1

Preliminaries

In this chapter we introduce some notation, including definitions and basic properties of source coding, entropy, and redundancy. In particular, we define the average redundancy and the worst-case redundancy for prefix codes.

1.1 Source Code and Entropy

Let us start with some notation. Throughout, we write $x = x_1 x_2 \dots$ for a nonempty sequence of unspecified length over a finite alphabet $\mathcal{A}$. We also write $x_i^j = x_i, \dots, x_j \in \mathcal{A}^{j-i+1}$ for a consecutive subsequence of length $j - i + 1$. Sometimes we use the abbreviation $x^n = x_1^n$. Finally, throughout we write $\mathbb{Z}$, $\mathbb{Q}$, and $\mathbb{R}$ for integer, rational, and real numbers respectively.

A *(binary) code* is a one-to-one (or injective) mapping:

$$C\colon \mathcal{A} \to \{0,1\}^+$$

from a finite alphabet $\mathcal{A}$ (the *source*) to the set $\{0,1\}^+$ of nonempty binary sequences (where we use the notation $S^+ = \cup_{i=1}^{\infty} S^i$) One can extend this concept to m-ary codes $C\colon \mathcal{A} \to \{0,1,\dots,m-1\}^+$, if needed. We write $L(C,x)$ (or simply $L(x)$) for the length of $C(x)$. Finally, a code is a prefix code if no codeword is a prefix of another codeword.

In this book, we mostly deal with a sequence of codes:

$$C_n\colon \mathcal{A}_n \to \{0,1\}^+,$$

where $\mathcal{A}_n$ is a sequence of alphabets. In particular, if this sequence is of the form $\mathcal{A}_n = \mathcal{A}^n$, where $\mathcal{A} = \{0,\dots,m-1\}$, the code is called a fixed-to-variable (FV) code, discussed in Chapter 2. If $\mathcal{A}_n \subseteq \mathcal{A}^+$ and $\{0,1\}^+$ is replaced by $\{0,1\}^M$ for some M, we deal with variable-to-fixed (VF) codes; otherwise we have a general variable-to-variable (VV) code. We discuss VF codes in Chapters 3 and 4 and VV codes in Chapter 5.

We denote by P a probability distribution on the alphabet $\mathcal{A}$. The elements of the source can be then interpreted as a random variable X with probability distribution $P(X = x) = P(x)$. Such a source is also called *probabilistic source*. For example, the code length $L(X)$ is then a random variable too, and the expected code length $\mathbf{E}[L(X)]$ is an important parameter of a probabilistic source code.

The source *entropy* of a probabilistic source is defined by

$$H(X) := H(P) = -\mathbf{E}[\log P(X)] = -\sum_{x\in\mathcal{A}} P(x)\log P(x),$$

https://doi.org/10.1017/9781108565462.002 Published online by Cambridge University Press

where we write log for the logarithm of unspecified base; however, throughout the book, the base is usually equal to 2, unless specified otherwise.

Finally, we introduce a few other concepts related to entropy that we will use throughout the book.

Definition 1.1 (i) Let $\mathcal{A}$ and $\mathcal{A}'$ be two finite alphabets and P a probability distribution on $\mathcal{A} \times \mathcal{A}'$, and (X, Y) a random vector with $P(X = a, Y = a') = P(a, a')$. Then the Joint Entropy $H(X, Y)$ is defined as

$$H(X, Y) = -\mathbf{E}[\log P(X, Y)] = -\sum_{a\in\mathcal{A}} \sum_{a'\in\mathcal{A}'} P(a, a') \log P(a, a'). \tag{1.1}$$

(ii) The Conditional Entropy $H(Y|X)$ is

$$\begin{aligned} H(Y|X) &= -\mathbf{E}[\log P(Y|X)] = \sum_{a\in\mathcal{A}} P(a) H(Y|\, X = a) \\ &= -\sum_{a\in\mathcal{A}} \sum_{a'\in\mathcal{A}'} P(a, a') \log P(a'|a), \end{aligned} \tag{1.2}$$

where $P(a'|a) = P(a, a')/P(a)$ and $P(a) = \sum_{a'\in\mathcal{A}'} P(a, a')$.
(iii) The Relative Entropy or Kullback Leibler Distance or Kullback Leibler Divergence between two distributions P and Q defined on the same probability space (or finite alphabet $\mathcal{A}$) is

$$D(P||Q) = \mathbf{E}\left[\log \frac{P(X)}{Q(X)}\right] = \sum_{a\in\mathcal{A}} P(a) \log \frac{P(a)}{Q(a)}, \tag{1.3}$$

where, by convention, $0 \log(0/Q) = 0$ and $P \log(P/0) = \infty$.

