CS 294-2
Bomb
Fall 2004

Grover's Algorithm + Quantum Zeno Effect + Vaidman

10/12/04
Lecture 11

Grover's algorithm

Recall that Grover's algorithm for searching over a space of size N works as follows: consider the two dimensional subspace that consists of two states: the target state $|a\rangle$ and $|\psi_0\rangle = \sum_x \frac{1}{\sqrt{N}}|x\rangle$. Start with the state $|\psi_0\rangle$, and repeat for $O(\sqrt{(}N)$ steps: reflect about $|a\rangle$ and then reflect about $|\psi_0\rangle$.

To implement the reflection about $|\psi_0\rangle$, we use the Diffusion operator D (assume $N = 2^n$), which works as follows. First, apply H_{2^n}, which maps $|\psi_0\rangle \mapsto |00\ldots0\rangle$. Then reflect around $|00\ldots0\rangle$. Finally, apply H_{2^n} to return to the original basis. (Note that this is simply a reflection around the zero vector in the Hadamard basis.)

Let's write out this diffusion operator explicitly to give another way of understanding Grover's algorithm:

Claim: The Diffusion operator D has two properties:

1. It is unitary and can be efficiently realized.
2. It can be seen as an "inversion about the mean."

Proof:

1. For $N = 2^n$, D can be decomposed and rewritten as:

$$\begin{aligned}
D &= H_N \begin{pmatrix} -1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} H_N \\
&= H_N \left(\begin{pmatrix} -2 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} + I \right) H_N \\
&= H_N \begin{pmatrix} -2 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} H_N + I \\
&= \begin{pmatrix} -2/N & -2/N & \cdots & -2/N \\ -2/N & -2/N & \cdots & -2/N \\ \vdots & \vdots & \ddots & \vdots \\ -2/N & -2/N & \cdots & -2/N \end{pmatrix} + I
\end{aligned}$$

$$= \begin{pmatrix} -2/N+1 & -2/N & \cdots & -2/N \\ -2/N & -2/N+1 & \cdots & -2/N \\ \vdots & \vdots & \ddots & \vdots \\ -2/N & -2/N & \cdots & -2/N+1 \end{pmatrix}$$

Observe that D is expressed as the product of three unitary matrices (two Hadamard matrices separated by a conditional phase shift matrix). Therefore, D is also unitary. Regarding the implementation, both the Hadamard and the conditional phase shift transforms can be efficiently realized within $O(n)$ gates.

2. Consider D operating on a vector $|\alpha\rangle$ to generate another vector $|\beta\rangle$:

$$D \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_i \\ \vdots \\ \alpha_N \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_i \\ \vdots \\ \beta_N \end{pmatrix}$$

If we let μ be the mean amplitude, then the expression $2\mu - \alpha_i$ describes a reflection of α_i about the mean. Thus, the amplitude of $\beta_i = -\frac{2}{N}\sum_j \alpha_j + \alpha_i = -2\mu + \alpha_i$ can be considered an "inversion about the mean" with respect to α_i.

■

The quantum search algorithm iteratively improves the probability of measuring a solution. Here's how:

1. Start state is $|\psi_0\rangle = \sum_x \frac{1}{\sqrt{N}}|x\rangle$
2. Invert the phase of $|a\rangle$ using f
3. Then invert about the mean using D
4. Repeat steps 2 and 3 $O(\sqrt{N})$ times, so in each iteration α_a increases by $\frac{2}{\sqrt{N}}$

This process is illustrated in Figure 0.1.

Suppose we just want to find a with probability $\frac{1}{2}$. Until this point, the rest of the basis vectors will have amplitude at least $\frac{1}{\sqrt{2N}}$. In each iteration of the algorithm, α_a increases by at least $\frac{2}{\sqrt{2N}} = \sqrt{\frac{2}{N}}$. Eventually, $\alpha_a = \frac{1}{\sqrt{2}}$. The number of iterations to get to this α_a is $\leq \sqrt{N}$.

More applications Grover's algorithm is often called a "database" search algorithm, where you can query in superposition. Other things you can do with a similar approach:

1. Find the minimum.
2. Approximately count elements, or generate random ones.
3. Speed up the collision problem.
4. Speed up the test for matrix multiplication. In this problem we are given three matrices, A, B, and C, and are told that the product of the first two equals the third. We wish to verify that this is indeed true. An efficient (randomized) way of doing this is picking a random array r, and checking to see whether $Cr = ABr = A(Br)$. Classically, we can do the check in $O(n^2)$ time, but using a similar approach to Grover's algorithm we can speed it up to $O(n^{1.75})$ time.

