Communication Complexity I: Deterministic Lower Bounds

Mihai Pătraşcu
IBM Almaden

March 30, 2009

This is part of a series of posts introducing communication complexity. In this first episode, we show how to give deterministic lower bounds by looking at rectangles.

No background is assumed. I believe a high school student should be able to follow this as a gentle introduction to lower bounds. (I mean a high school student who knows the Eastern European level of "basic Math." For example, we use $(1 + \frac{1}{n})^n \to e$ at some point.)

If there are steps that you want clarified, you are welcome to post a question.

Set disjointness. We start with a simple, abstract problem: set disjointness. Let's say Alice has a set S, Bob has a set T, and they are trying to communicate somehow, to determine whether S and T intersect. To quantify the problem, say $|S| = n$, $|T| = m$, and both sets come from a universe $[u] = \{1, \dots, u\}$.

For now, we will only be interested in deterministic communication protocols (as opposed to protocols using randomization, or protocols that work in the average case of an input distribution). In "symmetric" communication complexity, we are interested in the dense case, $n = m = \Theta(u)$, and we ask how many bits of communication Alice and Bob must exchange to solve the problem correctly.

The trivial upper bound is $u + 1$: Alice sends a bit vector of size u, specifying her set, and Bob replies with the answer. Alternatively, Bob can send his bit vector, and Alice computes the answer. What we show below will imply that any deterministic protocol must use $\Omega(u)$ bits of communication in the worst case, so the trivial protocol is asymptotically optimal.

In the asymmetric case, assume $n \ll m$, and $m = \Theta(u)$. Because the input sizes are rather different, it is more interesting to measure the number of bits sent by Alice and Bob separately, instead of measuring the total communication. Denote by a the number of bits sent by Alice throughout the protocol, and b the number of bits sent by Bob.

Clearly, there is a trade-off between a and b. The trivial protocols from above give two points on the trade-off curve. Alice can describe her set using $a = \lg \binom{u}{n} = O(n \lg \frac{u}{n})$ bits, and Bob replies with $b = 1$ bit. Alternatively, Bob can describe his set using $b = u$ bits, and Alice replies with $a = 1$ bit.

Pause for a second, and try to come up with the best trade-off curve interpolating between these two extreme points.

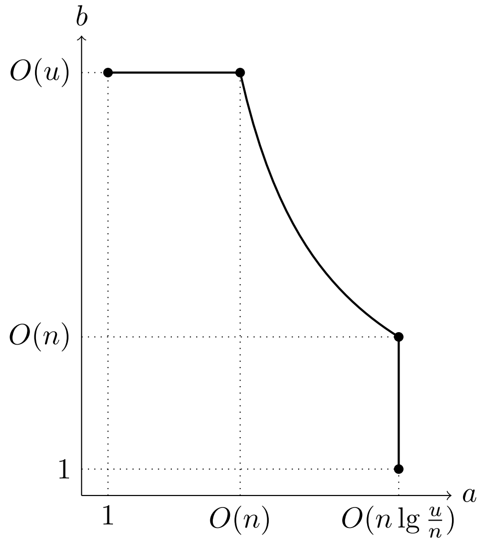

Figure 1: Trade-off between Alice's and Bob's communication.

The trade-off curve can be seen in Figure 1. Here is how it is obtained. Values of $a = o(n)$ are ineffective, i.e. Bob cannot save asymptotically over sending u bits. Similarly for values of $b = o(n)$.

The interesting portion of the trade-off is the curved one, which is described by: $b = u/2^{O(a/n)}$. To achieve this trade-off, let $k \geq a$ be a parameter. We break the universe $[u]$ into k blocks of roughly equal size, u/k. Now:

1. Alice begins by specifying the blocks containing her elements. This takes $a = \lg \binom{k}{n} = O(n \lg \frac{k}{n})$ bits. (If there are fewer than n distinct blocks, throw in some arbitrary ones.)
2. For every block containing an element of Alice, Bob sends a bit vector of size u/k, specifying which elements are in T. This takes $b = n \cdot \frac{u}{k}$ bits.
3. Alice replies with one more bit giving the anwer.

We have $a = O(n \lg \frac{k}{n})$ and $b = n\frac{u}{k}$. Now eliminate the parameter: $k = n \cdot 2^{O(a/n)}$ and thus $b = u/2^{O(a/n)}$.

Lower bounds. We are now going to prove that this trade-off is the best possible for determinstic protocols. This proof originates from [Miltersen, Nisan, Safra, Wigderson STOC'95], the paper that defined asymmetric communication complexity.

To prove lower bounds for set disjointness, we are going to look carefully at the truth table of the function. This is a gigantic object, a 2D array with $\binom{u}{n}$ rows (corresponding to Alice's possible inputs), and $\binom{u}{m}$ columns (corresponding to Bob's inputs). Every entry is 1 if the set described by the column is disjoint from the set described by the row, and 0 otherwise.

The starting point of most communication lower bounds is the concept of (combinatorial) rectangles. A rectangle is a matrix minor of the truth table, i.e. a set $A \times B$, where A is a subset of the rows and B is a subset of the columns. (Warning: this does not look like a rectangle in the geometric sense, since the rows in A and columns in B may not be consecutive.)

Suppose you are an outside observer, who doesn't know the inputs of Alice and Bob, but watches the communication taking places and tries to guess the inputs. After seeing some transcript of the communication, what have you learned about the input? It turns out, the possible problem instances that lead to a fixed communication trascript are always a combinatorial rectangle!

