and $p_x(r_1)$ to Bob to verify. This takes $O(\log(2 * N^2)) = O(\log(n))$ and ϵ falls exponentially with the number of trials as in the public example.*