Lecturer: Yongshan Ding, Pranav Gokhale *Scribe: Shankar G. Menon*

Lecture 08: More on Algorithms

November 1, 2018

1 Bernstein-Vazirani Algorithm

In (the non-recursive) Bernstein-Vazirani Algorithm [BV97], we will see that there is no exponential speed up compared to the classical algorithm but a polynomial one.

Problem: Given an oracle access to $f : \{0,1\}^n \to \{0,1\}$ and a promise that the function $f(x) = s \cdot x$ $(\sum_{i=1}^{n} s_i \cdot x_i)$ in F_2^n, where s is a secret string that the algorithm is trying to learn.

Classically we will have to try it out brute force way by giving n inputs, i.e.

$$f(100...0_n) = s_1$$
$$f(010...0_n) = s_2$$
$$\vdots$$
$$f(000...1) = s_n$$

In quantum computation we can do this in one query as shown in figure below.

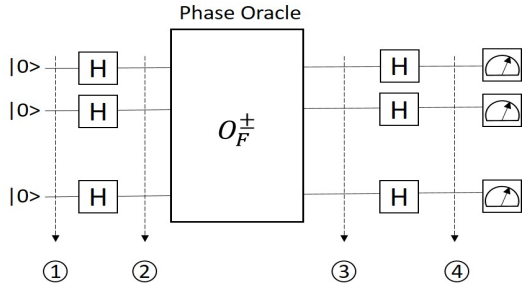

Figure 1: The schematic of Bernstein-Vazirani Algorithm

The states at different level are

① $\to |0\rangle^{\otimes n}$

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

③ $\to \frac{1}{\sqrt{2^n}} \sum_x (-1)^{f(x)} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_x (-1)^{s \cdot x} |x\rangle = \frac{1}{2}(|0\rangle + (-1)^{s_1} |1\rangle) \otimes \frac{1}{2}(|0\rangle + (-1)^{s_2} |1\rangle) \otimes ..$

So depending on s_i being 0 or 1, we will get $|+\rangle$ or $|-\rangle$.

④ $\to$ 0 if $s_i = 0$ and 1 if $s_i = 1$.

Here we assume that the such an oracle exist without worrying about the practical possibilities.

Implementing the Phase Oracle

The phase oracle can be implemented using a **phase kickback** circuit. In phase kick back circuit, a phase of π is kicked into the control bit.

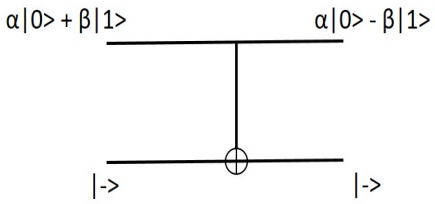

Figure 2: The kickback circuit

The control qubit is in arbitrary state $\alpha|0\rangle + \beta|1\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix}$ and the target qubit is in $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \begin{pmatrix}\frac{1}{\sqrt{2}}\\ -\frac{1}{\sqrt{2}}\end{pmatrix}$ So the 2-qubit system has joint state: $\begin{pmatrix}\alpha\\ \beta\end{pmatrix} \otimes \begin{pmatrix}\frac{1}{\sqrt{2}}\\ -\frac{1}{\sqrt{2}}\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}\alpha\\ -\alpha\\ \beta\\ -\beta\end{pmatrix}$

We know a CNOT gate has the form: $\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix}$

So after the CNOT gate, we have the state:

$$\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix} \cdot \frac{1}{\sqrt{2}}\begin{pmatrix}\alpha\\ -\alpha\\ \beta\\ -\beta\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}\alpha\\ -\alpha\\ -\beta\\ \beta\end{pmatrix} = \begin{pmatrix}\alpha\\ -\beta\end{pmatrix} \otimes \begin{pmatrix}\frac{1}{\sqrt{2}}\\ -\frac{1}{\sqrt{2}}\end{pmatrix}$$

Remarkably, by performing the CNOT, we have essentially changed the state of the control qubit, i.e. adding a phase to it. Therefore, the CNOT gate accomplishes the phase kickback.

