Lecture 17 – Grover's algorithm and the structure of quantum speed-ups

Outline

- unstructured search
- Grover's algorithm
- Bennett, Bernstein, Brassard, Vazirani (BBBV) lower bound
- Interference vs. quantum

Unstructured search

Given oracle access to $f:\{0,1\}^n \to \{0,1\}$ s.t $\exists\, s \in \{0,1\}^n$ $f(s)=1$ and $\forall\, x \neq s$ $f(x)=0$, find s!

Classically this takes time 2^n. Why?

Deterministic case: suppose your algorithm makes T queries to the oracle. As long as $T < 2^n$ can always find s such that all queries return 0.

Probabilistic case extends this idea to include randomness as well. If $\geq 2/3$ of all paths with T queries produce s then all those paths must be querying s. Clearly, there can be at most T points common to all those paths. So if $T < 2^n$ we can pick an s that will not be queried.

What about the quantum case?

Grover's quantum algorithm solves problem with $\sqrt{2^n} = 2^{n/2}$ quantum queries

Grover's algorithm

As before, the idea will be to leverage superposition and interference.

The quantum oracle is as before $U_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle$

Consider the equal superposition over all query points

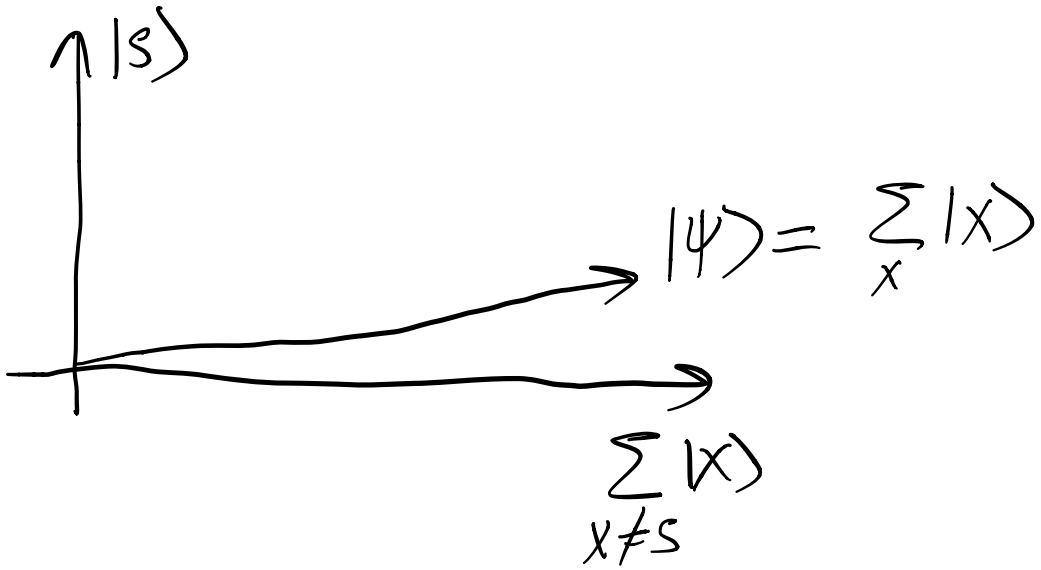
$$|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle$$

Can express this as

$$|\psi\rangle = \frac{1}{\sqrt{2^n}} |s\rangle + \frac{1}{\sqrt{2^n}} \sum_{x \neq s} |x\rangle$$

