CS191 – Fall 2014
Lecture 19: Grover's Algorithm

Birgitta Whaley
(Dated: Friday October 31, 2014)

The problem is to search for an item in an unstructured database. For example, suppose you are given a telephone number in LA and need to find out who it belongs to. You will have to go through all the phone numbers and check the names of the registered owners in each case... Searching an item in an unsorted database with size N costs a classical computer $O(N)$ running time, since on average $N/2$ entries need to be checked.

Can a quantum computer search for a needle in a haystack much more efficiently than its classical counterpart? Grover, in 1996, affirmatively answered this question by proposing a search algorithm that consults the database only $O(\sqrt{N})$ times. In contrast to algorithms based on the quantum Fourier transform, with exponential speedups, the search algorithm only provides a quadratic improvement. However, the algorithm is quite important because it has broad applications, and because the same technique can in principle be used to improve solutions of NP-complete problems.

One might think of having better improvements over the search algorithm. However, it turns out that Grover's search algorithm is optimal. At least $\Omega(\sqrt{N})$ queries are needed to solve the problem.

Grover's algorithm uses parallelism and amplitude amplification. The latter is nicely illustrated by a geometrical analysis of the quantum search algorithm that we will develop here.

I. THE QUANTUM ORACLE

Here's the search problem: You are given a boolean function $f : \{1, \ldots, N\} \to \{0, 1\}$, and are promised that for exactly one $a \in \{1, \ldots, N\}$, $f(a) = 1$. Think of this as a table of size N, where exactly one element has value 1, and all the others are 0. f is effectively an oracle that can check/recognize the solution when this is given it as input. So f acts like a detector of the target solution. In the current analysis we shall assume that there is only 1 solution, but the arguments can be generalized to a finite number of solutions.

We construct a two register state, with the database register first and the oracle register second. Then our oracle acts as follows (cf. the Deutsch-Jozsa algorithm)

$$O|x\rangle|q\rangle = |x\rangle|q \oplus f(x)\rangle,$$

where we assume f can be computed classically in polynomial time. Then we can also apply the oracle with

1. the database register in superposition:

$$\sum_x \alpha_x |x\rangle|0\rangle \to \sum_x \alpha_x |x\rangle|f(x)\rangle$$

and

2. with the oracle register in superposition:

$$\begin{aligned}
\sum_x \alpha_x |x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) &\mapsto \sum_x \alpha_x \left(\frac{|x\rangle|f(x)\rangle - |x\rangle\left|\overline{f(x)}\right\rangle}{\sqrt{2}}\right) \\
&= \sum_x \alpha_x |x\rangle \left(\frac{|f(x)\rangle - \left|\overline{f(x)}\right\rangle}{\sqrt{2}}\right) \\
&= \sum_x \alpha_x |x\rangle (-1)^{f(x)} \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)
\end{aligned}$$

Here $\overline{f(x)}$ means the binary complement of $f(x)$, i.e., if $f(x) = 1$, then $\overline{f(x)} = 0$. Note that we have used the same phase kick-back as in Deutsch-Jozsa to go from lines 2 to 3. (Check: if $f(x) = 0$ the oracle qubit is $|0\rangle - |1\rangle = (-1)^{f(0)}(|0\rangle - |1\rangle)$, while if $f(x) = 1$ the oracle qubit is $|1\rangle - |0\rangle = (-1)^{f(1)}(|0\rangle - |1\rangle)$.)

