Week 6: Phase Estimation and the RSA Cryptosystem

COMS 4281 (Fall 2024)

Admin

1. Practice worksheet out, and quiz #3 will be out tonight.
2. Midterm on October 21. More details soon.

Last time

- Discrete Fourier Transform F_N is a **unitary matrix** mapping standard basis $\{|0\rangle, \ldots, |N-1\rangle\}$ to Fourier basis $\{|f_0\rangle, |f_1\rangle, \ldots, |f_{N-1}\rangle\}$ where

$$|f_j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{2\pi ijk}{N}} |k\rangle \ .$$

- The Quantum Fourier Transform is a fast **quantum algorithm** that implements the DFT F_N for $N = 2^n$, and runs in time $\mathrm{poly}(n) = \mathrm{poly}(\log N)$.

Brief linear algebra review

Eigenvalues

If $M \in \mathbb{C}^{N \times N}$ is a matrix, $|\psi\rangle \in \mathbb{C}^N$ is a vector, and $\lambda \in \mathbb{C}$ satisfying

$$M |\psi\rangle = \lambda |\psi\rangle$$

then we say that $|\psi\rangle$ is an **eigenvector** of M with **eigenvalue** λ.

Eigenvalues of unitary matrices

Fact: The eigenvalues of a unitary matrix U are all of the form $e^{2\pi i\theta}$ for some $\theta \in [0, 2\pi)$.

Eigenvalues of unitary matrices

Fact: The eigenvalues of a unitary matrix U are all of the form $e^{2\pi i\theta}$ for some $\theta \in [0, 2\pi)$.

Proof: Suppose that $U|\psi\rangle = \lambda|\psi\rangle$ for some eigenvector $|\psi\rangle$ and some eigenvalue λ.

Eigenvalues of unitary matrices

Fact: The eigenvalues of a unitary matrix U are all of the form $e^{2\pi i\theta}$ for some $\theta \in [0, 2\pi)$.

Proof: Suppose that $U|\psi\rangle = \lambda|\psi\rangle$ for some eigenvector $|\psi\rangle$ and some eigenvalue λ.

Taking inner products of $\lambda|\psi\rangle$ with itself, on one hand we get

$$(\lambda^* \langle\psi|)(\lambda|\psi\rangle) = |\lambda|^2 \langle\psi|\psi\rangle = |\lambda|^2 \; .$$

Eigenvalues of unitary matrices

Fact: The eigenvalues of a unitary matrix U are all of the form $e^{2\pi i\theta}$ for some $\theta \in [0, 2\pi)$.

Proof: Suppose that $U|\psi\rangle = \lambda|\psi\rangle$ for some eigenvector $|\psi\rangle$ and some eigenvalue λ.

Taking inner products of $\lambda|\psi\rangle$ with itself, on one hand we get

$$(\lambda^* \langle\psi|)(\lambda|\psi\rangle) = |\lambda|^2 \langle\psi|\psi\rangle = |\lambda|^2 .$$

On the other hand,

$$(\lambda^* \langle\psi|)(\lambda|\psi\rangle) = (\langle\psi| U^\dagger)(U|\psi\rangle) = \langle\psi| U^\dagger U |\psi\rangle = \langle\psi|\psi\rangle = 1$$

because $U^\dagger U = I$ (one of definitions of being unitary).

Eigenvalues of unitary matrices

Fact: The eigenvalues of a unitary matrix U are all of the form $e^{2\pi i\theta}$ for some $\theta \in [0, 2\pi)$.

Proof continued: Therefore

$$|\lambda|^2 = 1$$

and the only such λ's possible are of the form $e^{2\pi i\theta}$.

Some examples

Example: What are the eigenvalues and eigenvectors of

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Some examples

Example: What are the eigenvalues and eigenvectors of

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

We see that

$$Z|0\rangle = |0\rangle \qquad Z = |1\rangle = -|1\rangle \ .$$

Therefore standard basis are the eigenvectors and ± 1 are corresponding eigenvalues.

Some examples

Example: What are the eigenvalues and eigenvectors of

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} .$$

Some examples

Example: What are the eigenvalues and eigenvectors of

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} .$$

We can compute this by hand, or we can also remember that

$$X |+\rangle = |+\rangle \qquad X |-\rangle = - |-\rangle$$

so the Hadamard basis are the eigenvectors and ± 1 are the corresponding eigenvalues.

Some examples

Example: What are the eigenvalues and eigenvectors of

$$CNOT = \begin{pmatrix} 1 & 0 & & \\ 0 & 1 & & \\ & & 0 & 1 \\ & & 1 & 0 \end{pmatrix}.$$

Some examples

Example: What are the eigenvalues and eigenvectors of

$$CNOT = \begin{pmatrix} 1 & 0 & & \\ 0 & 1 & & \\ & & 0 & 1 \\ & & 1 & 0 \end{pmatrix}.$$

1. $|0,0\rangle$ with eigenvalue 1
2. $|0,1\rangle$ with eigenvalue 1
3. $|1,+\rangle$ with eigenvalue 1
4. $|1,-\rangle$ with eigenvalue -1

Phase Estimation Algorithm

Application of QFT: Phase Estimation

Phase Estimation Algorithm (PEA) is one of the most important subroutines in quantum computing.