An example of Bernstein-Vazirani algorithm with s=101 is given below with phase oracle built using an XOR oracle.

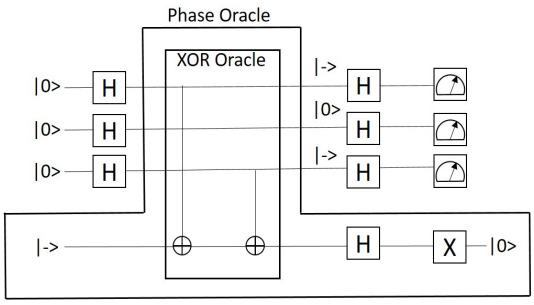

Figure 3: Example circuit for s=101

Here Bernstein-Vazirani algorithm only has polynomial speed up. But a modified version of this algorithm, Recursive Bernstein-Vazirani Algorithm has exponential speed up. Usually algorithms assume that the oracles can be implemented, but in many cases implementing oracles can be difficult. Because of the same we wont be discussing about the oracle in coming algorithms since its a given black box.

2 Simon's Algorithm

Simon's Algorithm [Sim97] is an exponential speed up algorithm in $\mathbb{Z}_2^n$.

Problem: given an oracle access to a function $f : \{0,1\}^n \rightarrow \{0,1\}^m$ $m \geq n$, with a promise that function f is "s-perioduc" for some secret string (s ϵ $\{0,1\}^n$).

The term s-periodic means $f(x \oplus s) = \sum_{i=1}^{n}(x_i + s_i)\ mod 2 = f(x)$

This can be thought of mapping to colors i.e. $f : \{0,1\}^n \rightarrow colors$. for example consider n=3 with f(x) given by f(000) = red, f(001) = green, f(010) = blue, f(011)= yellow, f(100) = yellow, f(101) = blue, f(110) = green, f(111) = red. One can notice that f(000) and f(111) has the same color, similarly f(110) and f(001) and others. These values differ by the addition of s mod 2. So here s is given by 111. This can be easily verified by adding 111 to any of the input and noting that the f(x+111) will have the same color as f(x). This relation can be better understood from the figure below.

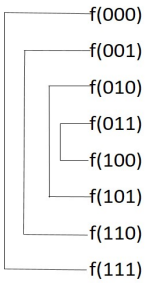

Figure 4: The mapping of function to colors and periodicity

Classically it is very hard to simulate this. The problem is analogous to finding two people in a room having same birthday. If there are n days in a year then $\sqrt{n}$ number of people are required to have a 50% chances of two people having same birthday. Here we are finding a pair having the same birthday which is similar to two values having same color. In our case we will have to make $\sqrt{N} = \sqrt{2^n}$ queries to get a collection with substantial probability.

Quantum circuit.

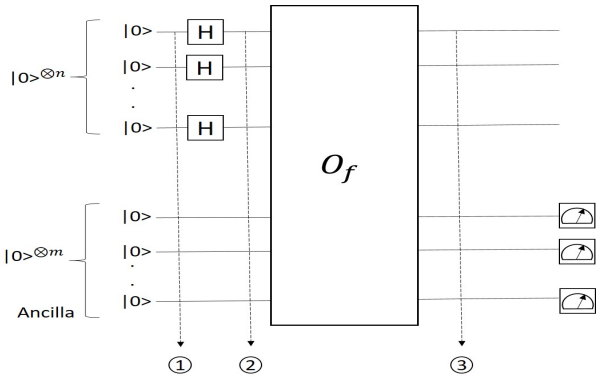

Figure 5: Quantum circuit for Simons Algorithm

The new idea Simon came up with is to measure the Ancilla. The output of the ancilla after the phase oracle will fall into one of the colors as we measure it.

