Note: in this course, log denotes $\log_2$.

Shannon's computation

Suppose we wish to compress a binary message $x_1^n = (x_1, \dots, x_n) \in \{0,1\}^n$. Assume x_1^n is generated by n iid random variables $X_1^n = (X_1, \dots, X_n)$ where each X_i is Bernouilli of parameter p, for some $p \in (0,1)$. We write P for the probability mass function of the X_i, i.e $P(x) = \mathbb{P}(X_i = x)$ for $x \in \{0,1\}$.

Idea: give more likely strings shorter descriptions.

Question: how is the probability distributed among all such x_1^n?

Let P^n denote the joint pmf of X_1^n. Then

$$\begin{aligned}
\mathbb{P}(X_1^n = x_1^n) = P^n(x_1^n) = \prod_{i=1}^n P(x_i) &= 2^{\log \prod_{i=1}^n P(x_i)} \\
&= 2^{\sum_{i=1}^n \log P(x_i)} \\
&= 2^{k \log p + (n-k)\log(1-p)} \\
&= 2^{-n\left[-\frac{k}{n}\log p - \frac{n-k}{n}\log(1-p)\right]} \\
&\approx 2^{-n[-p\log p - (1-p)\log(1-p)]}. && \text{(LLN)}
\end{aligned}$$

Where we have defined k to be the number of 1's in x_1^n. Now we define

$$h(p) = -p\log p - (1-p)\log(1-p)$$

so for large n we have

$$\mathbb{P}(X_1^n = x_1^n) \approx 2^{-nh(p)}$$

with high probability.

This means that for large n, the space $\{0,1\}^n$ of all possible messages consists of:

1. non typical strings that have negligible probability of showing up;
2. approximately $2^{nh(p)}$ each of similar probability.

Note that the *binary entropy function* $h(p)$ has a maximum at $p = \frac{1}{2}$ with $h(1/2) = 1$ and is symmetric through $p = \frac{1}{2}$.

Back to data compression. Consider the following algorithm. Let $B_n \subseteq \{0,1\}^n$ consist of the "typical" strings. Given x_1^n to compress:

- If $x_1^n \notin B_n \to$ declare "error";
- If $x_1^n \in B_n$, then describe it by describing its index j in B_n, where $1 \leq j \leq |B_n|$. This takes $\log|B_n| \approx nh(p)$ bits