Week 2 supplementary note

In the previous week's notes, we've shown that $\mathrm{Q}(f) = \Omega(\sqrt{\mathrm{bs}(f)})$ follows from showing $\mathrm{Q}(\textsc{PromiseOR}_n) = \Omega(\sqrt{n})$, but we did not prove the latter. Let's now attempt to do so.

The approach we'll use was first used by [BBBV97], and is called the *hybrid method.* For an algorithm Q and a fixed input $x \in \{0,1\}^n$, the hybrid method lets us roughly talk about "where the algorithm queried" when Q was run on x. It lets us do this even though a quantum algorithm queries the in superposition, so there is no true answer about where the algorithm queried; still, we will be able to say *something* about the relative query weights that were placed on each bit of the input x.

Before we tackle the hybrid method, let's look at a classical analogue of it, which can be used to lower bound randomized query complexity. This analogue takes the form of the following theorem.

Theorem 1. *Let R be a randomized query algorithm on n bits, and let $x \in \{0,1\}^n$. Let m_i^t be the probability that, when R is run on x, it queries bit i during query number t. Further, let $m_i = \sum_t m_i^t$ be the expected number of times R queries bit i of x before terminating. Suppose $B \subseteq [n]$ is a block such that R distinguishes between x and x^B to error ϵ (that is, $\Pr[R(x) = 1] \geq 1 - \epsilon$ and $\Pr[R(x^B) = 0] \geq 1 - \epsilon$ or vice versa). Then*

$$\sum_{i \in B} m_i \geq 1 - 2\epsilon.$$

Before we prove this theorem, let's see why it's useful. Note that $\sum_{i\in[n]} m_i$ is the expected number of queries R makes on x, so it is at most the worst-case number of queries of R. If R computes f to error $1/3$ using $\mathrm{R}(f)$ queries, then we have $\sum_{i\in[n]} m_i \leq \mathrm{R}(f)$. Also, for each sensitive block B of x, we have $\sum_{i\in B} m_i \geq 1/3$. If there are $\mathrm{bs}(f,x)$ different disjoint sensitive blocks for x, then the sum of the m_i within each block is at least $1/3$, so the sum of the m_i in all the blocks is at least $\mathrm{bs}(f,x)/3$. Picking x to be the input with largest block sensitivity, this shows that $\mathrm{R}(f) \geq \mathrm{bs}(f)/3$. In fact, if we use $\mathrm{R}_\epsilon(f)$ to denote the worst-case randomized query complexity of f to error ϵ (instead of to error $1/3$), then this argument (using Theorem 1) shows that $\mathrm{R}_\epsilon(f) \geq (1-2\epsilon)\,\mathrm{bs}(f)$. In other words, this argument analyzing "where the algorithm looked" for a single block in a single input is enough to prove the block sensitivity lower bound.

OK, let's prove Theorem 1.

Proof. Recall that a randomized query algorithm R is a probability distribution over deterministic decision trees. The sum $\sum_{iinB} m_i$ is the expected number of times a decision tree $D \sim R$ queries some $i \in B$ when it is run on the input x. This expectation is at least the probability p that a tree $D \sim R$ queries some $i \in B$ at least once when run on x. We now lower bound p.

Now, the key observation is that any decision tree which does not query any $i \in B$ must behave identically on x and x^B, and in particular, must output the same answer. That is to say, with probability $1-p$ the algorithm R does not query in B when run on x, so with probability $1-p$ it must also not query in B when run on x^B. Further, if q is the conditional probability that R outputs 1 when run on x given that it did not query in B, then $\Pr[R(x) = 1]$ is at least $(1-p)q$ and

$\Pr[R(x^B)=1]$ is also at least $(1-p)q$. Assume without loss of generality that $q \geq 1/2$ (otherwise, consider the probability $1-q$ that R outputs 0 on x conditioned on not querying in B instead of 1). Then $\Pr[R(x)=1] \geq (1-p)/2$ and $\Pr[R(x^B)=1] \geq (1-p)/2$. However, one of x and x^B is a 0 input, so one of these probabilities must be at most ϵ. Hence $(1-p)/2 \leq \epsilon$, or $p \geq 1-2\epsilon$, as desired. □

Hopefully Theorem 1 is reasonably intuitive: it just says that if an algorithm detects the difference between x and x^B, then it must query a bit inside the set B. Next, we will state the quantum version of this theorem.