Note that $|s\rangle$ and $\sum_{x \neq s} |x\rangle$ are orthogonal

$\Rightarrow$ we have a nice geometric picture



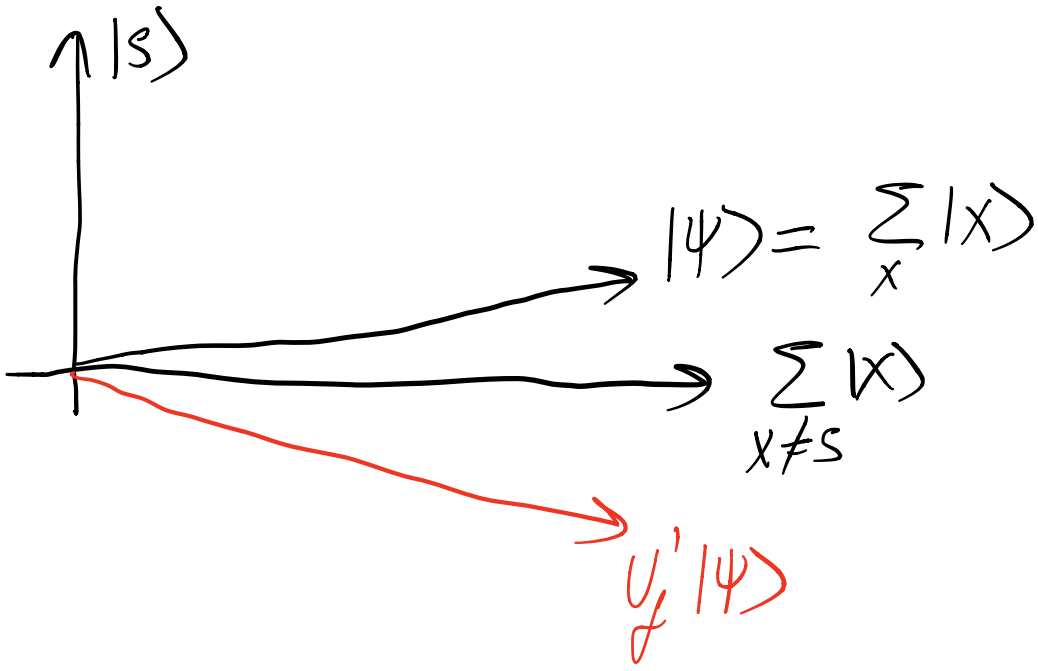
We know how to go from U_f to U_f' that performs the mapping

$$U_f' |x\rangle = (-1)^{f(x)} |x\rangle$$

How would this act on $|\psi\rangle$?

$$U_f' |\psi\rangle = \sum_x (-1)^{f(x)} |x\rangle = -|s\rangle + \sum_{x \neq s} |x\rangle$$

In the geometric picture this looks as follows



This is the reflection of $|\psi\rangle$ with respect to $\sum_{x \neq s} |x\rangle$

Can we reflect back? I.e. obtain the reflection of $U_f'|\psi\rangle$ with respect to $|\psi\rangle$.

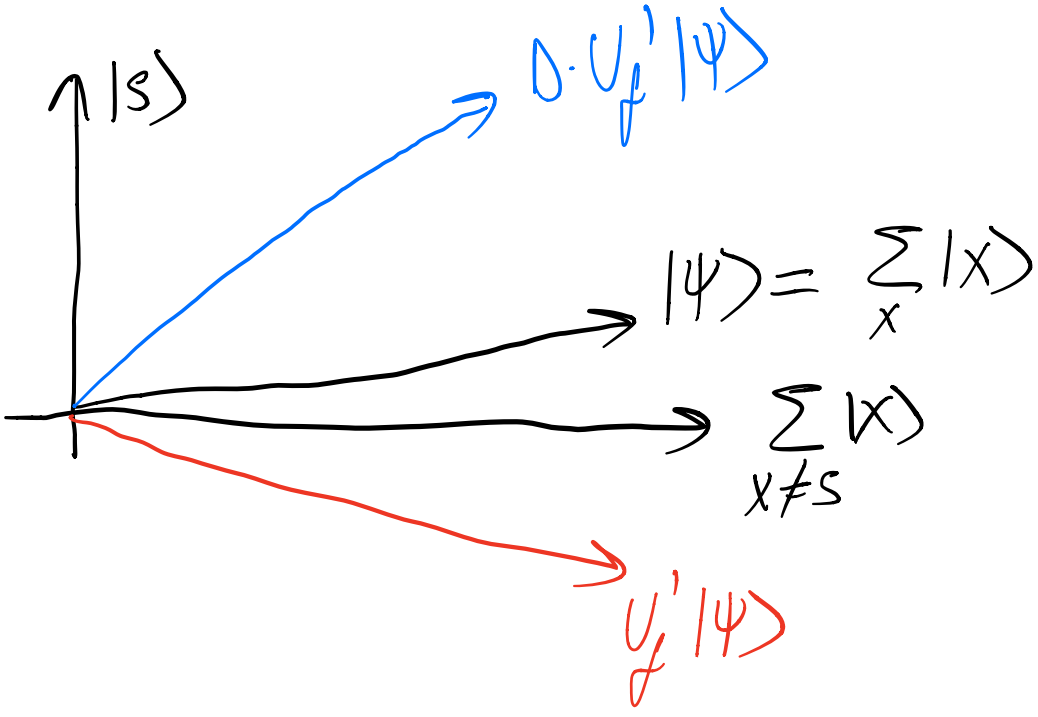
Yes! Note that when we reflected about $\sum_{x \neq s} |x\rangle$ we applied a unitary that maps $|s\rangle \to -|s\rangle$. We could just as well have applied a unitary that maps $|x\rangle \to -|x\rangle$, $x \neq s$, since up to global phase it's the same.

