ICTP Winter College on Optics: Quantum Photonics and Information
February 2020
Lecture one: Elements of Classical Information Theory

Vahid Karimipour[1]

Department of Physics, Sharif University of Technology, Tehran, Iran,

Abstract

This note is intended for those participants of the school who have no prior exposure to classical information theory. It tries to teach the basic elements of classical information, namely the meaning of Shannon entropy and how it is related to the measure of information.

1 Information and surprise

Suppose a friend of you "informs" you of an "event" which is going to happen, something like "There has been an accident in the road" or "there is going to be a bus strike today". We ask "how much" information you gain by hearing this news. Intuitively the more surprised you become, the more information have gained. The information is more valuable if you have not "expected" that event. So denote the event by E and the probability that it may have happened by p and the information we gain by $h(p)$. Note that h should be a function of p and we expect that $h(p)$ be a decreasing function of p. Moreover we expect that when your friend gives you news about two independent events E and F, which we denote by the pair (E, F), the information that we gain is the sum of the information that we gain about the two individual events. The pair of events

[1]email:vahid.karimipour@gmail.com

occur with probability pq, where p is the probability of E and q is the probability of F. Then we expect that

$$h(pq) = h(p) + h(q). \tag{1}$$

The only function which satisfies this property is the log function and so we write

$$h(p) = \alpha \log p \tag{2}$$

where α should be a negative number in order for h to be a decreasing function of p. Like any other measure we should calibrate our measure of information and we calibrate it so that for a binary event with two outcomes with probability $1/2$ it gives one unit of information, that is

$$h(1/2) = 1 \qquad \longrightarrow h(p) = -\log_2\ p. \tag{3}$$

One unit of information is called one "bit". If your friend plays the same game with the random variable

$$X = \{x_1, x_2, \cdots x_N\} \qquad Prob(x_i) = p_i \tag{4}$$

and gives you the news about the outcomes, on the average, each time he or she will give you this much information to you

$$H(X) = -\sum_i p_i \log p_i. \tag{5}$$

This is called Shannon entropy or Shannon Information. The relation with entropy will become clear later. It is bounded as follows:

$$0 \leq H(X) \leq \log_2 N \tag{6}$$

2 How many questions

We can interpret H as the average number of binary questions (with Yes or No answer) that you must ask before you determine exactly one specific element of a set. For example take

$$X = \{000, 111\} \qquad H(X) = 1, \tag{7}$$

then with only one question as follows:

Q1: Is the first digit 0 or 1? you will determine the element. In the following example

$$X = \{000, 011, 101, 110\} \qquad H(X) = 2, \tag{8}$$

you will find the element with only two questions: Q1: Is the first digit 0 or 1?

Q2: Is the first digit 0 or 1?

3 Randomness

Another interpretation of H(X) is the amount of entropy or disorder and it comes (actually much sooner than Shannon) from Statistical Mechanics. We judge figure ()-a as an ordered pattern and figure ()-b as a disordered pattern. The reason is that there is only one configuration of type a (hence zero entropy) and millions of configurations of type b (hence maximum entropy).

4 How many physical qubits

The best justification for the definition of Shannon entropy comes from the number of physical bits that you need for communicating a message chosen from the given alphabet X. Consider the alphabet $X = \{A, B, C, D\}$ where all the letters appear with equal probability 1/4. We can encode these letters as follows

$$\begin{aligned} A &\longrightarrow 00 \\ B &\longrightarrow 01 \\ C &\longrightarrow 10 \\ D &\longrightarrow 11. \end{aligned} \tag{9}$$

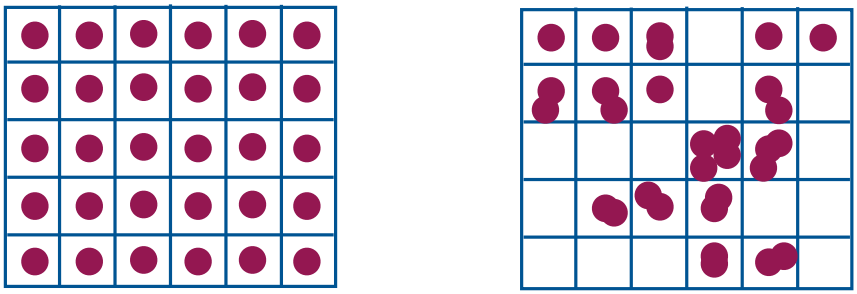

Figure 1: Left hand configuration is an (ordered) low entropy configuration, since there is only one way to distribute the balls in the cells so that each cell contains exactly one ball. The right hand side is (disordered) high entropy configuration, since there are a huge number of ways to distribute the balls so that each cell contains an arbitrary number of balls.

where the average number of bits necessary for encoding is 2. Now suppose that the probabilities are

$$P(A) = \frac{1}{2} \qquad P(B) = \frac{1}{4} \qquad P(C) = \frac{1}{8} \qquad P(D) = \frac{1}{8}. \tag{10}$$

Then we can encode the letters as follows

$$\begin{aligned} A &\longrightarrow 0 \\ B &\longrightarrow 10 \\ C &\longrightarrow 110 \\ D &\longrightarrow 111. \end{aligned} \tag{11}$$

where the average number of bits that we use is

$$\langle l \rangle = \sum_{i=1}^{4} l_i \times p_i = 1 \times \frac{1}{2} + 2 \times \frac{1}{4} + 3 \times \frac{1}{8} + 3 \times \frac{1}{8} = \frac{7}{4}, \tag{12}$$

which is exactly equal to $H(X)$. In fact

$$\begin{aligned} H(X) &= \sum_{i=1}^{4} p_i \log_2(\frac{1}{p_i}) = \frac{1}{2} \times \log_2(2) + \frac{1}{4} \times \log_2(4) + \frac{1}{8} \times \log_2(8) + \frac{1}{8} \times \log_2(8) \\ &= \frac{1}{2} \times 1 + \frac{1}{4} \times 2 + \frac{1}{8} \times 3 + \frac{1}{8} \times 3 = \frac{7}{4}. \end{aligned} \tag{13}$$

