Quantum algorithms (CO 781/CS 867/QIC 823, Winter 2011)
Andrew Childs, University of Waterloo

LECTURE 18: The adversary method

So far, we have discussed several different kinds of quantum algorithms. In the last few lectures, we will discuss ways of establishing limitations on the power of quantum algorithms. We begin in this lecture with a very broadly applicable method called the *quantum adversary method*.

Quantum query complexity

Many of the algorithms we have covered work in the setting of query complexity, where the input for a problem is provided by a black box. This setting is convenient since the black box provides a handle for proving lower bounds: we can often show that many queries are required to compute some given function of the black-box input. In contrast, it is notoriously difficult to prove lower bounds on the complexity of computing some function of explicit input data.

We briefly formalize the model of query complexity. Consider the computational task of computing a function $f\colon S \to T$, where $S \subset \Sigma^n$ is a set of strings over some input alphabet Σ. If $S = \Sigma^n$ then we say f is *total*; otherwise we say it is *partial*. The input string $x \in S$ is provided to us by a black box that computes x_i for any desired $i \in \{1, \ldots, n\}$. A query algorithm begins from a state that does not depend on the oracle string x. It then alternate between queries to the black box and other, non-query operations. Our goal is to compute $f(x)$ using as few queries to the black box as possible.

Of course, the minimum number of queries (which we call the *query complexity* of f) depends on the kind of computation we allow. There are at least three natural models:

- $D(f)$ denotes the deterministic query complexity, where the algorithm is classical and must always work correctly.
- R_ϵ denotes the randomized query complexity with (two-sided) error probability at most ϵ. Note that this it does not depend strongly on ϵ since we can boost the success probability by repeating the computation several times and take a majority vote. Therefore $R_\epsilon(f) = \Theta(R_{1/3}(f))$ for any constant ϵ, so sometimes we simply write $R(f)$.
- Q_ϵ denotes the quantum query complexity, again with (two-sided) error probability at most ϵ. Similarly to the randomized case, $Q_\epsilon(f) = \Theta(Q_{1/3}(f))$ for any constant ϵ, so sometimes we simply write $Q(f)$.

We know that $D(\textsc{or}) = n$ and $R(\textsc{or}) = \Theta(n)$. Grover's algorithm shows that $Q(\textsc{or}) = O(\sqrt{n})$. In this lecture we will use the adversary method to show that $Q(\textsc{or}) = \Omega(\sqrt{n})$, a tight lower bound.

Quantum queries

A quantum query algorithm begins from x-independent state $|\psi\rangle$ and applies a sequence of unitary operations $U_1, \ldots, U_t$ interspersed with queries O_x, resulting in the state

$$|\psi_x^t\rangle := U_t O_x \ldots U_2 O_x U_1 O_x |\psi\rangle. \tag{1}$$

To make this precise, we need to specify the action of the oracle O_x.

For simplicity, we will mostly consider the case where the input is a bit string, i.e., $\Sigma = \{0,1\}$. Perhaps the most natural oracle model is the bit flip oracle $\hat{O}_x$, which acts as

$$\hat{O}_x|i,b\rangle = |i, b \oplus x_i\rangle \qquad \text{for } i \in \{1,\ldots,n\},\, b \in \{0,1\}. \tag{2}$$

This is simply the linear extension of the natural reversible oracle mapping $(i,b) \mapsto (i, b \oplus x_i)$. Note that the algorithm may involve states in a larger Hilbert space; implicitly, the oracle acts as the identity on any ancillary registers.

It is often convenient to instead consider the phase oracle, which is obtained by conjugating the bit-flip oracle by Hadamard gates: by the well-known phase kickback trick, $O_x = (I \otimes H)\hat{O}_x(I \otimes H)$ satisfies

$$O_x|i,b\rangle = (-1)^{bx_i}|i,b\rangle \qquad \text{for } i \in \{1,\ldots,n\},\, b \in \{0,1\}. \tag{3}$$

Note that this is slightly wasteful since $O_x|i,0\rangle = |i,0\rangle$ for all i; we could equivalently consider a phase oracle O'_x defined by $O'_x|0\rangle = |0\rangle$ and $O'_x|i\rangle = (-1)^{x_i}|i\rangle$ for all $i \in \{1,\ldots,n\}$. However, it is essential to include the ability to not query the oracle by giving the oracle some eigenstate of known eigenvalue, independent of x. If we could only perform the phase flip $|i\rangle \mapsto (-1)^{x_i}|i\rangle$ for $i \in \{1,\ldots,n\}$, then we could not tell a string x from its bitwise complement $\bar{x}$.