Theorem 2. *Let Q be a T-query quantum algorithm on n bits, and let $x \in \{0,1\}^n$. Consider running Q on x until just before query number t, and then measuring the index register (the one specifying the bit $i \in [n]$ to query next). Let m_i^t be the probability that the outcome of this measurement is i. Let $m_i = \sum_t m_i^t$. Suppose $B \subseteq [n]$ is a block such that Q distinguishes between x and x^B to error ϵ (that is, $\Pr[Q(x)=1] \geq 1-\epsilon$ and $\Pr[Q(x^B)=0] \geq 1-\epsilon$ or vice versa). Then*

$$\sum_{i\in B} m_i \geq \frac{(1-2\epsilon)^2}{4T}.$$

Note that just like in the classical case, we have $\sum_i m_i = T$; that is, the sum of all m_i is the total number of queries the algorithm Q makes. If we interpret m_i to be the expected number of times Q queries bit i when run on x, then this theorem is saying that to detect a difference between x and x^B, a quantum algorithm needs to query at least $\Omega(1/T)$ total times instead of the block B, instead of the $\Omega(1)$ required by classical algorithms. That is, the longer the quantum algorithm runs for, the fewer *total* queries it needs to make inside of B to detect whether this block B has been flipped.

Just like in the classical case, we can use Theorem 2 to get a lower bound on $\mathrm{Q}(f)$ in terms of $\mathrm{bs}(f)$. To see this, consider the input x maximizing $\mathrm{bs}(f,x)$, which has $k = \mathrm{bs}(f)$ disjoint sensitive blocks. Fix a T-query quantum algorithm Q computing f to error ϵ, and define m_i relative to that Q and x. Then for each sensitive block B of those k blocks, we have $\sum_{i\in B} m_i \geq (1-2\epsilon)^2/4T$. Since the blocks are disjoint, we get $T = \sum_i m_i \geq k(1-2\epsilon)^2/4T$, or $T^2 \geq \mathrm{bs}(f)(1-2\epsilon)^2/4$. Taking square roots, this is $T \geq (1-2\epsilon)/2 \cdot \sqrt{\mathrm{bs}(f)}$, which means that $\mathrm{Q}_\epsilon(f) \geq (1-2\epsilon)/2 \cdot \sqrt{\mathrm{bs}(f)}$, where $\mathrm{Q}_\epsilon(f)$ denotes the quantum query complexity of f to worst-case error ϵ.

Corollary 3. *Let f be a (possibly partial) Boolean function. Then*

$$\mathrm{Q}_\epsilon(f) \geq \frac{1-2\epsilon}{2}\sqrt{\mathrm{bs}(f)}.$$

Note that this corollary also gives us $\mathrm{Q}(\textsc{PromiseOR}_n) = \Omega(\sqrt{n})$, as we wanted last class. We now prove Theorem 2.

Proof. Recall that a T-query quantum algorithm Q is a sequence of unitary matrices $U_0, U_1, \ldots, U_T$. The behavior of Q on an input x is by applying

$$U_T U^x U_{T-1} U^x \ldots U^x U_0 \left|\psi_{init}\right\rangle,$$

where $\left|\psi_{init}\right\rangle = \left|0\right\rangle_O \left|0\right\rangle_W \left|1\right\rangle_I \left|0\right\rangle_B$. Let $\left|\psi_1^x\right\rangle := U_0\left|\psi_{init}\right\rangle$, and for each $t \in \{1,2,\ldots,T\}$ define $\psi_{t+1}^x := U_t U^x \left|\psi_t^x\right\rangle$. Then $\left|\psi_t^x\right\rangle$ is the state of the algorithm Q when run on x right before query t is made, and $\left|\psi_{T+1}^x\right\rangle$ is the final state of the algorithm before being measured.

Let Π_i be the projector that maps $|\psi\rangle$ on registers O, W, I, B to its component where register I contains $|i\rangle$. That is, Π_i is the matrix $I_O \otimes I_W \otimes |i\rangle\langle i|_I \otimes I_B$, where I_O, I_W, and I_B are the identity matrices on registers O, W, and B respectively, and $\otimes$ denotes the Kronecker (tensor) product of the matrices. Then by definition, $m_i^t = \|\Pi_i \, |\psi_t^x\rangle\,\|^2$, where $\|\cdot\|$ denotes the 2-norm.

