Binary linear systems and the Bernstein-Vazirani Algorithm

Quantum Algorithms using Qniverse

Jothishwaran C A

Department of Electronics and Communication Engineering
Indian Institute of Technology Roorkee

Glossary

- **<u>Bit</u>:** A binary digit, a number that takes two values '0' and '1'. A bit of unknown value also called a **Boolean Variable.**

  The Bit also refers to the most fundamental unit of Classical Information. –> Information Theory

- **<u>Boolean Function</u>:** A function of one (or many) Boolean variables. Boolean functions are constructed using classical logic gates like AND, OR and NOT.

- **<u>Measurement</u>:** The only known method to extract information from a qubit. Measuring a single qubit yields a classical bit of information. The act of observing a qubit state is performed through a **Measurement**.

`.' → multiplication [context dependent]

Linear Systems: Introduction

- A linear system is defined as follows:

→ finite dimensions

$$A \cdot \boldsymbol{x} = \boldsymbol{b}$$ → solved using Gauss-Jordan elimination

↓

Complexity : O(n

Here, $\boldsymbol{x}, \boldsymbol{b}$ are vectors ($\boldsymbol{x}$ is unknown) and A is an operator defined over the vector space containing the aforementioned vectors.

- The above system has a unique solution if the rows of A are all linearly independent, i.e. The matrix must be of full rank or A is invertible.

- If $a_1, a_2, \dots, a_n$ are the rows of the matrix A, then the vector $\boldsymbol{b}$ contains the values:

$$\boldsymbol{b} = \begin{pmatrix} a_1 \cdot \boldsymbol{x} \\ a_2 \cdot \boldsymbol{x} \\ \vdots \\ a_{n-1} \cdot \boldsymbol{x} \\ a_n \cdot \boldsymbol{x} \end{pmatrix}$$

Linear systems are directly related to linear functions. $F(x) = a \cdot x$

Binary Linear Systems

Galois Field of order 2

- A linear system where the vectors and operator are defined over $GF(2)$ or binary numbers where the addition and multiplication are defined as follows:

$$0+0=1+1=0 \;\;;\;\; 0+1=1+0=1$$
$$0\cdot 0=1\cdot 0=0\cdot 1=0 \;\;;\;\; 1\cdot 1=1$$

$\Rightarrow$ add $\longleftrightarrow$ XOR gate

$\Rightarrow$ mult. $\longleftrightarrow$ AND gate

- In this case, the linear system has a direct connection to linear functions over binary vectors. Such functions are known as linear Boolean functions.

- A linear Boolean function is defined as $F(\boldsymbol{x}) = \boldsymbol{s}\cdot\boldsymbol{x} \;:\; \boldsymbol{s},\boldsymbol{x} \in \{0,1\}^n$ therefore if one is given an unknown linear Boolean function, one would need $'n'$ linearly independent values of the function to find out the value of $\boldsymbol{s}$.

$s = (s_1, s_2 \ldots s_n) \rightarrow$ Coefficient

$x = (x_1, x_2 \ldots x_n) \hookrightarrow$ Input

$\Rightarrow s\cdot x = s_1x_1 + s_2x_2 + \ldots + s_nx_n$

The Bernstein-Vazirani Problem

- The Bernstein-Vazirani problem is stated as follows: (not a decision problem)

s is unknown

Given oracle access to an unknown linear Boolean function, how easily can one find the 'secret' linear coefficient ***s*** *of the Boolean function.*

- The Bernstein-Vazirani algorithm offers a constant time quantum algorithm to solve the above problem.

$O(1)$

- Additionally, there is decision problem variant of the same problem, where the quantum algorithm can be shown to give a super-polynomial speed-up.

Both algos, classical & quantum are not in P

Linear Boolean functions: An example in $\{0,1\}^2$

Two bit functions

inputs $x = x_0 \; x_1$

x_0	x_1
0	0
0	1
1	0
1	1

Linear function $F(x) = S \cdot x$; $S \in \{0,1\}^2$ $S = 00, 01, 10, 11$

$$S \cdot x = \underbrace{00 \cdot x}_{L_0(x)} \text{ or } \underbrace{01 \cdot x}_{L_1(x)} \text{ or } \underbrace{10 \cdot x}_{L_2(x)} \text{ or } \underbrace{11 \cdot x}_{L_3(x)}$$

$$0 \text{ or } x_1 \text{ or } x_0 \text{ or } x_0 + x_1$$

Linear Boolean functions: An example in $\{0,1\}^2$

Two bit functions

Ignore

inputs $x = x_0 \, x_1$

x_0	x_1	$L_0(x)$		$L_1(x)$		$L_2(x)$		$L_3(x)$	
0	0	0	1	0	1	0	1	0	1
0	1	0	1	1	-1	0	1	1	-1
1	0	0	1	0	1	1	-1	1	-1
1	1	0	1	1	-1	1	-1	0	1

