CS 880: Quantum Information Processing 10/28/10

Lecture 22: Communication Complexity

Instructor: Dieter van Melkebeek Scribe: Kenneth Rudinger

Last lecture we saw how quantum communication protocols can lead to constant speedups over classical communication protocols. Given the task of transmitting an n bit classical string, we still require n qubits if there is no prior entanglement; $\frac{n}{2}$ qubits are required if prior entanglement is allowed. Thus, for such a task we *cannot* do better than speedups of constant factor. However, in different communication settings, such as communication complexity, we *can* achieve better than constant speedups. This lecture discusses the communication complexity of three different problems (equality, disjointness, and inner product); we will see how quantum communication can lead to better-than-constant speedups.

1 Communication Complexity Overview

Instead of concerning ourselves with the task of transmitting a string over a quantum communication channel, we now concern ourselves with attempting to compute a function where two parties, Alice and Bob, are each given exclusive access to half of the input, and one (or both parties) wish to know the output of the function.

In other words, consider the following general function f:

$$f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\} : (x, y) \to f(x, y) \tag{1}$$

Alice is given access to the string x, while Bob is given access to the string y. Our goal is to find a communication protocol of minimum cost (one that minimizes the quantity of data transmitted between Alice and Bob) which allows one or both parties to know $f(x, y)$. For any given function f, there are several different variations in the kind of communication allowed, which may (or may not) affect the complexity of computing f. We now describe the possible variants.

1.1 Allowed Directions of Communication

There are three kinds of allowed directions of communication. They are the following, listed in order of increasing restrictiveness.

1. Two-way: In two-way communication, both Alice and Bob may transmit and receive information.

2. One-way: In one-way communication, only one of the two parties (say Bob) may receive information from the other.

3. Simultaneous: In simultaneous communication, both Alice and Bob may transmit information to a third party, Charlie, but cannot receive any information.

1.2 Communication Models

These are the three computational models that we concern ourselves with:

1. Deterministic: All parties' actions are determined by their part of the input and the communication they have received thus far.
2. Randomized: All parties may use randomness to decide on their actions. There are two kinds of randomized models we examine.
   (a) Private random bits: The parties have their own sources of randomness; each may generate random bits, but there is a communication cost for one to know what the other's random bits are.
   (b) Public random bits: The random bits are generated publicly; both Alice and Bob have access to the same random inputs.
3. Quantum: Lastly, we examine quantum communication, in which quantum algorithms and quantum communication channels are allowed when examining a problem's communication complexity. There are two kinds of quantum communication we consider.
   (a) Prior entanglement allowed. Alice and Bob are each given half of one or more EPR pairs.
   (b) Prior entanglement not allowed. EPR pairs do not come for free.

We also note that one may consider quantum communication in which Alice and Bob can share entanglement but only a classical channel of communication. However, we will not further concern ourselves with this particular model of communication.

1.3 Error

Lastly, we may consider communications protocols with different allowed errors.

1. Exact. No error is allowed
2. Bounded error. Error is allowed but bounded. It may be bounded from one or both sides.

2 Problems

We focus on the following three functions $f(x, y)$:

1. Equality predicate:
$$EQ(x,y) = \begin{cases} 1 \text{ if } x = y \\ 0 \text{ otherwise} \end{cases}$$
2. Disjointness predicate:
$$DISJ(x,y) = \begin{cases} 0 \text{ if } \exists\, i \text{ such that } x_i = y_i = 1 \\ 1 \text{ otherwise} \end{cases}$$

3. Inner Product:

$$IP(x,y) = \sum_{i=1}^{n} x_i y_i \mod 2$$

2.1 Equality Predicate

2.1.1 Deterministic Protocol

In the trivial protocol Alice sends all of her input to Bob, who can then evaluate any predicate on the combined input. If Alice also needs to know the answer, Bob then sends that bit to Alice. The resulting communication cost is n or $n+1$. For the equality predicate no determinstic protocol can do better.

2.1.2 Randomized Protocol

Private coins Here is a randomized protocol with private coins. Alice picks a prime number p of $O(\log n)$ bits at random, and sends $x \bmod p$ and p to Bob, at a cost of $O(\log n)$. Bob computes $y \bmod p$ and compares it to $x \bmod p$ for equality. If the two are different, Bob knows with certainty $x \neq y$; if the two are equal, it is likely that $x = y$. This protocol works only needs one-way communication. For simultaneous communication, it turns out that the complexity is $\Omega(\sqrt{n})$; this bound is tight, as there exists a simultaneous protocol that has a cost $O(\sqrt{n})$. One way to obtain a protocol of cost $O(\sqrt{n}\log n)$ uses error-correcting codes.