5 Compression of Classical Information; typical sequences

Suppose that you are tossing a coin one thousand times and you want to communicate to your friend the result of your throwing. If the coin is balanced, then you have to send him sequences of results like this

$$00101011101.....110001111000111. \tag{14}$$

There are 2^{1000} sequences like this and you need 1000 bits to fully communicate this information to your friend. Now suppose that your coin is a biased one with the following probabilities:

$$Prob(0) = 0.999 \qquad Prob(1) = 0.001. \tag{15}$$

Then sequences like (14) will rarely occur. Instead mostly "typical" sequences like

$$\begin{aligned} &10000 \cdots 00000 \\ &01000 \cdots 00000 \\ &\cdots \\ &\cdots \\ &\cdots \\ &00000 \cdots 00001. \end{aligned} \tag{16}$$

will occur and you need to just say which one of these sequences has occurred. In some cases none of these occur and then you say nothing. You can not convey any information in those cases. For doing this, you need only to say the number of the particular sequence 1, 2, 3, or 1000 and for this you only need $\log_2 1000 \sim 10$

bits. If the probability were as follows:

$$prob(0) = p, \qquad prob(1) = 1 - p \tag{17}$$

Then the average number of 0's and 1's are

$$N_0 \approx Np \qquad N_1 \approx N(1-p) \tag{18}$$

and the the number of "Typical Strings" is

$$\mathcal{Q} = \binom{N}{N_0} = \frac{N!}{N_0!(N-N_0)!} = \frac{N!}{(NP)!(N(1-p)!)} \tag{19}$$

Using Stirling's formula

$$\log N! \approx N \log N - N \tag{20}$$

you can prove that $\mathcal{Q} \approx e^{NH(X)}$. So you need on the average not N bits but $NH(X)$ bits which is smaller. The actual amount of information can be communicated by only $NH(X)$ bits.

- Problem: Prove this statement.

For more general alphabets we have

$$\mathcal{Q}_m = \frac{m!}{(mp_1)!(mp_2)!\cdots(mp_N)!} \tag{21}$$

and

$$\log_2 \mathcal{Q}_m = \log_2\left(\frac{m!}{(mp_1)!(mp_2)!\cdots(mp_n)!}\right) \approx m\left(\sum_{i=1}^{N} p_i \log_2 \frac{1}{p_i}\right) \equiv mH(X) \tag{22}$$

where

$$H(X) := \sum_{i=1}^{N} p_i \log_2\left(\frac{1}{p_i}\right) \tag{23}$$

and

$$\mathcal{Q}_m \approx 2^{mH(X)} \tag{24}$$

■ Problem: Prove this statement.

This means that Alice can just label these typical strings by their number which ranges from 1 to $\mathcal{Q}$ and can send this number which requires $mH(X)$ bits instead of m bits.

In practice, typical strings are not defined so tightly and strings where the actual number of each letter is off the average number by a few standard deviations are also labeled as typical. These strings are also encoded and communicated. It can be shown however that for very large blocks the number of these loosely defined typical strings scales as

$$2^{m\left[H(X)+\delta_m A(X)\right]} \tag{25}$$

where $A = -\sum_i \sqrt{p_i(1-p_i)}\log p_i$ and

$$\lim_{m\longrightarrow\infty} \delta_m = 0. \tag{26}$$

6 Properties of Shannon Entropy

Shannon entropy can be defined in a straightforward way for more than one variables:

$$H(X,Y) = -\sum_{x,y} p(x,y)\log_2 p(x,y) \tag{27}$$

For independent random variables, where $p(x,y) = p(x)q(y)$, we find

$$H(X,Y) = H(X) + H(Y) \tag{28}$$

In general however.

$$H(X,Y) \leq H(X) + H(Y). \tag{29}$$

For two random variables one can define the conditional probability (the probability that x happens provided that y has happend) as:

$$P(x|y) = \frac{P(x,y)}{P(y)}, \tag{30}$$

which comes from the obvious intuitive relation $P(x,y) = P(x|y)P(y)$. $P(x|y)$ is a new probability distribution for the variable X.

Entropy of X when we know that one specific $y \in Y$ has happened is now given by:

$$H(X|y) = -\sum_x \log p(x|y) \log p(x|y) \tag{31}$$

Averaging over y gives the conditional entropy, i.e. What is the entropy of X if we know Y.

$$H(X|Y) = \sum_y P(y) H(X|y) = \cdots H(X,Y) - H(Y) \tag{32}$$

Since $H(X,Y) \leq H(X) + H(Y)$, this last relation implies that

$$H(X|Y) \leq H(Y),$$

which is quite natural since it means that the entropy of X has decreased once we know something about another related variable Y. If the two random variables X and Y are independent, then $H(X|Y) = H(X)$ which is again plausible since knowing about Y gives no information about X.

Mutual information: It may happen that our knowledge about some variable Y gives us some information about another (possibly related) variable X. In such cases, knowing about Y reduces the uncertainly or entropy of X. For example, if we know that today is going to be rainy then there is a higher probability that there is a traffic jam in downtown. These two events are correlated. We can define the mutual information as the amount by which the entropy of X is lowered when we know Y. This is formally defined as

$$I(X:Y) := H(X) - H(X|Y) = H(X) - [H(X,Y) - H(Y)] = H(X) + H(Y) - H(X,Y). \tag{33}$$

	Probabilites	
	Head	*Tale*
1	1/6	0
2	0	1/6
3	1/6	0
4	0	1/6
5	1/6	0
6	0	1/6

	Probabilites	
	Head	*Tale*
1	1/12	1/12
2	1/12	1/12
3	1/12	1/12
4	1/12	1/12
5	1/12	1/12
6	1/12	1/12

Table 1: The four possible functions from $\{0, 1\}$ to $\{0, 1\}$.