Note that $|\psi_t^x\rangle$ is a quantum state, and therefore a unit vector. We will examine the squared distance between $|\psi_t^x\rangle$ and $|\psi_t^{x^B}\rangle$, that is, the states of the algorithm just before query t is made when run on x vs. when run on x^B. We have

$$\begin{aligned}\left\| |\psi_t^x\rangle - |\psi_t^{x^B}\rangle \right\|^2 &= \left(\langle\psi_t^x| - \langle\psi_t^{x^B}|\right)\left(|\psi_t^x\rangle - |\psi_t^{x^B}\rangle\right)\\ &= \langle\psi_t^x|\psi_t^x\rangle + \langle\psi_t^{x^B}|\psi_t^{x^B}\rangle - \langle\psi_t^x|\psi_t^{x^B}\rangle - \langle\psi_t^{x^B}|\psi_t^x\rangle\\ &= 2 - 2\Re\left(\langle\psi_t^x|\psi_t^{x^B}\rangle\right),\end{aligned}$$

where $\Re(\cdot)$ denotes the real part of a complex number. This identity will come in handy.

The idea will be that this distance must be small at the beginning of the algorithm and large at the end, and can only change when the algorithm "queries inside" the block B where x and x^B differ. In this case where $t = 1$, we have $|\psi_1^x\rangle = |\psi_1^{x^B}\rangle$, since these both equal $U_0\,|\psi_{init}\rangle$. Hence the distance between them is 0. Also, in the case where $t = T+1$, the vectors $|\psi_{T+1}^x\rangle$ and $|\psi_{T+1}^{x^B}\rangle$ are the final states of the quantum algorithm when run on x and on x^B respectively. Now, write $|\psi_{T+1}^x\rangle = |\phi_0^x\rangle + |\phi_1^x\rangle$, where $|\phi_0^x\rangle$ and $|\phi_1^x\rangle$ are the orthogonal vectors corresponding to the output register being 0 and 1 respectively. Note that the norm of $|\phi_0^x\rangle$ and of $|\phi_1^x\rangle$ sum to 1, since they are orthogonal components of a unit vector. Then we have

$$\langle\psi_{T+1}^x|\psi_{T+1}^{x^B}\rangle = \big(\langle\phi_0^x| + \langle\phi_1^x|\big)\big(|\phi_0^{x^B}\rangle + |\phi_1^{x^B}\rangle\big) = \langle\phi_0^x|\phi_0^{x^B}\rangle + \langle\phi_1^x|\phi_1^{x^B}\rangle\,,$$

since all the vectors $|\phi_0^z\rangle$ are orthogonal to all the vectors $|\phi_1^z\rangle$. Using Cauchy-Schwartz, we get

$$|\,\langle\psi_{T+1}^x|\psi_{T+1}^{x^B}\rangle\,| \le \|\phi_0^x\| \cdot \|\phi_0^{x^B}\| + \|\phi_1^x\| \cdot \|\phi_1^{x^B}\| = ab + (1-a)(1-b),$$

where $a = \|\phi_0^x\|$ and $b = \|\phi_0^{x^B}\|$. Note that a^2 is the probability that the quantum algorithm outputs 0 on input x, which is at most ϵ; on the other hand, b^2 is the probability that the quantum algorithm outputs 0 on x^B, which is at least $1-\epsilon$. It is not hard to see that the maximum possible value of $ab + (1-a)(1-b)$ under these constraints is $2\sqrt{\epsilon(1-\epsilon)}$. This gives

$$\left\| |\psi_{T+1}^x\rangle - |\psi_{T+1}^{x^B}\rangle \right\|^2 \ge 2 - 4\sqrt{\epsilon(1-\epsilon)}.$$

