

Lecture 2: Entropy and Lossless Compression

ENGR 76 lecture notes — April 2, 2026

David Tse, Stanford University

Overview

In the previous lecture we introduced the definition of entropy of the simple case of binary-valued random variable. In this lecture we give a general definition for any discrete random variable, and we will further interpret entropy and connect it to lossless compression: how to represent a random source exactly using as few bits as possible on the average.

1 Entropy as expected surprise

Let

$$X \in \{0, 1, \dots, M-1\}$$

be a discrete random variable with pmf

$$P(X = i), \quad i = 0, 1, \dots, M-1.$$

We define the entropy of X is

$$H(X) = \sum_{i=0}^{M-1} P(X = i) \log_2 \frac{1}{P(X = i)}.$$

It is useful to interpret this as an expectation. Define the *surprise* of the outcome $X = i$ by

$$f(i) = \log_2 \frac{1}{P(X = i)}.$$

Then

$$H(X) = E[f(X)].$$

So entropy is the *expected surprise* of the source.

1.1 Binary case

If $X \in \{0, 1\}$ and

$$P(X = 0) = p, \quad P(X = 1) = 1 - p,$$

then

$$H(X) = p \log_2 \frac{1}{p} + (1 - p) \log_2 \frac{1}{1 - p}.$$

1.2 Why $\log_2(1/p)$? as the surprise?

This choice has the right qualitative behavior:

- If $p = 1$, then the event is certain, so the surprise should be 0:

$$\log_2 1 = 0.$$

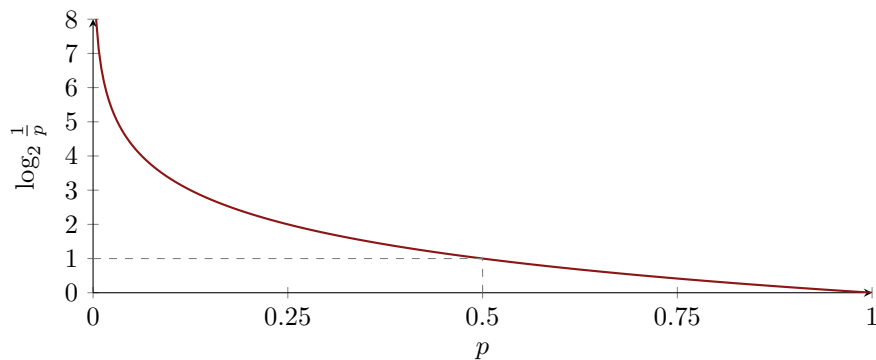


Figure 1: The surprise $\log_2(1/p)$ as a function of the probability p . The surprise is 0 when $p = 1$, equals 1 bit when $p = \frac{1}{2}$, and grows without bound as $p \rightarrow 0$.

- If $p = \frac{1}{2}$, then the surprise is

$$\log_2 2 = 1,$$

corresponding to one bit.

- As $p \rightarrow 0$, the event becomes very unlikely, so the surprise grows without bound:

$$\log_2 \frac{1}{p} \rightarrow \infty.$$

At the same time,

$$p \log_2 \frac{1}{p} \rightarrow 0 \quad \text{as } p \rightarrow 0,$$

so events of vanishing probability make vanishing contribution to the entropy.

2 Three basic properties of entropy

Property 1: Nonnegativity

$$H(X) \geq 0.$$

This follows because every term

$$P(X = i) \log_2 \frac{1}{P(X = i)}$$

is nonnegative.

Property 2: Label invariance

Entropy depends only on the set of probabilities $P(X = i)$, not on the labels or meaning of the symbols.

For example, a DNA source taking values in $\{a, g, c, t\}$ has the same entropy as a source taking values in $\{0, 1, 2, 3\}$, provided the probability assignments are the same.

Property 3: Maximum at the uniform distribution

Among all distributions on M symbols, entropy is maximized by the uniform distribution

$$P(X = i) = \frac{1}{M}, \quad i = 0, 1, \dots, M - 1.$$

In that case,

$$H(X) = \sum_{i=0}^{M-1} \frac{1}{M} \log_2 M = \log_2 M.$$

In particular, if $M = 2^n$, then

$$H(X) = \log_2 2^n = n \text{ bits.}$$

This agrees with the idea that when all 2^n possibilities are equally likely, n bits are needed to specify one of them.