Let us study a concrete example. Consider two different pairs of a dice and a coin which are tossed together. The probabilities are shown in the two tables below, (1). In the left hand side, if somebody tells us that the coin is Head, we are sure that the dice is odd. So there is a lot of mutual information between these two random variables (or two events). On the other hand if the probabilities are like the right hand table, then we gain no information about the oddness or evenness of the dice if we know that the coin is Head or Tail, in this case the mutual information is zero. Yet a third case is shown in table (**??**), where there is some mutual information between the two events.

- Problem: Calculate the mutual information about the coin and the dice for the above three tables. Do the same calculation if the event that you are interested in is not the exact number of the dice, but its parity (i.e. its evenness or oddness.)

In yet another case the probabilities are shown in table (2).

We will explore more properties of Shannon entropy later on. Many of the important and interesting properties of Shannon entropy are obtained from an inequality which we will discuss in the following exercise:

	Probabilites	
	Head	*Tale*
1	1/8	1/24
2	1/24	1/8
3	1/8	1/24
4	1/24	1/8
5	1/8	1/24
6	1/24	1/8

Table 2: The four possible functions from $\{0, 1\}$ to $\{0, 1\}$.

- Problem: Use the inequality $\log x \leq x - 1$ and then prove that for any two probability distributions $p(x)$ and $q(x)$,

$$-\sum_x p(x) \log p(x) \leq -\sum_x p(x) \log q(x). \tag{34}$$

Hints: 1-Draw a graph to prove the inequality $\log x \leq x - 1$ and then set $x = \frac{p}{q}$.

From this many of the properties of Shannon Entropy, stated above, can be derived. In each case, one has to choose an appropriate probability distribution $q(x)$.

- Problem: Use the result of the previous problem and show that $H(X) \leq \log_2 N$, where N is the size of the sample space of X (the number of different random values that X can take.)

- Problem: Use the result of the previous problem and show that $H(X, Y) \leq H(X) + H(Y)$

7 Classical Channels

Abstractly, a Classical channel has an input X and an output Y. It converts x to y by the conditional probability $P(y|x)$. $P(y|x)$ is the probability that an element x is turned into y due to error. Figure (**??**) shows the basic structure of a classical channel. Example: Figure () shows a binary symmetric channel.

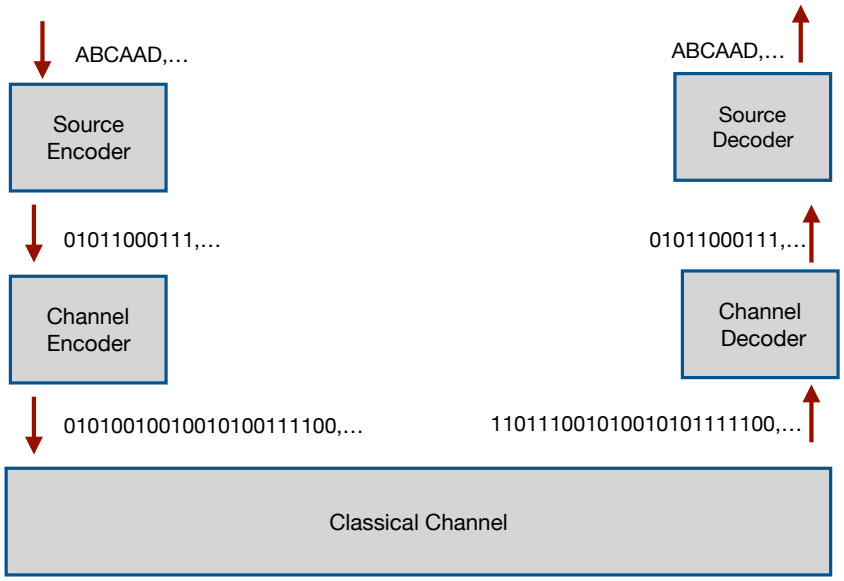

Figure 2: The scheme of a classical channel. The source encoder encodes the letters to sequences of 0's and 1's in an optimal way (Here the goal is to compress information and remove redundancy: Shannon's first theorem). The channel encoder encodes small blocks of 0's and 1's of size k into larger blocks of size n. (Here the goal is to combat the errors by adding sum redundancy deliberately: Shannon second theorem.)

- Simple repetition Code:

$$0 \longrightarrow 000 \qquad 1 \longrightarrow 111 \tag{35}$$

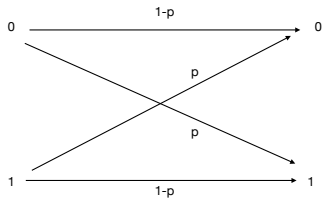

Figure 3: The simple and most common type of classical channel, the Binary Symmetric Channel in which each bit of 0 or 1 change with probability p.

Probability of error decreases from p to $\approx 3p^2$, at a price of reducing the rate R from 1 to 1/3.

$$P(error) = 3p^2(1-p) + p^3 \approx 3p^2 \tag{36}$$

$$R = \frac{k}{N} = \frac{1}{3}. \tag{37}$$

The basic parts of a classical channel are:

$$\text{Source Coding :} \qquad ABDUSSALAMICTP \longrightarrow 0010010011100100111..... \tag{38}$$

where the alphabet is encoded (compressed) into sequences of 0's and 1's. and

$$\text{Channel Coding :} \qquad (0010010011100)_k \longrightarrow (1001010001001010010101010101)_n \tag{39}$$

where blocks of length k of 0's and 1's are encoded into larger blocks of size n in order to correct possible errors if they occur. The philosophy is shown in figure (5). Any codeword can typically change into one of the words inside a sphere which contains $2^{nH(X)}$ words. If the receiver, receives one of these words, she can correct it back to the main codeword which sits at the center of the sphere. Error correcting is successful if there is enough space for all the spheres, that is if:

$$2^k \times 2^{nH(p)} \leq 2^n \tag{40}$$

or

$$k \leq n(1 - H(p)) \longrightarrow R \leq 1 - H(p) \tag{41}$$

Therefore we can do error-free communication if the rate R is less than $1 - H(p)$ which is called the capacity of the channel.

