1
Introduction to information theory

This chapter introduces some of the basic concepts of information theory, as well as the definitions and notation of probability theory that will be used throughout the book. The notion of entropy, which is fundamental to the whole topic of this book, is introduced here. We also present two major questions of information theory, those of data compression and error correction, and state Shannon's theorems.

Sec. 1.1 introduces the basic notations in probability. The notion of entropy, and the entropy rate of a sequence are discussed in Sections 1.2 and 1.3. A very important concept in information theory is the mutual information of two random variables, which is introduced in Section 1.4. Then we move to the two main aspects of the theory, the compression of data, in Sec. 1.5, and the transmission of data in, Sec. 1.6.

1.1 Random variables

The main object of this book will be the behaviour of large sets of **discrete random variables**. A discrete random variable X is completely defined[1] by the set of values it can take, $\mathcal{X}$, which we assume to be a finite set, and its **probability distribution** $\{p_X(x)\}_{x\in\mathcal{X}}$. The value $p_X(x)$ is the probability that the random variable X takes the value x. The probability distribution $p_X : \mathcal{X} \to [0,1]$ is a non-negative function that satisfies the normalization condition

$$\sum_{x\in\mathcal{X}} p_X(x) = 1 \; . \tag{1.1}$$

We shall denote by $\mathbb{P}(A)$ the probability of an **event** $A \subseteq \mathcal{X}$, so that $p_X(x) = \mathbb{P}(X = x)$. To lighten the notation, when there is no ambiguity, we shall use $p(x)$ to denote $p_X(x)$.

If $f(X)$ is a real-valued function of the random variable X, the **expectation value** of $f(X)$, which we shall also call the **average** of f, is denoted by

$$\mathbb{E}\, f = \sum_{x\in\mathcal{X}} p_X(x) f(x) \; . \tag{1.2}$$

While our main focus will be on random variables taking values in finite spaces, we shall sometimes make use of **continuous random variables** taking values in $\mathbb{R}^d$ or in some smooth finite-dimensional manifold. The probability measure for an

[1] In probabilistic jargon (which we shall avoid hereafter), we take the probability space $(\mathcal{X}, \mathsf{P}(\mathcal{X}), p_X)$, where $\mathsf{P}(\mathcal{X})$ is the σ-field of the parts of $\mathcal{X}$ and $p_X = \sum_{x\in X} p_X(x)\,\delta_x$.

'infinitesimal element' $\mathrm{d}x$ will be denoted by $\mathrm{d}p_X(x)$. Each time when p_X admits a density (with respect to the Lebesgue measure), we shall use the notation $p_X(x)$ for the value of this density at the point x. The total probability $\mathbb{P}(X \in \mathcal{A})$ that the variable X takes a value in some (measurable) set $\mathcal{A} \subseteq \mathcal{X}$ is given by the integral

$$\mathbb{P}(X \in \mathcal{A}) = \int_{x \in \mathcal{A}} \mathrm{d}p_X(x) = \int \mathbb{I}(x \in \mathcal{A})\ \mathrm{d}p_X(x)\ , \tag{1.3}$$

where the second form uses the **indicator function** $\mathbb{I}(s)$ of a logical statement s, which is defined to be equal to 1 if the statement s is true, and equal to 0 if the statement is false.

The expectation value $\mathbb{E}\, f(X)$ and the variance $\mathrm{Var}\, f(X)$ of a real-valued function $f(x)$ are given by

$$\mathbb{E}\, f(X) = \int f(x)\ \mathrm{d}p_X(x) \quad , \qquad \mathrm{Var}\, f(X) = \mathbb{E}\{f(X)^2\} - \{\mathbb{E}\, f(X)\}^2\ . \tag{1.4}$$

Sometimes, we may write $\mathbb{E}_X f(X)$ to specify the variable to be integrated over. We shall often use the shorthand **pdf** for the **probability density function** $p_X(x)$.