These constructions can easily be generalized to the case of a d-ary input alphabet, say $\Sigma = \mathbb{Z}_d$ (identifying input symbols with integers modulo d). Then for $b \in \Sigma$, we can define an oracle $\hat{O}_x$ by

$$\hat{O}_x|i,b\rangle = |i, b + x_i\rangle \qquad \text{for } i \in \{1,\ldots,n\},\, b \in \mathbb{Z}_d. \tag{4}$$

Taking the Fourier transform of the second register gives a phase oracle $O_x = (I \otimes F^\dagger_{\mathbb{Z}_d})\hat{O}_x(I \otimes F_{\mathbb{Z}_d})$ satisfying

$$O_x|i,b\rangle = \omega_d^{bx_i}|i,b\rangle \qquad \text{for } i \in \{1,\ldots,n\},\, b \in \mathbb{Z}_d \tag{5}$$

where $\omega_d := e^{2\pi i/d}$.

Quantum adversaries

Motivation for the quantum adversary method, as well as its name, comes from the following construction. Suppose the oracle is operated by an adversary who holds a quantum state determining the oracle string, which is in some superposition $\sum_{x\in S} a_x|x\rangle$ over the possible oracles. To implement each query, the adversary performs the "super-oracle"

$$O := \sum_{x\in S} |x\rangle\langle x| \otimes O_x. \tag{6}$$

An algorithm does not have direct access to the oracle string, and hence can only perform unitary operations that act as the identity on the adversary's superposition. After t steps, an algorithm maps the overall state to

$$|\psi^t\rangle := (I \otimes U_t)O \ldots (I \otimes U_2)O(I \otimes U_1)O\left(\sum_{x\in S} a_x|x\rangle \otimes |\psi\rangle\right) \tag{7}$$

$$= \sum_{x\in S} a_x|x\rangle \otimes |\psi^t_x\rangle. \tag{8}$$

The main idea of the approach is that for the algorithm to learn x, this state must become very entangled. To measure the entanglement of the pure state $|\psi^t\rangle$, we can consider the reduced density matrix of the oracle,

$$\rho^t := \sum_{x,y\in S} a_x^* a_y \langle\psi_x^t|\psi_y^t\rangle\, |x\rangle\langle y|. \tag{9}$$

Initially, the state ρ^0 is pure. Our goal is to quantify how mixed it must become (i.e., how entangled the overall state must be) before we can compute f with error at most ϵ. To do this we could consider, for example, the entropy of ρ^t. However, it turns out that other measures are easier to deal with.

In particular, we have the following basic fact about the distinguishability of quantum states (for a proof, see for example section A.9 of KLM):

Fact. *Given one of two pure states $|\psi\rangle, |\phi\rangle$, we can make a measurement that determines which state we have with error probability at most ϵ if and only if $|\langle\psi|\phi\rangle| \le 2\sqrt{\epsilon(1-\epsilon)}$.*

Thus it is convenient to consider measures that are linear in the inner products $\langle\psi_x^t|\psi_y^t\rangle$.

The adversary method

To obtain an adversary lower bound, we choose a matrix $\Gamma \in \mathbb{R}^{|S|\times|S|}$, with rows and columns indexed by the possible black-box inputs. The entry $\Gamma_{x,y}$ is meant to characterize how hard it is to distinguish between x and y. We say Γ is an *adversary matrix* if

1. $\Gamma_{xy} = \Gamma_{yx}$,
2. $\Gamma_{xy} \ge 0$, and
3. if $f(x) = f(y)$ then $\Gamma_{xy} = 0$.

The third condition is intuitively plausible because if $f(x) = f(y)$, then we do not need to distinguish between x and y.

Given an adversary matrix Γ, we can define a weight function

$$W^j := \sum_{x,y\in S} \Gamma_{xy} a_x^* a_y \langle\psi_x^j|\psi_y^j\rangle. \tag{10}$$

Note that this is a simple function of the entries of ρ^j. The idea of the lower bound is to show that W^j starts out large, must become small in order to compute f, and cannot change by much if we make a query.