Phase Estimation Algorithm (PEA) is one of the most important subroutines in quantum computing.

Goal of PEA:

- Ability to run controlled versions of U^k for $k = 1, 2, \ldots$.
- An **eigenstate** $|\psi\rangle$ where $U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$,

estimate θ.

Question: The eigenvalue $e^{2\pi i\theta}$ looks like a global phase... how can you possibly estimate it?

Question: The eigenvalue $e^{2\pi i\theta}$ looks like a global phase... how can you possibly estimate it?

Answer: It becomes a **relative** phase once you run the controlled-U gate in superposition:

$$\begin{aligned} cU\left|+\right\rangle\left|\psi\right\rangle &= \frac{1}{\sqrt{2}}(\left|0\right\rangle\left|\psi\right\rangle + \left|1\right\rangle U\left|\psi\right\rangle) \\ &= \frac{1}{\sqrt{2}}(\left|0\right\rangle\left|\psi\right\rangle + e^{2\pi i\theta}\left|1\right\rangle\left|\psi\right\rangle) \\ &= \frac{1}{\sqrt{2}}(\left|0\right\rangle + e^{2\pi i\theta}\left|1\right\rangle)\left|\psi\right\rangle \end{aligned}$$

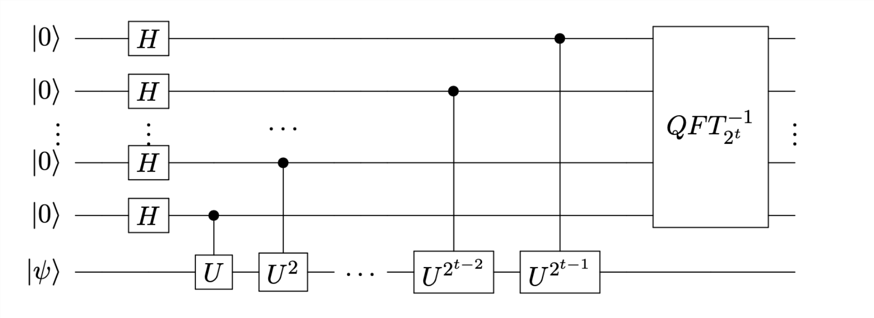
Phase Estimation Algorithm

Assume for simplicity that θ can be represented using exactly t bits. In other words the binary representation of θ looks like

$$\theta = 0.\theta_1\theta_2 \cdots \theta_t$$

where $\theta_1, \theta_2, \ldots \in \{0, 1\}$. This is equivalent to

$$\theta = \frac{\theta_1}{2} + \frac{\theta_2}{2^2} + \cdots + \frac{\theta_t}{2^t}.$$



Measuring the first t qubits will yield $|\theta_1, \theta_2, \ldots, \theta_t\rangle$.

Phase Estimation Algorithm Analysis

Let's analyze a special case where $t = 2$, and $\theta = \frac{\theta_1}{2} + \frac{\theta_2}{4}$ for $\theta_1, \theta_2 \in \{0, 1\}$.

(On the board...)

Phase Estimation Algorithm Analysis

Question: What if the phase θ cannot be exactly expressed as t bits?

Phase Estimation Algorithm Analysis

Question: What if the phase θ cannot be exactly expressed as t bits?

Answer: If we use $t + k$ ancilla qubits, and measure only the first t ancilla qubits, we will get the best t-bit approximation $\widetilde{\theta}$ of θ with probability $1 - 2^{-k}$.

Phase Estimation Algorithm Analysis

Question: What happens if $|\psi\rangle$ is not an eigenvector of U?

Phase Estimation Algorithm Analysis

Question: What happens if $|\psi\rangle$ is not an eigenvector of U?

Answer: The set $\{|\phi_j\rangle\}$ of eigenvectors of U forms a basis for $\mathbb{C}^{2^n}$ (if U is n-qubit unitary). We can write $|\psi\rangle$ as

$$|\psi\rangle = \sum_j \alpha_j |\phi_j\rangle$$

for some coefficients α_j.

Running Phase Estimation on $|\psi\rangle$ with ancilla qubits $|0 \cdots 0\rangle$ yields a state that is close to

$$\approx \sum_j \alpha_j |\phi_j\rangle \otimes |\widetilde{\theta}_j\rangle$$

where $\widetilde{\theta}_j$ is an approximation of the eigenphase θ_j, i.e. $U|\phi_j\rangle = e^{2\pi i \theta_j} |\phi_j\rangle$.