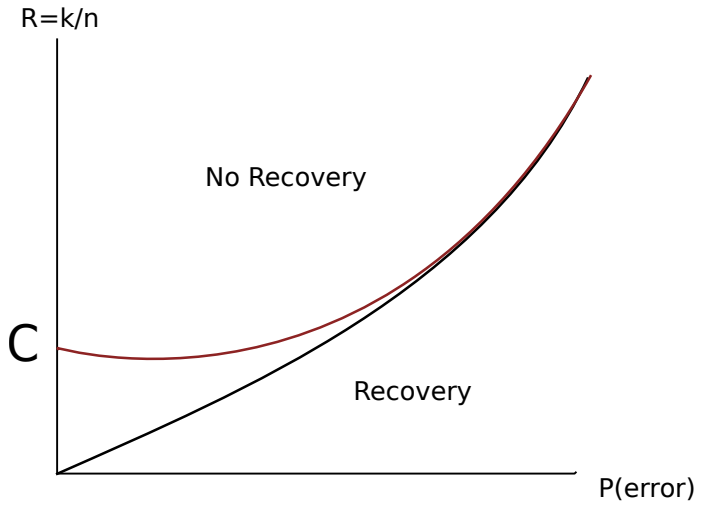

Figure 4: Using repetition code the to decrease the error probability decreases the rate of information transmission at the same time. According to Shannon's second theorem, there is a block coding such that for sufficiently large blocks, the error probability goes to zero, provided that the rate is below C, the capacity of the channel.

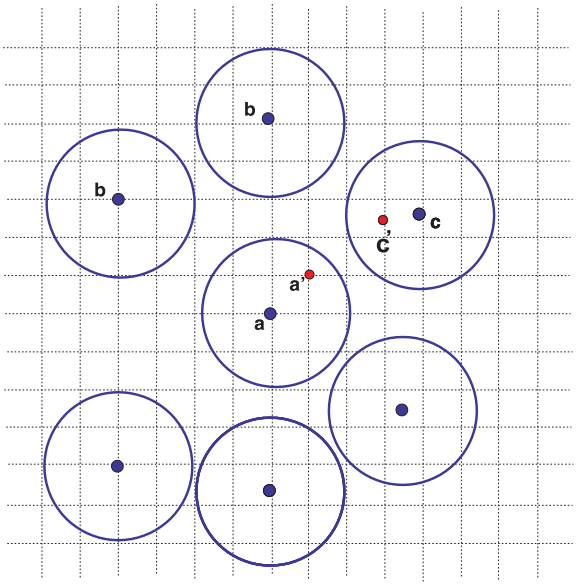

Figure 5: If in the receiver we receive a', we correct it to a. In each sphere there are a number of $2^{nH(p)}$ typical errors. Error free transmission is possible if the spheres do not overlap.

Lecture Two: Quantum Primitives, From Deutsch Algorithm to Quantum Teleportation

Abstract

This note is intended for those participants of the school who have not prior exposure to quantum computation. It tries to teach the basic elements of quantum computation, quantum circuits and introduces the simplest quantum algorithms. It also surveys quantum teleportation in its simplest setting and leaves quantum key distribution to other lectures, namely Cyber Security.

8 Classical Computation

A classical bit takes two values 0 and 1 from the set $\{0,1\}$. The set of values taken by n bits is $\{0,1\}^{\times n} = \{000..00, \cdots 111..1\}$. Ultimately the basic function of a classical computer is to calculate an arbitrary function $f : \{0,1\}^{\times n} \longrightarrow \{0,1\}^{\times m}$. Naturally any such function is the combination of functions of the form $f : \{0,1\}^{\times n} \longrightarrow \{0,1\}$. Such a function which has only two outputs ("Yes" or "No") defines what is called a "Decision Problem". "Decision Problems" are the most important problems in computer science. Examples are:

P1: Is a given integer N a prime number?

P2: Does a given number N has a factor less than K?

P3: Given a Boolean function $F(x_1, x_2, \cdots x_N)$, is there an input $(x_1, x_2, \cdots x_N)$ which satisfies F? This is called the satisfiability problem.

9 Classical Circuits

Consider a function $f : \{0,1\}^n \longrightarrow 1$. Then we can write

$$f(x_1, x_2, \cdots x_n) = \begin{cases} f(0, x_2, \cdots x_n) & \text{if} \quad x_1 = 0 \\ f(1, x_2, \cdots x_n) & \text{if} \quad x_1 = 1 \end{cases} \tag{42}$$

Therefore we can write

$$f(x_1, x_2, \cdots x_n) = \left[\overline{x_1} \wedge f(0, x_2, \cdots x_n)\right] \vee \left[x_1 \wedge f(1, x_2, \cdots x_n)\right] \tag{43}$$

This means that any computable function can be calculated using a circuit constructed from the classical gates

$$\text{AND} \equiv \wedge \qquad \text{OR} \equiv \vee \qquad \text{and} \qquad \text{NOT} \equiv \bar{} \qquad . \tag{44}$$

This is a universal set of classical gates for classical computation. To this set we should also add the gate fanout which copies each bit into two bits of the same value. This gate is forbidden in quantum computation.

10 Easy and hard problems

roughly any input x of a function has a size, if the number of steps needed for solving that problem grows like

$$T(N) \sim N^k \tag{45}$$

for some k, then we consider this problem as an easy or tractable problem, and if it grows like

$$T(N) \sim e^{\alpha N} \tag{46}$$

then this problem is considered a hard problem. NP hard problems are those for which we can easily verify a solution, once somebody finds this solution (i.e. Finding a needle in a haystack or the satisfiability problem). In contrast there are hard problems for which even verification of the solution is not easy (i.e. Finding the shortest hay in a haystack, or the traveling salesman problem.) The difficulty of these problems often arises from two factors:

1- the exponentially large size of the space of possible solutions, and

2- the absence of any physical or mathematical insight as to how to reduce the size of this space (i.e. no symmetry to reduce the ground space of a given Hamiltonian.)

