LECTURE 10 (February 19)

TODAY Do quantum speedups need structure ?
Search problems

So far, we have seen some problems where quantum algorithms are exponentially faster

For example, Simon's problem
- f is one-to-one
- $\exists s$ s.t. $f(x \oplus s) = f(x)$ $\forall x$

OR, Forrelation
- input is very Fourier correlated
- input is not very Fourier correlated

On the other hand, for Unstructured search problem
- if $\exists x$ s.t. $f(x) = 0$
- no such x exists

we had a quadratic advantage

- Can we identify some general principles required to have an exponential quantum advantage ?

Promise vs Total Problems

Simon's problem & Forrelation are promise problems while
Unstructured search is a total problem

Are exponential speedups possible for total problems ? NO!

Theorem For any total problem that has a quantum algorithm with q queries there is a randomized algorithm that makes q^3 queries.

Infact, $\exists$ deterministic algorithm that makes q^4 queries.

You will be asked to prove this in an optional homework exercise

Unstructured vs Structured Problems

Simon's problem and Forrelation used a structured oracle

E.g. in Simon's, oracle was a problem with an algebraic structure

in Forrelation, oracle was Fourier correlated

ON the other hand, for unstructured search, one can take the oracle to be an unstructured oracle

i.e. $O_n : \{0,1\}^n \to \{0,1\}$ that the algorithm is making queries to is a random function

(OR equivalently, the truth table of the oracle whose bits are queried by the algorithm is a random string of length 2^n)

Why do we want an unstructured oracle ?

① Recall that the results are in a black-box access model
The hope is that if the oracle is a random function, the algorithm should not be able to use any specific properties of the black-box and such an assumption is closer to real world

② One can instantiate the random function oracle with a cryptographic hash function to get a real-world problem that we believe could be a candidate for quantum advantage

This is again not a proof but probably a more convincing heuristic evidence

Are quantum speedups possible for unstructured promise problems ?

i.e. given access to $\boxed{O}$ where $O_n : \{0,1\}^n \to \{0,1\}$ is a random function

can we have an exponential quantum advantage w.h.p. over the choice of f ?

This is not possible for ~~total~~ problems as we already saw, but is it possible for promise problems ?

Conjecture (Folklore go's)

On $1-\varepsilon$ fraction of the oracles the acceptance probability of a quantum algorithm that makes q queries and outputs a bit, can be approximated up to an additive error ε by a classical algorithm making $\text{poly}(d, \frac{1}{\varepsilon})$ queries.

$\Rightarrow$ On 99% of the oracles, $\exists$ a classical algorithm that effectively computes the same answer with only a polynomial overhead in queries

So, no quantum advantage is possible for unstructured decision problems assuming this conjecture!

Does the story end here? What about search or sampling problems?
Are unstructured speedups possible?

Separation of BQP-search from BPP-search wrt random oracles [Yamakawa-Zhandry '22]

Result $\exists$ NP^O-search problem that is in BQP^O but not in BPP^O whp. over the choice of O

So unstructured speedups seem to be possible here!

Open problem Can we find other example of such search problems?

Yamakawa-Zhandry search problem

Inverting a specific one-way function

Let Σ be an alphabet that we will choose to be $\mathbb{F}_q^n$ where q is a prime power with $q = O(n^2)$

$O: \Sigma \longrightarrow \{0,1\}$ be a random oracle

$C \subseteq \Sigma^n$ be a subspace of $(\mathbb{F}_q^n)^n$ that forms an error-correcting code with certain properties that we describe later

Problem Let $f: C \longrightarrow \{0,1\}^n$ be defined as

$$f(c_1, \ldots c_n) = (O(c_1), \ldots O(c_n))$$

Domain is the set of codewords

Find a preimage of 0^n

Theorem Choosing C appropriately, whp over choice of O

(1) $\exists$ a quantum algorithm that can approximately prepare a uniform superposition over all solutions with poly(n) queries
i.e. it can prepare the state

$$\propto \sum_{x \in f^{-1}(0)} |x\rangle$$

(2) any classical algorithm requires 2^{n^c} queries to find a pre-image

This problem is in NP given access to the oracle [Why?]