An error correcting code is a mapping

$$E : \{0,1\}^n \to \{0,1\}^m$$

with certain characteristics, namely the absolute and relative Hamming distance, respectively denotes by Δ and δ, and the rate ρ:

$$\Delta = \min\{\text{number of positions in which } E(x) \text{ and } E(y) \text{ differ for } x,y \in \{0,1\}^n \text{ and } x \neq y\}$$

$$\delta = \frac{\Delta}{m}$$

$$\rho = \frac{n}{m}$$

The purpose of this error-correcting code E is to amplify the difference between each of our distinct strings that reside in $\{0,1\}^n$, by mapping them all to a larger space where they can differ by more bits; even elements that differ by only a single bit in $\{0,1\}^n$ will be mapped to strings that differ by many bits in $\{0,1\}^m$. The number of differing bits is at least Δ. If there is some degradation in the fidelity of some string $E(x)$ (that is, some of its bits are flipped), we may detect this error if fewer than Δ bits are flipped; if fewer than $\frac{\Delta}{2}$ bits are flipped, we may recover x in its entirety, as the degraded string will still be closer (Hamming distance-wise) to $E(x)$ than to $E(y)$ for all other y in $\{0,1\}^n$.

It turns out that there exist families of error-correcting codes with constant $\delta > 0$ and $\rho > 0$. Such families are sometime referred to as "good" error-correcting codes. One such family is known as the Justesen code. We will omit the details of the code, except to say that it is efficiently computable and decadable.

We can use any good family of error correcting codes to develop a simultaneous protocol in the following manner. We know that if $x = y$, then $E(x) = E(y)$. However, if $x \neq y$, then not only will $E(x) \neq E(y)$, but $E(x)$ and $E(y)$ will differ in a fraction at least δ of the bits. Thus, Alice and Bob compute $E(x)$ and $E(y)$, respectively. Both select bit positions at random, and send the bit positions and their corresponding bit values to Charlie. If there is an overlap in the i^{th} position number, then Charlie gets to compare the i^{th} bit of $E(x)$ and $E(y)$, giving a good indication of whether or not $x = y$. Charlie outputs 1 iff there is agreement in all the overlapping positions.

How many such pairs need to be sent to Charlie for a good chance of overlap? By the birthday paradox, we know that sending $\emptyset(\sqrt{m})$ random places to Charlie (which is $O(\sqrt{n})$) will yield a good probability of overlap. Since each position requires $O(\log n)$ bits of communication for the index, this yields an $O(\sqrt{n}\log n)$ simultaneous protocol that has a good probability of successfully computing $EQ(x, y)$.

Public Coins With shared public coins, Alice and Bob can select the same random bit from $E(x)$ and $E(y)$, and transmit it to Charlie. Charlie makes the comparison, and knows with good probability whether or not $x = y$. Thus, using a good error correcting code and $O(\log n)$ public coins, we obtain a protocol of cost $O(1)$.

In fact, at the expense of more public coins, we can use simpler error correcting codes with worse rate. In particular, we can use the Hadamard code, which maps a string x to the sequence of inner products of x with every string of the same length. Both Alice and Bob have access to some random string r of length n. They compute $IP(x, r)$ and $IP(y, r)$, respectively, and send their one bit answers to Charlie. If $IP(x, r) \neq IP(y, r)$, then Charlie knows $x \neq y$. If $IP(x, r) = IP(y, r)$, then Charlie knows that $x = y$ with error bounded by 1/2 (which can be arbitrarily improved by having Alice and Bob compute and transmit their inner products with new random numbers $O(1)$ times).

2.1.3 Quantum Protocol

Based on the classical simultaneous protocol using a good error correcting code, we can construct a quantum simultaneous communication protocol of cost $O(\log n)$, utilizing Alice's and Bob's ability to transmit qubits in superpositions. Alice and Bob prepare the following states:

$$|\psi_{Alice}\rangle = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} |i\rangle \, |(E(x))_i\rangle$$

$$|\psi_{Bob}\rangle = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} |i\rangle \, |(E(y))_i\rangle$$

These states are transmitted to Charlie. If $x = y$, $\langle\psi_{Alice}|\psi_{Bob}\rangle = 1$; otherwise, $|\langle\psi_{Alice}|\psi_{Bob}\rangle| \leq 1 - \delta$. We leave it as an exercise (similar to problem 2 on HW 2) to exhibit a quantum operation Charlie can perform on these states that accepts with probability $(1 + |\langle\psi_{Alice}|\psi_{Bob}\rangle|^2)/2$. Since each of these states contains $O(\log n)$ qubits, this gives a simultaneous quantum protocol without entanglement with a cost of $O(\log n)$.

2.2 Disjointness Predicate

Now we examine the various communicatio settings to evaluate the disjointness predicate, $DISJ(x, y)$.

2.2.1 Deterministic Protocol

As in the case of the equality predicate, n bits must be in order to compute $DISJ(x, y)$.

2.2.2 Randomized Protocol

As there might only be only one intersection between x and y which would make $DISJ(x, y) = 1$, we cannot take advantage of the the birthday paradox. In fact, it can be shown that any randomized protocol has cost $\Omega(n)$, even in the public coin setting.