So to reflect about $|\psi\rangle$ we should perform an operation that takes $|\psi\rangle \to -|\psi\rangle$. How?

We know $|\psi\rangle = H^{\otimes n}|0^n\rangle$

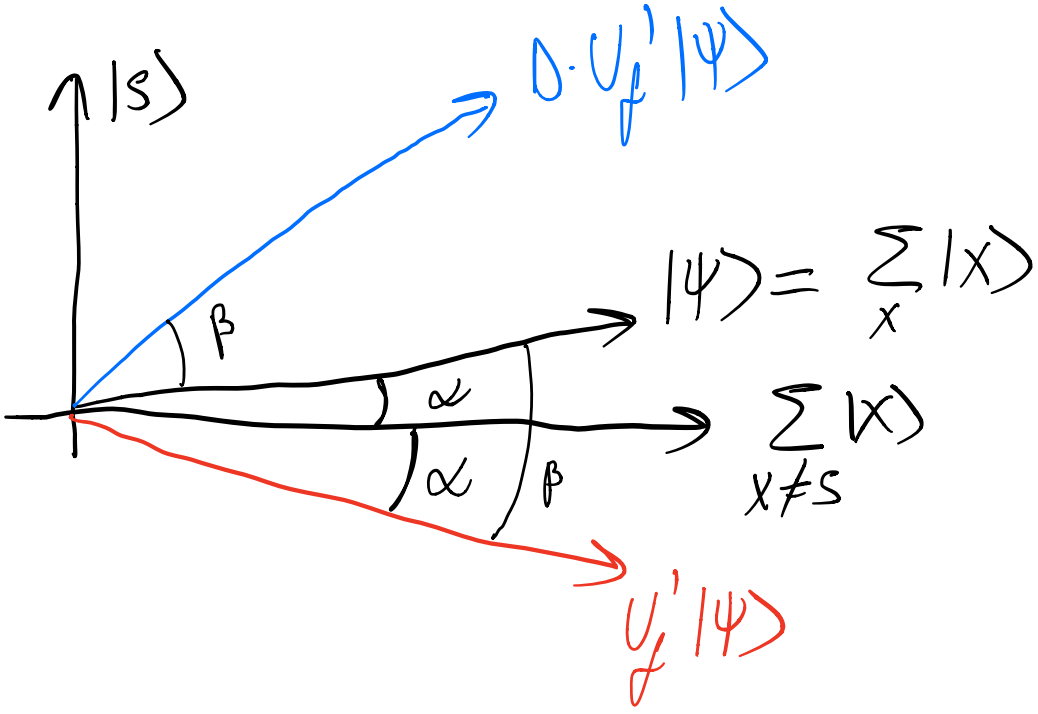
So just take $|0^n\rangle \to -|0^n\rangle$ and conjugate this operation by H.

Call this operation D – Grover diffusion operator



The state $D \cdot U_f'|\psi\rangle$ is closer to $|s\rangle$ than $|\psi\rangle$!

