REVISITING SHANNON ENTROPY

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

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

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

ALICE $\longrightarrow$ x_i $\longrightarrow$ 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 2 bits to encode the four values of x_1, x_2, x_3 and x_4.

How?? → 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 ($p_1 = 1/2$) with a single bit 0 and x_2 ($p_2 = 1/4$) with two bits 10 and similarly x_3 and x_4 ($p_3 = p_4 = 1/8$) 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 not 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.

$x_1 \to 0$
$x_2 \to 10$
$x_3 \to 110$
$x_4 \to 111$

If the first bit Bob gets is 1, he knows it is not x_1, else it is x_1
second bit is also 1, he knows it is not x_1 and x_2, else it is x_2
third bit is also 1, he knows it is not x_1, x_2 and x_3, else it is x_3

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

$$\frac{1}{2} \times 1 + \frac{1}{4} \times 2 + \frac{1}{8} \times 3 + \frac{1}{8} \times 3 = 7/4 < 2$$

(Alice can ON AVERAGE get away with less than 2 bits)

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

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

ALICE $\longrightarrow$ BOB

$X_1\ X_2\ X_3\ X_4\ X_5 \ldots X_n$

$X_i \in \{0, 1\}$

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

$$H(X) \equiv H(X_i) = -p \log_2 p - (1-p)\log_2(1-p)$$

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 $\longrightarrow$ $X_1\ X_2\ X_3\ X_4 \ldots X_n$ $\longrightarrow$ 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 $H(X) = 7/4$ for each X_i, and therefore needs only $n\,H(X)$ bits, where $n\,H(X) < 2n$ as $H(X) \leq \log(4)$ or $H(X) \leq 2$.