CAS CS 535: Complexity Theory

Lecturer: Mark Bun Fall 2023

Lecture Notes 25:

Quantum Circuits, BQP

Reading.

- Arora-Barak § 10.1-10.3

Last time: Hardness of Approximation, Proof of PCP Mini

Quantum computation is the field that studies the power of (abstract) computational devices based on the principles of quantum mechanics. There are many reasons to care about quantum computing (even if you don't believe full-fledged quantum computers will ever be built!):

- Quantum computers pose a credible challenge to the "strong extended Church-Turing thesis" which asserts that every physically realizable computation device can be simulated by a deterministic TM with polynomial slowdown.
- Quantum mechanics impacts our efforts in classical computing whether we like it or not. At the lowest level, hardware manufacturers have to work to suppress quantum mechanical effects in order for classical chips to behave as expected. At the highest level, efficient quantum algorithms like Shor's algorithm for factoring reveal a need to understand the limits of quantum devices in order to design conservative, future-proof cryptography.
- Existing quantum devices are known to enable certain computational tasks that are hard or impossible classically. For instance, the Bennett-Brassard protocol enables one party to securely transmit a secret encryption key to another, with security that does not rely on assumptions about the computational power of an eavesdropper monitoring the communication. This protocol actually has a relatively simple implementation!
- There are many important classical computing results whose proofs go by way of a detour through quantum computing. The survey of Drucker and de Wolf `https://arxiv.org/abs/0910.3376` has several great examples; some more recent ones include formula size lower bounds based on fast quantum algorithms for formula evaluation `https://dl.acm.org/doi/10.1145/3055399.3055472`, and fast optimization/machine learning algorithms based on "dequantizing" quantum algorithms `https://arxiv.org/abs/1807.04271`.
- Insights from quantum computing can shed light on our understanding of physics itself. For one, if we can't actually build quantum computers, a coherent explanation as to *why* would likely reveal a hitherto unknown breakdown of quantum mechanics. Quantum computational complexity also offers insight into the tension between the predictions of quantum mechanics and of general relativity around black holes.

1 Quantum Mechanics

The famous "two-slit" experiment illustrates some of the key principles of quantum mechanics that we hope to exploit. Imagine an experimental setup where a device emits particles at a wall with either one or two slits cut in it. Behind the wall is an array of detectors that count how many times they're hit by particles passing through the wall.

Classical particles Suppose the device emits classical particles (e.g., baseballs). If one slit is open, then a dense patch of hits appears behind it. And if two slits are open, then the dense patches "sum up" and the total number of hits is roughly twice the number of hits as when one slit was open.

Classical waves If instead the device emits a beam of light or a ripple through water, then the two waves propagating through the slits interfere. In some locations, they'll interfere constructively (adding up) whereas in other locations they'll interfere destructively (canceling out) leading to an alternating pattern of light and dense areas on the back wall.

Quantum particles If the device emits a single photon (quantum unit of light) at a time then a similar interference pattern appears as if each photon was individually behaving as a wave. That is, each photon has the ability to self-interfere, as if it's exploring all possible paths to the wall simultaneously, each with some signed probability ("amplitude"), with the property that amplitudes of the same sign add up, while amplitudes of opposite signs cancel each other out. Even more bizarrely, if one places detectors at the slits, then the interference pattern disappears, and the photons behave as if they were classical particles – thus, making observations changes the outcome of the experiment.

Much of the power of quantum computing comes from carefully managing interference. Intuitively, this means manipulating quantum states so that they constructively interfere to increase the probability of outcomes we want, while destructively interfering to decrease the probability of outcomes we don't.