2.2.3 Quantum Protocol

As it is our goal to find an i such that $x_i = y_i = 1$, our protocol is essentially a Grover search on $O(\log n)$ qubits. However, both Alice and Bob have their own unique parts of the input, so we cannot run Grover immediately. Instead, whenever Grover's algorithm needs to make a query $\sum_i \alpha_i |i\rangle$, we ship that state with a few more qubits back and forth between Alice and Bob so as to obtain $\sum_i \alpha_i |i\rangle |x_i y_i\rangle$. To do so, the following algorithm is executed.

1. Alice begins with the state $|\psi_0\rangle = \sum_i \alpha_i |i\rangle |0\rangle |0\rangle |0\rangle$.
2. By XORing x_i to the second register, Alice makes the state $|\psi_1\rangle = \sum_i \alpha_i |i\rangle |x_i\rangle |0\rangle |0\rangle$, which she sends to Bob.
3. Bob follows the second step in a similar fashion, writing y_i to the third register, yielding $|\psi_2\rangle = \sum_i \alpha_i |i\rangle |x_i\rangle |y_i\rangle |0\rangle$.
4. Bob then transforms $|\psi_2\rangle$ to the following state: $|\psi_3\rangle = \sum_i \alpha_i |i\rangle |x_i\rangle |y_i\rangle |x_i y_i\rangle$.
5. Bob XORs y_i again to the third register, yielding $|\psi_4\rangle = \sum_i \alpha_i |i\rangle |x_i\rangle |0\rangle |x_i y_i\rangle$; this state he sends to Alice.
6. Alice XORs x_i again to the second register, yielding the desired state $|\psi_5\rangle = \sum_i \alpha_i |i\rangle |0\rangle |0\rangle |x_i y_i\rangle$.

Now following Grover, $O(\sqrt{n})$ queries are made, ultimately yielding an evaluation of $DISJ(x, y)$. Because it costs $O(\log n)$ qubits of communication per query, the final cost is found to be $O(\sqrt{n} \log n)$. There does exist a method for eliminating the $O(\log n)$ factor in the cost, yielding an overall cost of $O(\sqrt{n})$, which turns out to be optimal up to constant factors.

2.3 Inner Product

Now we turn to the inner product function, $IP(x, y)$. It turns out that cost for deterministic, random, and quantum protocols are all $\Omega(n)$, even in the bounded-error two-way communication setting. The following is a proof of this claim for the case of exact protocols.

Proof. Suppose we can evaluate $IP(x, y)$ with m qubits of communication on inputs of length n. Then we can perform phase kickback, and apply the following transformation using m qubits of communication.

$$|x\rangle |y\rangle \to (-1)^{x \cdot y} |x\rangle |y\rangle .$$

Then Alice can transmit n classical bits to Bob using m qubits of communication as follows. Given an input x, Alice prepares the state $|x\rangle$; Bob creates a uniform superposition $\frac{1}{\sqrt{N}}\sum_y |y\rangle$. We can then apply our above-described inner product protocol in superposition to yield:

$$|x\rangle \frac{1}{\sqrt{N}} \sum_y |y\rangle \rightarrow |x\rangle \frac{1}{\sqrt{N}} \sum_y (-1)^{x \cdot y} |y\rangle$$

Now Alice has the first register with the state $|x\rangle$; Bob has the second register with the state $\frac{1}{\sqrt{N}}\sum_y (-1)^{x\cdot y} |y\rangle$. Note that the latter is simply the Hadamard transform of $|x\rangle$. Bob can then apply a Hadamard gate to each qubit in his register to invert the transformation:

$$\frac{1}{\sqrt{N}} \sum_y (-1)^{x \cdot y} |y\rangle \rightarrow H^{\otimes n} \frac{1}{\sqrt{N}} \sum_y (-1)^{x \cdot y} |y\rangle = |x\rangle .$$

Thus Bob can recover x (which is n bits long) with m bits of communication. By Holevo's theorem, $m \geq n$, and we are done.

3 Summary, Promise Problems

We summarize our findings of the separations in the various forms of communication protocols in the following table:

	Classical	Quantum	Comments
Two-way	$\Omega(n)$	$O(\sqrt{n})$	From disjointness predicate; a quadratic separation
One-way	-	-	No separation known
Simultaneous	$\Omega(\sqrt{n})$	$O(\log n)$	From equality predicate; an exponential separation

The above results are the best separations known for total functions, i.e., functions that are defined for all strings. Stronger separations are known for promise problems. In fact, there exists a promise problem that has a one-way quantum communication cost of $O(\log n)$ but requires $\Omega(n^{\frac{1}{3}})$ classical communication, even in the two-way setting.

The following is a (continuous) example of such a problem. Alice is given $\mathbf{x} \in \mathbb{R}^n$; Bob is given a subspace $H \subseteq \mathbb{R}^n$, where $\dim H = \frac{n}{2}$. The promise is that either $\mathbf{x} \in H$ or $\mathbf{x} \in H^{\perp}$; the task is to determine which subspace (H or $H^{\perp}$) $\mathbf{x}$ resides in. The quantum protocol is the following: Alice encodes $\mathbf{x}$ in $\log n$ qubits, and sends the qubits to Bob. Bob then measures the projection of $\mathbf{x}$ in H, determining where $\mathbf{x}$ resides.