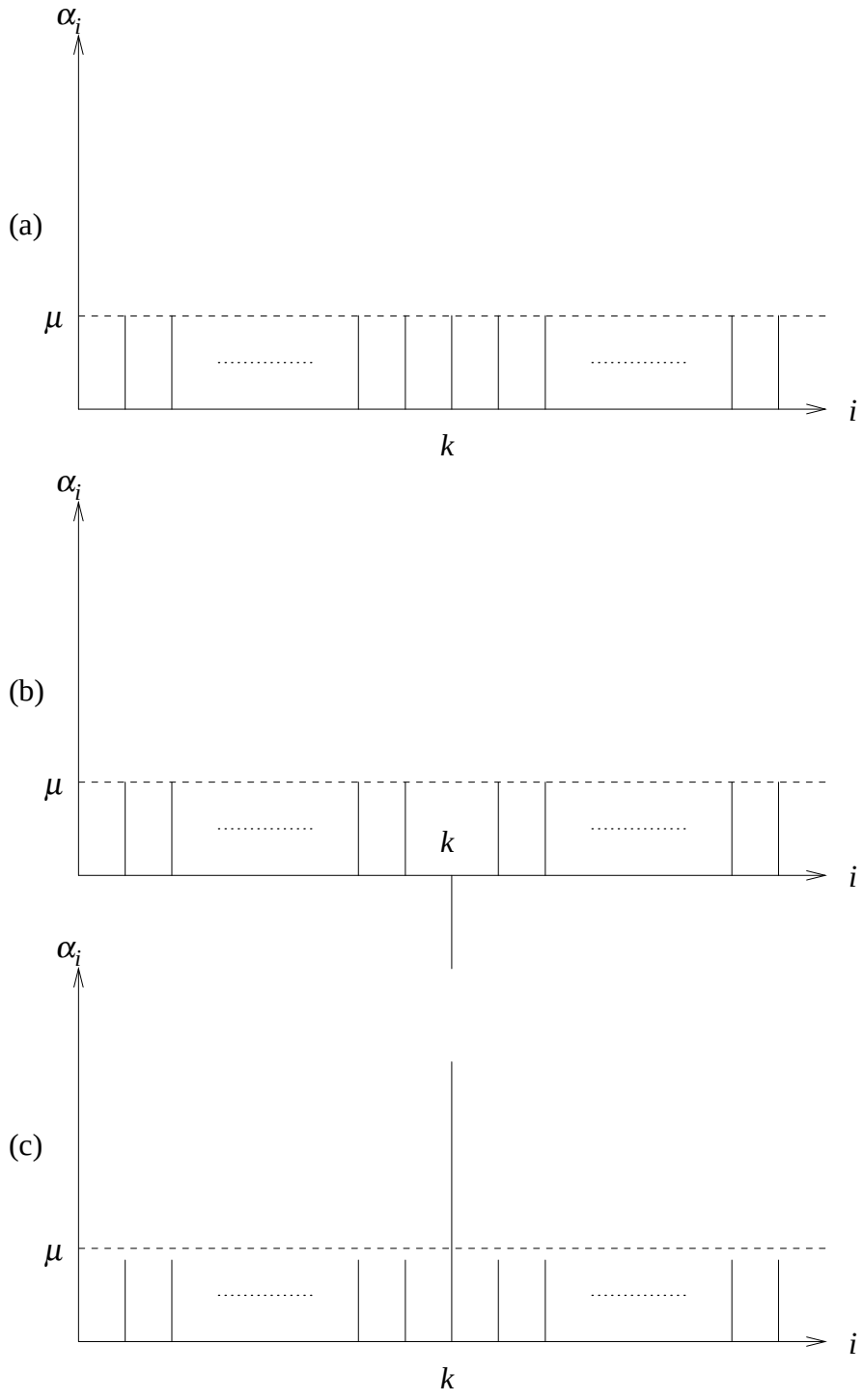

Figure 0.1: The first three steps of Grover's algorithm. We start with a uniform superposition of all basis vectors in (a). In (b), we have used the function f to invert the phase of α_k. After running the diffusion operator D, we amplify α_k while decreasing all other amplitudes.

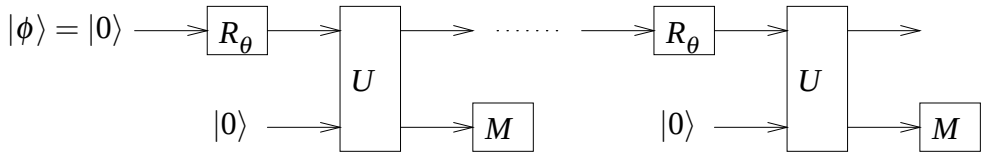

Figure 0.2: This figure shows the circuit used for finding a bomb. There are N steps, and in each step we rotate the control bit and run U.