(iv) The Mutual Information of X and Y is the relative entropy between the joint distribution of X and Y and the product distribution $P(X)P(Y)$; that is,

$$I(X; Y) = \mathbf{E}\left[\log \frac{P(X, Y)}{P(X)P(Y)}\right] = \sum_{a\in\mathcal{A}} \sum_{a'\in\mathcal{A}'} P(a, a') \log \frac{P(a, a')}{P(a)P(a')}. \tag{1.4}$$

(v) Rényi's Entropy of order b ($-\infty \le b \le \infty$, $b \neq 0$) is defined as

$$H_b(X) = -\frac{\log \mathbf{E}[P^b(X)]}{b} = -\frac{1}{b} \log \sum_{a\in\mathcal{A}} P^{b+1}(a), \tag{1.5}$$

provided $b \neq 0$ is finite, and by

$$H_{-\infty} = \min_{i\in\mathcal{A}}\{P(i)\}, \tag{1.6}$$

$$H_{\infty} = \max_{i\in\mathcal{A}}\{P(i)\}. \tag{1.7}$$

Observe that $H(X) = \lim_{b\to 0} H_b(X)$.

If P_n is a sequence of probability measures on $\mathcal{A}^n$ (for $n \ge 1$), we also define the entropy rate h, if it exists, as

$$h = \lim_{n\to\infty} \frac{H(P_n)}{n} = \lim_{n\to\infty} \frac{-\sum_{x^n\in\mathcal{A}^n} P_n(x^n) \log P_n(x^n)}{n}.$$

https://doi.org/10.1017/9781108565462.002 Published online by Cambridge University Press

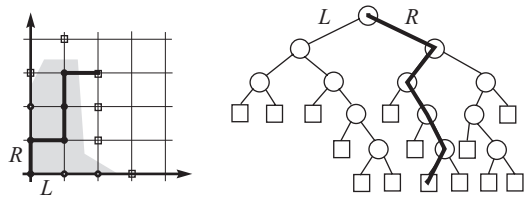

Figure 1.1 Lattice paths and binary trees.

1.2 Prefix Codes and Their Properties

As discussed, a *prefix code* is a code for which no codeword $C(x)$ for $x \in \mathcal{A}$ is a prefix of another codeword. For such codes, there is mapping between a codeword $C(x)$ and a path in a tree from the root to a terminal (external) node (e.g., for a binary prefix code, a move to the left in the tree represents 0 and a move to the right represents 1), as shown in Figure 1.1. We also point out that a prefix code and the corresponding path in a tree define a lattice path in the first quadrant also shown in Figure 1.1. Here left L and right R traversals in the binary tree correspond to "left" or "up" movement in the lattice. If some additional constraints are imposed on the prefix codes, this translates into certain restrictions on the lattice path indicated as the shaded area in Figure 1.1. (See Section 3.4 for some embellishments on this topic.)

The prefix condition imposes some restrictions on the code length. This fact is known as Kraft's inequality discussed next.

Theorem 1.2 (Kraft's inequality) *Let $|\mathcal{A}| = N$. Then for any binary prefix code C we have*

$$\sum_{x\in\mathcal{A}} 2^{-L(C,x)} \le 1.$$

Conversely, if positive integers $\ell_1, \ell_2, \ldots, \ell_N$ satisfy the inequality

$$\sum_{i=1}^{N} 2^{-\ell_i} \le 1, \tag{1.8}$$

then there exists a prefix code with these codeword lengths.

Proof This is an easy exercise on trees. Let $\ell_{\max}$ be the maximum codeword length. Observe that at level $\ell_{\max}$, some nodes are codewords, some are descendants of codewords, and some are neither. Since the number of descendants at level $\ell_{\max}$ of a codeword located at level ℓ_i is $2^{\ell_{\max}-\ell_i}$, we obtain

$$\sum_{i=1}^{N} 2^{\ell_{\max}-\ell_i} \le 2^{\ell_{\max}},$$

which is the desired inequality. The converse part can also be proved, and the reader is asked to prove it in Exercise 1.7. ∎

Using Kraft's inequality we can now prove the first theorem of Shannon (which was first established by Khinchin) that bounds from below the average code length.

https://doi.org/10.1017/9781108565462.002 Published online by Cambridge University Press

Theorem 1.3 *Let C be a prefix code on the alphabet $\mathcal{A}$, P a probability distribution on $\mathcal{A}$, and X a random variable with $P(X = a) = P(a)$. Then the average code length $\mathbf{E}[L(C,X)]$ cannot be smaller than the entropy $H(P)$; that is,*

$$\mathbf{E}[L(C,X)] \geq H(P),$$

where the expectation is taken with respect to the distribution P and the logarithms in the definition of $H(P)$ are the binary logarithms.

