CMSC 858L: Quantum Complexity

Instructor: Daniel Gottesman

Spring 2023

13 More Adversary Applications and Quantum Communication Complexity

13.1 References

The optimal separation between classical and quantum query complexity is from Aaronson, Ben-David, Kothari, Rao, and Tal, "Degree vs. Approximate Degree and Quantum Implications of Huang's Sensitivity Theorem," 2010.12629 and Ambainis, Balodis, Belovs, Lee, Santha, and Smotrovs, "Separations in Query Complexity Based on Pointer Functions," 1506.04719.

The adversary method, including the theorem presented in this lecture, was introduced by Ambainis, "Quantum lower bounds by quantum arguments," quant-ph/0002066. The relationship between quantum communication complexity and oracles was shown by Buhrman, Cleve, and Wigderson, "Quantum vs. classical communication and computation," quant-ph/9802040, and also gave a $O(\sqrt{N}\log N)$ algorithm for the OR/AND tree problem given as an example.

13.2 Correction

It turns out that the optimal separation between the classical deterministic query complexity is known, up to log factors: It is $D(f) = O(Q(f)^4)$ and $D(f) = \tilde{\Omega}(Q(f)^4)$. (The tilde means "up to log factors".)

13.3 More applications of the adversary method

The same technique that we saw for the unstructured search problem can be used to prove the following theorem:

Theorem 1. *Let f be a function of N binary variables $X_0, \ldots, X_{N-1}$ and X and Y be two subsets of possible input values such that $\forall\, x \in X, y \in Y, f(x) \neq f(y)$. Suppose there exists a relation $R \subseteq X \times Y$ with the following properties:*

1. *$\forall\, x \in X$, there are* at least *m different values of $y \in Y$ such that $(x, y) \in R$*
2. *$\forall\, y \in Y$, there are* at least *m' different values of $x \in X$ such that $(x, y) \in R$*
3. *$\forall\, x = (x_0, \ldots, x_{N-1}) \in X, i \in \{0, \ldots, N-1\}$, there are* at most *$l$ different values of $y = (y_0, \ldots, y_{N-1}) \in Y$ such that $(x, y) \in R$ and $x_i \neq y_i$.*
4. *$\forall\, y = (y_0, \ldots, y_{N-1}) \in Y, i \in \{0, \ldots, N-1\}$, there are* at most *$l'$ different values of $x = (x_0, \ldots, x_{N-1}) \in X$ such that $(x, y) \in R$ and $x_i \neq y_i$.*

Then any quantum algorithm for f uses at least $\Omega(\sqrt{mm'/(ll')})$ queries.

The inputs again represent different oracles and the function is the oracle property we are supposed to compute. The individual variables X_i are the values $O(i)$. The sets X and Y can be thought of as separating the inputs into sets with different function values. R sets up connections between elements with different function values. The first two conditions on the relation say that there are lots of things related every $x \in X$ and every $y \in Y$. The last two conditions say that there are not too many things related to $x \in X$ that are different in a particular coordinate value and similarly for $y \in Y$.

So, to see how this works in practice, consider the data search/ OR problem. We will let X be the all-0 input $(0, \ldots, 0)$, so $f(x) = 0$ for $x \in X$. We will let Y be the set of all Hamming weight 1 inputs O_i. $f(y) = 1$ for $y \in Y$. We will just let the relation R be complete, so all $(x, y) \in X \times Y$ are related.

Then $m = N = |Y|$ and $m' = 1 = |X|$. For $x \in X$, each coordinate has exactly 1 $y \in Y$ that differs from it in that coordinate (O_i differs from the all-0s input on coordinate i), and each $y \in Y$ has at most 1 $x \in X$ that differs from it on coordinate i. Thus $l = l' = 1$. The theorem then says we need $\Omega(\sqrt{N})$ queries.

Another example is an OR/AND tree. Let N be a perfect square, let $g(X_0, \ldots, X_k)$ be the OR function of the $k + 1$ inputs, and let $f(X_0, \ldots, X_{N-1})$ be the AND of $g(X_a, \ldots, X_{a+\sqrt{N}-1})$ for $a = 0, \ldots, \sqrt{N} - 1$. That is, $f = 1$ iff there is at least one marked element in each block of $\sqrt{N}$ inputs to the oracle.

To apply the theorem, we let X be the set of oracles that have exactly 1 marked element for each block of $\sqrt{N}$ oracle inputs and let Y be the set of oracles that have exactly one block of $\sqrt{N}$ oracle inputs that has no marked elements, and all other blocks have exactly 1 marked element. Then $f(x) = 1$ for $x \in X$ and $f(y) = 0$ for any $y \in Y$. These two sets are basically as close as we can get while still having different function values. You can see that changing just one oracle value for Y in the block that has no marked elements will result in an element of X. The relation R is defined to be all such pairs (x, y) which differ on exactly one variable.

For this R, take any $x \in X$. It has $\sqrt{N}$ 1 values, one in each block of $\sqrt{N}$. By changing any one of them, we get an element of Y, so $m = \sqrt{N}$. However, on any particular coordinate, there is at most one related element of Y that differs on that coordinate, so $l = 1$. Similarly, for each $y \in Y$, it is related to exactly $\sqrt{N}$ values of x, that differ from it on one of the coordinates in the unmarked block, but there is no more than one that differs on any particular coordinate. Thus, $m' = \sqrt{N}$ and $l' = 1$.

Then applying the theorem tells us that the quantum query complexity is at least $\Omega(\sqrt{N})$.

