Information: From Classical to Quantum

Jiacan Yu
Kapitza Society,University of Rochester

April 22, 2022

Abstract

This paper introduces the concept of entropy and shows how it can help with answering the following questions: how much a piece of message can be compressed, how much information we get if we receive a message from a noisy channel, and what the maximum rate of transmission of information is with a noisy channel. Then we will generalizes entropy from the context of classical information to quantum information. We will see that assuming the initial quantum system is not entangled with the environment, the total entropy will increase.

1 Introduction

A message is a string of letters chosen from $\{a_1, a_2, ..., a_k\}$ independently, with priori probability $p(a_i)$. For example, let's say Alice and Bob speak a language that has an alphabet of 4 letters: $\{a, b, c, d\}$. In this language, a occurs with probability 1/2, b occurs with probability 1/4, c and d occurs with probability 1/8. Then any thing written in this language can be considered as a message. If the letters in a message follow the priori probability distribution very well, this message is called a typical string. Now suppose Alice send Bob a message. Bob don't know anything about this message, and he is only allowed to look at the first letter. If Bob sees the first letter is a, he gets 1 bit of information, because now the probability space of the possible messages $\frac{1}{2^1}$ of the probability space before Bob looks. If Bob sees the first letter is d, he gets 3 bits of information, because now the size of probability space decreases by $\frac{1}{2^3}$. We can see that although the number of letters that Bob sees is the same, the amount of information he gets is different. We can also see that two messages of the same length may have different amount of information. We may ask: is it possible to compress messages, that is, can we find a way to code them while the information contained is not changed?

2 How Much a Piece of Message can be Compressed

Consider strings chosen from $\{0, 1\}$ with $p(0) = 1 - p$ and $p(1) = p$. Let's say we want to compress these strings. When they are very long, by the law of large number, almost all of them are typical. So to compress them, we only need to code the typical strings. A typical string of length n will contain about np 1's. Therefore the number of typical strings is around $\binom{n}{np}$. Taking log base two and applying Sterling approximation (all the log in the paper means log base two), we get:

$$log\binom{n}{np} \approx nH(p) \tag{1}$$

where

$$H(p) \equiv -plog(p) - (1-p)log(1-p) = -\langle log\, p\rangle \tag{2}$$

$H(p)$ is entropy. There are about $2^{nH(p)}$ typical strings, each occurring with equal probability. To code them with a binary code, we need around $2^{nH(p)}$ distinct codes, so the length of the codes should be around $nH(p)$. Since $0 \leq H(p) \leq 1$ for $0 \leq p \leq 1$, we say it is possible to compress most of messages. In fact, $H(p) = 1$ only when $p = 1/2$. That is, only when $p = 1/2$, we still need n letters to specify a message. In other cases, a message can be specified by a shorter code. This is where the idea of compressing messages is brilliant: most of useful messages are long, so by theorem, they will follow the priori probability distribution very well. Therefore, just code the typical strings is enough.

If the message is not binary, it can be shown in a similar way that the number of typical strings is

$$2^{nH(X)} \tag{3}$$

where

$$H(X) \equiv \sum_x -p(x) log\, p(x) = -\langle log\, p(x) \rangle \tag{4}$$

$H(X)$ is entropy. Since we need $nH(X)$ bits of information to specify a message, entropy measures our priori ignorance per letter. To see this better, we can look at the probability of a message and apply the central limit theorem.

Let x be a n-letter message $x_1 x_2 ... x_n$. It occurs with a priori probability

$$P(x) = p(x_1)p(x_2)...p(x_n) \tag{5}$$

Take a log base two on both sides,

$$log\, P(x) = \sum_i^n log\, p(x_i) \tag{6}$$

Multiply by -1/n,

$$-\frac{1}{n} log\, P(x) = -\frac{1}{n} \sum_i^n log\, p(x_i) = -\langle log\, p(x) \rangle = H(X) \tag{7}$$