The only way which remains is to explore the entire space of solutions. In problems where the space of potential solutions has been reduced and has a structure (due to symmetry), we can use classical parallelism. Essentially this means that the search space is broken into pieces so that each region is searched by one computer in parallel. However when the search space is exponentially large, this method becomes ineffective.

11 Quantum Computers

Both the input and output of a classical and quantum computers are classical. Their task is also the same, computing a function. The only difference is that the quantum computer has an exponentially large "Stomach" (to use an analogy) compared to the classical computer. This exponentially large stomach is the Hilbert space of N qubits whose dimension is 2^N and an arbitrary state $|\psi\rangle$ can simultaneously exist in all of these basis states, like

$$|\Psi\rangle = \sum_{x=0}^{2^N-1} \psi(x)|x\rangle. \tag{47}$$

Note that the space of N classical bits is also exponentially large, that is, there are 2^N possible configurations for N classical bits, but a classical computer cannot simultaneously exist in all these configurations. Therefore when a uniform superposition like $\frac{1}{\sqrt{N}}\sum_x |x\rangle$ is fed into the Quantum Computer (QC), they are all processes simultaneously. This is called Quantum Parallelism. This is shown in figure (**??**). The input state evolves by a big unitary operator (a quantum circuit) to the output state. The big unitary operator can be broken to one and two unitary qubit gates. Like classical circuit, there is also a universal set of gates (not unique) from which any big unitary operator can be built. If we want precise construction, the universal set i

$$\{U, CNOT\}, \tag{48}$$

where the controlled NOT gate acts as

$$CNOT|i, j\rangle = |i, i + j\rangle. \tag{49}$$

and by U we mean that the ability of building any single qubit gate. If we are consent with arbitrary precision, then one universal set of gates is

$$\{H, T, CNOT\} \tag{50}$$

where

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}. \tag{51}$$

Of course when we measure the output we obtain only one of the basis states by chance. The art of writing a quantum algorithm is to design the QC in such a way that the output holds the true solution with a high probability. Figure (**??**) shows the essential idea of a quantum algorithm. A uniform superposition of all the possible solutions is fed into the Quantum Computer (QC). We know that there is a solution x_0, although we do not know which one. However the quantum circuit is designed in a way which outputs a state so that by measuring the output state, we can find x_0 with high probability. All this is done without knowing x_0. We will see this in several examples.

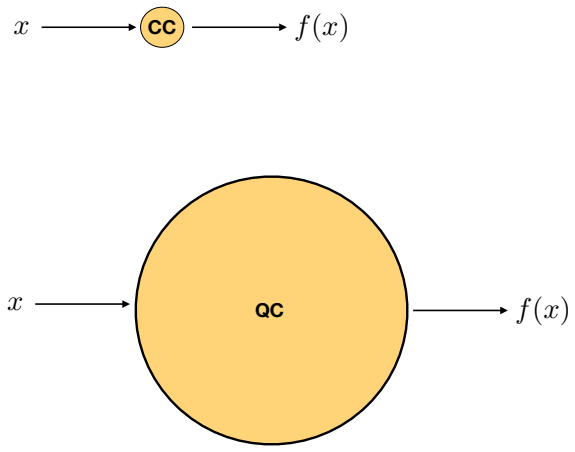

Figure 6: In both classical and quantum computers, the inputs and outputs are classical bits. To use an analogy, the only very big difference is that a classical computer has a very small stomach and the quantum computer has an exponentially large stomach.

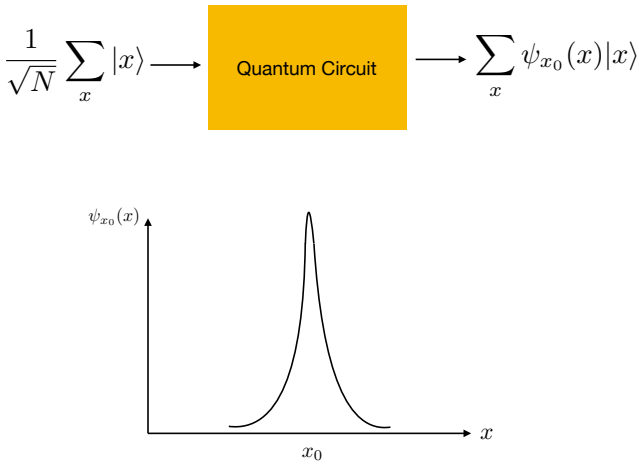

Figure 7: Quantum Parallelism: A uniform superposition of all the input data is processed simultaneously by a quantum computer. The quantum algorithm is designed so that upon measurement the probability of obtaining the correct answer (here x_0) is exponentially higher than the other answers.

12 Deutsch Algorithm

The first indication that quantum computers can do extraordinary tasks is due to the work of David Deutsch. Consider an oracle (A black box) which calculates a function $f : \{0,1\} \longrightarrow \{0,1\}$. You do not know which function the black box calculates and you are asked to determine the type of the function by just looking at the input and output. There are four different functions, indicated in the following table. Clearly you have to read the block box twice in order to determine the function. Let us now consider a simpler problem. Call the functions f_1 and f_4 constant functions and the functions f_2 and f_3 balanced functions. The problem is to determine whether the oracle computes a constant function or a balanced function. The invertible form of the function is defined by

$$|x, y\rangle \longrightarrow |x, y \oplus f(x)\rangle. \tag{52}$$

If we now feed the following state to the oracle, it evolves as follows:

$$|x,-\rangle \equiv \frac{1}{\sqrt{2}}(|x,0\rangle - |x,1\rangle) \longrightarrow \frac{1}{\sqrt{2}}(|x,f(x)\rangle - |x,f(x)+1\rangle) \equiv (-1)^{f(x)}|x\rangle|-\rangle. \tag{53}$$

which means that a superposition evolves as

$$\sum_x |x,-\rangle \longrightarrow \sum_x (-1)^{f(x)}|x\rangle|-\rangle. \tag{54}$$

Therefore the second register contains $|-\rangle$, and the first register contains $\sum_x (-1)^{f(x)}|x\rangle$ which is $|+\rangle$ for constant functions and $|-\rangle$ for balanced functions. Thus by measuring only the first register once, we can determine the type of function, whether it is constant or balanced.

