CAS CS 535: Complexity Theory

Lecturer: Mark Bun Fall 2023

Lecture Notes 26:

Grover's Algorithm

Reading.

- Arora-Barak § 10.4

Last time: Quantum Circuits, **BQP**

Recap of Quantum Computing

- An n-qubit quantum system is described by $|\varphi\rangle = \sum_{s\in\{0,1\}^n} \alpha_s |s\rangle$, where each $\alpha_s \in \mathbb{C}$ and $\sum_{s\in\{0,1\}^n} |\alpha_s|^2 = 1$.
- A quantum system evolves via unitary transformations: $|\varphi_t\rangle = U_t |\varphi_{t-1}\rangle$ where U_t is a unitary matrix.
- If a quantum state $|\varphi\rangle$ is measured, each basis state s is observed with probability $|a_s|^2$.

A *quantum circuit* is specified by a sequence of local (each applying to $O(1)$ qubits) unitary operations, essentially WLOG from a finite "universal" gate set. The circuit is applied to an initial state of the form $|x\rangle |0^m\rangle$ where x is the input, and the measured final state determines the output.

1 Unordered Search

Let's study one of the most important and versatile quantum algorithms, discovered by Lov Grover circa 1996, for solving the "unordered search" problem. The cleanest way to think about this problem is through the following abstract formulation.

Problem 1 (Unordered Search)**.** Let $x \in \{0,1\}^N$ be a bit string under the promise that there is exactly one index i such that $x_i = 1$. Identify i using (as few as possible) "queries" of the form "What is the value of bit x_j?" (for $j \in [N]$).

You can convince yourself that every deterministic algorithm for this algorithm requires N queries in the worst case. Similarly, a randomized algorithm that succeeds with high probability still needs $\Omega(N)$ queries. Grover Search is a quantum algorithm that solves this problem using only $O(\sqrt{N})$ queries (and moreover, is optimal for this problem.)

Before describing Grover Search, let us clarify what we mean by giving a quantum algorithm "query" access to x. Analogously to how we can think of a randomized algorithm as proposing a distribution over queries to ask (and receiving the corresponding distribution over answers), we can model a quantum algorithm as proposing a superposition over queries and receiving a superposition of answers. As a first attempt at modeling this, we might imagine x specifying an operator O_x such that $O_x |i\rangle = |x_i\rangle$. However, this is not reversible. One way to get a reversible operation is by adding the answer x_i to an auxiliary "target" register: $O_x \left|i, y\right\rangle = \left|i, y \oplus x_i\right\rangle$.

Observe that if we set the target register to $\left|-\right\rangle = \frac{1}{\sqrt{2}}(\left|0\right\rangle - \left|1\right\rangle)$, then we get

$$O_x \left|i, -\right\rangle = \frac{1}{\sqrt{2}}(\left|i, x_i\right\rangle - \left|i, 1 - x_i\right\rangle) = (-1)^{x_i} \left|i, -\right\rangle .$$

This is sometimes called the "phase kickback trick" as it has the effect of loading the desired answer into the phase (rotation angle in the complex plane) of the state. It means that we can equivalently use the operation

$$P_x \left|i\right\rangle := (-1)^{x_i} \left|i\right\rangle$$

as our model for querying the implicit input x.

Theorem 1 (Grover's Algorithm)**.** *For every N there is a quantum circuit (with access to P_x queries) that, for every $x \in \{0,1\}^N$ with exactly one index i such that $x_i = 1$, finds i with probability at least $1 - O(1/N)$. Moreover, the circuit invokes P_x at most $O(\sqrt{N})$ times and has size $O(\sqrt{N} \log N)$.*

This is perhaps the most basic version of Grover's result, and by now there are lots of extensions and generalizations. One of the most important is that one can remove the condition that there is at most one index i for which $x_i = 1$ without significantly increasing the number of queries.

1.1 Application to Satisfiability

How might we use Grover's algorithm in this abstract query model to solve a standard computational problem with an explicitly given input? Let's go back to our old friend, SAT, thinking of a CNF formula φ as being evaluated on an n-bit string $i \in \{0,1\}^n$. To use Grover's algorithm to find a satisfying assignment to φ, let $x_i = \varphi(i)$. This can be computed by a classical poly-size circuit, which can be written as a unitary operation U_φ where $U_\varphi \left|i, 0, 0\right\rangle = \left|i, \varphi(i), z_i\right\rangle$ and z_i is the contents of the classical formula evaluator's workspace. One can implement the phase query oracle P_x by applying U_φ, then the operation $\left|b\right\rangle \mapsto (-1)^b \left|b\right\rangle$ to the second register, then applying U_φ^{-1}. (This step of "uncomputing" is necessary because leaving unclean workspace around can mess up the desired interference effects of a quantum algorithm.)

