14.1 Entropy and Optimal Coding

0. Problem Setting: Why Entropy Appears

Alice observes a random variable X taking values in a finite alphabet

$$[N] = \{1, \dots, N\},$$

and wishes to communicate its realization to Bob using a binary string.

A *code* is a function

$$c : [N] \to \{0,1\}^*,$$

and the length of the codeword for symbol i is

$$\ell(i) = |c(i)|.$$

Let $p_i = P(X = i)$. The objective is to minimize the expected code length

$$\min_c \sum_{i=1}^{N} p_i \, \ell(i).$$

1. Injective and Prefix-Free Codes

Injective. If $c(i) = c(j)$ for $i \neq j$, decoding is impossible. Thus, injectivity is a minimal requirement.

Prefix-free. A code is prefix-free if no codeword is a prefix of another. For example, $\{0, 01\}$ is not prefix-free.

This condition guarantees instantaneous (uniquely decodable) decoding when multiple symbols are transmitted sequentially.

2. Inefficiency of Uniform-Length Codes

A naive choice is to use a uniform code of length $\lceil \log_2 N \rceil$ bits for all symbols. However, if the distribution is highly non-uniform (e.g. $P(X = 1) = 0.99$), assigning shorter codewords to more frequent symbols is more efficient.

3. Optimal Coding and Huffman Bound

3.1 Optimization Problem

$$c^* = \arg\min_c \sum_{i=1}^{N} p_i \, \ell(c(i)),$$

where the minimum is taken over all valid (injective, prefix-free) codes.

3.2 Huffman Coding Guarantee

Let

$$H_2(P) = \sum_{i:p_i>0} p_i \log_2 \frac{1}{p_i}$$

denote the binary entropy of P. Huffman coding satisfies

$$H_2(P) \;\le\; \sum_{i=1}^{N} p_i\, \ell(c^*(i)) \;\le\; H_2(P) + 1.$$

The $+1$ term arises because code lengths must be integers.

4. Entropy with Natural Logarithm

Define the entropy using the natural logarithm as

$$H(P) = \sum_{i:p_i>0} p_i \log \frac{1}{p_i}.$$

This is a rescaling of $H_2(P)$:

$$H(P) = (\ln 2)\, H_2(P).$$

The interpretation remains the same; only the unit changes (bits vs. nats).

5. Kraft Inequality

5.1 Statement

For a prefix-free binary code with lengths $\ell_1, \ldots, \ell_N$,

$$\sum_{i=1}^{N} 2^{-\ell_i} \le 1.$$

5.2 Interpretation

A codeword of length ℓ occupies a fraction $2^{-\ell}$ of the binary tree. Prefix-free codes correspond to disjoint regions, so the total cannot exceed 1.

5.3 Proof Sketch

There are 2^L binary strings of length L. A codeword of length ℓ_i prefixes exactly $2^{L-\ell_i}$ such strings. Prefix-freeness implies disjointness, hence

$$\sum_i 2^{L-\ell_i} \le 2^L,$$

which yields the inequality after dividing by 2^L.

6. Entropy Lower Bound via KL Divergence

Define

$$q(x) = \frac{2^{-\ell(x)}}{\sum_y 2^{-\ell(y)}}.$$

Since KL divergence is nonnegative,

$$D(P\|Q) = \sum_x p(x) \log \frac{p(x)}{q(x)} \geq 0.$$

Rearranging,

$$\sum_x p(x) \log \frac{1}{p(x)} \leq \sum_x p(x) \log \frac{1}{q(x)}.$$

Moreover,

$$\log \frac{1}{q(x)} = \ell(x) + \log\Big(\sum_y 2^{-\ell(y)}\Big).$$

By Kraft's inequality, the logarithmic term is non-positive, hence

$$H(P) \leq \mathbb{E}[\ell(X)].$$

7. Achievability: Upper Bound via Shannon Code

Choose

$$\ell_i = \left\lceil \log_2 \frac{1}{p_i} \right\rceil.$$

Then

$$-\log_2 p_i \leq \ell_i < -\log_2 p_i + 1.$$

Moreover,

$$\sum_i 2^{-\ell_i} \leq \sum_i p_i = 1,$$

so a prefix-free code exists.

Multiplying the inequality by p_i and summing,

$$\sum_i p_i(-\log_2 p_i) \leq \sum_i p_i \ell_i \leq \sum_i p_i(-\log_2 p_i) + \sum_i p_i.$$

Since $\sum_i p_i = 1$,

$$\sum_i p_i \ell_i \leq H_2(P) + 1.$$

8. Final Result

Combining both bounds,

$$\boxed{H_2(P) \;\leq\; \mathbb{E}[\ell(X)] \;\leq\; H_2(P) + 1}$$