CS6846 – Quantum Algorithms and Cryptography

Grover's Algorithm

Instructor: Shweta Agrawal, IIT Madras
Email: shweta@cse.iitm.ac.in

Problem: Let $f: \{0,1\}^n \to \{0,1\}$.

Promise that $\exists$ at most 1 point x^* s.t. $f(x^*) = 1$.

Find x^*.

Problem': Given $g: \{0,1\}^n \to \{0,1\}^m$ and some $y \in \{0,1\}^m$, find x s.t. $g(x) = y$. (g is injective).

f it has g, y hardwired.

$f(x) = 1$ iff $g(x) = y$.

GROVER'S ALGORITHM:

Notation $|x\rangle \xrightarrow{O_f} (-1)^{f(x)} |x\rangle$

① Prepare $|\psi\rangle = \sum_x \frac{1}{2^{n/2}} |x\rangle$.

② Repeat the following 2 steps k times:

$k \downarrow O(2^{n/2})$

(a): Apply O_f

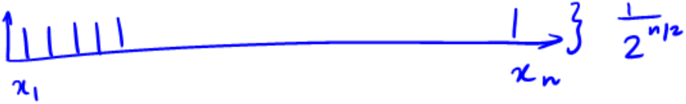
(b) Apply "Reflection about the mean" map

$$\sum_x \alpha_x |x\rangle \longrightarrow \sum_x (2\mu - \alpha_x) |x\rangle$$

where $\mu = \sum_x \frac{\alpha_x}{2^n}$.

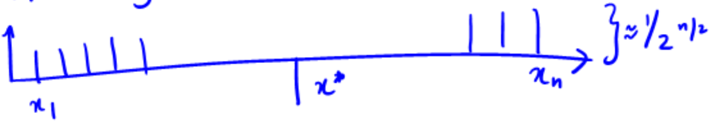
3. Measure all n bits.

Analysis:

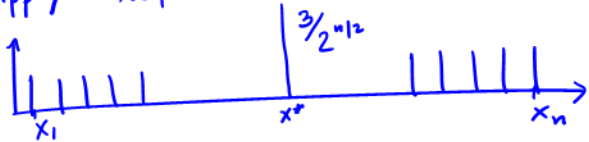
① 

k

② Apply O_f:



③ Apply reflection map. $\alpha_x \to (2\mu - \alpha_x)$



After k iterations

amplitude of x^* is $\frac{2k-1}{2^{n/2}}$

By setting $k = 2^{n/2}$, this amplitude becomes constant.

Can repeat to amplify this probability.

Classical $O(2^n)$ Birthday Algo $O(2^{n/2})$

Quantum $O(2^{n/2}) \rightarrow O(2^{n/3})$

Want $O(2^{n/3})$ algorithm:

1. Make k queries randomly to f.
2. Store in database
3. Define $g(x) = 1$ if $f(x)$ is in the database.
4. Optimize for k.

Claim: Mean reflection map is a valid unitary operation.

We'll show that this map is equivalent to

$$\underbrace{H^{\otimes n}\left(2|0\rangle\langle 0| - I\right)H^{\otimes n}}_{\begin{pmatrix} 1 & & & & & 0 \\ & -1 & & & & \\ & & -1 & & & \\ & & & -1 & & \\ & & & & \ddots & \\ 0 & & & & & -1 \end{pmatrix}}.$$

$$2\, \underbrace{H^{\otimes n}|0\rangle}_{|\psi\rangle} \langle 0| H^{\otimes n} - I.$$

Consider arbitrary state

$$\sum_x \alpha_x |x\rangle$$

$$\left(2|\psi\rangle\langle\psi| - I\right)\left(\sum \alpha_x |x\rangle\right)$$

$$= 2|\psi\rangle \underbrace{\sum_x \alpha_x \langle\psi|x\rangle} - \sum_x \alpha_x |x\rangle.$$

μ? Recall $\mu = \sum_x \dfrac{\alpha_x}{2^n}$

$$\sum_x \alpha_x / 2^{n/2} = \mu \cdot 2^{n/2}.$$

$$= 2\left(\sum_x \frac{1}{\cancel{2^{n/2}}}|x\rangle\right) \mu \cdot \cancel{2^{n/2}} - \sum \alpha_x |x\rangle.$$

$$= \sum_x \left(2\mu - \alpha_x\right)|x\rangle$$

This is the map that we applied in the algorithm.