13.4 Quantum Communication Complexity

Now for a digression. Let's talk about a different kind of complexity. We've already seen circuit complexity, the size of a circuit to solve some computational problem, and query complexity, the number of oracle queries needed to learn some property of the oracle. *Communication complexity* of a problem is the amount of communication needed to compute some function whose inputs are distributed. Again, we don't care about the gate complexity, only the amount of communication. As with query complexity, it is actually possible to prove things about communication complexity.

There are bewildering number of different variants of communication complexity. Let's focus on 2-party communication complexity, where Alice and Bob have inputs x and y and are trying to compute $f(x, y)$. Even in this context, there are many different choices to make. Obviously, one choice is quantum vs. classical. But there are actually a number of natural ways to make communication complexity quantum. You can say Alice and Bob can send qubits and we want to count the number of qubits. Or you say Alice and Bob have lots of entangled states but they only communicate classically and we count the number of classical bits sent. Or we could count both entangled states and classical bits. Or we could have entangled states but also quantum communication. Luckily, some of these variants are equivalent: In particular, using 1 maximally entangled state and two classical bits, we can use quantum teleportation to send one qubit. We can also use one transmitted qubit plus one maximally entangled state to send 2 classical bits using superdense coding. So when there is plenty of entanglement, the transmission of 1 qubit is equivalent to the transmission of 2 classical bits, and therefore these two types of quantum communication complexity are equivalent up to a factor of 2. But when entanglement is limited, there is a tradeoff between entanglement and communication.

You can also play around with the transmission rules. For instance, maybe both Alice and Bob can send

qubits to each other, or alternatively Alice can send to Bob but not vice-versa. Or Alice and Bob don't communicate directly but instead both send a message to a third party who is supposed to process their transmissions and compute the function. And if Alice and Bob can both send messages, can they go back and forth, or just send once and that's it? Basically all of these different variants can be *separated*: i.e., there exist problems for which the communication complexity of the variants is not the same. And, finally, you can consider deterministic or zero-error communication complexity or bounded-error communication complexity, or even communication complexity with one-sided error (0 error for a yes instance, for instance, but not for a no instance); you can also do this with query or gate complexity.

The most common, default model of quantum communication complexity is where Alice and Bob send qubits and they can both send to each other back and forth in as many rounds as necessary. The measure of quantum communication complexity is how many qubits need to be transmitted in total between Alice and Bob to compute $f(x, y)$ with high probability (greater than 2/3).

The reason I am sticking this topic in here is that, in this model, there is a way to convert a quantum query complexity algorithm into a quantum communication complexity protocol with communication equal to the number of queries times $2(\log N+1)$ (for N possible inputs to the query algorithm). Specifically, given a quantum algorithm to compute $f(O)$ for an oracle $O(z)$, we get a quantum communication complexity protocol to compute $g(x, y)$, where $g(x, y) = f(O)$ where x and y are N-bit vectors and the oracle O is defined by the N-bit vector $x \wedge y$ (the bitwise AND of x and y). (You can use any function $\mathbb{Z}_2 \times \mathbb{Z}_2 \to \mathbb{Z}_2$, but the AND function seems to work particularly well.)

As a concrete example, suppose we start with the unstructured search problem, find a marked element z_0 such that $O(z_0) = 1$. The corresponding communication complexity problem is to find i such that $x_i = y_i = 1$. (Or more simply, to learn if there is such an i.) This is quite a natural problem, known as the *disjointness* problem: For instance, suppose you wish to set up a meeting with your advisor. Here the N bits of x and y represent time slots in your calendar and that of your advisor. The ith bit is 0 if that time slot is unavailable and 1 if it is free. You are looking for a time slot where both are free.

How many bits do you need to send? One of you could send their whole schedule, indicating when they are free and when they are busy, but is that the best thing to do? Classically, this is basically the best you can do — the classical communication complexity is $\Theta(N)$, but if you can send qubits, you can do better.

The idea is to use Grover's algorithm, essentially treating your advisor's schedule as the oracle. Suppose at some point, you have the state $\sum_i \alpha_i |i\rangle$. Then you (quantumly) consult your schedule and compute that in an ancilla register, so the state is $\sum_i \alpha_i |i\rangle|x_i\rangle$. Now you send this state to your advisor, and they perform the unitary $U|i\rangle|c\rangle = (-1)^{cz_i}|i\rangle|c\rangle$ and send the system back. You then erase the ancilla to get $\sum_i \alpha_i (-1)^{x_i z_i}|i\rangle$. You've just performed the oracle O given by the vector $x \wedge y$, as desired. You can thus run Grover's algorithm, using this subroutine in place of the oracle. Each "oracle" call uses $\log N + 1$ qubits transmitted in each direction. At the end, you measure, and with high probability you find a marked element for O, namely a location i such that $x_i y_i = 1$, as desired.

The $\log N$ overhead can be removed, so the actual quantum communication complexity for this problem is $O(\sqrt{N})$, which turns out to be optimal. To prove a lower bound on the quantum communication complexity, however, we can't rely on the oracle bounds (since there might be communication complexity problems that are not converted query algorithms), and different techniques are needed to lower bound communication complexity. I won't go into those, however.

Disjointness provides a quadratic improvement in communication between quantum vs. classical. An exponential separation is known as well, for a more artifical promise problem. Exponential separations are also known between most (maybe all) pairs of variant quantum communication complexity problems mentioned before. As far as I can see, it looks like it is still an open question whether it is possible to get an exponential improvement in communication complexity between randomized classical communication and quantum communication for a total function, at least for the standard model.