This can be seen by induction on the communication protocol. Before any communication, any inputs look plausible to the outside observer, i.e. the plausible inputs are the the entire truth table matrix. Say that in the next step, Alice sends a bit. This breaks the plausible inputs of Alice in 2 disjoint classes: the inputs for which she would send a 1, and the inputs for which she would send a 0. Observing the bit she sent, your belief about the input changes to a subset for Alice, and does not change in any way for Bob. Thus, your belief changes to a subrectangle, that drops some of the rows of the old rectangle. By analogous reasoning, when Bob sends a bit, your belief changes by dropping some columns.

Communication lower bounds follow by proving two facts. *First*, if the communication protocol is short, the rectangle reached at the end is "large". (Intuitively, there weren't many steps of the induction to shrink it.) But in the final rectangle, all entries must be identical, either zero or one, because the protocol finishes by announcing the answer to the problem. In other words, if the protocol finished by announcing an answer, and there's still a plausible input for which the answer is different, the protocol is sometimes wrong.

Second, one shows combinatorially that any large rectangle is bichromatic (contains both zeros and ones).

We are now going to show lower bounds for set disjointness, implementing these two steps.

Step 1: Richness and large rectangles. How do we prove a useful result saying that the final rectangle must be "large?" For this, [MNSW] introduce the notion of "richness" (I don't know the story behind the name). We say (the truth table of) a function is $[U, V]$-rich if it contains at least V columns, each of which has at least U 1-entries.

Lemma 1. *If Alice and Bob compute a $[U, V]$-rich function by communicating a total of a, respectively b bits, the function contains a rectangle of size $(U/2^a) \times (V/2^{a+b})$ with only 1-entries.*

Proof. By induction on the length of the protocol. Let's say that we are currently in a rectangle $A \times B$ that is $[U, V]$-rich. We have two cases:

- Bob communicates the next bit. Let's say B_0 is the set of columns for which he sends zero, and B_1 is the set for which he sends one. Since $A \times B$ contains V columns with at least U ones, either $A \times B_0$ or $A \times B_1$ contain $\frac{V}{2}$ columns with at least U ones. We continue the induction in the $[U, \frac{V}{2}]$-rich rectangle.

- Alice communicates the next bit, breaking A into A_0 and A_1. Each of the V columns that made $A \times B$ rich has at least $\frac{U}{2}$ ones in either $A_0 \times B$ or $A_1 \times B$. Thus, in either $A_0 \times B$ or $A_1 \times B$ we have find at least $\frac{V}{2}$ columns that have at least $\frac{U}{2}$ ones. We can continue the induction in a rectangle that is $[\frac{U}{2}, \frac{V}{2}]$-rich.

At the end of the protocol, we reach a monochromatic rectangle that is $[U/2^a, V/2^{a+b}]$-rich. Since the rectangle has nonzero richness, it contains some ones, and therefore it contains only ones. Furthermore, it must have size at least $U/2^a$ by $V/2^{a+b}$ to accomodate the richness. □

Step 2: Rectangles in set disjointness. Remember that we are interested in the case of a dense T; let's assume $m = u/2$ for concreteness. A column of the truth table is a set T of $m = u/2$ elements from the universe $[u]$. For each T, there are $\binom{u/2}{n}$ sets S which are disjoint from T. Thus, the problem is $\left[\binom{u/2}{n}, \binom{u}{u/2}\right]$-rich. By the richness lemma, we obtain a one-rectangle of size $\binom{u/2}{n}/2^a$ by $\binom{u}{u/2}/2^{a+b}$.

Assume we have a rectangle $\{S_1, S_2, \ldots\} \times \{T_1, T_2, \ldots\}$ that is all ones. Then, every S_i is disjoint from every T_j. Defining $\mathcal{S} = S_1 \cup S_2 \cup \ldots$ and $\mathcal{T} = T_1 \cup T_2 \cup ...$, it must be that $\mathcal{S}$ is disjoint from $\mathcal{T}$. Therefore:

$$|\mathcal{S}| + |\mathcal{T}| \leq u \tag{1}$$

But also, the size of the rectangle is at most $\binom{|\mathcal{S}|}{n}$ by $\binom{|\mathcal{T}|}{u/2}$, because every row is an n-subset of $\mathcal{S}$, and every column an m-subset of $\mathcal{T}$. Therefore:

$$\binom{|\mathcal{S}|}{n} \geq \binom{u/2}{n}/2^a \tag{2}$$

$$\binom{|\mathcal{T}|}{u/2} \geq \binom{u}{u/2}/2^{a+b} \tag{3}$$

We now pull the following binomial inequality out of our hat:

$$\binom{A}{C} \Big/ \binom{B}{C} \geq (A/B)^C$$

Then (2) becomes $\left(\frac{u}{2|\mathcal{S}|}\right)^n < 2^a$, and thus $|\mathcal{S}| > u/2^{O(a/n)}$. From (1) we have $|\mathcal{T}| \leq u - |\mathcal{S}| \leq u \cdot (1 - 2^{-O(a/n)})$. Applying the binomial inequality to (3), we have:

$$\left(\frac{u}{|\mathcal{T}|}\right)^{u/2} \leq 2^{a+b} \Rightarrow \left(1 + 2^{-O(a/n)}\right)^{\Theta(u)} \leq 2^{a+b} \Rightarrow e^{u/2^{O(a/n)}} \leq 2^{a+b} \Rightarrow b \geq u/2^{O(a/n)}$$

The first implication used $\frac{1}{1-\varepsilon} > 1 + \varepsilon$. The second implication used $\left(1 + \frac{1}{A}\right)^B = \left(\left(1 + \frac{1}{A}\right)^A\right)^{B/A} = e^{\Theta(B/A)}$. This is exactly the desired trade-off.

Acknowledgements. The writing was improved based on comments by an anonymous commenter.