Example 1.1 A fair die with M faces has $\mathcal{X} = \{1, 2, \dots, M\}$ and $p(i) = 1/M$ for all $i \in \{1, \dots, M\}$. The average of x is $\mathbb{E}\, X = (1 + \dots + M)/M = (M+1)/2$.

Example 1.2 *Gaussian variable.* A continuous variable $X \in \mathbb{R}$ has a Gaussian distribution of mean m and variance σ^2 if its probability density is

$$p(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{[x-m]^2}{2\sigma^2}\right)\ . \tag{1.5}$$

We have $\mathbb{E}X = m$ and $\mathbb{E}(X-m)^2 = \sigma^2$.

Appendix A contains some definitions and notation for the random variables that we shall encounter most frequently

The notation of this chapter refers mainly to discrete variables. Most of the expressions can be transposed to the case of continuous variables by replacing sums $\sum_x$ by integrals and interpreting $p(x)$ as a probability density.

Exercise 1.1 *Jensen's inequality.* Let X be a random variable taking values in a set $\mathcal{X} \subseteq \mathbb{R}$ and let f be a convex function (i.e. a function such that $\forall x, y$ and $\forall \alpha \in [0,1]$: $f(\alpha x + (1-\alpha)y) \le \alpha f(x) + (1-\alpha) f(y)$). Then

$$\mathbb{E}f(X) \ge f(\mathbb{E}X)\ . \tag{1.6}$$

Supposing for simplicity that $\mathcal{X}$ is a finite set with $|\mathcal{X}| = n$, prove this equality by recursion on n.

1.2 Entropy

The **entropy** H_X of a discrete random variable X with probability distribution $p(x)$ is defined as

$$H_X \equiv -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x) = \mathbb{E} \log_2 \left[\frac{1}{p(X)} \right] \,, \tag{1.7}$$

where we define, by continuity, $0 \log_2 0 = 0$. We shall also use the notation $H(p)$ whenever we want to stress the dependence of the entropy upon the probability distribution of X.

In this chapter, we use the logarithm to base 2, which is well adapted to digital communication, and the entropy is then expressed in **bits**. In other contexts, and in particular in statistical physics, one uses the natural logarithm (with base $e \approx 2.7182818$) instead. It is sometimes said that, in this case, entropy is measured in **nats**. In fact, the two definitions differ by a global multiplicative constant, which amounts to a change of units. When there is no ambiguity, we shall use H instead of H_X.

Intuitively, the entropy H_X is a measure of the uncertainty of the random variable X. One can think of it as the missing information: the larger the entropy, the less a priori information one has on the value of the random variable. It roughly coincides with the logarithm of the number of typical values that the variable can take, as the following examples show.

Example 1.3 A fair coin has two values with equal probability. Its entropy is 1 bit.

Example 1.4 Imagine throwing M fair coins: the number of all possible outcomes is 2^M. The entropy equals M bits.

Example 1.5 A fair die with M faces has entropy $\log_2 M$.

Example 1.6 *Bernoulli process.* A Bernoulli random variable X can take values $0, 1$ with probabilities $p(0) = q$, $p(1) = 1 - q$. Its entropy is

$$H_X = -q \log_2 q - (1 - q) \log_2 (1 - q) \,, \tag{1.8}$$

which is plotted as a function of q in Fig. 1.1. This entropy vanishes when $q = 0$ or $q = 1$ because the outcome is certain; it is maximal at $q = 1/2$, when the uncertainty of the outcome is maximal.

Since Bernoulli variables are ubiquitous, it is convenient to introduce the function $\mathcal{H}(q) \equiv -q \log q - (1 - q) \log(1 - q)$ for their entropy.