We can repeat this process a number of times to get closer and closer to $|s\rangle$. How many times?



$$\alpha = \arcsin \frac{1}{\sqrt{2^n}} \qquad \beta = 2\cdot\alpha = 2\arcsin \frac{1}{\sqrt{2^n}}$$

Each iteration rotates state by β. Say we do it m times

$$\Rightarrow \quad m\cdot\beta = \frac{\pi}{2} - \arcsin \frac{1}{\sqrt{2^n}}$$

$$\Rightarrow \quad (1+2m)\arcsin \frac{1}{\sqrt{2^n}} = \frac{\pi}{2}$$

$$\Rightarrow \quad (1+2m) = \frac{\pi}{2\arcsin\left(1/\sqrt{2^n}\right)}$$

$$\Rightarrow \quad m = \frac{\pi}{4\arcsin\left(1/\sqrt{2^n}\right)} - \frac{1}{2}$$

$$\underset{(-1\le x\le 1)}{\arcsin(x)} = x + \frac{x^3}{6} + \cdots$$

$$\Rightarrow \boxed{m \approx \frac{\pi}{4 \cdot (1/\sqrt{2^n})} = O(\sqrt{2^n})}$$

Can we do better quantumly? In particular, can we do it in polynomial time? If yes then $NP \subseteq BQP$ (technically $UP \subseteq BQP$) so this should make us suspicious!

↓ unique NP

BBBV lower bound

A year before Grover's algorithm BBBV showed that number of required quantum queries is $\Omega(\sqrt{2^n})$.

Proof sketch

The state of a Q alg after t queries will be

$$\boxed{|\psi_t\rangle = \sum_{x,w} \alpha_{x,w,t} |x\rangle|w\rangle}$$

The $|x\rangle$ register is the query register and $|w\rangle$ is the "workspace" register.

Let $M_x = \sum_{t=1}^{T} \sum_{w} |\alpha_{x,w,t}|^2$ be the magnitude of query x after T queries.

Note that $\sum_x M_x = \sum_{t=1}^{T} \underbrace{\sum_{x,w} |\alpha_{x,w,t}|^2}_{1} = T$

$\Rightarrow$ average magnitude is $\frac{T}{2^n}$

There must be an x^* s.t $M_{x^*} \leq \frac{T}{2^n}$

$\Leftrightarrow$ $\sum_{t=1}^{T} \underbrace{\sum_{w} |\alpha_{x^*,w,t}|^2}_{|\alpha_{x^*,t}|^2} \leq \frac{T}{2^n}$

$\Rightarrow$ $\sum_{t=1}^{T} |\alpha_{x^*,t}|^2 \leq \frac{T}{2^n}$

By Cauchy-Schwarz

$$\sum_{t=1}^{T} |\alpha_{x^*,t}| \cdot 1 \leq \underbrace{\sqrt{\sum_{t=1}^{T} |\alpha_{x^*,t}|^2}}_{\leq \sqrt{\frac{T}{2^n}}} \cdot \underbrace{\sqrt{\sum_{t=1}^{T} 1}}_{\sqrt{T}}$$

$$\Rightarrow \boxed{\sum_{t=1}^{T} |\alpha_{x^*,t}| \leq \frac{T}{\sqrt{2^n}}}$$

Imagine 2 situations now. In one a q algorithm queries f s.t $f(x) = 0, \forall x$. In the second, only after the T'th oracle call $f(x^*) = 1$ and otherwise $f(x) = 0$. How much do the states of the algorithm differ by in the 2 situations?

Clearly, upto $T-1$ states are identical. After T'th query states differ only in the components containing x^*.

$$\Rightarrow \| |\psi_T^1\rangle - |\psi_T^2\rangle \| \leq 2 |\alpha_{x^*,T}|$$

↓ Euclidean distance

Similarly for states that differ in the last 2 queries the difference will be at most $2|\alpha_{x^*,T}| + 2|\alpha_{x^*,T-1}|$

By induction the difference between $f(x)=0 \ \forall x$ and $f(x^*)=1, f(x)=0 \ \forall x \neq x^*$ will be

$$2\sum_{t=1}^{T} |\alpha_{x^*,t}| \leq 2 \cdot \frac{T}{\sqrt{2^n}}$$

$\Rightarrow$ probability of correctly finding $x^* \leq O(T^2/2^n)$

If we want this to be constant $\Rightarrow$ $T = \Omega(\sqrt{2^n})$

This also shows that $\exists$ oracle O s.t $NP^O \not\subseteq BQP^O$

What about general interference computation?