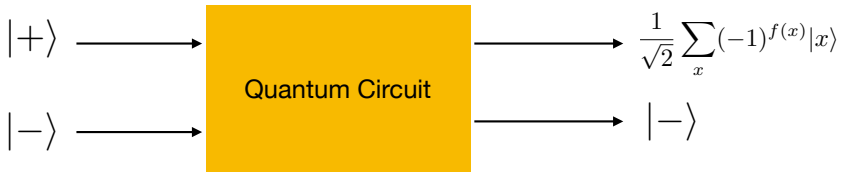

Figure 8: The first and the second register are prepared in $|+\rangle$ and $|-\rangle$ respectively. After calling the oracle, the second register is still in $|-\rangle$, but the first changes to $|+\rangle$ or $|-\rangle$ depending on whether the function is constant or balanced.

- prove that

$$H^{\otimes n}|\mathbf{x}\rangle = \frac{1}{\sqrt{N}} \sum_{\mathbf{y}=0}^{N-1} (-1)^{\mathbf{x}\cdot\mathbf{y}}|\mathbf{y}\rangle, \tag{55}$$

f_1	
0	0
1	0

f_2	
0	0
1	1

f_3	
0	1
1	0

f_4	
0	1
1	1

Table 3: The four possible functions from $\{0,1\}$ to $\{0,1\}$.

13 Deutsch-Jozsa Algorithm

The Deutsch-Jozsa algorithm is the generalization of the Deutsch algorithm to functions from many bits to one bit:

$$f : Z_2^{\times n} \longrightarrow Z_2 \tag{56}$$

The problem is still to find whether the oracle is calculating a constant function of a balanced function. The number of constant function is 2, however the number of balanced functions is huge. Referring to table () and imagining the generalization of this table to the present case, we see that it is equal to the number of different ways that you can distribute 2^{n-1} 0's in 2^n rows. Therefore the number of balanced functions is equal to $\binom{2^n}{2^{n-1}}$. Classically one should read the oracle a large number of times, i.e. on the average on the order of this huge number, in order to answer this question. Using the same algorithm (quantum circuit) as that of Deutsch (with appropriate modifications), we can answer this question by calling the oracle only once. We feed to the oracle a uniform superposition of all the inputs which evolves as follows:

$$\frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}} |\mathbf{x}\rangle \otimes |-\rangle \longrightarrow \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}} (-1)^{f(\mathbf{x})} |\mathbf{x}\rangle \otimes |-\rangle \longrightarrow \tag{57}$$

The second register remains at $|-\rangle$, but the first register carries the information about the function. For a constant function, this register is equal to $\frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}} |\mathbf{x}\rangle$ (modulo a phase) and for a balanced function it is again a uniform superposition but with half of the sings positive and the other half negative. If we now act by $H^{\otimes n}$ on this register we find that for a constant function we obtain $|00\cdots0\rangle$ and for a balanced function we obtain everything but $|00\cdots0\rangle$. Therefore by measuring the first register, we can determine the type of the function

by reading the oracle exactly once. To see this note that

- Exercise: Prove this last statement. Hint: Use the following equality.

$$H^{\otimes n}|\mathbf{0}\rangle = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}} |\mathbf{x}\rangle \tag{58}$$

In view of the large number of balanced functions one may ask why the Deutsch-Josza algorithm is not an exponential speed up compared to classical computers. Why it is the Shor algorithm and not the Deutsch-Josza algorithm which is hallmark of quantum algorithms? The answer is that there are "randomized classical algorithms" which can solve this problem with exponentially small probability of error. One just reads the oracle k times, i.e. feeds k random inputs into the oracle. If any two of them are different, one immediately declares (correctly) that the function is balanced. If all of them are equal, he declares the function to be constant (perhaps incorrectly). An error happens if the function is balanced and by bad luck one has picked up only those input variables whose output has been all zoro or one. The probability for this happening is

$$P_k(Error) = 2 \times (\frac{1}{2})^k = \frac{1}{2^{k-1}} \tag{59}$$

which exponentially goes to zero with k, which is the number of times one reads the oracle.

14 A glimpse at the Grover Algorithm

Along with Shor Algorithm (which we will not discuss), Grover algorithm is the most important quantum algorithm to date. Although it leads only to a quadratic speedup and not to an exponential one, it is very important due to the wide application of search in a wide variety of problems. The search problem can be formulated in the following general way. There is a function f such that it gives 0 for all input variables and

gives 1 only for one of the inputs w.

$$f(x) = \begin{cases} 0 & x \neq w \\ 1 & x = w \end{cases} \tag{60}$$

The problem is to find w. Classically we need on the order of $\frac{N}{2}$ times to call the function in order to find w. We only sketch the basic concepts related to this algorithm. First we invoke our previous familiar result that the operator which calls the function f, if it is cleverly called to act on $|\mathbf{x}\rangle|-\rangle$, acts as follows:

$$U|\mathbf{x}\rangle|-\rangle = (-1)^{f(\mathbf{x})}|\mathbf{x}\rangle|-\rangle \tag{61}$$

Therefore on a linear superposition $|s\rangle = \frac{1}{\sqrt{N}}\sum_{\mathbf{x}}|\mathbf{x}\rangle$ it only negates the phase of $|w\rangle$. Measuring the new state at this stage doesn't help us at all to determine w, since the probabilities are all the same. However we can now invert the state around its average $\overline{\psi} := \frac{1}{N}\sum_x \psi_x$. This is accomplished by the following unitary operator:

$$R := 2|s\rangle\langle s| - I \tag{62}$$

■ Exercise: Prove this statement.