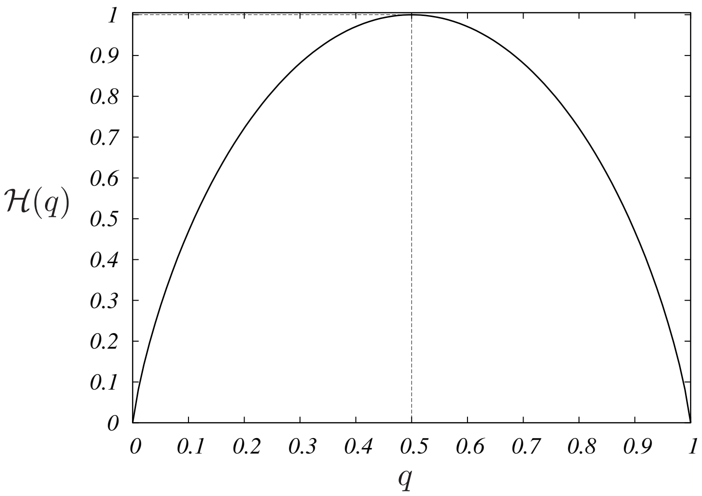

Fig. 1.1 The entropy $\mathcal{H}(q)$ of a binary variable with $p(X=0)=q$ and $p(X=1)=1-q$, plotted versus q.

Exercise 1.2 An unfair die with four faces and $p(1)=1/2,\ p(2)=1/4,\ p(3)=p(4)=1/8$ has entropy $H=7/4$, smaller than that of the corresponding fair die.

Exercise 1.3 DNA is built from a sequence of bases which are of four types, A, T, G, C. In the natural DNA of primates, the four bases have nearly the same frequency, and the entropy per base, if one makes the simplifying assumption of independence of the various bases, is $H=-\log_2(1/4)=2$. In some genuses of bacteria, one can have big differences in concentrations: for example, $p(G)=p(C)=0.38$, $p(A)=p(T)=0.12$, giving a smaller entropy $H\approx 1.79$.

Exercise 1.4 In some intuitive way, the entropy of a random variable is related to the 'risk' or 'surprise' which is associated with it. Let us see how these notions can be made more precise.

Consider a gambler who bets on a sequence of Bernoulli random variables $X_t\in\{0,1\}$, $t\in\{0,1,2,\dots\}$, with mean $\mathbb{E}X_t=p$. Imagine he knows the distribution of the X_t's and, at time t, he bets a fraction $w(1)=p$ of his money on 1 and a fraction $w(0)=(1-p)$ on 0. He loses whatever is put on the wrong number, while he doubles whatever has been put on the right one. Define the average doubling rate of his wealth at time t as

$$W_t=\frac{1}{t}\,\mathbb{E}\,\log_2\left\{\prod_{t'=1}^{t}2w(X_{t'})\right\}. \tag{1.9}$$

It is easy to prove that the expected doubling rate $\mathbb{E}W_t$ is related to the entropy of X_t: $\mathbb{E}W_t=1-\mathcal{H}(p)$. In other words, it is easier to make money out of predictable events.

Another notion that is directly related to entropy is the **Kullback–Leibler** (KL) **divergence** between two probability distributions $p(x)$ and $q(x)$ over the same finite space $\mathcal{X}$. This is defined as

$$D(q||p) \equiv \sum_{x\in\mathcal{X}} q(x) \log \frac{q(x)}{p(x)}, \tag{1.10}$$

where we adopt the conventions $0 \log 0 = 0$ and $0 \log(0/0) = 0$. It is easy to show that *(i)* $D(q||p)$ is convex in $q(x)$; *(ii)* $D(q||p) \geq 0$; and *(iii)* $D(q||p) > 0$ unless $q(x) \equiv p(x)$. The last two properties derive from the concavity of the logarithm (i.e. the fact that the function $-\log x$ is convex) and Jensen's inequality (eqn (1.6)): if $\mathbb{E}$ denotes the expectation with respect to the distribution $q(x)$, then $-D(q||p) = \mathbb{E} \log[p(x)/q(x)] \leq \log \mathbb{E}[p(x)/q(x)] = 0$. The KL divergence $D(q||p)$ thus looks like a distance between the probability distributions q and p, although it is not symmetric.