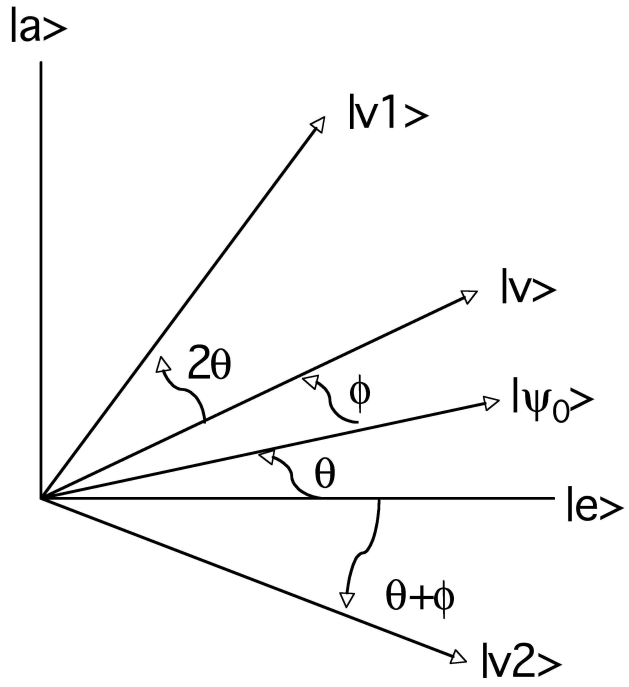

FIG. 1: To rotate $|v\rangle$ by 2θ to $|v1\rangle$, we reflect around $|e\rangle$, reaching $|v2\rangle$, and then reflect around $|\psi_0\rangle$.

So the oracle marks the solutions to the search problem by a minus sign, i.e., by a phase (note, no measurement). While it is quite sloppy, we will follow many authors and summarize the action of the oracle on an arbitrary state

$$|\nu\rangle = \sum_x \alpha_x |x\rangle$$

as

$$O|\nu\rangle \equiv \sum_x (-1)^{f(x)} \alpha_x |x\rangle,$$

where we have omitted the oracle register on both sides of the equation because it is in the same state. But don't forget that the oracle register is essential for this action of O on the database register to be realized!

II. GEOMETRIC ANALYSIS OF SEARCH

Grover's algorithm finds a in $O(\sqrt{N})$ steps. Consider the two dimensional subspace that consists of two states: $|a\rangle$ and the uniform superposition $|\psi_0\rangle = \sum_x \frac{1}{\sqrt{N}}|x\rangle$. Let θ be the angle between $|\psi_0\rangle$ and $|e\rangle$, where $|e\rangle$ is the vector that is orthogonal to $|a\rangle$ (in the direction of $|\psi_0\rangle$) in this subspace. See Figure 1.

$|a\rangle$ is the target and we can regard $|\psi_0\rangle$ as the least biased initial state. So we want to increase θ to go from $|\psi_0\rangle$ to $|a\rangle$. How do we accomplish this?

One way to rotate a vector is to make two reflections. In particular, we can rotate a vector $|v\rangle$ by 2θ to the new vector $|v1\rangle$ by first reflecting about $|e\rangle$ and then reflecting about $|\psi_0\rangle$. This transformation is also illustrated in Figure 1. The first reflection transforms an arbitrary vector $|v\rangle$ to $|v2\rangle$ and the second reflection transforms $|v2\rangle$ to $|v1\rangle$.

Each step of our algorithm is thus a rotation by 2θ from the previous location $|v\rangle$. This means that we need at most $\frac{\pi/2}{2\theta}$ iterations for the algorithm to complete. Now, what is θ?

$$\langle\psi_0|a\rangle = \cos(\pi/2 - \theta) = \sin(\theta)$$

but

$$\langle\psi_0|a\rangle = \frac{1}{\sqrt{N}} \sum_x \langle x|\, a\rangle = \frac{1}{\sqrt{N}} \delta_{xa} = \frac{1}{\sqrt{N}}$$

Then since $\sin\theta \approx \theta$, we know that $\theta \approx \frac{1}{\sqrt{N}}$. Thus, we need $O(\frac{\pi/2}{2/\sqrt{(N)}}) = O(\sqrt{N})$ iterations for the algorithm to complete. In the end, we get very close to $|a\rangle$, and then with high probability, a measurement of the state in the computational basis (all qubits) yields a.