Then the central limit theorem tells that: if n is large enough, for any $\delta > 0$, we have

$$H(X) - \delta < -\frac{1}{n} log\, P(x) < H(X) - \delta \tag{8}$$

and the total probability of typical strings exceeds $(1 - \epsilon)$ $\forall \epsilon > 0$. Or, in other words, typical strings occur with probability $1 - \epsilon$, and each typical string has probability $P(x)$ such that

$$2^{-n(H-\delta)} \geq P(x) \geq 2^{-n(H+\delta)} \tag{9}$$

Let $N(\epsilon, \delta)$ be the number of typical strings. Since the total probability of typical strings is between 1 and $1 - \epsilon$, we have:

$$1 > 2^{-n(H-\delta)} N(\epsilon, \delta) \geq \sum_x P(x) \geq 2^{-n(H+\delta)} N(\epsilon, \delta) > 1 - \epsilon \tag{10}$$

Hence, we have a bound for $N(\epsilon, \delta)$:

$$2^{n(H-\delta)} > N(\epsilon, \delta) > (1 - \epsilon) 2^{n(H+\delta)} \tag{11}$$

To encode a message with a binary code, its length needs to be $n(H + \delta)$

3 How Much Information we Get if we Receive a Message from a Noisy Channel

Now suppose we have a noisy channel: if a string x is sent via the channel, there will be some independent random error during the transmission, so when x is received, it becomes y. Suppose we are very familiar with the properties of the channel so that we know $p(x|y)$. We want to answer the question: how much we know about x when y is received. We can't say for sure that x is y, since it is not very probable that no error happens during the transmission. But we do know more about x, because x is likely a string that is the same with y except for a few letters that are changed by error. In fact, we can update the distribution of x using Bayes rule.

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \tag{12}$$

where $p(y|x)$ and $p(x)$ are already known, and $p(y)$ can be computed by

$$p(y) = \sum_x p(y|x)p(x) \tag{13}$$

With this new distribution of x we can define conditional entropy similar with how we define entropy before:

$$H(X|Y) \equiv -\langle log\, p(x|y)\rangle \tag{14}$$

Now just $nH(X|Y)$ more bits of information is need to specify x. Hence, the information about x that I gain when I learn y is quantified by how much ignorance is reduced by learning y.

$$I(X:Y) \equiv H(X) - H(X|Y) \tag{15}$$

This is how $I(X:Y)$, mutual information, is defined. It is easy to see that if X and Y are independent random variables, $I(X:Y) = 0$

4 Maximum Rate of Transmission of Information Using a Noisy Channel

We just showed that although the channel is noisy, the receiver can still get some information. So if we encode our message with a longer code, which is supposed to be more robust to errors, the receiver should be able to specify the message with certainty. What is the lower bound of the bits of the code such that the receiver is still able to decode the message without error? Suppose the message we want to send is binary. Suppose the channel flips a bit with probability p. If we encode a k bits message with more bits, say n bits. We define the rate of the code as

$$R \equiv \frac{k}{n} \tag{16}$$

For a n-bit encoded input, the channel will make about np errors, so the input string diffuses to an output string that is randomly chosen from in the error sphere centered at the input string that contains about $\binom{n}{np} \approx 2^{nH(p)}$ output strings. The left figure of fig. 1 shows how this can be visualized.

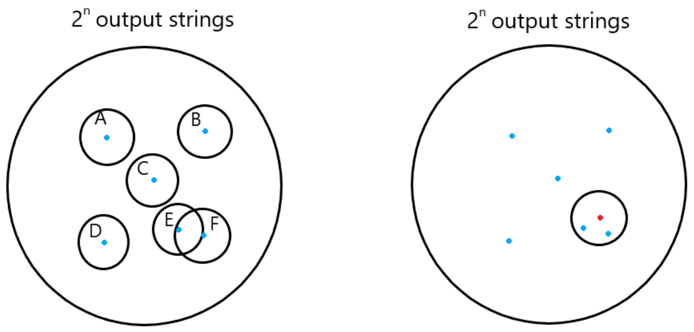

Figure 1: On the left figure, the biggest circle represents the output space consisted of all the possible n-bit output strings. The blue points are the n-bit strings that are chosen randomly to encode those binary k-bit messages. They are called input strings or codewords. For simplicity these codewords are represented by letters A, B, C, and so on. The smaller circles represent the error spheres that contains all the output strings that are the same with the codewords except for about np bits that are flipped due to the errors of the channel. The right figure shows the received string Y and its Hamming sphere. The blue points still represent the codewords.

If two error spheres overlap, and if the output string happens to be a string inside the overlapped region, we will not be able to figure out what the message is with certainty, since the message could be any one of the messages whose error sphere contains the output string.