2 States and Operators

We'll describe "the rules" of quantum computing via an analogy with an abstract view of classical randomized computing. Consider a classical computing device with 2 bits of memory, labeled x_1, x_2. At any point in its computation, the device can be in one of four possible states, each of which we'll denote by $|x_1x_2\rangle$ and associate with a standard basis vector in $\mathbb{R}^4$:

$$|00\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix}, |01\rangle = \begin{pmatrix}0\\1\\0\\0\end{pmatrix}, |10\rangle = \begin{pmatrix}0\\0\\1\\0\end{pmatrix}, |11\rangle = \begin{pmatrix}0\\0\\0\\1\end{pmatrix}$$

More generally, you can think of the "state" of a randomized computation as being a probability distribution over these four basis, for example

$$|v\rangle = \frac{1}{2}|00\rangle + \frac{1}{2}|10\rangle = |10\rangle = \begin{pmatrix}1/2\\0\\1/2\\0\end{pmatrix}.$$

One "step" of a randomized computation maps each basis state to a distribution over (convex combination) of other basis states, and extends linearly to such a mapping from distributions to distributions. For example, consider the instruction:

$$\text{Set } x_2 = \begin{cases} x_1 \text{ with probability } 1/2, \\ x_2 \text{ with probability } 1/2. \end{cases}$$

This instruction maps $|01\rangle$ to $\frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle$. In general, we can encode capture it behavior in the 4×4 matrix

$$S = \begin{pmatrix} 1 & 1/2 & 0 & 0 \\ 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 1 \end{pmatrix}$$

where updating a state $|v\rangle$ to a new state $|w\rangle$ by applying this instruction corresponds to matrix multiplication $|w\rangle = S|v\rangle$. Observe that the columns of this matrix are exactly the distributions induced by updating the basis state vectors, and therefore have entries that sum to 1. That is, S is a stochastic matrix; in general, it maps probability distributions to probability distributions.

One can think of a randomized computation as starting in some initial state $|v_0\rangle$ (possibly depending on an input to a problem), applying a sequence of computation steps captured by stochastic matrices $S_1, \ldots, S_T$, resulting in some final state $|v_T\rangle$. The final output of the computation is obtained by sampling a a basis state from this final distribution.

A quantum computation does the same thing, except instead of its state being a non-negative real-valued vector representing probabilities, its state is a complex-valued vector whose components are called "amplitudes." This state is called a "superposition" of basis states.

	Randomized n-bit system	Quantum n-qubit system
State	$\|v\rangle = \sum_{s\in\{0,1\}^n} p_s \|s\rangle$ s.t. $p_s \geq 0$, $\sum_{s\in\{0,1\}^n} p_s = 1$	$\|\varphi\rangle = \sum_{s\in\{0,1\}^n} \alpha_s \|s\rangle$ s.t. $\sum_{s\in\{0,1\}^n} \|\alpha_s\|^2 = 1$
Evolution	$\|v_t\rangle = S_t \|v_{t-1}\rangle$ S_t is stochastic	$\|\varphi_t\rangle = U_t \|\varphi_{t-1}\rangle$ U_t is unitary
Observation	Observe s w.p. p_s	If "measured," observe s w.p. $\|\alpha_s\|^2$.

Unitary evolution A unitary matrix U is one for which $\|U|\varphi\rangle\|_2 = 1$ for every complex unit vector $|\varphi\rangle$. That is, just as how stochastic matrices are those that preserve vectors representing legitimate probability distributions, unitary matrices preserve vectors of legitimate amplitudes. An equivalent characterization of the unitary matrices is as those for which $U^\dagger U = \mathrm{Id}$, where $U^\dagger = (U^\top)^*$ is the conjugate transpose of U. This, in particular, means unitary matrices are invertible (with inverse $U^\dagger$), and thus quantum computing is *reversible*.

Measurement In our abstract view of randomized computing, observing the state of a computation collapses it down to a definite realization of a basis state $|s\rangle$. We can perform additional computation after this observation, but it evolves according to the conditional distribution conditioned on having observed $|s\rangle$. This is similarly the case for quantum computing; by performing a measurement, we can observe a definite value of the system, but the information contained in the amplitude vector is now lost.