3 Lossless compression

We now connect entropy to coding.

Lossless compression

A **lossless compression scheme** maps each source symbol x to a binary string, called a **codeword**, in such a way that the original symbol can be recovered exactly.

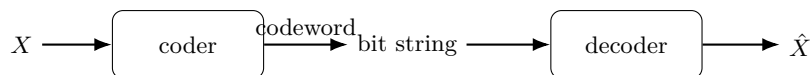


Figure 2: Lossless compression maps a source symbol to a binary codeword and then decodes it back exactly.

4 A fixed-length example

Consider a source

$$X \in \{a, g, c, t\}$$

with

$$P(a) = P(g) = P(c) = P(t) = \frac{1}{4}.$$

A natural fixed-length code is shown in Table 1.

nucleotide	probability	codeword	codeword length
a	$\frac{1}{4}$	00	2
g	$\frac{1}{4}$	01	2
c	$\frac{1}{4}$	10	2
t	$\frac{1}{4}$	11	2

Table 1: A fixed-length code for the uniform source on $\{a, g, c, t\}$.

Each codeword has length 2, so the expected codeword length is

$$\bar{\ell} = 2 \text{ bits.}$$

The entropy is

$$H(X) = 4 \cdot \frac{1}{4} \log_2 4 = 2 \text{ bits.}$$

So in this case the expected codeword length matches the entropy.

5 Expected codeword length

If the symbols i are assigned codewords of lengths ℓ_i , then the **expected codeword length** is

$$\bar{\ell} = \sum_i P(X = i) \ell_i.$$

This is the average number of bits used per source symbol.

6 A variable-length example

Now consider a nonuniform source on the same alphabet:

$$P(a) = \frac{1}{2}, \quad P(g) = \frac{1}{4}, \quad P(c) = \frac{1}{8}, \quad P(t) = \frac{1}{8}.$$

A variable-length code is shown in Table 2.

nucleotide	probability	codeword	codeword length
a	$\frac{1}{2}$	0	1
g	$\frac{1}{4}$	10	2
c	$\frac{1}{8}$	111	3
t	$\frac{1}{8}$	110	3

Table 2: A variable-length code for a nonuniform source.

Hence

$$\bar{\ell} = \frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 3 + \frac{1}{8} \cdot 3 = 1.75 \text{ bits.}$$

The entropy is

$$H(X) = \frac{1}{2} \log_2 2 + \frac{1}{4} \log_2 4 + \frac{1}{8} \log_2 8 + \frac{1}{8} \log_2 8 = 1.75 \text{ bits.}$$

So again,

$$\bar{\ell} = H(X).$$

This example illustrates the basic principle:

more likely symbols should be assigned shorter codewords.

nucleotide	probability	codeword	codeword length
<i>a</i>	$\frac{1}{2}$	0	1
<i>g</i>	$\frac{1}{4}$	10	2
<i>c</i>	$\frac{1}{8}$	011	3
<i>t</i>	$\frac{1}{8}$	010	3

Table 3: An example of a code that is not prefix-free.

7 Prefix-free codes

A code is **prefix-free** if no codeword is a prefix of another codeword.

For example, the code in Table 2 is prefix-free.

By contrast, the code in Table 3 is not prefix-free, because the codeword 0 is a prefix of both 011 and 010.

Why prefix-free is important

In practice, we don't code only one symbol, we code many symbols in succession. Think about the image example, where an image has 12 million pixels and we need to code the intensities of all of them. If a code is prefix-free, then the bit sequence formed by concatenation of the codewords of the symbols can be decoded from left to right without ambiguity: as soon as one codeword ends, the decoder knows it has identified a complete source symbol. Otherwise there is ambiguity in the parsing. For example, a bit string 010 represent one symbol "t" or a sequence of two symbols "ag".

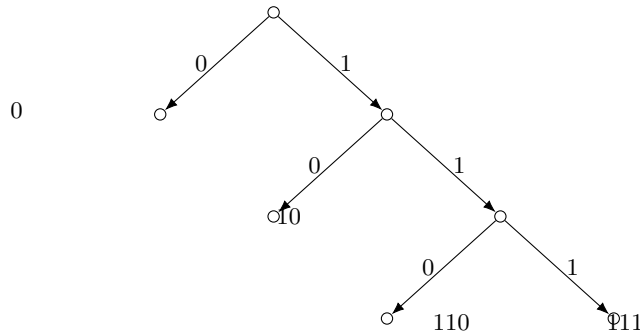


Figure 3: A code tree for the prefix-free code 0, 10, 110, 111. Prefix-free means all codewords are leaves of the code tree.