We want to find the upper bound of the rate for this channel. We can start by establishing restrictions on the rate. Without considering that the error spheres might overlap, an obvious restriction is that the "volume" of output space should be larger than the sum of the "volume" of the all the error spheres. The "volume" of the output space means the number of all the possible output strings, which is 2^n. The "volume" of the error sphere means the number of strings in the error sphere, which is about $2^{nH(p)}$. Since we want to encode 2^k or 2^{nR} messages, the number of error spheres is 2^{nR}. Hence, the restriction is

$$2^{nH(p)}2^{nR} \leq 2^n \tag{17}$$

It is easy to see this means

$$R \leq 1 - H(p) \equiv C(p) \tag{18}$$

where we define $C(p)$ as the channel capacity. We will see that the rate of a code can be made as close to the channel capacity as wanted.

Suppose we randomly choose codewords from the output space to encode the messages. This is not a good way to encode messages, but it helps. Suppose the message E is sent and the output string is Y, as indicated by the red dot in the right figure. To decode, we draw a Hamming Sphere around the output string Y that contains all the strings that are the same with Y except for $np + \delta$ bits. The volume of the Hamming sphere is $2^{n(H(p)+\delta}$. Since this volume is slightly larger than that of the error sphere, when n goes to infinity, we will always find one or more codewords in the Hamming sphere. If there is exactly one codeword in the Hamming sphere, then it has to be the message that is sent. If there is more than one codeword in the Hamming sphere, we have a decoding error. How large is the the probability of decoding error? The Hamming sphere occupies a fraction f of the whole output space.

$$f = \frac{2^{n(H(p)+\delta)}}{2^n} = 2^{-n(C(p)-\delta)} \tag{19}$$

Since the codewords are chosen randomly, the probability for a particular codeword aside from E falling in the Hamming sphere is f. The probability that one or more than one of the $2^{nR} - 1$ codewords aside from E falling in the Hamming sphere is smaller than

$$2^{nR} f = 2^{-n(C(p)-R-\delta} \tag{20}$$

Since δ is very small and n is very large, R can be chosen very close to C, while probability of decoding error is exponentially small.

What we have shown is that there exist a randomly generated code such that the probability of decoding error is very small just for some codeword E and for some output string Y that belongs to its error sphere. What about other output strings and codewords? *In fact, since the codewords are chosen randomly, their distribution should be even in the output space.* This means what we just proved should not be a special case. The average probability of decoding error over all the outputs and codewords should be similar with the probability of decoding error given the output is Y and the message sent is E, which is small. Let p_i denote the probability of decoding error when codeword i is sent. For any positive ϵ and a n large enough, there exist a randomly chosen code such that

$$\frac{1}{2^{nR}} \sum_{i=1}^{2^{nR}} p_i \leq \epsilon \tag{21}$$

There may be some codewords that have very large probability of decoding error, which is not wanted. But we can throw away these codewords so that the probability of decoding error of EACH codeword is smaller than 2ϵ, and we will see that this will barley decrease the rate of the code.

Let $N_{2\epsilon}$ denote the number of codewords with $p_i > 2\epsilon$,

$$\frac{1}{2^{nR}}(N_{2\epsilon})2\epsilon < \frac{1}{2^{nR}} \sum_{i=1}^{2^{nR}} p_i \leq \epsilon \Rightarrow N_{2\epsilon} < 2^{nR-1} \tag{22}$$

Throwing the $N_{2\epsilon}$ bad codewords away, the rate of the new code becomes

$$\text{Rate} = R - \frac{1}{n} \tag{23}$$

When n is very large, the new rate is close to R, which is close to the channel capacity.