<u>Careful Analysis:</u>

Notation: Let α^t be the amplitude of x^* after the t^{th} step, where one step consists of applying O_f followed by Reflection step (Grover diffusion gate).

Let β^t be amplitude of any other $x \in \{0,1\}^n$ after t^{th} step.

Let μ^t be mean of all amplitudes after the oracle gate on the t^{th} step.

Let $N = 2^n$.

$$\mu^t = \frac{(N-1)\beta^t - \alpha^t}{N}.$$

Proposition: Suppose $\alpha^t \leq \frac{1}{2}$ & $N \geq 4$.

Then $\alpha^{t+1} \geq \alpha^t + \frac{1}{\sqrt{N}}$.

Proof:

$$(\alpha^t)^2 + (N-1)(\beta^t)^2 = 1.$$

$$\Rightarrow 1 \le \frac{1}{4} + (N-1)(\beta^t)^2.$$

By re-arranging,

$$\beta^t \geqslant \sqrt{\frac{3}{4(N-1)}}$$

Therefore

$$\mu^t = \frac{-\alpha^t + (N-1)\beta^t}{N}$$

$$\geqslant \frac{-\frac{1}{2} + (N-1)\sqrt{\frac{3}{4(N-1)}}}{N}.$$

$$= \frac{1}{2} \frac{(N-1)\sqrt{\frac{3}{N-1}} - 1}{N}.$$

$$= \frac{1}{2} \frac{\sqrt{3N-3} - 1}{N}.$$

$$\geq \frac{1}{2} \frac{1}{\sqrt{N}} \quad \text{when } N \geq 4.$$

This implies:

$$\alpha^{t+1} = 2\mu^t + \alpha^t$$

$$\geq \frac{1}{\sqrt{N}} + \alpha^t.$$

Proposition: For any t,

$$\alpha^{t+1} \leq \alpha^t + \frac{2}{\sqrt{N}}.$$

Proof: For any given step t,

$$\mu^t = \frac{-\alpha^t + (N-1)\beta^t}{N} \leq \frac{N-1}{N}\beta^t.$$

We also know that

$$(N-1)(\beta^t)^2 \leq 1.$$

$$\beta^t \leq \frac{1}{\sqrt{N-1}}$$

$$\alpha^{t+1} = 2\mu^t + \alpha^t$$

$$\alpha^{t+1} = 2\mu^t + \alpha^t$$

$$\leq 2\,\frac{N-1}{N}\,\beta^t + \alpha^t.$$

$$\leq 2\,\frac{N-1}{N}\,\frac{1}{\sqrt{N-1}} + \alpha^t$$

$$= 2\,\frac{\sqrt{N-1}}{N} + \alpha^t$$

$$\leq \frac{2}{\sqrt{N}} + \alpha^t$$

as desired.

Now we show that $\alpha > 0.1$ after $O(\sqrt{N})$ steps.

If $N < 16$, $\alpha^t = \frac{1}{\sqrt{N}} \geq 0.25$.

(check).

If $N \geq 16$, as long as $t \leq \sqrt{N}/8$

we get $\alpha^t \leq \alpha^0 + \frac{2}{\sqrt{N}} t$.

$$\leq \alpha^0 + \frac{2}{\sqrt{N}} \frac{\sqrt{N}}{8}$$

$$= \frac{1}{\sqrt{N}} + \frac{1}{4}$$

$$\leq 1/2.$$

By the first prop for $\frac{\sqrt{N}}{8}$ steps,

$$\alpha^{\sqrt{N}/8} \geq \frac{\sqrt{N}}{8} \frac{1}{\sqrt{N}} = \frac{1}{8} > 0.1$$

Probability of getting x^* is $> (0.1)^2$
0.01

If I repeat 110 times,

$$\Pr(\text{one answer is } x^*) = 1 - (\Pr(\text{not } x^*))$$
$$\geq 1 - 0.99^{110}$$
$$\geq 2/3.$$