How to choose C? Assume that each symbol of the each codeword c is distinct

① List Recoverability : This will be helpful in ensuring classical hardness

② Decoding from random errors in the dual code This will be helpful in designing the quantum algorithm

} Such codes exist

Quantum Algorithm How to efficiently prepare the state

$$|\psi\rangle \propto \sum_{c \in C : f(c)=0} |c\rangle \quad ?$$

Consider the following two states :

$$|1_C\rangle \propto \sum_{c \in C} |c\rangle \qquad \text{and} \qquad |1_{pre}\rangle \propto \sum_{x \in \Sigma^n : O(x)=0^h} |x\rangle$$

Only supported on codewords

Only supported over 0^n preimages but over the entire alphabet Σ^n and not just the domain $C \subseteq \Sigma^n$

If we could somehow take pointwise product of these two states and normalize it we would be done !

The problem is that this is not a unitary or even a linear operation apriori so, we will need to use something else

We will apply the Quantum Fourier Transform and use another property of the code

Let us recall the Quantum Fourier Transform and its properties

This is similar to Hadamard which takes $x \in \{0,1\}^n$ and

$$H^{\otimes n}|x\rangle \longrightarrow \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} (-1)^{x \cdot y} |y\rangle$$

Quantum Fourier Transform QFT

This can be implemented efficiently

This is a unitary transformation that maps

$$QFT|x\rangle \longrightarrow \frac{1}{\sqrt{|\Sigma|^n}} \sum_{y \in \Sigma^n} \omega_q^{\phi(x \cdot y)} |y\rangle$$

$\phi : \mathbb{F}_q \to \mathbb{F}_p$ is

① linear

② $\forall x \in \Sigma \setminus \{0\}$, $\sum_{y \in \Sigma} \omega_p^{\phi(x \cdot y)} = 0$

where $x \in \Sigma^n$ where $\Sigma = \mathbb{F}_q^n$ where $q = p^r$ for prime p

Notation $|f\rangle = \sum_x f(x)|x\rangle$

Then, we denote by $|\hat{f}\rangle = QFT|f\rangle = \sum_x \hat{f}(x)|x\rangle$

$f(x)$ is amplitude in standard basis while $\hat{f}(x)$ is the amplitude in the Fourier basis

Two very useful properties of QFT

① Pointwise product becomes convolution after QFT [Exercise]

$$|f \cdot g\rangle = \sum_x f(x) g(x) |x\rangle$$

$$\widehat{|f \cdot g\rangle} = QFT|f \cdot g\rangle = \frac{1}{\sqrt{|\Sigma|^n}} \sum_x \hat{f} \star \hat{g}(x)|x\rangle \quad \text{where} \quad \hat{f} \star \hat{g}(x) = \sum_{y+z=x} \hat{f}(y)\hat{g}(z)$$

② QFT of uniform superposition over subspace C is uniform superposition over dual subspace $C^\perp$

$$\frac{1}{\sqrt{|C|}} \sum_{x \in C} |x\rangle \xrightarrow{QFT} \frac{1}{\sqrt{|C^\perp|}} \sum_{x \in C^\perp} |x\rangle$$

where $C^\perp = \{d \mid c \cdot d = 0 \ \forall c\}$

E.g. $\frac{1}{\sqrt{2^{n-1}}} \sum_{\substack{x \in \{0,1\}^n \\ x_n = 0}} |x\rangle \xrightarrow{H^{\otimes n}} |0\rangle^{\otimes n-1} |+\rangle$

$C = \{x \mid x_n = 0\}$ $\quad C^\perp = \{x \mid x_1 = x_2 = \ldots = x_{n-1} = 0\}$

NEXT TIME Algorithm & its analysis