The importance of the entropy, and its use as a measure of information, derives from the following properties:

1. $H_X \geq 0$.
2. $H_X = 0$ if and only if the random variable X is certain, which means that X takes one value with probability one.
3. Among all probability distributions on a set $\mathcal{X}$ with M elements, H is maximum when all events x are equiprobable, with $p(x) = 1/M$. The entropy is then $H_X = \log_2 M$. To prove this statement, note that if $\mathcal{X}$ has M elements, then the KL divergence $D(p||\overline{p})$ between $p(x)$ and the uniform distribution $\overline{p}(x) = 1/M$ is $D(p||\overline{p}) = \log_2 M - H(p)$. The statement is a direct consequence of the properties of the KL divergence.
4. If X and Y are two **independent** random variables, meaning that $p_{X,Y}(x,y) = p_X(x)p_Y(y)$, the total entropy of the pair X, Y is equal to $H_X + H_Y$:

$$\begin{aligned} H_{X,Y} &= -\sum_{x,y} p_{X,Y}(x,y) \log_2 p_{X,Y}(x,y) \\ &= -\sum_{x,y} p_{X,Y}(x,y) \left(\log_2 p_X(x) + \log_2 p_Y(y)\right) = H_X + H_Y\,. \end{aligned} \tag{1.11}$$

5. For any pair of random variables, one has in general $H_{X,Y} \leq H_X + H_Y$, and this result is immediately generalizable to n variables. (The proof can be obtained by using the positivity of the KL divergence $D(p_1||p_2)$, where $p_1 = p_{X,Y}$ and $p_2 = p_X p_Y$.)
6. *Additivity for composite events.* Take a finite set of events $\mathcal{X}$, and decompose it into $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2$, where $\mathcal{X}_1 \cap \mathcal{X}_2 = \emptyset$. Denote by $q_1 = \sum_{x\in\mathcal{X}_1} p(x)$ the probability of $\mathcal{X}_1$, and denote by q_2 the probability of $\mathcal{X}_2$. For each $x \in \mathcal{X}_1$, define as usual the conditional probability of x, given that $x \in \mathcal{X}_1$, by $r_1(x) = p(x)/q_1$ and define $r_2(x)$ similarly as the conditional probability of x, given that $x \in \mathcal{X}_2$. The total entropy can then be written as the sum of two contributions $H_X = -\sum_{x\in\mathcal{X}} p(x) \log_2 p(x) = H(q) + \widetilde{H}(q, r)$, where

$$H(q) = -q_1 \log_2 q_1 - q_2 \log_2 q_2 \tag{1.12}$$

$$\widetilde{H}(q,r) = -q_1 \sum_{x \in \mathcal{X}_1} r_1(x) \log_2 r_1(x) - q_2 \sum_{x \in \mathcal{X}_1} r_2(x) \log_2 r_2(x)\,. \tag{1.13}$$

The proof is straightforward and is done by substituting the laws r_1 and r_2 by their definitions. This property can be interpreted as the fact that the average information associated with the choice of an event x is additive, being the sum of the information $H(q)$ associated to a choice of subset, and the information $\widetilde{H}(q,r)$ associated with the choice of the event inside the subset (weighted by the probability of the subset). This is the main property of the entropy, which justifies its use as a measure of information. In fact, this is a simple example of the chain rule for conditional entropy, which will be illustrated further in Sec. 1.4.

Conversely, these properties, together with appropriate hypotheses of continuity and monotonicity, can be used to define the entropy axiomatically.

