CS 593/MA 592 - Intro to Quantum Computing
Spring 2024
Thursday, February 15 - Lecture 6.2

Today's scribe: Reed Phillips

Reading: Chapter 6 of Nielsen and Chuang, second half.

Agenda:

1. Grover's algorithm in general
2. Quantum lower bounds

1 Grover's Algorithm in General

In the previous lecture, we covered Grover's algorithm for cases where the function $f : \{0,1\}^n \to \{0,1\}$ has a single solution, i.e. $\#f^{-1}(1) = 1$. In this lecture, we extend it to cases where there is more than one solution and provide a general version for where the number of solutions is unknown.

1.1 Known number of solutions m

Suppose we know that $\#f^{-1}(1) = m$, for some $m \geq 1$. The same idea as the $m = 1$ case will still apply: each application of the Grover operator $R_{|\psi\rangle}R_f$ will still rotate the state closer to the subspace of solutions. We just have to modify how many times we apply $R_{|\psi\rangle}R_f$ before measuring.

Starting from

$$|\psi\rangle = \frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} |x\rangle,$$

we write $|\psi\rangle$ as a sum of the "good" part $|W\rangle$ and the "bad" part $|W^\perp\rangle$:

$$|\psi\rangle = \underbrace{\left(\sum_{x:f(x)=1} \frac{1}{2^{n/2}} |x\rangle \right)}_{|W\rangle} + \underbrace{\left(\sum_{x:f(x)=0} \frac{1}{2^{n/2}} |x\rangle \right)}_{|W^\perp\rangle}$$

Let $\theta_0/2$ be the angle between $|\psi\rangle$ and $|W^\perp\rangle$. We can compute $\sin(\theta_0/2)$ by computing the cosine of the complementary angle between $|\psi\rangle$ and $|W\rangle$ via the inner product:

$$\begin{aligned}
\sin(\theta_0/2) &= \frac{\langle \psi | W \rangle}{\||\psi\rangle\| \cdot \||W\rangle\|} \\
&= \frac{\sum_{x:f(x)=1} \frac{1}{2^n}}{1 \cdot \sqrt{\sum_{x:f(x)=1} \left(\frac{1}{2^{n/2}}\right)^2}} \\
&= \frac{m/2^n}{\sqrt{m}/2^{n/2}} \\
&= \frac{\sqrt{m}}{2^{n/2}}
\end{aligned}$$

By the same Taylor series idea as the $m = 1$ case, we can conclude that $\theta_0/2 \approx \sqrt{m}/2^{n/2}$. To rotate the state close enough to $|W\rangle$ to have a high probability of success (that is, a high probability of seeing an x with $f(x) = 1$ when we measure in the computational basis), we need $\Theta(1/\theta_0) = \Theta(2^{n/2}/\sqrt{m})$ applications of the Grover operator.

1.2 Unknown number of solutions

Suppose we don't know the number of solutions $\#f^{-1}(1) = m$ ahead of time. Ideally, we'd still be able to get a runtime of $O(2^{n/2}/\sqrt{m})$, or at the very least $O(2^{n/2})$.

How should we proceed if m is unknown? The reason knowing m is useful is that it tells us how many times to apply the Grover operator before measuring; specifically, the starting angle from $|W^\perp\rangle$ is $\theta_0(m)/2 = \sin^{-1}(\sqrt{m}/2^{n/2})$, and each application of the Grover operator rotates $\theta_0(m)$ further from $|W^\perp\rangle$. Optimally, we could rotate the state until it lined up (nearly) perfectly with $|W\rangle$, and then be near-guaranteed to read off a solution when measuring. Denote the version of Grover's algorithm optimized for a particular m by GS_m.

Suppose we didn't know m exactly, but knew that $2^i \leq m \leq 2^{i+1}$ for some i. If we run $GS_{2^{i+1}}$, then we might have under-rotated if m was closer to 2^i. But by how much? Assuming m is small enough that we can reasonably make the approximation $\theta_0(m)/2 = \sqrt{m}/2^{n/2}$, we can approximate $\theta_0(2^{i+1}) = 2 \cdot 2^{(i+1-n)/2}$. Compared to $\theta_0(2^{i+1})$:

$$\begin{aligned}\theta_0(2^i) &= 2 \cdot 2^{(i-n)/2} \\ &= 2 \cdot 2^{(i+1-n)/2} \cdot 2^{-1/2} \\ &= \theta_0(2^{i+1}) \cdot \frac{1}{\sqrt{2}}\end{aligned}$$

The circuit $GS_{2^{i+1}}$ applies the Grover operator just enough times that, for $m = 2^{i+1}$, the state rotates by $\pi/2$. If m is more like 2^i, then applying the Grover operator that many times only rotates the state by $\frac{\pi}{2\sqrt{2}}$. However, this still isn't too bad! If the angle between the final state and $|W\rangle^\perp$ is $\frac{\pi}{2\sqrt{2}}$, then by the Born rule the chance we measure something that's not a solution is only $\cos^2\left(\frac{\pi}{2\sqrt{2}}\right) \approx 0.197$. That means we've still got roughly a $4/5$ chance of measuring a solution.

