Ashwin Nayak
Vazirani Class #16
October 16, 1997

How sensitive is a quantum computer to small perturbations?

While building quantum circuits for the different problems we have seen so far, we often assumed that we have access to arbitrary rotation and/or phase change gates. This seems to call for infinite precision in the implementation of such gates, which is clearly an unrealistic demand. (From the computational perspective, however, arbitrary precision would be very useful—we could smuggle in more computational power through it. For example, Adleman shows how we can factor in deterministic polynomial time if we could do infinite precision real arithmetic in a single step of computation.) The question that arises then is whether we can approximate the gates used in our quantum circuits with *finite* precision ones, and yet expect them to work almost as well. Fortunately, as we will see in this lecture, quantum circuits are quite robust against such approximation. In fact, we will show that a computation that runs for T steps needs no more than $O(\log T)$ bits of precision in the gates involved.

This robustness against small errors, however, also seems to limit the power of quantum computers. Consider the following problem. We are given a black-box subroutine to compute a function $f : \{0,1\}^n \rightarrow \{0,1\}$, and the goal is to find an input x such that $f(x) = 1$. We could think of f as a formula over n variables, in which case the problem becomes just a reformulation of the satisfiability problem. A natural question to ask is whether we can exploit the exponential "parallelism" that a quantum computer affords to search for a solution x efficiently, given access to the function f. It turns out that any quantum algorithm must take at least $\Omega(2^{n/2})$ steps in the worst case in order to solve this problem. The reason, as we shall see, is essentially the one mentioned above: the inability of a quantum machine to "detect" small perturbations. This result is a clear indication that quantum parallelism alone cannot yield an efficient solution to an **NP**-hard problem; one would necessarily have to exploit additional structure inherent in the problem to be able to beat the lower bound.

Remark: The lower bound of $2^{n/2}$ also happens to be tight up to a constant factor. This surprising result is due to Grover, who discovered an algorithm for the case when there is exactly one x such that $f(x) = 1$ (which happens to be the hardest case). We will see this algorithm in a later lecture.

1 Perturbed computations

Suppose we wish to set up a superposition $|\psi\rangle \in \mathcal{C}^{2^n}$, but the quantum circuit we have only produces a superposition $|\psi'\rangle$ which is ϵ-close to $|\psi\rangle$, i.e., is at distance at most ϵ from $|\psi\rangle$ in the L_2 norm. Could the circuit still be of any use? It could, if the distribution $\mathcal{D}(|\psi'\rangle)$ resulting from measuring $|\psi'\rangle$ is sufficiently close to $\mathcal{D}(|\psi\rangle)$. For example, if $\|\, \mathcal{D}(|\psi'\rangle) - \mathcal{D}(|\psi\rangle)\, \|_1 \leq \delta$, then the chances of noticing the difference in much less than $1/\delta$ runs of the circuit are very small. The distributions are indeed close if $|\psi\rangle$ and $|\psi'\rangle$ are close as vectors in $\mathcal{C}^{2^n}$. More precisely, we can show the following:

Lemma 1 *If* $\| \, |\psi'\rangle - |\psi\rangle \, \|_2 \leq \epsilon$, *then* $\| \, \mathcal{D}(\psi') - \mathcal{D}(\psi) \, \|_1 \leq \sqrt{2}\,\epsilon$.

The proof of this lemma is left as an exercise.

Now consider a computation consisting of T steps, involving the application of the (unitary) transformations $U_1, U_2, \cdots, U_T$ to a superposition $|\phi_0\rangle$. Suppose, instead, that we use "perturbed" unitary transformations U_i' to carry out the computation. Can we expect the circuit constructed out of these to still work reasonably well? It turns out that due to the unitary nature of the transformations involved, the errors generated by the perturbations are not magnified as the computation proceeds; they accumulate only additively! To see this, consider the following $T+1$ transformations obtained by composing, for the ith transformation ($0 \leq i \leq T$), U_j at the jth step for $j \leq i$, and then U_j' for the time steps $j > i$. Let $|\psi_i\rangle$ be the result of action of these $T+1$ transormations on the superposition $|\phi_0\rangle$. (The different computations are shown below.)

Step								
1	U_1'	U_1	U_1	U_1		U_1	U_1	U_1
2	U_2'	U_2'	U_2	U_2	$\cdots$	U_2	U_2	U_2
3	U_3'	U_3'	U_3'	U_3		U_3	U_3	U_3
4	U_4'	U_4'	U_4'	U_4'		U_4	U_4	U_4
$\vdots$		$\vdots$			$\cdots$		$\vdots$	
$T-2$	U_{T-2}'	U_{T-2}'	U_{T-2}'	U_{T-2}'		U_{T-2}	U_{T-2}	U_{T-2}
$T-1$	U_{T-1}'	U_{T-1}'	U_{T-1}'	U_{T-1}'	$\cdots$	U_{T-1}'	U_{T-1}	U_{T-1}
T	U_T'	U_T'	U_T'	U_T'		U_T'	U_T'	U_T
Result	$\lvert\psi_0\rangle$	$\lvert\psi_1\rangle$	$\lvert\psi_2\rangle$	$\lvert\psi_3\rangle$	$\cdots$	$\lvert\psi_{T-2}\rangle$	$\lvert\psi_{T-1}\rangle$	$\lvert\psi_T\rangle$