Vaidman's Bomb

To illustrate some of the concepts behind Grover's algorithm, we'll briefly consider a problem known as Vaidman's bomb. In this problem, we have a package that may or may not contain a bomb. However, the bomb is so sensitive that simply looking to see if the bomb exists will cause it to explode. So, can we determine whether the package contains a bomb without setting it off? Paradoxically, quantum mechanics says that we can. In particular, we will demonstrate that there is a sequence of N measurements such that if the package contains a bomb, we will look with probability $1/N$, and if the package does not contain a bomb, we will look with certainty.

The Quantum Zeno Effect

To achieve this goal, we'll take advantage of a phenomenon known as the Quantum Zeno Effect (also referred to as the "watched pot" or the "watchdog" effect). Consider a quantum state consisting of a single qubit. This qubit starts at $|0\rangle$, and at every step we will rotate it toward $|1\rangle$ by $\theta = \pi/2N$. After one rotation, we have $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$, where $\beta = \sin\theta \approx 1/N$. After N steps, the state will be $|1\rangle$, so any measurement will return $|1\rangle$ with high probability.

Now what if we decide to measure the state after each rotation? After the first rotation, we will measure $|0\rangle$ with high probability, *but this measurement collapses the state back to* $|0\rangle$. Thus, each measurement has a high probability of yielding $|0\rangle$; the probability of getting $|1\rangle$ by the end is approximately $N\frac{1}{N^2} = \frac{1}{N}$, as opposed to the extremely high probability in the previous case.

Essentially, the Quantum Zeno Effect says that if we have a quantum state that is in transition toward a different state, making frequent measurements can delay that transition by repeatedly collapsing the qubit back to its original state.

Looking for the Bomb

To determine whether Vaidman's bomb exists without actually looking at it, we want to take advantage of the Quantum Zeno Effect. We'll have a control qubit that indicates whether or not we plan to look at the contents of the package, and we'll have a measurement that collapses this qubit back to $|0\rangle$ in the case that a bomb is present.

We'll assume that we have a device that can measure whether a bomb is present. We will model this device as a quantum circuit U that has one input (the control qubit $|\phi\rangle$) and one output (a qubit that is $|1\rangle$ if a bomb is definitely present). If there is no bomb, then U maps $|\phi\rangle|0\rangle \mapsto |\phi\rangle|0\rangle$; in other words, U behaves as the identity. If there is a bomb, then U maps $|0\rangle|0\rangle \mapsto |0\rangle|0\rangle$ (there's a bomb, but we didn't look) and $|1\rangle|0\rangle \mapsto |1\rangle|1\rangle$ (we looked at the bomb); that is, U behaves as a CNOT gate.

We want to figure out whether there is a bomb (i.e., we want to test U's behavior) without setting off the bomb very often. Figure 0.2 shows the circuit we will use. We initialize the control qubit $|\phi\rangle$ to $|0\rangle$. In each step of the algorithm, we rotate the control qubit toward $|1\rangle$ by θ and then run U; we'll execute the algorithm for N steps.

Consider the case where there is no bomb; our initial input is $|0\rangle|0\rangle$. If we rotate the control qubit by θ, the input to the first U gate is $(\alpha|0\rangle + \beta|1\rangle)|0\rangle$, and the output of U is the same state (since U is the identity). Measuring the output qubit always returns $|0\rangle$ and doesn't alter the state; thus, each step rotates the qubit further until $\beta = 1$ at the last measurement.

Now consider the case where there is a bomb. Once again, our initial input is $|00\rangle$. After the first rotation, the input to

U is $(\alpha|0\rangle + \beta|1\rangle)|0\rangle$, and the output of U is $\alpha|0\rangle|0\rangle + \beta|1\rangle|1\rangle$. When we measure the last qubit, we have a $\beta^2 \approx 1/N^2$ probability of looking at the bomb and setting it off. Otherwise, we measure $|0\rangle$ for the output qubit, which means we didn't look at the bomb. However, *this measurement collapses the state back to* $|0\rangle|0\rangle$. Thus, subsequent steps in the algorithm will simply repeat this process. Overall, we only have a $N\beta^2 \approx 1/N$ chance of actually looking at the bomb.

Vaidman and Grover

To see the relationship to Grover's algorithm, consider a particularly unfortunate case where we have N packages, $N-1$ of which contain bombs. We want to find the one package that does not contain a bomb, though we don't mind setting off a few of the bombs in the process. Grover's algorithm has a property similar to Vaidman's method where the amplitude of one target basis vector is amplified while all others are constantly diminished or reset.

The important thing to note is that it's highly counterintuitive to be able to search in $\sqrt{N}$ steps. By querying in superposition, we manage to search using fewer steps than there are locations to search!