1.3 Sequences of random variables and their entropy rate

In many situations of interest, one deals with a random process which generates **sequences of random variables** $\{X_t\}_{t\in\mathbb{N}}$, each of them taking values in the same finite space $\mathcal{X}$. We denote by $P_N(x_1,\dots,x_N)$ the joint probability distribution of the first N variables. If $A \subset \{1,\dots,N\}$ is a subset of indices, we denote by $\overline{A}$ its complement $\overline{A} = \{1,\dots,N\} \setminus A$ and use the notation $\underline{x}_A = \{x_i, i \in A\}$ and $\underline{x}_{\overline{A}} = \{x_i, i \in \overline{A}\}$ (the set subscript will be dropped whenever it is clear from the context). The **marginal distribution** of the variables in A is obtained by summing P_N over the variables in $\overline{A}$:

$$P_A(\underline{x}_A) = \sum_{\underline{x}_{\overline{A}}} P_N(x_1,\dots,x_N)\,. \tag{1.14}$$

Example 1.7 The simplest case is when the X_t's are independent. This means that $P_N(x_1,\dots,x_N) = p_1(x_1)p_2(x_2)\dots p_N(x_N)$. If all the distributions p_i are identical, equal to p, the variables are **independent identically distributed**, and abbreviated as **i.i.d.** The joint distribution is

$$P_N(x_1,\dots,x_N) = \prod_{t=1}^{N} p(x_i)\,. \tag{1.15}$$

Example 1.8 The sequence $\{X_t\}_{t\in\mathbb{N}}$ is said to be a **Markov chain** if

$$P_N(x_1,\dots,x_N) = p_1(x_1)\prod_{t=1}^{N-1} w(x_t \to x_{t+1})\,. \tag{1.16}$$

Here $\{p_1(x)\}_{x\in\mathcal{X}}$ is called the **initial state**, and the $\{w(x \to y)\}_{x,y\in\mathcal{X}}$ are the **transition probabilities** of the chain. The transition probabilities must be non-negative and normalized:

$$\sum_{y\in\mathcal{X}} w(x\to y) = 1\,, \qquad \text{for any } x\in\mathcal{X}. \tag{1.17}$$

When we have a sequence of random variables generated by a process, it is intuitively clear that the entropy grows with the number N of variables. This intuition suggests that we should define the **entropy rate** of a sequence $\underline{x}_N \equiv \{X_t\}_{t\in\mathbb{N}}$ as

$$h_X = \lim_{N\to\infty} H_{\underline{X}_N}/N\,, \tag{1.18}$$

if the limit exists. The following examples should convince the reader that the above definition is meaningful.

Example 1.9 If the X_t's are i.i.d. random variables with distribution $\{p(x)\}_{x\in\mathcal{X}}$, the additivity of entropy implies

$$h_X = H(p) = -\sum_{x\in\mathcal{X}} p(x)\log p(x)\,. \tag{1.19}$$

Example 1.10 Let $\{X_t\}_{t\in\mathbb{N}}$ be a Markov chain with initial state $\{p_1(x)\}_{x\in\mathcal{X}}$ and transition probabilities $\{w(x\to y)\}_{x,y\in\mathcal{X}}$. Call $\{p_t(x)\}_{x\in\mathcal{X}}$ the marginal distribution of X_t and assume the following limit to exist independently of the initial condition:

$$p^*(x) = \lim_{t\to\infty} p_t(x)\,. \tag{1.20}$$

As we shall see in Chapter 4, this indeed turns out to be true under quite mild hypotheses on the transition probabilities $\{w(x\to y)\}_{x,y\in\mathcal{X}}$. It is then easy to show that

$$h_X = -\sum_{x,y\in\mathcal{X}} p^*(x)\,w(x\to y)\log w(x\to y)\,. \tag{1.21}$$

If you imagine, for instance, that a text in English is generated by picking letters randomly from the alphabet $\mathcal{X}$, with empirically determined transition probabilities $w(x \to y)$, then eqn (1.21) gives a rough estimate of the entropy of English.