Under this inversion the amplitudes change as:

$$\psi(x) \longrightarrow 2\overline{\psi} - \psi(x). \tag{63}$$

The Grover algorithm consists of applying the combination RU to the uniform superposition $|s\rangle$. The state changes as in figure (**??**) and it can be shown that after $O(\sqrt{N})$ operations (calling the function), the final state concentrate with a very high probability around $|w\rangle$.

What has been shown in these figures can be cast into mathematical form as follows: Let us separate the favorite state $|w\rangle$ and define the following normalized superposition

$$|r\rangle = \frac{1}{\sqrt{N-1}}\Big(\sum_{x_i \neq w}|x_i\rangle\Big). \tag{64}$$

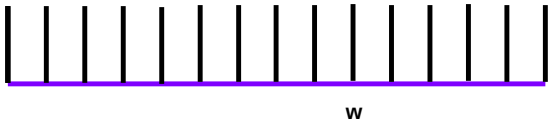

Figure 9: A uniform superposition of all the data is fed into the Grover Quantum Circuit.

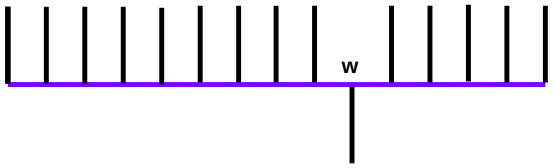

Figure 10: After reading the function f, the amplitude of w changes sign. But this is not enough for finding w by measurement.

This state is clearly orthogonal to $|w\rangle$. In the two dimensional subspace spanned by $|w\rangle$ and $|r\rangle$, we have a basis

$$|w\rangle = \begin{pmatrix} 1 \\ \\ 0 \end{pmatrix} \qquad , \qquad |r\rangle = \begin{pmatrix} 0 \\ \\ 1 \end{pmatrix} \tag{65}$$

and the state $|s\rangle$ can be expanded as

$$|s\rangle = \frac{1}{\sqrt{N}}|w\rangle + \sqrt{\frac{N-1}{N}}|r\rangle = \begin{pmatrix} \frac{1}{\sqrt{N}} \\ \\ \sqrt{\frac{N-1}{N}} \end{pmatrix}. \tag{66}$$

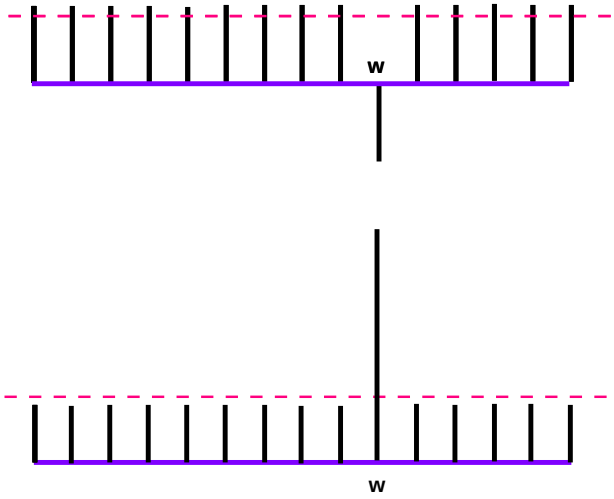

Figure 11: Inversion about the average (the dashed line), amplifies the amplitude of w and decreases the amplitude of other data. By repeating this process $O(\sqrt{N})$ times, the amplitude of w will become almost the only living amplitude.

The operator U_f (which calls the function f) is given by

$$U_f = I - 2|w\rangle\langle w| = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, \tag{67}$$

and the inversion around the average has the expression

$$R = 2|s\rangle\langle s| - I = \begin{pmatrix} \frac{2}{N} - 1 & \frac{2}{N}\sqrt{N-1} \\ \frac{2}{N}\sqrt{N-1} & 1 - \frac{2}{N} \end{pmatrix} \tag{68}$$

The Grover algorithm will then be

$$G = I_s U_f = \begin{pmatrix} 1 - \frac{2}{N} & \frac{2}{N}\sqrt{N-1} \\ -\frac{2}{N}\sqrt{N-1} & 1 - \frac{2}{N} \end{pmatrix}. \tag{69}$$

This is an orthogonal matrix and can be nicely parameterized as a simple rotation in the $w - r$ plane. Let us define

$$\cos\theta = 1 - \frac{2}{N}, \qquad \sin\theta = \frac{2}{N}\sqrt{N-1}, \tag{70}$$

Then

$$G = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}. \tag{71}$$

Since this is a rotation we find

$$G^m = \begin{pmatrix} \cos m\theta & \sin m\theta \\ -\sin m\theta & \cos m\theta \end{pmatrix} \tag{72}$$

and hence

$$\langle w|G^m|s\rangle = \frac{1}{\sqrt{N}}\cos m\theta + \sqrt{\frac{N-1}{N}}\sin m\theta. \tag{73}$$

Defining

$$\cos\alpha = \frac{1}{\sqrt{N}} \qquad , \qquad \sin\alpha = \sqrt{\frac{N-1}{N}} \tag{74}$$

we can write

$$\langle w|G^m|s\rangle = \cos(m\theta - \alpha). \tag{75}$$

This only means that if we take $m \approx \frac{\theta}{\alpha}$ we will find w with a very high probability.

- Exercise: Prove that $m \approx \frac{\pi}{4}\sqrt{N}$.