Now Grover's algorithm can be used to find a satisfying assignment i (if one exists) using $O(\sqrt{N}) = 2^{n/2}$ queries to the phase oracle. The number of elementary operations beats brute-force classical search by roughly a quadratic factor.

Note that Grover's algorithm is optimal amongst all quantum query algorithms for solving abstract unordered search. So if there is a quantum algorithm for SAT that beats classical brute-force by more than a quadratic factor, it's going to have to exploit the structure of the input instance φ beyond its "black-box" input-output behavior $i \mapsto \varphi(i)$. Many complexity theorists actually believe that quantum algorithms cannot solve $\mathbf{NP}$-complete problems in polynomial time.

2 Grover's Algorithm

Grover's algorithm is actually remarkably simple to state. Define the following "diffusion" operator

$$D := \frac{2}{N} \begin{pmatrix} 1 & 1 & \cdots \\ 1 & 1 & \cdots \\ \vdots & & \ddots \end{pmatrix} - \mathrm{Id}\,.$$

That is, D is the matrix with entries $2/N - 1$ down the diagonal, and $2/N$ everywhere else. (We'll see in a bit why D is unitary and how to implement it with basic gates.)

Grover Search Input: String $x \in \{0,1\}^N$ given implicitly by phase oracle P_x.

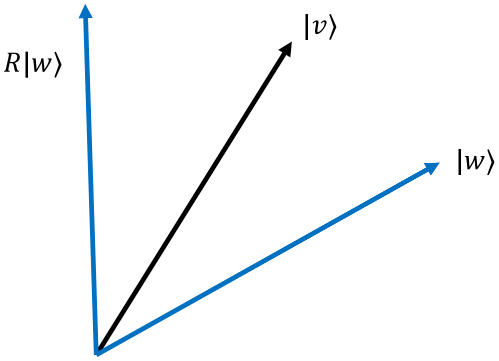
1. Initialize the uniform superposition $|u\rangle = \frac{1}{\sqrt{N}} \sum_{j=1}^{N} |j\rangle$.
2. For $t = 1, \ldots, T$:
   (a) Apply the phase oracle P_x
   (b) Apply the diffusion operator D
3. Measure, and check that the resulting i is a solution.

More compactly, the final state of the algorithm is $|\psi\rangle = (DP_x)^t |u\rangle$.
To understand this algorithm, let us interpret the basic operators P_x and D as *reflections*.

Definition 2. Let $|v\rangle \in \mathbb{C}^N$ be a complex vector. Matrix R is a reflection across $|v\rangle$ if

1. $R|v\rangle = |v\rangle$, and
2. $R|w\rangle = -|w\rangle$ for every $|w\rangle$ orthogonal to $|v\rangle$.

When applied to an arbitrary vector $|w\rangle$, such a reflection keeps the component parallel to $|v\rangle$ the same while negating the perpendicular component.



A bit of notation: A "ket" vector $|v\rangle$ is a column vector, whose conjugate transpose is the "bra" vector $\langle v|$. The inner product between two vectors becomes the "bra-ket" $\langle v|w\rangle$, while the outer product is the matrix $|v\rangle\langle w|$.

Claim 3. *If $|v\rangle$ is any unit vector, a reflection R through $|v\rangle$ takes the form $R = 2|v\rangle\langle v| - \mathrm{Id}$, and is unitary.*

Proof. First, we check that $R|v\rangle = 2|v\rangle\langle v|v\rangle - |v\rangle = |v\rangle$. Second, if $|w\rangle$ is orthogonal to $|v\rangle$, then $R|w\rangle = 2|v\rangle\langle v|w\rangle - |w\rangle = -|w\rangle$.

