Entropy, Information, and Lossless Compression

Uri Shaham

1 Entropy

1.1 Motivation

A file, message, or dataset is compressible when it contains statistical regularity. A fair coin flip requires one bit to describe, but a highly biased coin flip requires less than one bit on average if we encode many flips at once. Entropy is the quantity that makes this statement precise.

There is a dictionary between probability and bits:

$$\text{uncertainty} \quad \longleftrightarrow \quad \text{number of bits needed to describe an outcome.}$$

1.2 Self-Information and Entropy

Definition 1.1 (Self-information)**.** Let X be a discrete random variable with distribution $p(x) = \mathbb{P}[X = x]$. The self-information of outcome x is

$$\imath_X(x) = \log \frac{1}{p(x)}.$$

Rare outcomes carry more information than common outcomes.

Definition 1.2 (Entropy)**.** The entropy of X is the expected self-information:

$$H(X) = \mathbb{E}\left[\log \frac{1}{p(X)}\right] = -\sum_{x \in \mathcal{X}} p(x) \log p(x).$$

The convention is $0 \log(1/0) = 0$.

Example 1.3 (A biased coin)**.** Let $X \sim \text{Bernoulli}(p)$. Then

$$H(X) = H_2(p) = -p \log p - (1-p) \log(1-p).$$

Here $H_2(1/2) = 1$ bit, while $H_2(p) \to 0$ as $p \to 0$ or $p \to 1$. A deterministic bit carries no uncertainty.

Proposition 1.4 (Basic bounds)**.** *If X takes values in an alphabet $\mathcal{X}$ of size m, then*

$$0 \le H(X) \le \log m.$$

The lower bound is attained by a deterministic distribution. The upper bound is attained by the uniform distribution.

Proof sketch. Nonnegativity is immediate since $p(x) \leq 1$. For the upper bound, compare p to the uniform distribution $u(x) = 1/m$ using nonnegativity of KL divergence, proved below:

$$0 \leq D(p\|u) = \sum_x p(x) \log \frac{p(x)}{1/m} = \log m - H(X).$$

□

2 Joint Entropy, Conditional Entropy, and Mutual Information

Definition 2.1 (Joint and conditional entropy)**.** For jointly distributed X, Y, the joint entropy is

$$H(X, Y) = -\sum_{x,y} p(x, y) \log p(x, y),$$

and the conditional entropy is

$$H(X \mid Y) = -\sum_{x,y} p(x, y) \log p(x \mid y).$$

Note that we can write the conditional entropy as the expectation of the entropy of the conditional distribution $H(X \mid Y) = -\sum_y p(y)p(x \mid y) \log p(x \mid y) = \mathbb{E}_y H(X \mid Y = y)$.

Proposition 2.2 (Chain rule for entropy)**.**

$$H(X, Y) = H(X) + H(Y \mid X) = H(Y) + H(X \mid Y).$$

More generally,

$$H(X_1, \ldots, X_n) = \sum_{i=1}^{n} H(X_i \mid X_1, \ldots, X_{i-1}).$$

Proposition 2.3. *If X and Y are independent, $H(Y \mid X) = H(Y)$, so $H(X, Y) = H(X) + H(Y)$.*

The proofs of both are straightforward.

Definition 2.4 (Mutual information)**.** The mutual information between X and Y is

$$I(X; Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X).$$

As a quick corollary, convince yourself that the mutual information between independent variables is 0. The following proposition shows that the mutual information is in fact the KL divergence between the joint distribution, and the product of the marginals.

Proposition 2.5.

$$I(X; Y) = D_{\mathrm{KL}}(p(x, y) \,\|\, p(x)p(y)).$$

Proof.

$$D_{\mathrm{KL}}(p(x, y) \,\|\, p(x)p(y)) = \sum_{x,y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}.$$

Expand the log using $p(x,y) = p(x)p(y \mid x)$:

$$\sum_{x,y} p(x,y) \log \frac{p(x)p(y \mid x)}{p(x)p(y)} = \sum_{x,y} p(x,y) \left[\log p(y \mid x) - \log p(y)\right]$$
$$= \sum_{x,y} p(x,y) \log p(y \mid x) - \sum_{x,y} p(x,y) \log p(y). \tag{1}$$

First sum. This is $-H(Y \mid X)$ by definition.

Second sum. Marginalize out x using $\sum_x p(x,y) = p(y)$:

$$\sum_{x,y} p(x,y) \log p(y) = \sum_y p(y) \log p(y) = -H(Y).$$

Combining:

$$D_{\mathrm{KL}}(p(x,y) \,\|\, p(x)p(y)) = H(Y) - H(Y \mid X) = I(X;Y).$$

□

Interpretations.

- $H(X)$ is uncertainty about X before seeing anything.
- $H(X \mid Y)$ is uncertainty about X after observing Y.
- $I(X;Y)$ is the reduction in uncertainty about X from observing Y.