8 Uniquely decodable codes

Suppose a sequence of source symbols

$$X_1, X_2, \dots, X_n$$

is encoded as the concatenated bit string

$$c_1 c_2 \cdots c_n.$$

(Figure 4).

A code is **uniquely decodable** if the concatenated bit string determines the original source sequence uniquely.

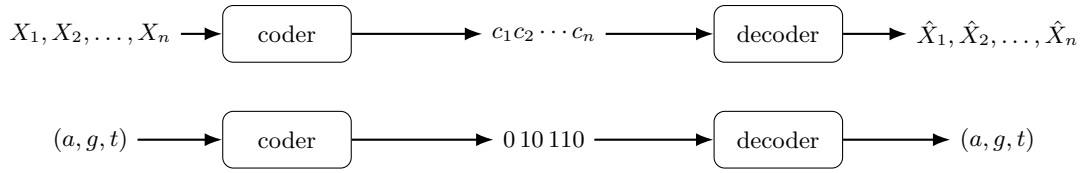


Figure 4: Top: in general, a sequence of source symbols is mapped to a concatenation of codewords and then decoded back into a source sequence. Bottom: a concrete example, where (a, g, t) is mapped to the concatenated bit string 010110. A code is uniquely decodable if the concatenated bit string determines the original source sequence uniquely.

Every prefix-free code is uniquely decodable:

$$\text{prefix-free} \implies \text{uniquely decodable}.$$

The converse is not true: not all uniquely decodable codes are prefix-free; however we will see next week that there is no loss of performance in using prefix-free codes.

Figure 5 summarizes the relationship among the different classes of codes:

$$\text{prefix-free codes} \subset \text{uniquely decodable codes} \subset \text{all codes}.$$

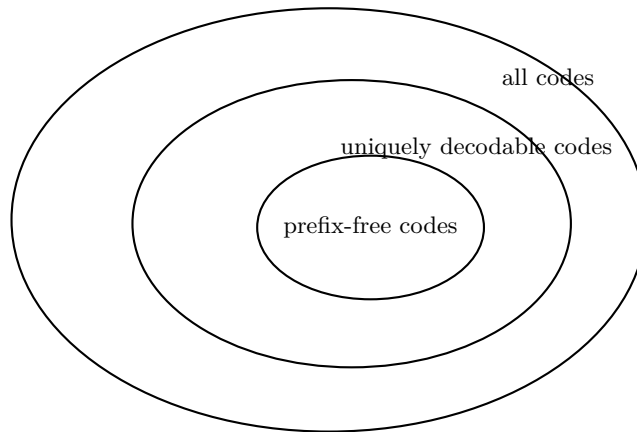


Figure 5: Relationship among classes of codes: every prefix-free code is uniquely decodable, and every uniquely decodable code is a code.

9 The source coding bound

Let $\bar{\ell}_{\min}$ denote the minimum expected codeword length over all prefix-free binary codes for the source X . Then

$$H(X) \leq \bar{\ell}_{\min} \leq H(X) + 1.$$

This theorem says:

- entropy is a lower bound on the average number of bits per symbol for prefix-free compression,
- and there always exists a prefix-free code whose average length is within one bit of entropy.

We will prove this theorem next week.

10 Summary

1. Entropy can be interpreted as expected surprise:

$$H(X) = E \left[\log_2 \frac{1}{P(X)} \right].$$

2. Entropy is nonnegative, label-invariant, and maximized by the uniform distribution.
3. In lossless compression, each source symbol is assigned a binary codeword.
4. The average number of bits used per symbol is

$$\bar{\ell} = \sum_i P(X = i) \ell_i.$$

5. Prefix-free codes are especially useful because they can be decoded unambiguously from left to right.
6. The optimal expected length over prefix-free codes satisfies

$$H(X) \leq \bar{\ell}_{\min} \leq H(X) + 1.$$

Takeaway

Entropy is not only a measure of uncertainty. It also determines the fundamental limit of lossless compression, up to an additive gap of at most one bit for prefix-free codes.