To see that it's unitary, write an arbitrary unit vector $|w\rangle = \alpha|v\rangle + \beta|v^\perp\rangle$, where $|\alpha|^2 + |\beta|^2 = 1$ and $|v^\perp\rangle$ is orthogonal to $|v\rangle$. Then $R|w\rangle = \alpha|v\rangle - \beta|v^\perp\rangle$ which is also a unit vector. □

Claim 4. *The phase oracle and diffusion operator are reflections:*

1. *If i is the unique index such that $x_i = 1$, then $-P_x$ is a reflection across $|i\rangle$.*
2. *The diffusion operator D is a reflection across the uniform superposition $|u\rangle$.*

Proof. For the phase oracle, note that $|1\rangle, \ldots, |N\rangle$ are an orthonormal basis for $\mathbb{C}^N$. So it suffices that $-P_x|i\rangle = |i\rangle$, while $-P_x|j\rangle = -|j\rangle$ for every $j \neq i$.

For the diffusion operator, it suffices to observe that

$$|u\rangle\langle u| = \frac{1}{N}\begin{pmatrix} 1 & 1 & \cdots \\ 1 & 1 & \cdots \\ \vdots & & \ddots \end{pmatrix},$$

so $D = 2|u\rangle\langle u| - \mathrm{Id}$. □

Thus, Grover's algorithm repeatedly applies these two reflections to "rotate" the starting vector $|u\rangle$ toward the destination $|i\rangle$. To see how this works, let

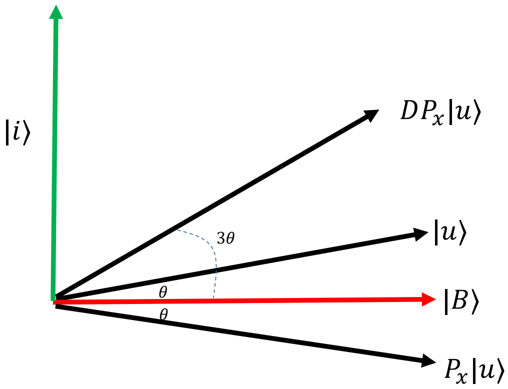
$$|B\rangle = \frac{1}{\sqrt{N-1}}\sum_{j \neq i}|j\rangle$$

be the uniform superposition over indices not equal to i. Thus, $|u\rangle = \sqrt{1 - 1/N}\,|B\rangle + 1/\sqrt{N}\,|i\rangle$, and moreover, every intermediate state of Grover's algorithm lies in the subspace V spanned by the orthogonal vectors $|i\rangle$ and $|B\rangle$. Restricted to this subspace V, the operator P_x itself is a reflection across the "bad" state $|B\rangle$.

Mathematically, if we let $\mathrm{Proj}_V = |i\rangle\langle i| + |B\rangle\langle B|$ be the projector onto V, we see that

$$P_x\,\mathrm{Proj}_V = (\mathrm{Id} - 2|i\rangle\langle i|)\,\mathrm{Proj}_V = |B\rangle\langle B| - |i\rangle\langle i| = 2|B\rangle\langle B| - \mathrm{Proj}_V\,.$$

So restricted to the two-dimensional (real) subspace V, we have that the first Grover iteration DP_x does the following to the uniform superposition $|u\rangle$:



where θ is the initial angle between $|u\rangle$ and $|B\rangle$. We can estimate this initial angle as

$$\cos\theta = \langle u|B\rangle = \sqrt{1-\frac{1}{N}} \approx 1 - \frac{1}{2N},$$

and using the Taylor approximation $\cos\theta \approx 1 - \theta^2/2$, we see $\theta \approx 1/\sqrt{N}$.

In general, every iteration of Grover's algorithm has the effect of taking a state $|\psi\rangle$, whose angle with $|B\rangle$ is ρ, and increasing it to $\rho + 2\theta$. Thus, after T iterations, the angle away from $|B\rangle$ is $(2T+1)\theta$. To get close to $|i\rangle$, which is an angle of $\pi/2$ away from $|B\rangle$, we thus expect having to take a number of iterations T for which $(2T+1)\theta \approx 1$, i.e., $T \approx \sqrt{N}$.

More precisely, after T iterations, the state of the algorithm is

$$\sin((2T+1)\theta)\,|i\rangle + \cos((2T+1)\theta)\,|B\rangle\,.$$

The probability of observing $|i\rangle$ after measuring is thus $\sin^2((2T+1)\theta)$.

We want this probability to be as close to 1 as possible, or equivalently, for $(2T+1)\theta$ to be as close to $\pi/2$ as possible. Taking $T = O(\sqrt{N})$ to be the integer for which $(2T+1)\theta$ is as close to $\pi/2$ as possible results in error at most $\sin^2(\theta) \le O(1/N)$.