The initial value of the weight function is

$$\begin{aligned} W^0 &= \sum_{x,y\in S} \Gamma_{xy} a_x^* a_y \langle\psi_x^0|\psi_y^0\rangle \qquad (11)\\ &= \sum_{x,y\in S} a_x^* \Gamma_{xy} a_y \qquad (12)\end{aligned}$$

since $|\psi_x^0\rangle$ cannot depend on x. To make this as large as possible, we take a to be a principal eigenvector of Γ. Then $W^0 = \|\Gamma\|$.

The final value of the weight function is constrain by the fact that we must distinguish x from y with error probability at most ϵ whenever $f(x) \neq f(y)$. To do this after t queries, we need $|\langle \psi_x^t | \psi_y^t \rangle| \leq 2\sqrt{\epsilon(1-\epsilon)}$ for all pairs $x, y \in S$ with $f(x) \neq f(y)$. Thus we have

$$W^t \leq \sum_{x,y \in S} \Gamma_{xy} a_x^* a_y 2\sqrt{\epsilon(1-\epsilon)} \tag{13}$$

$$= 2\sqrt{\epsilon(1-\epsilon)} \|\Gamma\|. \tag{14}$$

Here we can include the terms where $f(x) = f(y)$ in the sum since $\Gamma_{xy} = 0$ for such pairs.

It remains to understand how much the weight function can decrease at each step of the algorithm. We have

$$W^{j+1} - W^j = \sum_{x,y \in S} \Gamma_{xy} a_x^* a_y (\langle \psi_x^{j+1} | \psi_y^{j+1} \rangle - \langle \psi_x^j | \psi_y^j \rangle). \tag{15}$$

Now consider how the state changes when we make a query. We have $|\psi_x^{j+1}\rangle = U^{j+1} O_x |\psi_x^j\rangle$. Thus the elements of the Gram matrix of the states $\{|\psi_x^{j+1}\rangle : x \in S\}$ are

$$\langle \psi_x^{j+1} | \psi_y^{j+1} \rangle = \langle \psi_x^j | O_x^\dagger (U^{j+1})^\dagger U^{j+1} O_y | \psi_y^j \rangle \tag{16}$$

$$= \langle \psi_x^j | O_x O_y | \psi_y^j \rangle \tag{17}$$

since U^{j+1} is unitary and $O_x^\dagger = O_x$. Thus we have

$$W^{j+1} - W^j = \sum_{x,y \in S} \Gamma_{xy} a_x^* a_y \langle \psi_x^j | (O_x O_y - I) | \psi_y^j \rangle \tag{18}$$

Observe that $O_x O_y |i, b\rangle = (-1)^{b(x_i \oplus y_i)} |i, b\rangle$. Let $P_0 = I \otimes |0\rangle\langle 0|$ denote the projection onto the $b = 0$ states, and let P_i denote the projection $|i, 1\rangle\langle i, 1|$. (As with O_x, the projections P_i implicitly act as the identity on any ancilla registers, so $\sum_{i=0}^n P_i = I$.) Then $O_x O_y = P_0 + \sum_{i=1}^n (-1)^{x_i \oplus y_i} P_i$, so $O_x O_y - I = -2 \sum_{i:\, x_i \neq y_i}^n P_i$. Thus we have

$$|W^{j+1} - W^j| = 2 \left| \sum_{x,y \in S} \sum_{i:\, x_i \neq y_i} \Gamma_{xy} a_x^* a_y \langle \psi_x^j | P_i | \psi_y^j \rangle \right| \tag{19}$$

$$\leq 2 \sum_{x,y \in S} \sum_{i:\, x_i \neq y_i} \Gamma_{xy} |a_x^* a_y \langle \psi_x^j | P_i | \psi_y^j \rangle| \tag{20}$$

$$\leq 2 \sum_{x,y \in S} \sum_{i:\, x_i \neq y_i} \Gamma_{xy} \|a_x P_i |\psi_x^j\rangle\| \cdot \|a_y P_i |\psi_y^j\rangle\| \tag{21}$$

where the first inequality uses the triangle inequality and the second uses the Cauchy-Schwarz inequality.