We now write

$$\begin{aligned}
\sqrt{2-4\sqrt{\epsilon(1-\epsilon)}} &\le \left\| |\psi^x_{T+1}\rangle - |\psi^{x^B}_{T+1}\rangle \right\| \\
&= \sum_{t=1}^{T} \left\| |\psi^x_{t+1}\rangle - |\psi^{x^B}_{t+1}\rangle \right\| - \left\| |\psi^x_t\rangle - |\psi^{x^B}_t\rangle \right\| \\
&= \sum_{t=1}^{T} \left\| U_t U^x |\psi^x_t\rangle - U_t U^{x^B} |\psi^{x^B}_t\rangle \right\| - \left\| |\psi^x_t\rangle - |\psi^{x^B}_t\rangle \right\| \\
&= \sum_{t=1}^{T} \left\| (U^{x^B})^\dagger U^x |\psi^x_t\rangle - |\psi^{x^B}_t\rangle \right\| - \left\| |\psi^x_t\rangle - |\psi^{x^B}_t\rangle \right\| \\
&\le \sum_{t=1}^{T} \left\| (U^{x^B})^\dagger U^x |\psi^x_t\rangle - |\psi^{x^B}_t\rangle - |\psi^x_t\rangle + |\psi^{x^B}_t\rangle \right\| \\
&= \sum_{t=1}^{T} \left\| \left(U^x - U^{B^x}\right) |\psi^x_t\rangle \right\|.
\end{aligned}$$

Here the fourth line followed by using the fact that the unitary matrices preserve the 2-norm, and applying $(U^{x^B})^\dagger U_t^\dagger$ inside the norm (recall that $U^\dagger = U^{-1}$ for a unitary matrix U). The fifth line followed using triangle inequality. The last line followed by multiplying the unitary U^{x^B} inside the norm.

Next, we note that $|\psi^x_t\rangle = \sum_{i=1}^n \Pi_i |\psi^x_t\rangle$. Furthermore, the matrix $U^x - U^{x^B}$ maps any vector that has i in the index register to 0 if $i \notin B$; this is because for $i \notin B$, we U^x and U^{x^B} behave the same on such a vector. We can therefore continue our inequality chain:

$$\begin{aligned}
&= \sum_{t=1}^{T} \left\| \left(U^x - U^{x^B}\right) \sum_{i=1}^{n} \Pi_i |\psi^x_t\rangle \right\| \\
&= \sum_{t=1}^{T} \left\| \sum_{i=1}^{n} \left(U^x - U^{x^B}\right) \Pi_i |\psi^x_t\rangle \right\| \\
&= \sum_{t=1}^{T} \left\| \sum_{i\in B} \left(U^x - U^{x^B}\right) \Pi_i |\psi^x_t\rangle \right\| \\
&= \sum_{t=1}^{T} \sqrt{\sum_{i\in B} \left\| \left(U^x - U^{x^B}\right) \Pi_i |\psi^x_t\rangle \right\|^2} \\
&\le \sum_{t=1}^{T} \sqrt{\sum_{i\in B} 4m_i^t} \\
&\le \sqrt{T \sum_{t=1}^{T} \sum_{i\in B} 4m_i^t} \\
&= \sqrt{4T \sum_{i\in B} m_i}.
\end{aligned}$$

Several lines here require explanation. The fourth line follows because each of the vectors $\big(U^x - U^{x^B}\big)\Pi_i |\psi^x_t\rangle$ for different values of i are all orthogonal. The fifth line follows because, by triangle inequality,

$$\left\|\left(U^x - U^{x^B}\right)\Pi_i \left|\psi_t^x\right\rangle\right\| \leq \|U^x \Pi_i \left|\psi_t^x\right\rangle\| + \|U^{x^B}\Pi_i \left|\psi_t^x\right\rangle\| = 2\|\Pi_i \left|\psi_t^x\right\rangle\| = 2\sqrt{m_i^t}.$$

The sixth line follows by Cauchy-Schwartz on the sum over t, and the last line by rearranging the sums and by the definition of m_i.

We therefore conclude

$$\sum_{i \in B} m_i \geq \frac{1 - 2\sqrt{\epsilon(1-\epsilon)}}{2T}.$$

To finish the proof, we note that

$$2\sqrt{\epsilon(1-\epsilon)} = \sqrt{1-(1-2\epsilon)^2} \leq 1 - (1-2\epsilon)^2/2,$$

from which we get $1 - 2\sqrt{\epsilon(1-\epsilon)} \geq (1-2\epsilon)^2/2$. □

References

[BBBV97] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and Weaknesses of Quantum Computing. *SIAM Journal on Computing* (5 1997). DOI: 10.1137/S0097539796300933. arXiv: quant-ph/9701001 (p. 1).