Running Phase Estimation on $|\psi\rangle$ with ancilla qubits $|0 \cdots 0\rangle$ yields a state that is close to

$$\approx \sum_j \alpha_j |\phi_j\rangle \otimes |\widetilde{\theta}_j\rangle$$

where $\widetilde{\theta}_j$ is an approximation of the eigenphase θ_j, i.e. $U |\phi_j\rangle = e^{2\pi i \theta_j} |\phi_j\rangle$.

Measuring the last register yields $\widetilde{\theta}_j$ with probability $|\alpha_j|^2$.

RSA and the Factoring problem

RSA Cryptosystem

- Invented by Rivest, Shamir, and Adleman in 1977
- Most widely deployed public-key cryptosystem
- Enables public-key encryption as well as digital signatures

Public key encryption

1. Bob generates a *secret-key/public-key* pair (sk, pk), and publishes pk on the internet.
2. Alice uses pk and her message m to create a *ciphertext* c which she sends to Bob.
3. Bob gets c, and uses sk to decode m.
4. The adversary sees (pk, c), and should get no information about m.

RSA Cryptosystem

Bob

1. Pick random prime numbers p, q, and set $N = pq$.

RSA Cryptosystem

Bob

1. Pick random prime numbers p, q, and set $N = pq$.
2. Pick random prime number $1 \leq e \leq (p-1)(q-1)$.

RSA Cryptosystem

Bob

1. Pick random prime numbers p, q, and set $N = pq$.
2. Pick random prime number $1 \leq e \leq (p-1)(q-1)$.
3. Compute integer d where $ed = 1 \bmod (p-1)(q-1)$.

RSA Cryptosystem

Bob

1. Pick random prime numbers p, q, and set $N = pq$.
2. Pick random prime number $1 \leq e \leq (p-1)(q-1)$.
3. Compute integer d where $ed = 1 \bmod (p-1)(q-1)$.
4. Set public key $pk = (e, N)$, and secret key $sk = d$.

RSA Cryptosystem

Alice gets a message $1 \leq m < N$. She computes and sends $c = m^e \mod N$, and send c to Bob.

RSA Cryptosystem

Alice gets a message $1 \leq m < N$. She computes and sends $c = m^e \mod N$, and send c to Bob.

Bob computes $m' = c^d$ mod N to decode the message.

RSA Cryptosystem

Alice gets a message $1 \leq m < N$. She computes and sends $c = m^e \mod N$, and send c to Bob.

Bob computes $m' = c^d$ mod N to decode the message.

This works because $c^d = (m^e)^d = m^{ed}$, and modulo N this equals m by *Fermat's Little Theorem*.

RSA Cryptosystem

Adversary sees the public key $pk = (e, N)$ and the encrypted message (ciphertext) c.

It does not know the primes p, q, nor the secret key $sk = d$.

RSA Cryptosystem

Adversary sees the public key $pk = (e, N)$ and the encrypted message (ciphertext) c.

It does not know the primes p, q, nor the secret key $sk = d$.

If it knew the prime factorization $N = pq$ it could compute the secret key!

Factoring problem

Input: Positive integer N.

Output: Prime factorization of N as $p_1^{a_1} p_2^{a_2} \cdots$.

Factoring problem

Input: Positive integer N.

Output: Prime factorization of N as $p_1^{a_1} p_2^{a_2} \cdots$.

The prime factorization of N is unique by the **Fundamental Theorem of Arithmetic**.

To find a factorization of N, it suffices to be able to find *some* nontrivial divisor of N.

Factoring problem

It is widely believed that Factoring is hard for classical computers.
The best classical algorithm, known as the **General Number Field Sieve**, takes time roughly

$$\exp\left(O(\log N)^{1/3}\right) .$$

This is essentially **exponential** in the number of digits of N.

Shor's algorithm

A quantum algorithm to solve Factoring in $\mathrm{poly}(\log N)$ steps.

Discovered by Peter Shor in 1993. He was inspired by Simon's Algorithm.

Shor's algorithm

A quantum algorithm to solve Factoring in $\mathrm{poly}(\log N)$ steps.

Discovered by Peter Shor in 1993. He was inspired by Simon's Algorithm.

Shor's Algorithm is also a hybrid classical-quantum algorithm.

1. **Classical part**: reduce the factoring problem to **order finding**.
2. **Quantum part**: solve order finding.

Order Finding

Input: given positive integers N, x such that

1. $1 \leq x < N$
2. $\gcd(N, x) = 1$ (i.e. they do not have any nontrivial factors in common)

Order Finding

Input: given positive integers N, x such that

1. $1 \leq x < N$
2. $\gcd(N, x) = 1$ (i.e. they do not have any nontrivial factors in common)

Output: find smallest integer r such that $x^r = 1 \bmod N$ (called the **order** of $x \bmod N$).

Next time

A quantum algorithm to solve Order Finding