Note that one must not iterate beyond this point. Subsequent iterations will rotate the vector $|v\rangle$ away from $|a\rangle$ again. For large N, we need to iterate $r = \pi\sqrt{N}/4$ times and the corresponding probability of error is $O(1-\cos^2\theta) = O(\sin^2\theta) = O(N^{-1})$.

How do we implement the two reflections?

1. Reflection about $|e\rangle$ is easy. $|e\rangle$ is the vector orthogonal to $|a\rangle$ so all we need to do is flip the phase of the component of the database wavefunction in the direction of $|a\rangle$, i.e., we send any component $|a\rangle$ to $-|a\rangle$ and leave all other components as is. To accomplish this, we just act with the oracle:

$$\begin{aligned} O|v\rangle &= \sum_x (-1)^{f(x)}\alpha_x|x\rangle \\ &= \sum_{x\neq a}\alpha_x|x\rangle - \alpha_a|a\rangle \\ &= \sum_x \alpha_x|x\rangle - 2\alpha_a|a\rangle \\ \Rightarrow \hat{O}_a &= \hat{I} - 2|a\rangle\langle a| = R_{|e\rangle} \end{aligned}$$

   The notation $R_{|e\rangle}$ means a reflection about the vector $|e\rangle$.

2. What about the reflection about $|\psi_0\rangle$? This uniform superposition state is just the zero vector from the computational basis transformed to the Hadamard basis.So we first apply H_{2^n}, which maps $|\psi_0\rangle \mapsto |00\ldots0\rangle$, then reflect around $|00\ldots0\rangle$, and finally, apply H_{2^n} to return to the original basis. The reflection about the zero vector in the computation basis can easily be seen to be given by

$$-I + 2|0\rangle\langle 0|$$

   using similar analysis to that above for reflection about $|e\rangle$. The overall reflection about $|\psi_0\rangle$ is then given by the product of the three transformations:

$$\begin{aligned} D &= (H^{\overrightarrow{n}})^\dagger[-I + 2|0\rangle\langle 0|]H^{\overrightarrow{n}} \\ &= -I + 2|\psi_0\rangle\langle\psi_0| \\ &= -R_{|\psi_0\rangle} \end{aligned}$$

   where we have used the fact that H is self-adjoint ($H = H^\dagger$) and that $|\psi_0\rangle = H^{\overrightarrow{n}}|0\rangle$. This operator is referred to as the 'diffusion operator' for reasons that will be explained in the next lecture.

To make one iteration step we combine the two reflections, yielding the Grover operator

$$G = DO_a = -R_{|\psi_0\rangle}R_{|e\rangle}.$$

We apply this Grover operator $O(\sqrt{N})$ times to rotate from $|\psi_0\rangle$ to $|a\rangle$. Notice the difference between the two reflections which arises from the fact that we have written one in terms of the vector about which we reflect ($R_{|\psi_0\rangle}$) and the other in terms of the vector perpendicular to the vector about which we reflect ($R_{|a\rangle}$).

What about efficiency of implementation? Observe that D is expressed as the product of three unitary matrices: two Hadamard matrices separated by a conditional phase shift matrix (minus one times a diagonal matrix with values everywhere one except for the uniform superposition state when the value gains a relative phase shift $e^{i\pi}$).

Therefore, D is also unitary. Regarding the implementation, note that both the Hadamard and the conditional phase shift transforms can be efficiently realized within $O(n)$ gates.

This geometric analysis of the algorithm as rotating from the uniform superposition state $|\Psi_0\rangle$ to the target state $|a\rangle$, at each iteration we are increasing the amount of the target state $|a\rangle$ in the evolving superposition state $|\nu\rangle$. We refer to this as "amplitude amplification" and will discuss it a little more in the next lecture.

III. REFERENCES AND FURTHER READING

1. Benenti et al., Ch. 3.10

2. Kaye et al., Ch. 8.1

3. Nielsen and Chuang, Quantum Computation and Quantum Information, Ch. 6

4. Literature: Grover, quant-ph/9605043, quant-ph/9706033; diffusion transform and other motivations from physics, quant-ph/0109116