① $|0\rangle^{\otimes(n+m)}$

② $(\frac{1}{\sqrt{2^n}} \sum_{x \epsilon \{0,1\}^n} |x\rangle) \otimes |0\rangle^{\otimes m}$

③ $(\frac{1}{\sqrt{2^n}} \sum_{x \epsilon \{0,1\}^n} |x\rangle) \otimes |f(x)\rangle$

Once we make a measurement f(x) falls into one of the states. Since f(x) is like a color, we can write it as $(\sum_{x:f(x)=c} |x\rangle) \otimes |c\rangle$ where $|c\rangle$ is the color that we measure upto some normalization constant. Since the period is two, then we have two states associated with the same color of the form $\frac{1}{2}(|x_c\rangle + |x_c + s\rangle) \otimes |c\rangle$. When we measure $|c\rangle$ we will either get $|x_c\rangle$ or $|x_c + s\rangle$ as the input also collapse into one of the corresponding states. But every time we might get a different color associated with the same state. To solve the issue we add Hadamard gate to the circuit as shown below.

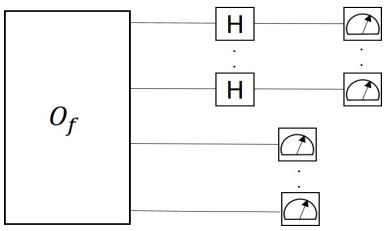

Figure 6: Circuit modification

The states can now be written as

$$H^{\otimes n}\big(\frac{1}{\sqrt{2}}|x_c\rangle + |x_c \oplus s\rangle\big) \otimes |c\rangle$$

The first part of the equation can be rewritten as

$$\begin{aligned}
&H^{\otimes n}\big(\frac{1}{\sqrt{2}}|x_c\rangle + |x_c \oplus s\rangle\big) \otimes |c\rangle \\
&= \frac{1}{\sqrt{2^n}}\big(\frac{1}{\sqrt{2}}\sum_{y\in\{0,1\}^n}(-1)^{x_c\cdot y}|y\rangle + \frac{1}{\sqrt{2}}\sum_{y\in\{0,1\}^n}(-1)^{(x_c\oplus s)\cdot y}|y\rangle\big) \\
&= \frac{1}{\sqrt{2}\sqrt{2^n}}\sum_{y\in\{0,1\}^n}(-1)^{x_c\cdot y}\big(1+(-1)^{s\cdot y}\big)|y\rangle
\end{aligned}$$

The last line is just factoring out the common phase from both terms. Now if we look at the term in the parenthesis: $(1 + (-1)^{s\cdot y})$, we observe that it equals to:

$$\begin{cases} 0 & \text{if } s\cdot y = 1 \\ 2 & \text{if } s\cdot y = 0 \end{cases}$$

Therefore only the states with non-zero coefficients are left, i.e. only those satisfying $s \cdot y = 0$ So the full state becomes:

$$\sqrt{\frac{2}{2^n}}\sum_{y:s\cdot y=0}(-1)^{x_c\cdot y}|y\rangle$$

Now we get a uniformly distributed y such that $s \cdot y = 0$. This is a linear equation with n unknowns. By repeating the process for n times, we can solve for s by normal linear algebra technique. Here we are solving Simons algorithm in binary boolean string (i.e. the field $\mathbb{Z}_2^n$). The integer version (i.e. $\mathbb{Z}_N$) is a subroutine of Shor's Algorithm.

3 Noisy Intermediate-Scale Quantum (NISQ) Algorithms

Here noisy refers to small number of qubits not enough for error correction. And the intermediate scale refers to having qubits ranging from 100 to 10000 and with a quantum volume

in the range of 100 qubit width and 1000 qubit depth. The two main NISQ algorithms that will be discussed are:

1. Variational Quantum Eigensolver (VQE) [PMS+14]
2. Quantum Approximate Optimization Algorithm (QAOA) [FGG14]

Both of the algorithms solve the same problem which is finding the lowest Eigenvalue of a hermitian matrix H. An n × n matrix is polynomial in n, but our interest which is $2^n \times 2^n$ is polynomial in exponential which is much harder to solve. One of the interesting hermitian matrix to work with is Hamiltonian. Hamiltonian is an operator which tells us about the energy of the system at time t. Since H is a hermitian matrix: that is, it real eigen values and orthonormal eigen vectors.

$$H\left|E_n\right\rangle = E_n\left|E_n\right\rangle$$

Out of all these eigen values, the one with lowest energy is called ground state energy. The challenge is to find this ground state of the $2^n \times 2^n$ matrix Hamiltonian.