A more realistic model can be obtained using a Markov chain *with memory.* This means that each new letter x_{t+1} depends on the past through the values of the k previous letters $x_t, x_{t-1}, \dots, x_{t-k+1}$. Its conditional distribution is given by the transition probabilities $w(x_t, x_{t-1}, \dots, x_{t-k+1} \ \to \ x_{t+1})$. Computing the corresponding entropy rate is easy. For $k = 4$, one obtains an entropy of 2.8 bits per letter, much smaller than the trivial upper bound $\log_2 27$ (there are 26 letters, plus the space symbol), but many words so generated are still not correct English words. Better estimates of the entropy of English, obtained through guessing experiments, give a number around 1.3.

1.4 Correlated variables and mutual information

Given two random variables X and Y taking values in $\mathcal{X}$ and $\mathcal{Y}$, we denote their joint probability distribution as $p_{X,Y}(x, y)$, which is abbreviated as $p(x, y)$, and we denote the conditional probability distribution for the variable y, given x, as $p_{Y|X}(y|x)$, abbreviated as $p(y|x)$. The reader should be familiar with the classical Bayes' theorem

$$p(y|x) = p(x, y)/p(x) \ . \tag{1.22}$$

When the random variables X and Y are independent, $p(y|x)$ is independent of x. When the variables are dependent, it is interesting to have a measure of their degree of dependence: how much information does one obtain about the value of y if one knows x? The notions of conditional entropy and mutual information will answer this question.

We define the **conditional entropy** $H_{Y|X}$ as the entropy of the law $p(y|x)$, averaged over x:

$$H_{Y|X} \equiv - \sum_{x \in \mathcal{X}} p(x) \sum_{y \in \mathcal{Y}} p(y|x) \log_2 p(y|x) \, . \tag{1.23}$$

The joint entropy $H_{X,Y} \equiv - \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p(x, y) \log_2 p(x, y)$ of the pair of variables x, y can be written as the entropy of x plus the conditional entropy of y given x, an identity known as the **chain rule**:

$$H_{X,Y} = H_X + H_{Y|X} \, . \tag{1.24}$$

In the simple case, where the two variables are independent, $H_{Y|X} = H_Y$, and $H_{X,Y} = H_X + H_Y$. One way to measure the correlation of the two variables is to use the **mutual information** $I_{X,Y}$, which is defined as

$$I_{X,Y} \equiv \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p(x, y) \log_2 \frac{p(x, y)}{p(x)p(y)} \ . \tag{1.25}$$

It is related to the conditional entropies by

$$I_{X,Y} = H_Y - H_{Y|X} = H_X - H_{X|Y} \ . \tag{1.26}$$

This shows that the mutual information $I_{X,Y}$ measures the reduction in the uncertainty of x due to the knowledge of y, and is symmetric in x, y.

Proposition 1.11 $I_{X,Y} \geq 0$. *Moreover,* $I_{X,Y} = 0$ *if and only if* X *and* Y *are independent variables.*

Proof Write $I_{X,Y} = \mathbb{E}_{x,y} - \log_2\{p(x)p(y)/p(x,y)\}$. Consider the random variable $u = (x,y)$ with probability distribution $p(x,y)$. As the function $-\log(\,\cdot\,)$ is convex, one can apply Jensen's inequality (eqn (1.6)). This gives the result $I_{X,Y} \geq 0$ □

Exercise 1.5 A large group of friends plays the following game ('telephone without cables'). The person number zero chooses a number $X_0 \in \{0,1\}$ with equal probability and communicates it to the person number one without letting the others hear, and so on. The first person communicates the number to the second person, without letting anyone else hear. Call X_n the number communicated from the n-th to the $(n+1)$-th person. Assume that, at each step a person may become confused and communicate the wrong number with probability p. How much information does the n-th person have about the choice of the first person?

We can quantify this information through $I_{X_0,X_n} \equiv I_n$. Show that $I_n = 1 - \mathcal{H}(p_n)$ with p_n given by $1 - 2p_n = (1-2p)^n$. In particular, as $n \to \infty$,

$$I_n = \frac{(1-2p)^{2n}}{2\log 2}\left[1 + O((1-2p)^{2n})\right] . \tag{1.27}$$

The 'knowledge' about the original choice decreases exponentially along the chain.