Proof Let $K = \sum_x 2^{-L(x)} \leq 1$ and $L(x) := L(C,x)$. Then by Kraft's inequality, $K \leq 1$. Furthermore, by using the inequality $-\log_2 x \geq \frac{1}{\ln 2}(1-x)$ for $x > 0$ we get (recall that $\log = \log_2$)

$$\begin{aligned}
\mathbf{E}[L(C,X)] - H(P)] &= \sum_{x\in\mathcal{A}} P(x)L(x) + \sum_{x\in\mathcal{A}} P(x)\log P(x) \\
&= -\sum_{x\in\mathcal{A}} P(x)\log\frac{2^{-L(x)}/K}{P(x)} - \log K \\
&\geq \frac{1}{\ln 2}\left(\sum_{x\in\mathcal{A}} P(x) - \frac{1}{K}\sum_{x\in\mathcal{A}} 2^{-L(x)}\right) - \log K \\
&= -\log K \geq 0,
\end{aligned}$$

as proposed. ∎

Observe that this theorem implies the existence of at least one element $\widetilde{x} \in \mathcal{A}$ such that

$$L(\widetilde{x}) \geq -\log P(\widetilde{x}). \tag{1.9}$$

Furthermore, we can complement Theorem 1.3 by the following property.

Lemma 1.4 (Barron) *Let C be a prefix code and $a > 0$. Then*

$$P(L(C,X) < -\log P(X) - a) \leq 2^{-a}.$$

Proof We argue as follows (again $\log = \log_2$):

$$\begin{aligned}
P(L(X) < -\log P(X) - a) &= \sum_{x:\, P(x)<2^{-L(x)-a}} P(x) \\
&\leq \sum_{x:\, P(x)<2^{-L(x)-a}} 2^{-L(x)-a} \\
&\leq 2^{-a}\sum_x 2^{-L(x)} \leq 2^{-a},
\end{aligned}$$

where we have used Kraft's inequality. ∎

What is the best prefix code with respect to code length? We are now in a position to answer this question. One needs to solve the following constrained optimization problem:

$$\min_L \sum_x L(x)P(x) \quad \text{subject to} \quad \sum_x 2^{-L(x)} \leq 1. \tag{1.10}$$

This optimization problem has an easy *real-valued* solution through Lagrangian multipliers, and one finds that the optimal code length is $L(x) = -\log P(x)$, provided the *integer character of the length is ignored* (see Exercise 1.8). If it is not ignored, then interesting

https://doi.org/10.1017/9781108565462.002 Published online by Cambridge University Press

things happen. First, the excess of the code length over $-\log P(x)$ is called the redundancy and is discussed in this book, in particular in Chapter 2. Furthermore, to minimize the redundancy, that is, to make $-\log P(x)$ as close to an integer as possible, ingenious algorithms were designed, which we will discuss in Part I of this book, in particular in Chapter 5.

In this book, we mostly deal with prefix codes, with the exception of Chapter 6 where we discuss nonprefix one-to-one codes.

To start with, we just mention a very simple prefix code, namely the Shannon code. It assigns to $x \in \mathcal{A}$ a codeword with code length

$$L(x) = \lceil -\log P(x) \rceil.$$

By Theorem 1.2, such a prefix code always exists, since

$$\sum_{x\in\mathcal{A}} 2^{-\lceil -\log P(x)\rceil} \le \sum_{x\in\mathcal{A}} P(x) = 1.$$

1.3 Redundancy

In general, one needs to round the length to an integer, thereby incurring some cost. This cost is usually called *redundancy*. More precisely, redundancy is the excess of real code length over its ideal (optimal) code length, which is assumed to be $-\log P(x)$. There are several possible specifications of this general definition. For a *known* distribution P, which we assume throughout Part I, the *pointwise redundancy* $R^C(x)$ for a code C and the *average redundancy* $\overline{R}^C$ are defined as

$$\begin{aligned} R^C(x) &= L(C,x) + \log P(x), \\ \overline{R} &= \mathbf{E}[L(C,X)] - H(P)]. \end{aligned}$$

Furthermore, we define the *maximal* or *worst-case* redundancy R^* as

$$R^* = \max_{x\in\mathcal{A}}[L(C,x) + \log P(x)] = \max_{x\in\mathcal{A}} R^C(x).$$

The pointwise redundancy can be negative, but the average and worst-case redundancies cannot due to the Shannon theorem (Theorem 1.3) and (1.9), respectively.