5 Von Neumann Entropy

Now we generalize entropy from the context of classical information to quantum information. Imagine a source that prepares messages of n "letters", where each "letter" is chosen from an ensemble of quantum states $\{\boldsymbol{\rho}(x)\}$ with a priori probability $p(x)$. If the observer is ignorant about what a letter is, the letter is represented by a density matrix

$$\boldsymbol{\rho} = \sum_x p(x)\boldsymbol{\rho}(x) \tag{24}$$

It can be diagonalized as

$$\boldsymbol{\rho} = \sum_a \lambda_a |a\rangle \langle a| \tag{25}$$

The Von Neumann Entropy $H(\boldsymbol{\rho})$ is defined as the negative of the expectation of the log of the probability distribution, which is the same with the entropy in classical information. (Here the log is still log base 2)

$$H(\boldsymbol{\rho}) \equiv -\sum_a \lambda_a log \lambda_a = -tr(\boldsymbol{\rho} log \boldsymbol{\rho}) \tag{26}$$

Von Neumann Entropy plays many roles. First, it is the minimum number of qubits per letter needed to encode the information. Second, it is the maximum amount of classical information per qubit that we can gain about the preparation by making the best possible measurement. Third, it quantifies the entanglement between quantum systems.

We will introduce three of its mathematical properties.

First, if $\boldsymbol{\rho}$ is a pure state, $H(\boldsymbol{\rho}) = 0$

$$\boldsymbol{\rho} = |a\rangle \langle a| \Rightarrow H(\boldsymbol{\rho}) = -1log1 = 0 \tag{27}$$

Second, entropy does not change under unitary evolution. This is because unitary evolution is basically a change of basis. Since $H(\boldsymbol{\rho})$ only depend on the eigenvalues, it is not changed by unitary evolution.

Third, sbuadditivity.

$$H(\boldsymbol{\rho}_{AB}) \leq H(\boldsymbol{\rho}_A) + H(\boldsymbol{\rho}_B) \tag{28}$$

where $\boldsymbol{\rho}_A$ and $\boldsymbol{\rho}_B$ are obtained by taking the corresponding partial trace of $\boldsymbol{\rho}_{AB}$. This is analogous to classical information in which

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

Figure 2 helps to see this inequality.

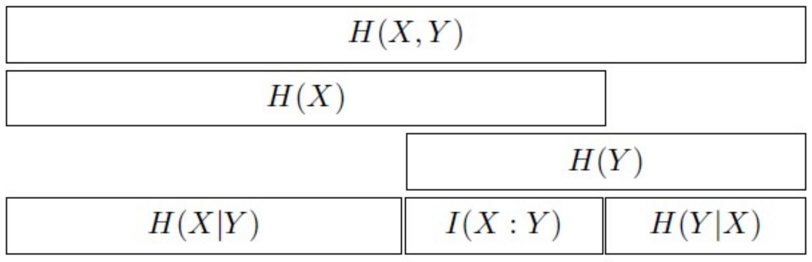

Figure 2: A helpful picture that shows the relationship between different kinds of entropy and mutual information. This picture is from [1] *PHY265 Lecture Notes: Introducing Quantum Information* by professor Alice Quillen.

The equation holds when $\boldsymbol{\rho}_{AB} = \boldsymbol{\rho}_A \otimes \boldsymbol{\rho}_B$, that is, when system A and B are entangled.

Assuming a system A and its environment E are initially unentangled, the second law of thermodynamics can be derived. In the beginning, the entropy of the whole system is

$$H(\boldsymbol{\rho}_{AE}) = H(\boldsymbol{\rho}_A) + H(\boldsymbol{\rho}_E) \tag{30}$$

Let the system evolve:

$$\boldsymbol{\rho}_{AE} \longrightarrow \boldsymbol{\rho}'_{AE} = \boldsymbol{U}_{AE}\boldsymbol{\rho}_{AE}\boldsymbol{U}_{AE}^{-1} \tag{31}$$

Since entropy does not change under unitary evolution, $H(\boldsymbol{\rho}_{AE}) = H(\boldsymbol{\rho}'_{AE})$, but in most of the time, the system and the environment become entangled.

$$H(\boldsymbol{\rho}_A) + H(\boldsymbol{\rho}_E) = H(\boldsymbol{\rho}'_{AE}) \leq H(\boldsymbol{\rho}'_A) + H(\boldsymbol{\rho}'_E) \tag{32}$$

If we think of the sum of the entropy of system A and its environment E as the total entropy, then we see that the entropy never decrease.

References

[1] Alice Quillen. *PHY265 Lecture Notes: Introducing Quantum Information*. University of Rochester. (2022).

[2] John Perskill. *Lecture Notes for Physics 229: Quantum Information and Computation*. California Institute of Technology. (1998).

[3] John Perskill. *Quantum Information Chapter 10. Quantum Shannon Theory*. California Institute of Technology. (2018).