The kind of measurement we've described is called "measurement in the computational basis" but more general measurements are available (projective, and more generally, positive operator-valued measurements).

Let's look at a concrete example of how complex-valued amplitudes (actually, just negative ones) let us do things counterintuitive to classical probability. Consider the one qubit unitary operation

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},$$

and let us examine the behavior of evolving a qubit under this operation and measuring. First observe that

$$H|0\rangle = \frac{1}{2}(|0\rangle + |1\rangle) =: |+\rangle$$

$$H|1\rangle = \frac{1}{2}(|0\rangle - |1\rangle) =: |-\rangle$$

so upon measuring, in both cases we observe 0 with probability $1/2$ and 1 with probability $1/2$. But now suppose our initial state is the uniform superposition $|0\rangle$ and $|1\rangle$, namely $|+\rangle$ itself. Then

$$H|+\rangle = |0\rangle,$$

so now we definitely observe the state 0 and never observe the state 1. This is because applying H causes constructive interference on the state 0, but destructive interference on the state 1.

3 Quantum Circuits

Of course, if we sit down to implement a quantum computer (or a classical one, for that matter) we don't have arbitrary unitary matrices at our disposal. Just as we build classical circuits out of gates that operate locally on a constant number of bits, we want to think of our unitary operations as compositions of local unitary operators.

One important issue is that there is a continuum of unitary matrices operating on, say, 2 qubits. Fortunately, just as how arbitrary classical functions can be implemented by AND/OR/NOT gates operating on a constant number of bits, so too can arbitrary unitary operations be approximated by a finite gate set. The Solovay-Kitaev theorem tells us there is a universal gate set of size 3:

Theorem 1. *Every unitary matrix on* 1 *or* 2 *qubits can be approximated to error* ε *(in operator norm) using a circuit of size* $\operatorname{poly}\log(1/\varepsilon)$ *built from gates* CNOT, H, T *where*

$$\mathrm{CNOT}|x, y\rangle = |x, x \oplus y\rangle$$

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$T \begin{pmatrix} 1 & 0 \\ 0 & e^{-i\pi/4} \end{pmatrix}$$

Our convention for quantum circuits is as follows. There are n registers containing the input x (encoded as basis state $|x\rangle$) as well as m additional "workspace" registers each initialized to $|0\rangle$. The internal nodes of a quantum circuit consist of quantum gates from some universal gate set operating on at most a constant number of qubits at a time. At the end of the computation, a measurement is performed on (a subset) of the qubits of the final state, yielding a probabilistic answer.

Note that this convention assumes measurement only happens at the end of computation. The principle of "deferred measurement" shows this is without loss of generality; we can simulate intermediate measurements with measurements that happen only at the end with a modest blowup in circuit size.

Definition 2. A language L is in $\mathbf{BQTIME}(T(n))$ if there exists a sequence of logspace uniform quantum circuits $\{C_n\}$ such that

- $|C_n| = O(T(n))$.
- For every x, $\Pr[C(|x\rangle\,|0^{m(|x|)}\rangle) = L(x)] \geq 2/3$.

The class $\mathbf{BQP} = \cup_{c=1}^{\infty}\mathbf{BQTIME}(n^c)$.

3.1 Relationship to other classes

Claim 3. $\mathbf{BPP} \subseteq \mathbf{BQP}$.

Proof sketch. A $\mathbf{BPP}$ machine can be simulated by a uniform family of classical poly-size circuits operating on an input, together with polynomially many random bits. The gates in this circuit can be simulated by reversible (hence unitary) gates as follows: Replace gate $g(x, y)$ with $g(|x, y, z\rangle) = |x, y, g(x, y) \oplus z\rangle)$. Initialize the extra inputs with workspace registers, and simulate the machine's coin tosses with Hadamard gates. □

Claim 4. $\mathbf{BQP} \subseteq \mathbf{PP}(\subseteq \mathbf{PSPACE})$.