Now for each $i \in \{1, \ldots, n\}$, let Γ_i be a matrix with

$$(\Gamma_i)_{xy} = \begin{cases} \Gamma_{xy} & \text{if } x_i \neq y_i \\ 0 & \text{if } x_i = y_i, \end{cases} \tag{22}$$

and define vectors v_i with components $(v_i)_x = \|a_x P_i |\psi_x^j\rangle\|$. Then we have

$$|W^{j+1} - W^j| \le 2 \sum_{x,y \in S} \sum_{i=1}^{n} (v_i)_x (\Gamma_i)_{xy} (v_i)_y \tag{23}$$

$$= 2 \sum_{i=1}^{n} v_i^\dagger \Gamma_i v_i \tag{24}$$

$$\le 2 \sum_{i=1}^{n} \|\Gamma_i\| \cdot \|v_i\|^2. \tag{25}$$

Finally, since

$$\sum_{i=1}^{n} \|v_i\|^2 = \sum_{i=1}^{n} \sum_{x \in S} \|a_x P_i |\psi_x^j\rangle\|^2 \tag{26}$$

$$\le \sum_{x \in S} a_x^2 \| |\psi_x^j\rangle\|^2 \tag{27}$$

$$= \sum_{x \in S} a_x^2 \tag{28}$$

$$= 1, \tag{29}$$

we have

$$|W^{j+1} - W^j| \le 2 \max_{i \in \{1,\ldots,n\}} \|\Gamma_i\|. \tag{30}$$

Since $W^0 = \|\Gamma\|$, we have

$$W^t \ge \|\Gamma\| - 2t \max_{i \in \{1,\ldots,n\}} \|\Gamma_i\|. \tag{31}$$

Thus, to have $W^t \le 2\sqrt{\epsilon(1-\epsilon)}\|\Gamma\|$, we require

$$t \ge \frac{1 - 2\sqrt{\epsilon(1-\epsilon)}}{2} \operatorname{Adv}(f). \tag{32}$$

where

$$\operatorname{Adv}(f) := \max_{\Gamma} \frac{\|\Gamma\|}{\max_{i \in \{1,\ldots,n\}} \|\Gamma_i\|} \tag{33}$$

with the maximum taken over all adversary matrices Γ.

Example: Unstructured search

As a simple application of this method, we prove the optimality of Grover's algorithm. It is sufficient to consider the problem of distinguishing between the case of no marked items and the case of a unique marked item (in an unkown location). Thus, consider the partial function where S consists of the strings of Hamming weight 0 or 1, and f is the logical OR of the input bits. (Equivalently, we consider the total function OR but only consider adversary matrices with zero weight on strings of Hamming weight more than 1.)

For this problem, adversary matrices have the form

$$\Gamma = \begin{pmatrix} 0 & \gamma_1 & \cdots & \gamma_n \\ \gamma_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_n & 0 & \cdots & 0 \end{pmatrix} \tag{34}$$

for some nonnegative coefficients $\gamma_1, \ldots, \gamma_n$. Symmetry suggests that we should take $\gamma_1 = \cdots = \gamma_n$. This can be formalized, but for the present purposes we can take this as an ansatz.

Setting $\gamma_1 = \cdots = \gamma_n = 1$ (since an overall scale factor does not affect the bound), we have

$$\Gamma^2 = \begin{pmatrix} n & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \cdots & 1 \end{pmatrix} \tag{35}$$

which has norm $\|\Gamma^2\| = n$, and hence $\|\Gamma\| = \sqrt{n}$. We also have

$$\Gamma_1 = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix} \tag{36}$$

and similarly for the other Γ_i, so $\|\Gamma_i\| = 1$. Thus we find $\mathrm{Adv}(\textsc{or}) \geq \sqrt{n}$, and it follows that $Q_\epsilon(\textsc{or}) \geq \frac{1-2\sqrt{\epsilon(1-\epsilon)}}{2} n$. This shows that Grover's algorithm is optimal up to a constant factor (recall that Grover's algorithm finds a unique marked item with probability $1 - o(1)$ in $\frac{\pi}{4}\sqrt{n} + o(1)$ queries).

Other adversaries

The adversary method described above is a generalization of the method originally formulated by Ambainis, which considered only a relation between yes and no inputs and did not allow arbitrary positive weights. More recently, it was realized that one can use negative weights and still obtain a lower bound, and that sometimes this bound can be dramatically better. In fact, it was shown by Reichardt that this bound is essentially tight: up to constant factors, the negative adversary method characterizes quantum query complexity.