Mutual information is degraded when data is transmitted or processed. This is quantified as follows

Proposition 1.12 ***Data-processing inequality***. *Consider a Markov chain* $X \to Y \to Z$ *(so that the joint probability of the three variables can be written as* $p_1(x)w_2(x \to y)w_3(y \to z)$*). Then* $I_{X,Z} \leq I_{X,Y}$. *In particular, if we apply this result to the case where* Z *is a function of* Y, $Z = f(Y)$, *we find that applying* f *degrades the information:* $I_{X,f(Y)} \leq I_{X,Y}$.

Proof We introduce the mutual information of two variables conditioned on a third one: $I_{X,Y|Z} = H_{X|Z} - H_{X|(YZ)}$. The mutual information between a variable X and a pair of variables (YZ) can be decomposed using the following chain rule: $I_{X,(YZ)} = I_{X,Z} + I_{X,Y|Z} = I_{X,Y} + I_{X,Z|Y}$. If we have a Markov chain $X \to Y \to Z$, X and Z are independent when we condition on the value of Y, and therefore $I_{X,Z|Y} = 0$. The result follows from the fact that $I_{X,Y|Z} \geq 0$. □

The conditional entropy also provides a lower bound on the probability of guessing a random variable. Suppose you want to guess the value of the random variable X, but you observe only the random variable Y (which can be thought of as a noisy version of X). From Y, you compute a function $\widehat{X} = g(Y)$, which is your estimate for X. What is the probability $\mathrm{P_e}$ that you guessed incorrectly? Intuitively, if X and Y are strongly correlated, one can expect that $\mathrm{P_e}$ is small, whereas it increases for less well-correlated variables. This is quantified as follows.

Proposition 1.13 ***Fano's inequality***. *Consider a random variable* X *taking values in the alphabet* $\mathcal{X}$, *and the Markov chain* $X \to Y \to \widehat{X}$, *where* $\widehat{X} = g(Y)$ *is an estimate* *for the value of X. Define the probability of making an error as* $\mathrm{P_e} = \mathbb{P}(\widehat{X} \neq X)$. *This is bounded from below as follows:*

$$\mathcal{H}(\mathrm{P_e}) + \mathrm{P_e} \log_2(|\mathcal{X}| - 1) \geq H(X|Y) \,. \tag{1.28}$$

Proof Define a random variable $E = \mathbb{I}(\widehat{X} \neq X)$, equal to 0 if $\widehat{X} = X$ and to 1 otherwise, and decompose the conditional entropy $H_{X,E|Y}$ using the chain rule in two ways: $H_{X,E|Y} = H_{X|Y} + H_{E|X,Y} = H_{E|Y} + H_{X|E,Y}$. Then notice that (i) $H_{E|X,Y} = 0$ (because E is a function of X and Y); (ii) $H_{E|Y} \leq H_E = \mathcal{H}(\mathrm{P_e})$ and (iii) $H_{X|E,Y} = (1 - \mathrm{P_e})H_{X|E=0,Y} + \mathrm{P_e} H_{X|E=1,Y} = \mathrm{P_e} H_{X|E=1,Y} \leq \mathrm{P_e} \log_2(|\mathcal{X}| - 1)$. □

Exercise 1.6 Suppose that X can take k values, and that its distribution is $p(1) = 1 - p$, $p(x) = p/(k-1)$ for $x \geq 2$. If X and Y are independent, what is the value of the right-hand side of Fano's inequality? Assuming that $1 - p > \frac{p}{k-1}$, what is the best guess one can make about the value of X? What is the probability of error? Show that Fano's inequality holds as an equality in this case.

1.5 Data compression

Imagine an information source which generates a sequence of symbols $\underline{X} = \{X_1, \dots, X_N\}$ taking values in a finite alphabet $\mathcal{X}$. We assume a probabilistic model for the source, meaning that the X_i are random variables. We want to store the information contained in a given realization $\underline{x} = \{x_1 \dots x_N\}$ of the source in the most compact way.