Notice that since all the operations involved in the computation are unitary,

$$\begin{aligned} \| \, |\psi_{i+1}\rangle - |\psi_i\rangle \, \| &= \| \, U_T' \cdots U_{i+2}' U_{i+1} U_i \cdots U_1 |\phi_0\rangle - U_T' \cdots U_{i+2}' U_{i+1}' U_i \cdots U_1 |\phi_0\rangle \, \| \\ &= \| \, U_T' \cdots U_{i+2}' (U_{i+1} - U_{i+1}') U_i \cdots U_1 |\phi_0\rangle \, \| \\ &= \| \, (U_{i+1} - U_{i+1}') |\phi_i\rangle \, \| \, , \end{aligned}$$

where $|\phi_i\rangle = U_i \cdots U_1 |\phi_0\rangle$ is the superposition we expect to get if the correct transformations are applied. This implies that (going via the triangle inequality)

$$\| \, |\psi_T\rangle - |\psi_0\rangle \, \| \leq \sum_{0 \leq i < T} \| \, |\psi_{i+1}\rangle - |\psi_i\rangle \, \|$$

$$= \sum_{0 \le i < T} \| \, (U_{i+1} - U'_{i+1}) |\phi_i\rangle \, \| \qquad (1)$$

In other words, at each stage, if we employ a perturbed tranformation in the place of another transformation, we can assume that we are working with intended tranformation for it, and just add a correction term (the corresponding error) to our result.

2 Errors in defective circuits

Define the norm of a linear operator A as

$$\| \, A \, \| \stackrel{\mathrm{def}}{=} \max_{v : \| v \| = 1} \| \, Av \, \| \, .$$

We can specialize the result summarized by (1) to the case when the transformations U_i' are close to the U_i's in the sense that $\| \, U_i' - U_i \, \| \le \delta$. In this case, the net error in the computation amounts to at most δT, since each correct superposition $|\phi_i\rangle$ has norm 1.

Suppose each transformation U_i is a basic rotation by θ gate (or a conditional phase change gate with the same parameter), as is the case in a circuit that we may like to implement, and suppose that U_i is approximated by a rotation by $\theta + \delta\theta$ gate U_i'. For any unit length vector $v \in \mathcal{C}^{2^n}$, we then have $\| \, U_i' v - U_i v \, \| \le \delta\theta$. So the T steps of computation lead to an error of at most $\delta\theta T$ in the final superposition, i.e., an error of at most $\sqrt{2}\,\delta\theta T$ in the corresponding distributions. To make this smaller than a given ϵ, we need $\delta\theta \le \epsilon / \sqrt{2} T$. This means that we need the only $O(\log T)$ bits of accuracy in the specification of the angle for any gate in the circuit.

3 A lower bound for the search problem

As mentioned earlier, the robustness of quantum circuits against small perturbations (reflected in (1)) also seems to limit their computational power. Recall the search problem introduced earlier: given a boolean function f of n bits, find an x such that $f(x) = 1$. We can show that, in the worst case, any algorithm must take $\Omega(2^{n/2})$ steps to locate a solution string with probability at least a constant $p > 0$.

Suppose algorithm A solves this problem in T steps. I.e., with probability at least p, it finds a solution for every n variable function in T steps (if such a solution exists). We will test A on the function $f_0 \equiv 0$ and construct a function f such that f is non-zero on a unique string x_0, and the algorithm A performs badly on input f. The string x_0 is in fact the one at which A probes f the least during its run on input f_0.

Let $\alpha_{t,y}$ be the amplitude with which A evaluates f_0 at y at time step t. Let x_0 be such that

$a_y = \sum_t |\alpha_{t,y}|^2$ is minimized. We know that $\sum_y a_y \leq T$, and therefore, $a_{x_0} = \min_y a_y \leq T/2^n$.

Now consider the function f which is non-zero only at x_0. We will show that the algorithm A takes at least $\Omega(2^{n/2})$ steps to locate x_0 when it is given f as input. Consider the run of the algorithm on input f. We can view the T actions U_i' of A as perturbations of its actions U_i on input f_0. The distance between the superpositions $|\psi_0\rangle$ and $|\psi_T\rangle$ resulting from the runs on f and f_0 respectively is then given by (1). Now, for each $0 \leq i < T$, notice that $||\,(U_{i+1} - U_{i+1}')|\phi_i\rangle\,|| \leq \sqrt{2}\,|\alpha_{i,x_0}|$, so that

$$||\,|\psi_T\rangle - |\psi_0\rangle\,|| \quad \leq \quad \sqrt{2}\sum_i |\alpha_{i,x_0}|\,.$$

Since $a_{x_0} = \sum_i |\alpha_{i,x_0}|^2 \leq T/2^n$, the sum $\sum_i |\alpha_{i,x_0}|$ can be at most $T/2^{n/2}$ (when all the amplitudes are equal to $1/2^{n/2}$). So we get

$$||\,|\psi_T\rangle - |\psi_0\rangle\,|| \quad \leq \quad \sqrt{2}\frac{T}{2^{n/2}}.$$

But A must (or can be modified to) distinguish between the two distributions $\mathcal{D}(\psi_0)$ and $\mathcal{D}(\psi_T)$ by at least p. This implies, by lemma 1 above, that

$$p \quad \leq \quad ||\,\mathcal{D}(\psi_0) - \mathcal{D}(\psi_T)\,||_1 \quad \leq \quad 2\frac{T}{2^{n/2}}$$

which translates into an exponential lower bound of $(p/2)\,2^{n/2}$ for T.