This argument shows that if we want to make Grover's algorithm work for an unknown m, then we only have to know some i such that $2^i \leq m \leq 2^{i+1}$. But, since we know $m \leq 2^n$, there aren't very many possible values of i: just the values 0 through $n-1$. We can just try all of them! Even better, since the runtime of GS_{2^i} is $O(\sqrt{2^n/2^i}) = O(\sqrt{2}^{n-i})$, the total runtime of these guesses is a geometric series:

$$\sqrt{2}^{n-0} + \sqrt{2}^{n-1} + \ldots + \sqrt{2}^{n-(n-1)} = \frac{\sqrt{2}^{n+1} - \sqrt{2}}{\sqrt{2}-1}$$

Which is still $O(\sqrt{2^n})$.

There's also the possibility that $m = 0$: there are no solutions. We can detect this by running the total algorithm a few times; since when $m > 0$, each time has at least a $4/5$ chance[1] of returning a solution when one exists, having it fail to return one many times in a row is strong evidence that no solutions exist.

1.3 Counting the number of solutions

This can also be done in $O(\sqrt{2^n})$ time, but we need phase estimation and the quantum Fourier transform to do it. More on this later.

2 Quantum Lower Bounds

We know that Grover's algorithm can solve unstructured search in $O(\sqrt{2^n})$ time. Is there anything better? It turns out that there isn't, but proving so rigorously is outside the scope of this lecture. For reference, Nielsen and Chuang provide a rigorous proof of the optimality of Grover search via the *polynomial method*.

[1] The $4/5$ chance was derived assuming m was small. It might not be, but if m is large there are easier ways to solve the problem. For instance, just compute $f(w)$ classically for a bunch of random w; if m is anywhere near large enough for the $4/5$ to not be a good estimate, m is large enough that you'll probably find a solution by just guessing.

Since their book was published, better techniques for proving lower bounds have been developed. The main tool is the *quantum adversary method*, introduced by Ambainis in 2000 and developed further in the years following. In fact, the best version of the quantum adversary method not only shows a lower bound on the number of queries needed to solve a given problem, but also provides an algorithm for meeting that lower bound.

We won't be using the full version of the quantum adversary method or proving that it works. Instead, we'll just provide the perspective from which it operates, and show that it can be used to show that Grover's algorithm is optimal.

So far, we've been given a function $f : \{0,1\}^n \to \{0,1\}$ in the form of an oracle $R_f : |x\rangle \mapsto (-1)^{f(x)}|x\rangle$. The perspective shift is to instead view the function as a bistring $S \in \{0,1\}^{2^n}$, and view our queries as revealing bits in that bitstring. For example, take the *AND* function, which takes in two bits and only returns 1 if both input bits are 1. We can represent it with the following truth table:

x_1	x_2	$x_1 \wedge x_2$
0	0	0
0	1	0
1	0	0
1	1	1

The corresponding bitstring S is just the third column of this table.

We can then phrase an arbitrary decision problem $L : \{0,1\}^{2^n} \to \{0,1\}$ as a function that says whether any particular S is a "yes" or "no" instance. For example, we could take L to be the *OR* function. Then the bitstring $000\ldots00$ would be the only "no" instance ($L(000\ldots00) = 0$), and all other bistrings would be "yes" instances ($L(x) = 1$ for $x \neq 000\ldots00$). In fact, this particular L is the "Grover decision problem", since being able to decide L means deciding whether or not $f(w) = 1$ has any solutions.

For the "basic" version of the quantum adversary method we'll be using, we will define the sets $YES_L = \{x : L(x) = 1\}$ and $NO_L = \{x : L(x) = 0\}$. We will also define the Hamming distance between two bitstrings x and y, $dist_{Hamm}(x,y)$, as the number of bits they differ on. The method is as follows:

Theorem 1 ("Super-basic adversary method," after Ryan McDonnell's lecture notes)**.** *Let L be a decision problem. Let $Y \subseteq YES_L$ and $Z \subseteq NO_L$ be arbitrary subsets. Then, if there exist m, m' such that:*

(i) For all $y \in Y$, there are at least m distinct values $z \in Z$ such that $dist_{Hamm}(y,z) = 1$

(ii) For all $z \in Z$, there are at least m' distinct values $y \in Y$ such that $dist_{Hamm}(y,z) = 1$

Then at least $\Omega(\sqrt{m \cdot m'})$ queries are necessary to decide L with high probability.

To apply this theorem to Grover search, let $L = OR$. Take $Z = \{0\}$ and $Y = \{x : x \text{ has exactly one bit which is } 1\}$. Then we can use $m = 1$ and $m' = 2^n$, and we get that at least $\Omega(\sqrt{2^n})$ queries are required. Grover's algorithm can decide L with $O(\sqrt{2^n})$ queries, so it is optimal.