MAXCUT/Clustering is another problem where we can use VQE to find the solution. In this problem we are required to maximize the weight of the edges crossing the cuts. A cut is made on an edge whenever it is connected to vertices of two different colors.

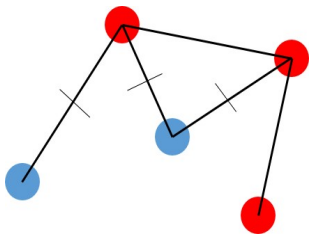

Figure 7: Schematic representation of maxcut, where we cut the edge whenever the connecting vertexes have different color.

This can be rewritten as as eigen value problem with Hamiltonian given by

$$H = \frac{1}{2}\sum_{\text{edges i,j}}(I - Z_i Z_j)$$

Here Z_i and Z_j are the paulis Z matrix for ith and jth vertex. The maxcut then would be same as the ground state eigen value of this Hamiltonian. For example we can consider just 2 edges connected by an edge, Then Hamiltonian is given by

$$H = \frac{1}{2}\left(\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix} - \begin{bmatrix}1&0\\0&-1\end{bmatrix}\otimes\begin{bmatrix}1&0\\0&-1\end{bmatrix}\right) = \begin{bmatrix}0&0&0&0\\0&-1&0&0\\0&0&-1&0\\0&0&0&0\end{bmatrix}$$

This matrix has eigen vector $|00\rangle$ (no crossing) with eigen value 0 and $|01\rangle$(with crossing) with eigen value -1. Here 0 corresponds to edges connecting the same color and -1 represents the one which connects different colors.

Variational Principle

Variational Principle says that for any given vector $|\psi\rangle$, $\langle\psi| H |\psi\rangle \geq E_0$
H is hermitian implies that H has a complete set of eigen vectors which are orthonormal to each other. Let $|E_0\rangle, |E_1\rangle, .. |E_n\rangle$ be the eigen vectors of H with Eigen values $E_0, E_1, ..E_n$ respectively with E_0 being the smallest. Then any given vectors can be written as a superposition of these eigen vectors i.e.

$$|\psi\rangle = C_0 |E_0\rangle + C_1 |E_1\rangle + ...C_n |E_n\rangle$$

So for any given $|\psi\rangle$, we have

$$\begin{aligned}\langle\psi| H |\psi\rangle &= (C_0 \langle E_0| + C_1 \langle E_1| + ...C_n \langle E_n|)H(C_0 |E_0\rangle + C_1 |E_1\rangle + ...C_n |E_n\rangle) \\ &= C_0^2 E_0 + C_1^2 E_1 + ...C_n^2 E_n \\ &\geq E_0\end{aligned}$$

So to find lowest eigenvalue, we keep checking E for different values of $|\psi\rangle$ till we get the minimum. In VQE we parametrize ψ using θ and find the value for θ which gives the smallest value for H which is similar to classical optimization. So it can be summarized as guess and check, where check is measuring the ¡H¿ and guess is using classical optimizer to select the next value of θ based on previous results. Here $|\psi(\theta)\rangle$ is the ansatz and we have two ways of getting the ansatz. There are two ways of getting the ansatz

1. Hardware/machine ansatz
2. Problem/Physics

Consider that you want to learn the ping pong. The above ansatz can be understood as different approaches that one can take to learn it. One strategy would be to go to a coach and learn from the first principles which is similar to 'Problem/physical ansatz'. The second approach is building up from what you already know, for example tennis. Both has its pros and cons and the idea is to select the ansatz based on the problem. If our choice of ansatz doesnt matter much, then we can go for the one which has the best machine performance. But if the choice of the ansatz matter then we have to make the circuit accordingly.

References

[BV97] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM Journal on computing*, 26(5):1411–1473, 1997.

[FGG14] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.

[PMS+14] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O'brien. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5:4213, 2014.

[Sim97] Daniel R Simon. On the power of quantum computation. *SIAM journal on computing*, 26(5):1474–1483, 1997.