The situation here is different!

Suppose $f: \{0,1\}^n \to \{-1,+1\}$ and

either $\exists!\ s \in \{0,1\}^n \quad f(s) = 1, \ \forall x \neq s \ f(x) = -1$

or $\forall x \in \{0,1\}^n \quad f(x) = -1$

Can we tell the situations apart with interf1?

interf1 (f, D, N)

$$S = \frac{1}{N} \left| \sum_{x \in D} f(x) \right|$$

return True if $S \geq 2/3$

False if $S \leq 1/3$

Define $D' = \{0,1\}^n \times \{0,1\}^n$

$f' : D' \to \{-1, +1\}$

$$f'(x,y) = \begin{cases} \text{if } f(x) = 1, & f'(x,y) = 1 \\ \text{if } f(x) = -1, & f'(x,y) = (-1)^{|y|} \end{cases}$$

Take $N = 2^n$

Perform interf1 (f', D', N). What happens?

Case 1: $\exists s \quad f(s) = 1$

$$\Rightarrow \sum_{x,y} f'(x,y) = \underbrace{\sum_{x \neq s, y} f'(x,y)}_{0} + \underbrace{\sum_{y} f'(s,y)}_{2^n}$$

$$\Rightarrow S = 2^n / 2^n = 1$$

Case 2: $\forall x \quad f(x) = -1 \Rightarrow S = 0$

$\Rightarrow$ Can perfectly tell the 2 situations apart!

Observation 1: This crucially relies on s being unique. If there were multiple s values for which $f(s)=1$ then normalization won't work!

We'd need to know how many s there are to choose N appropriately. But that defeats the purpose ☺

$\Rightarrow$ this shows $UP \subseteq AWPP$

not $NP \subseteq AWPP$

However, by a beautiful result known as the Valiant-Vazirani theorem $NP \subseteq BPP^{AWPP}$

Observation 2: What prevents us from doing the same thing quantumly? Normalization! Because of the Born rule, we wouldn't be able to achieve $N = 2^n$. In fact can show that if amplitudes are

normalized in p-norm, $p > 2$ then you do get that $NP \subseteq BQP_p$.

Why? Hand-wavy argument: unitaries preserve 2-norm. If states are normalized in p-norm, $p > 2$, can boost probability of unlikely events! $Pr \to Pr^{2/p}$ in one iteration. Repeating this poly(n) times can boost prob of events that start off with prob $2^{-\text{poly}(n)}$, like finding s!

The interplay between unitarity and Born's rule is essential in q.c. It shows that while q.c. is performing interference it is a very constrained kind of interference.

Q algorithms are useful for solving highly structured problems, i.e. problems with some sort of "global property" in the following sense

Let $F = (f(0^n), f(0^{n-1}1), \ldots, f(1^n))$

for some $f: \{0,1\}^n \to \{0,1\}^m$; $|F| = 2^n$.

We'll return to the case of decision problems for simplicity.

In the case of Simon or Shor, there was a global property for F, like $F_i = F_j$ iff $i = j + k$ for Yes inputs. For No inputs we also have a constraint that say $F_i \neq F_j$ unless $i = j$ (f is 1-to-1).

Two things are happening here. The (Hamming) distance between a Yes input and a No input is large, and the fraction of inputs under consideration

is far smaller than the set of all inputs!
This is a highly structured problem!
For if the problem is to distinguish between

$(0,0\ldots,0)$ and $(0,0\ldots1\ldots0)$

there's not much difference between yes and no instances!

Additionally we know that if input can be any F (from all $2^{m \cdot 2^n}$ strings) there's always a classical algorithm that solves the problem with poly(Q) queries, where Q is required number of quantum queries (Beals et al result).

The moral is that in terms of query complexity, things are extremely subtle ...