15 A glimpse at Shor's Algorithm

Given a large integer, the problem is to find an integer factor of this number. The steps of the algorithm are as follows:

Step 1: (Classical): Let $x \neq \pm$ be a number less than N such that

$$x^2 = 1 \quad mod\ N \tag{76}$$

, that is

$$x^2 - 1 = 0 \qquad mod\ N, \tag{77}$$

or

$$(x-1)(x+1) = kN \qquad for\ some\ k. \tag{78}$$

Naturally N has a prime factorization. Therefore the previous equation can be written in the form

$$(x-1)(x+1) = k \times p_1 p_2 p_3 \cdots p_L. \tag{79}$$

Since both $x-1$ and $x+1$ are less than N, none of them contains all the factors $p_1 p_2 p_3 \cdots p_L$ on the right hand side. The only possibility is that part of the factors come from $x-1$ and part of them come from $x+1$. So we use

Step 2: (Classical): Find the greatest common divisor of $x-1$ and N or $x+1$ and N. This can be done by the Euclid algorithm and is quite fast classically.

So the only problem is to find such an $x^2 = 1 \ \ mod\ N$. To do this, we choose a number $Y < N$ which is prime with respect to N (this can easily be done on a classical computer) and form the set

$$S = \{Y, Y^2, Y^3 \cdots\} \qquad mod\ N. \tag{80}$$

We now note that all the numbers in S are less than N (since everything is calculated mod N). Therefore S cannot be an infinite set and we will should have

$$Y^r = 1 \qquad for\ some\ j. \tag{81}$$

If $r = 2k$, then we have found our X which is Y^2. So the problem reduces to finding the period of the following function

$$f_Y(j) := Y^j \qquad mod\ N. \tag{82}$$

It is here that Shor's algorithm play the crucial role by using the quantum Fourier transform and finding the period of the function.

16 Quantum Teleportation

Up until now we have discussed only quantum computation. We can now briefly discuss a quantum communication task, namely teleportation. The goal of teleportation is to transfer the "unknown" state of an object to another object which is far away. The resource that we shall consume is entanglement. Let us see how this works for qubits. The scheme is depicted in figure (). The two persons or labs which are involved in this task are commonly called Alice and Bob and we will use this terminology.

Alice and Bob share a maximally entangled state $|\phi^+\rangle$. This state may have been sent to them by a third party a while ago.

$$|\phi^+\rangle = \frac{1}{\sqrt{2}}(|0,0\rangle + |1,1\rangle)_{ab} \tag{83}$$

This is one of the four maximally entangled states commonly called the Bell basis:

$$\begin{aligned} |\phi^+\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\ |\phi^-\rangle &= \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \\ |\psi^+\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \\ |\psi^-\rangle &= \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle), \end{aligned} \tag{84}$$

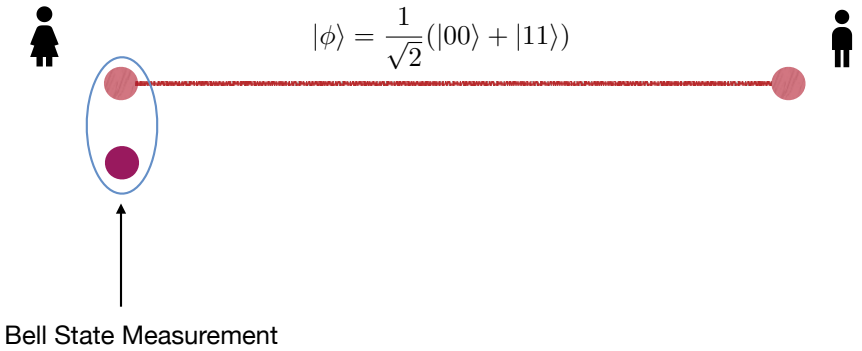

Figure 12: Alice and Bob share a maximally entangled state. Alice makes a Bell measurement on her two particles and sends the results of the measurement to Bob who will retrieve the state of the particle of Alice.

from which we find

$$\begin{aligned} |00\rangle &= \frac{1}{\sqrt{2}}(|\phi^+\rangle + |\phi^-\rangle) \\ |11\rangle &= \frac{1}{\sqrt{2}}(|\phi^+\rangle - |\phi^-\rangle) \\ |01\rangle &= \frac{1}{\sqrt{2}}(|\psi^+\rangle + |\psi^-\rangle) \\ |10\rangle &= \frac{1}{\sqrt{2}}(|\psi^+\rangle - |\psi^-\rangle). \end{aligned} \tag{85}$$

Alice has a particle in an unknown state $|\chi\rangle = a|0\rangle + b|1\rangle$ to her. She wants to create this state in Bob's lab. The combined state of Alice and Bob and Alice's particle is

$$\begin{aligned} |\Psi\rangle = |\chi\rangle|\phi^+\rangle &= (\alpha|0\rangle + \beta|1\rangle)_a \frac{1}{\sqrt{2}}(|0,0\rangle + |1,1\rangle)_{ab} \\ &= \frac{1}{\sqrt{2}}\left(\alpha|0,0,0\rangle + \beta|1,0,0\rangle + \alpha|0,1,1\rangle + \beta|1,1,1\rangle\right). \end{aligned} \tag{86}$$

Using () this state can also be expanded in the following form:

$$\begin{aligned} |\Psi\rangle &= \frac{1}{2}(|\phi^+\rangle(\alpha|0\rangle + \beta|1\rangle) + |\phi^-\rangle(\alpha|0\rangle - \beta|1\rangle) \\ &+ |\psi^+\rangle(\beta|0\rangle + \alpha|1\rangle) + |\psi^-\rangle(-\beta|0\rangle + \alpha|1\rangle)). \end{aligned} \tag{87}$$

Therefore if Alice makes a measurement on her two particles the state of Bob collapses to one of the following four states. The operator in front of each state is the operator which Bob should act on his collapsed state to recover the original state sent by Alice. To see which operator he should use, Alice needs to send him the result of her measurement. This can be done by sending to him only two classical bits.

$\|\phi^+\rangle$	$\alpha\|0\rangle + \beta\|1\rangle$	I
$\|\phi^-\rangle$	$\alpha\|0\rangle - \beta\|1\rangle$	Z
$\|\psi^+\rangle$	$\beta\|0\rangle + \alpha\|1\rangle$	X
$\|\phi^+\rangle$	$-\beta\|0\rangle + \alpha\|1\rangle$	Y

(88)