For example, for the Shannon code we have

$$0 \le \overline{R} \le R^* = \max_{x\in\mathcal{A}}[\lceil -\log P(x)\rceil + \log P(x)] < 1.$$

1.4 Exercises

1.1 Establish the following properties:

$$\begin{aligned} H(X,Y) &= H(X) + H(Y|X), \\ H(X_1,X_2,\ldots,X_n) &= \sum_{i=1}^{n} H(X_i|X_{i-1},\ldots,X_1), \\ I(X;Y) &= H(X) - H(X|Y) = H(Y) - H(Y|X), \\ I(X;X) &= H(X). \end{aligned}$$

https://doi.org/10.1017/9781108565462.002 Published online by Cambridge University Press

1.2 Prove that the following inequalities hold:

$$D(P\|Q) \geq 0, \tag{1.11}$$

$$I(X;Y) \geq 0, \tag{1.12}$$

$$H(X) \geq H(X|Y), \tag{1.13}$$

$$H(X_1,\ldots,X_n) \leq \sum_{i=1}^{n} H(X_i), \tag{1.14}$$

with equality in (1.11) if and only if $P(a) = Q(a)$ for all $a \in \mathcal{A}$, equality in (1.12) and (1.13) if and only if X and Y are independent, and equality in (1.14) if and only if $X_1,\ldots,X_n$ are independent.

1.3 Let $Y = g(X)$ where g is a measurable function. Prove

- $h(g(X)) \leq h(X)$;
- $h(Y|X) = 0$.

1.4 Random variables X, Y, Z form a Markov chain in that order (denoted by $X \to Y \to Z$) if the conditional distribution of Z depends on Y and is independent of X; that is,

$$P(X = x, Y = y, Z = z) = P(X = x)P(Y = y|X = x)P(Z = z|Y = y)$$

for all possible x, y, z. Prove the *data processing inequality* that states

$$I(X;Y) \geq I(X;Z)$$

if $X \to Y \to Z$.

1.5 Consider a probability vector $\mathbf{p} = (p_1,\ldots,p_n)$ such that $\sum_{i=1}^{n} p_i = 1$. What probability distribution $\mathbf{p}$ minimizes the entropy $H(\mathbf{p})$?

1.6 (Log sum inequality) (i) Prove that for nonnegative numbers $a_1,\ldots,a_n$ and $b_1,\ldots,b_n$,

$$\sum_{i=1}^{n} a_i \log\frac{a_i}{b_i} \geq \left(\sum_{i=1}^{n} a_i\right) \log\frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i}$$

with equality if and only if $\frac{a_i}{b_i} = const$.

(ii) Deduce form (i) that for $p_1,\ldots,p_n$ and $q_1,\ldots,q_n$ such that $\sum_{i=1}^{n} p_i = \sum_{i=1}^{n} q_i = 1$, we have

$$\sum_{i=1}^{n} p_i \log\frac{1}{q_i} \geq \sum_{i=1}^{n} p_i \log\frac{1}{p_i};$$

that is,

$$\min_{q_i} \sum_{i=1}^{n} p_i \log\frac{1}{q_i} = \sum_{i=1}^{n} p_i \log\frac{1}{p_i}.$$

(iii) Show that the following (potential) extension of (ii) is **not true**:

$$\sum_{i=1}^{n} p_i \left\lceil \log\frac{1}{q_i} \right\rceil \geq \sum_{i=1}^{n} p_i \left\lceil \log\frac{1}{p_i} \right\rceil,$$

where $\lceil x \rceil$ is the smallest integer greater than or equal to x.

https://doi.org/10.1017/9781108565462.002 Published online by Cambridge University Press

1.7 Prove the converse part of the Kraft inequality in Theorem 1.2.

1.8 Consider the optimization problem (1.10). Show that the optimal length is $L(x) = -\log P(x)$.

Bibliographical Notes

Information theory was born in 1948 when Shannon (1948) published a monumental work wherein he presented three theorems: on source coding, channel coding, and distortion theory. There are many good books discussing information theory. We recommend Gallager (1968) and Cover and Thomas (1991).

Lemma 1.4 was first proved by Barron (1985). The worst-case redundancy was introduced by Shtarkov (1987).

https://doi.org/10.1017/9781108565462.002 Published online by Cambridge University Press