CMSC 657: Introduction to Quantum Information Processing
Lecture 23

Instructor: Daniel Gottesman

Fall 2024

1 Entropy

Next we will talk about other kinds of manipulation one can do with quantum information. One of the basic tools of the trade is entropy. Actually, there are many different kinds of entropy, but the most common is the von Neumann entropy.

Definition 1. *Given a density matrix ρ, the* von Neumann entropy *is $S(\rho) = -\operatorname{Tr} \rho \log_2 \rho$.*

The function $\log \rho$ can be calculated either by diagonalizing ρ or by using the power series expansion of log. Note that when ρ has a 0 eigenvalue, $\rho \log \rho$ is still well defined (and 0 on the null subspace) because x goes to 0 faster than $\log x$.

When ρ is a thermal state, $S(\rho)$ is equal to the thermodynamic entropy of the system. $S(\rho)$ is also the natural generalization of a classical concept from information theory, the *Shannon entropy*: Suppose ρ is diagonal, with entries p_i, $\sum_i p_i = 1$ (since $\operatorname{Tr} \rho = 1$). This represents a classical probability distribution. Then $S(\rho) = -\sum_i p_i \log_2 p_i = H(\{p_i\})$. The Shannon entropy quantifies the amount of "information" in a probability distribution, meaning, roughly speaking, the amount that you learn by seeing the outcome of a random sample from the distribution. If the distribution is highly peaked, you don't need to see the outcome to have a pretty good guess at what happened, but if it is very spread out, you do.

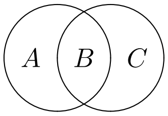
From now on, log is always base 2.

Frequently we have a composite system composed of subsystems $A, B, \ldots$. We may talk about the entropy of a subsystem $S(A) = S(\rho_A)$, where $\rho_A = \operatorname{Tr}_{B,C,\ldots} \rho$ is the density matrix of the subsystem. We can also talk about the entropy of multiple subsystems, such as $S(A, B)$, the entropy of the density matrix traced over all subsystems other than A and B.

The Von Neumann entropy has a number of useful properties, some of which are listed below:

1. On a Hilbert space of dimension D, $S(\rho) \leq \log D$. This is achieved when $\rho = I/D$.
2. $S(\rho) = 0$ iff ρ is a pure state.
3. **Triangle inequality or Araki-Lieb inequality:** $S(A, B) \geq |S(A) - S(B)|$.
4. **Subadditivity:** $S(A, B) \leq S(A) + S(B)$. Note: *Sub*-additivity, it could be less if the subsystems are correlated, and LHS can be 0 when the systems are entangled.
5. **Strong subadditivity:** $S(A, B, C) \leq S(A, B) + S(B, C) - S(B)$.
6. For a global pure state of A and B combined, $S(A) = S(B)$.
7. $S(\rho \otimes \sigma) = S(\rho) + S(\sigma)$.

Number 6 follows from 2 and 3: $S(A, B) = 0 \geq |S(A) - S(B)|$, so $S(A) = S(B)$.



The Venn diagram can be a mnemonic for strong subadditivity: AB and BC have an overlap in B, which is counted twice, so we subtract that off.

2 Compression

2.1 Classical Data Compression

Again, Alice wants to send information to Bob over a communications channel. This time the channel is perfect (no noise) and Alice wants to minimize the number of bits she has to send over the channel. (Again, this is useful for storage as well as communication.) If Alice's original message is n bits long, there are 2^n possible messages she could be sending. If all of them are equally likely, she has no choice but to send all n bits. However, if some messages are more likely than others, Alice can *compress* the information to transmit fewer bits on average.

For instance, suppose there are 4 possible classical messages. A occurs with probability 1/2, B with probability 1/4, and C and D each occur with probability 1/8. Since there are 4 messages, Alice would need 2 bits to send an arbitrary message without compression. However, suppose Alice encodes her messages, with the more common messages being shorter. For instance, suppose she sends the following:

$$A \longrightarrow 0 \quad (1)$$
$$B \longrightarrow 10 \quad (2)$$
$$C \longrightarrow 110 \quad (3)$$
$$D \longrightarrow 111. \quad (4)$$

Then the average number of bits Alice sends for the message is

$$\frac{1}{2}\cdot 1+\frac{1}{4}\cdot 2+\frac{1}{8}\cdot 3+\frac{1}{8}\cdot 3=\frac{7}{4}. \quad (5)$$

That is, Alice saves 1/4 of a bit on average by using this encoding. If she is sending only one message of this type, there is a good chance she will end up using more than 2 bits. However, if Alice is sending many messages drawn from this same distribution, the chances are very high she will use close to the average number of bits for the full transmission.

Notice that in this example, there is a 50% chance that the first bit Alice sends is 0. If Alice's first bit is 0, the second bit is the first bit of a new message, which also has a 50% chance of being 0, and if Alice's first bit transmitted is 1, there is a 50% chance that the second bit is 0 (when the message is B). Similarly, the third bit has a 50% chance of being 0 regardless of the values of the first two bits. The upshot is that when Alice sends n such messages, the compressed version is about $7n/4$ bits, and all bit strings of this length are equally likely. This is perfect compression, which can't always be achieved, but the goal is to reassign messages to strings to use all the possibilities as equally as possible.

In *block coding*, a standard type of compression scheme, Alice will collect many messages from the same source (i.e., messages each distributed according to the same probability distribution $\{p_i\}$ and independent of each other — a case known as i.i.d., independent identically distributed). For the *typical* sets of messages, a set which collectively has a high probability of occurring, Alice chooses bit strings to represent those strings of messages. She ignores the atypical messages or possibly reverts to some inefficient encoding of them.

In particular, suppose message i occurs with probability p_i. Then the probability of getting a particular sequence with n_i copies of message i (running over all i) is

$$\text{Prob.} = \prod_i p_i^{n_i}. \quad (6)$$

The most likely case is $n_i \approx p_i n$, and in fact the probability distribution is highly peaked around this case. Let us let the typical message strings be those with $|n_i - p_i n| = o(n)$. Then the total number of typical message strings is about $2^{nH(\{p_i\})}$, with H the Shannon entropy of the distribution. Thus, this block compression scheme needs to transmit only $nH(\{p_i\})$ bits to have a very high chance of successfully transmitting all messages.

(How do we get $nH(\{p_i\})$? Stirling's formula. The number of strings is about $n!/\prod_i (p_i n)!$, and $\log n! \approx n \log n$. Thus,

$$\begin{aligned}
\log[n!/\prod_i (p_i n)!] &= n \log n - \sum_i p_i n \log(p_i n) &(7)\\
&= n \log n - \sum_i p_i n (\log p_i + \log n) &(8)\\
&= n \log n - (\sum_i p_i) n \log n + nH(\{p_i\}) &(9)\\
&= nH(\{p_i\}), &(10)
\end{aligned}$$

since $\sum p_i = 1$.)

Theorem 1 (Shannon's Source Coding Theorem)**.** *Suppose we have a scheme to encode n messages from an i.i.d. source with entropy H with asymptotic probability of failure $\to 0$ as $n \to \infty$. Then the scheme uses at least $nH - o(n)$ bits. Moreover, there exist compression schemes that use only $nH + o(n)$ bits.*