

REVISITING SHANNON ENTROPY

We define Shannon entropy for a random variable X , which takes the values $\{x_1, x_2, x_3, \dots, x_n\}$ with probability $\{p_1, p_2, p_3, \dots, p_n\}$ as

$$H(X) = - \sum_i p_i \log_2 p_i$$

Shannon entropy quantifies the amount of information needed to transmit or store information.

$$\text{ALICE} \longrightarrow x_i \longrightarrow \text{BOB}$$

$X = \{x_1, x_2, x_3, x_4\}$ occurs with probability $\{p_1, p_2, p_3, p_4\}$

How many bits does Alice need? The answer is she needs $H(X)$ bits. A quick visual check tells us that ideally Alice would have needed $\lceil 2 \rceil$ bits to encode the four values of x_1, x_2, x_3 and x_4 .

How?? $\rightarrow 2$ bits will give

her $2^2 = 4$ possible values $\{00, 01, 10, 11\}$ which she can use for $\{x_1, x_2, x_3, x_4\}$

But Alice can get away with using only $H(X)$ bits, where $H(X) = - \sum_i p_i \log_2 p_i$, where p_i is the probability of x_i occurring. This is enshrined in Shannon's noiseless coding theorem.

Let us look at the example in class, where $\{p_1, p_2, p_3, p_4\} = \{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\}$.

Alice encodes x_1 ($p_1 = \frac{1}{2}$) with a single bit 0 and x_2 ($p_2 = \frac{1}{4}$) with two bits 10 and similarly x_3 and x_4 ($p_3 = p_4 = \frac{1}{8}$) with three bits as 110 and 111.

Why does she do this?? She needs to make sure Bob can identify all four x_i 's.

If she chooses 0 and 1 for x_1 and x_2 , she will be left with any two bit message that can distinguish x_3 and x_4 from x_1 and x_2 .

For example if she chooses: $x_1 = 0, x_2 = 1, x_3 = 10, x_4 = 11$, then it is possible Bob will confuse x_3 with the message $x_2 x_1$. You get the drift.

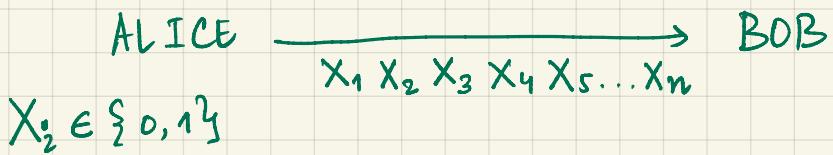
$$\begin{array}{ll} x_1 \rightarrow 0 & \\ x_2 \rightarrow 10 & \\ x_3 \rightarrow 110 & \\ x_4 \rightarrow 111 & \end{array} \left. \begin{array}{l} \text{If the first bit Bob gets is 1, he knows it is not } x_1, \text{ else it is } x_1 \\ \text{second bit is also 1, he knows it is not } x_1 \text{ and } x_2, \text{ else it is } x_2 \\ \text{third bit is also 1, he knows it is not } x_1, x_2 \text{ and } x_3, \text{ else it is } x_3 \end{array} \right\}$$

So, now what is the average size of her message?

$$\frac{1}{2} \times 1 + \frac{1}{4} \times 2 + \frac{1}{8} \times 3 + \frac{1}{8} \times 3 = \frac{7}{4} < 2 \quad (\text{Alice can ON AVERAGE get away with less than 2 bits})$$

$$\text{Also, } H(X) = -\frac{1}{2} \log(\frac{1}{2}) - \frac{1}{4} \log(\frac{1}{4}) - \frac{1}{8} \log(\frac{1}{8}) - \frac{1}{8} \log(\frac{1}{8}) = \frac{7}{4} < 2$$

Now suppose Alice has to send a string of n bits to Bob.



Each variable or bit X_i can take values 0 and 1 with probability p and $1-p$. Therefore each bit can be stored or compressed to

$$H(x) = H(x_i) = -p \log_2 p - (1-p) \log_2 (1-p)$$

Therefore for Alice to send a single bit to Bob, she will need $H(x)$ bit.

As such for Alice to send n bits to Bob, she will need $n H(x)$ bits.

Note each message here is a single bit with $2^1 = 2$ values ($\{0, 1\}$) i.e., $d=2$, therefore $H(x) \leq \log_2 d = 1$. Therefore, $n H(x) \leq n$.

How does $n H(x)$ relate to the previous problem?

Again, let us assume Alice sends a message to Bob but now it is a set of values.

ALICE $\longrightarrow X_1 \ X_2 \ X_3 \ X_4 \dots X_n \longrightarrow$ BOB
 $X_i = \{x_1, x_2, x_3, x_4\}$, which means each X_i can take four values.

So, naively Alice would need 2 bits for each of the n X_i 's, therefore need a total of $2n$ bits.

But from Shannon's noiseless coding theorem, she needs $H(x) = 2/4$ for each X_i , and therefore needs only $n H(x)$ bits, where $n H(x) < 2n$ as $H(x) \leq \log(4)$ or $H(x) \leq 2$.