Example 2.6 (Perfect and useless observations)**.** If $Y = X$, then $H(X \mid Y) = 0$ and $I(X;Y) = H(X)$. If X and Y are independent, then $H(X \mid Y) = H(X)$ and $I(X;Y) = 0$.

Definition 2.7 (Conditional Mutual Information)**.** The conditional mutual information of X and Y given Z is the mutual information between X and Y under the conditional distribution $p(x,y \mid z)$, averaged over Z:

$$I(X;Y \mid Z) = \mathbb{E}_z\big[I(X;Y \mid Z = z)\big] = \sum_z p(z) \sum_{x,y} p(x,y \mid z) \log \frac{p(x,y \mid z)}{p(x \mid z)p(y \mid z)}.$$

Equivalently, using $I(A;B) = H(A) - H(A \mid B)$ applied conditionally:

$$I(X;Y \mid Z) = H(X \mid Z) - H(X \mid Y,Z).$$

This definition allows us to prove the chain rule of mutual information.

Lemma 2.8 (Chain Rule of Mutual Information)**.** *For any three random variables X, Y, Z:*

$$I(X;Y,Z) = I(X;Z) + I(X;Y \mid Z).$$

Proof. We expand both sides using $I(A;B) = H(A) - H(A \mid B)$:

$$I(X;Z) + I(X;Y \mid Z) = H(X) - H(X \mid Z) + H(X \mid Z) - H(X \mid Y,Z)$$
$$= H(X) - H(X \mid Y,Z)$$
$$= I(X;Y,Z).$$

□

3 The Data-Processing Principle

Definition 3.1 (Markov chain)**.** We write $X \to Y \to Z$ when X and Z are conditionally independent given Y, equivalently

$$P_{XYZ}(x, y, z) = P_X(x)P_{Y|X}(y \mid x)P_{Z|Y}(z \mid y).$$

Theorem 3.2 (Data Processing Inequality)**.** *Let $X \to Y \to Z$ be a Markov chain. Then:*

$$I(X; Z) \leq I(X; Y).$$

In particular, no processing of Y can increase the information it contains about X.

Proof. We apply the chain rule of mutual information to expand $I(X; Y, Z)$ in two ways.

Expansion 1: condition on Y first.

$$I(X; Y, Z) = I(X; Y) + I(X; Z \mid Y).$$

Expansion 2: condition on Z first.

$$I(X; Y, Z) = I(X; Z) + I(X; Y \mid Z).$$

Equating the two expansions:

$$I(X; Y) + I(X; Z \mid Y) = I(X; Z) + I(X; Y \mid Z),$$

which rearranges to:

$$I(X; Y) - I(X; Z) = I(X; Y \mid Z) - I(X; Z \mid Y).$$

The Markov condition $X \perp\!\!\!\perp Z \mid Y$ gives $I(X; Z \mid Y) = 0$, so:

$$I(X; Y) - I(X; Z) = I(X; Y \mid Z) \geq 0,$$

where the inequality holds by non-negativity of mutual information. Therefore $I(X; Z) \leq I(X; Y)$. □

CS interpretation. If Y is data about an unknown quantity X, and $Z = f(Y)$ is any randomized or deterministic computation of Y, then Z cannot contain more information about X than Y did. Computation can reorganize information, but it cannot create new information about a hidden source.

4 Prefix-free codes

Definition 4.1 (Binary code)**.** A binary code for an alphabet $\mathcal{X}$ assigns each symbol $x \in \mathcal{X}$ a binary string $c(x) \in \{0, 1\}^*$. Its length is $\ell(x) = |c(x)|$.

Definition 4.2 (Prefix-free code)**.** A code is prefix-free if no codeword is a prefix of another codeword. Prefix-free codes can be decoded instantaneously from a stream.

For a distribution p on $\mathcal{X}$, the expected code length is

$$L = \mathbb{E}[\ell(X)] = \sum_x p(x)\ell(x).$$

Theorem 4.3 (Kraft inequality)**.** *There exists a binary prefix-free code with lengths $\ell(x)$ if and only if*

$$\sum_{x\in\mathcal{X}} 2^{-\ell(x)} \leq 1.$$

Intuitive proof. Think of all binary strings as leaves of an infinite binary tree. A codeword of length ℓ corresponds to a node at depth ℓ. If a code is prefix-free, then no codeword can be an ancestor of another codeword. So once we choose a codeword at depth ℓ, the entire subtree below it becomes unavailable for all other codewords. Now truncate the infinite tree at some depth $L \geq \max_i \ell_i$. The full binarytree of depth L has exactly

$$2^L$$

leaves. A codeword of length ℓ_i rules out all leaves below it, namely

$$2^{L-\ell_i}$$

leaves. Because the code is prefix-free, the subtrees ruled out by different codewords are disjoint. Therefore the total number of ruled-out leaves is at most the total number of leaves:

$$\sum_{i=1}^{m} 2^{L-\ell_i} \leq 2^L.$$

Dividing both sides by 2^L gives

$$\sum_{i=1}^{m} 2^{-\ell_i} \leq 1.$$

(Converse direction.) Given lengths $l_1 \leq l_2 \leq \cdots \leq l_n$ with $\sum_i 2^{-l_i} \leq 1$, we construct a prefix-free code greedily. Assign codewords one at a time in order of non-decreasing length. When assigning c_i, the codewords $c_1, \ldots, c_{i-1}$ already assigned have blocked

$$\sum_{j=1}^{i-1} 2^{l_i - l_j}$$

nodes at depth l_i (each earlier codeword c_j blocks $2^{l_i-l_j}$ nodes at depth l_i, namely those that have c_j as a prefix). The total number of blocked nodes is:

$$\sum_{j=1}^{i-1} 2^{l_i-l_j} = 2^{l_i} \sum_{j=1}^{i-1} 2^{-l_j} < 2^{l_i} \sum_{j=1}^{n} 2^{-l_j} \leq 2^{l_i},$$

where the last step uses the already proved direction of Kraft inequality. Since the total number of nodes at depth l_i is 2^{l_i}, at least one node remains available, and we can always assign a valid codeword.

□

5 Entropy as a lower bound on expected length

Theorem 5.1 (Shannon Code Upper Bound)**.** *Let X be a discrete random variable taking values in a finite alphabet with probabilities*

$$p_i = \mathbb{P}(X = x_i).$$

Then there exists a prefix-free binary code with expected length L satisfying

$$H(X) \leq L < H(X) + 1.$$

In particular, one may choose the Shannon code lengths

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

Proof. For each symbol x_i, define

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

Then

$$\ell_i \geq \log_2 \frac{1}{p_i}.$$

Equivalently,

$$2^{-\ell_i} \leq p_i.$$

Summing over all i gives

$$\sum_{i=1}^{m} 2^{-\ell_i} \leq \sum_{i=1}^{m} p_i = 1.$$

By the Kraft inequality, there exists a prefix-free binary code with codeword lengths $\ell_1, \ldots, \ell_m$. Now let L denote the expected length of this code:

$$L = \sum_{i=1}^{m} p_i \ell_i.$$

Since

$$\ell_i = \left\lceil \log_2 \frac{1}{p_i} \right\rceil < \log_2 \frac{1}{p_i} + 1,$$

we get

$$L = \sum_{i=1}^{m} p_i \ell_i < \sum_{i=1}^{m} p_i \left(\log_2 \frac{1}{p_i} + 1 \right).$$

Therefore

$$L < \sum_{i=1}^{m} p_i \log_2 \frac{1}{p_i} + \sum_{i=1}^{m} p_i = H(X) + 1.$$

It remains to show the lower bound. For any prefix-free binary code with lengths ℓ_i, Kraft's inequality gives

$$\sum_{i=1}^{m} 2^{-\ell_i} \leq 1.$$

Define

$$q_i = \frac{2^{-\ell_i}}{\sum_{j=1}^{m} 2^{-\ell_j}}.$$

Then $q_1, \ldots, q_m$ is a probability distribution. Using nonnegativity of KL divergence,

$$\sum_{i=1}^{m} p_i \log_2 \frac{1}{q_i} \geq \sum_{i=1}^{m} p_i \log_2 \frac{1}{p_i} = H(X).$$

But

$$q_i = \frac{2^{-\ell_i}}{K}, \qquad K = \sum_{j=1}^{m} 2^{-\ell_j} \leq 1,$$

so

$$\log_2 \frac{1}{q_i} = \ell_i + \log_2 K \leq \ell_i.$$

Therefore

$$H(X) \leq \sum_{i=1}^{m} p_i \log_2 \frac{1}{q_i} \leq \sum_{i=1}^{m} p_i \ell_i = L.$$

Combining the two bounds gives

$$H(X) \leq L < H(X) + 1.$$

□

5.1 Huffman coding

Huffman coding is the optimal prefix-code construction for a known discrete distribution. The algorithm repeatedly merges the two least likely symbols, assigning them sibling leaves in the code tree.

Example 5.2 (A small Huffman code)**.** Suppose $\mathcal{X} = \{a, b, c, d\}$ with probabilities

$$p(a) = 1/2, \quad p(b) = 1/4, \quad p(c) = p(d) = 1/8.$$

One optimal prefix code is

$$a \mapsto 0, \quad b \mapsto 10, \quad c \mapsto 110, \quad d \mapsto 111.$$

The expected length is

$$L = \frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 3 + \frac{1}{8} \cdot 3 = 1.75.$$

The entropy is also 1.75 bits, so this code is exactly optimal.