$L_1 - L_3$

→ Same number of 0's & 1's

→ The truth table has patterns

$$Y = (-1)^{L(x)}$$

Periodic sequences and the Fourier transform

Let there be a sequence of two numbers (not only bits)

$P: \; a_0 \; a_1$

$a_0 = a_1 \; ; \; \hat{a}_1 = 0$

or $a_0 > a_1 \; ; \; \hat{a}_1 > 0$

or $a_0 < a_1 \; ; \; \hat{a}_1 < 0$

or $a_0 = -a_1 \; ; \; \hat{a}_0 = 0$

$$\hat{a}_0 = \frac{a_0 + a_1}{\sqrt{2}}$$

$$\hat{a}_1 = \frac{a_0 - a_1}{\sqrt{2}}$$

$\hat{a}_0 \rightarrow$ average value of the sequence

$a_1 \rightarrow$ is the difference

$$P = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix}$$

$$\underbrace{\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}}_{M} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \hat{a}_0 \\ \hat{a}_1 \end{bmatrix} \Rightarrow M \cdot P = \hat{P}$$

M ↓ Hadamard matrix

$\hat{a}_0$ & $\hat{a}_1$

$$\hat{P} = \begin{bmatrix} \hat{a}_0 \\ \hat{a}_1 \end{bmatrix}$$

$Q = a_0 \; a_1 \; a_2$ for a sequence of Length N

$x^3 - 1 = 0$

$1, w, w^2$ we can create Fourier Basis using the roots of $x^N - 1 = 0$

Fourier transform for $N = 2^n$

for this special value of N

$(a_0 \quad a_1)(a_2 \quad a_3)$

$a_2 + a_3 = \hat{a}_2$

$\hat{a}_0 \quad \hat{a}_1 \quad \hat{a}_2 \quad \hat{a}_3$

$a_2 - a_3 = \hat{a}_3$

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{matrix} a_0 + a_1 + a_2 + a_3 \\ a_0 - a_1 + a_2 - a_3 \\ a_0 + a_1 - a_2 - a_3 \\ a_0 - a_1 - (a_2 - a_3) \end{matrix}$$

$$M_4$$

$$M_4 = \underset{\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}}{\underset{\downarrow}{M}} \otimes M = \begin{bmatrix} 1\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} & 1\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \\ 1\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} & -1\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \end{bmatrix}$$

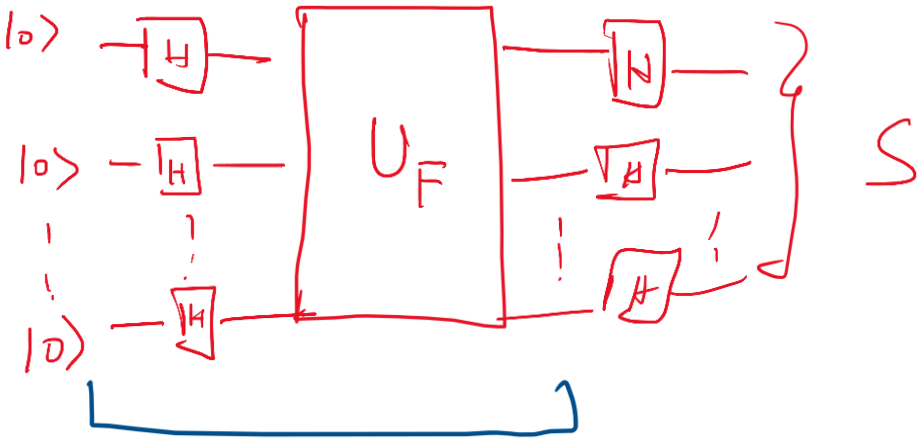
For $N = 2^n$; the new fourier basis

Walsh - Fourier basis
Walsh - Hadamard basis } Orthogonal basis defined by the linear functions over n-bits

Therefore, if there is a sequence

$P = (-1)^{L_i(x)}$, then the Walsh - Fourier Transform will give the value of 'i'

Bernstein-Vazirani Algorithm: Quantum Circuit



Sequence generation

$$U_F|x\rangle = \underbrace{(-1)^{F(x)}}_{Y(x)}|x\rangle$$

Reading Materials

→ Quantum Computing : Quantum Inf. & Comp. Nielson & Chuang

Quantum Computing : A gentle Intro. Reiffel & Pollack

→ Classical Crypto : Douglas Stinson