This is the basic problem of **source coding**. Apart from being an issue of the utmost practical interest, it is a very instructive subject. It in fact allows us to formalize in a concrete fashion the intuitions of 'information' and 'uncertainty' which are associated with the definition of entropy. Since entropy will play a crucial role throughout the book, we present here a little detour into source coding.

1.5.1 Codewords

We first need to formalize what is meant by 'storing the information'. We define a **source code** for the random variable $\underline{X}$ to be a mapping w which associates with any possible information sequence in $\mathcal{X}^N$ a string in a reference alphabet, which we shall assume to be $\{0, 1\}$:

$$\begin{aligned} w : \mathcal{X}^N &\rightarrow \{0,1\}^* \\ \underline{x} &\mapsto w(\underline{x}) \,. \end{aligned} \tag{1.29}$$

Here we have used the convention of denoting by $\{0, 1\}^*$ the set of binary strings of arbitrary length. Any binary string which is in the image of w is called a **codeword**.

Often, the sequence of symbols $X_1 \dots X_N$ is a part of a longer stream. The compression of this stream is realized in three steps. First, the stream is broken into blocks of length N. Then, each block is encoded separately using w. Finally, the codewords are glued together to form a new (hopefully more compact) stream. If the original stream consists of the blocks $\underline{x}^{(1)}, \underline{x}^{(2)}, \dots, \underline{x}^{(r)}$, the output of the encoding process will be the concatenation of $w(\underline{x}^{(1)}), \dots, w(\underline{x}^{(r)})$. In general, there is more than one way of parsing this concatenation into codewords, which may cause troubles when one wants to recover the compressed data. We shall therefore require the code w to be such that any concatenation of codewords can be parsed unambiguously. The mappings w satisfying this property are called **uniquely decodable codes**.

Unique decodability is certainly satisfied if, for any $\underline{x}, \underline{x}' \in \mathcal{X}^N$, $w(\underline{x})$ is not a prefix of $w(\underline{x}')$ (see Fig. 1.2). In such a case the code is said to be **instantaneous**. Hereafter, we shall focus on instantaneous codes, since they are both practical and slightly simpler to analyse.

Now that we have specified how to store information, namely using a source code, it is useful to introduce a figure of merit for source codes. If $l_w(x)$ is the length of the string $w(x)$, the average length of the code is

$$L(w) = \sum_{\underline{x} \in \mathcal{X}^N} p(\underline{x})\, l_w(\underline{x}) \ . \tag{1.30}$$

Example 1.14 Take $N = 1$, and consider a random variable X which takes values in $\mathcal{X} = \{1, 2, \dots, 8\}$ with probabilities $p(1) = 1/2$, $p(2) = 1/4$, $p(3) = 1/8$, $p(4) = 1/16$, $p(5) = 1/32$, $p(6) = 1/64$, $p(7) = 1/128$, and $p(8) = 1/128$. Consider the two codes w_1 and w_2 defined by the table below:

x	$p(x)$	$w_1(x)$	$w_2(x)$
1	1/2	000	0
2	1/4	001	10
3	1/8	010	110
4	1/16	011	1110
5	1/32	100	11110
6	1/64	101	111110
7	1/128	110	1111110
8	1/128	111	11111110

(1.31)

These two codes are instantaneous. For instance, looking at the code w_2, the encoded string 10001101110010 can be parsed in only one way, since each symbol 0 ends a codeword. It thus corresponds to the sequence $x_1 = 2, x_2 = 1, x_3 = 1, x_4 = 3, x_5 = 4, x_6 = 1, x_7 = 2$. The average length of code w_1 is $L(w_1) = 3$, and the average length of code w_2 is $L(w_2) = 247/128$. Notice that w_2 achieves a shorter average length because it assigns the shortest codeword (namely 0) to the most probable symbol (i.e. $x = 1$).

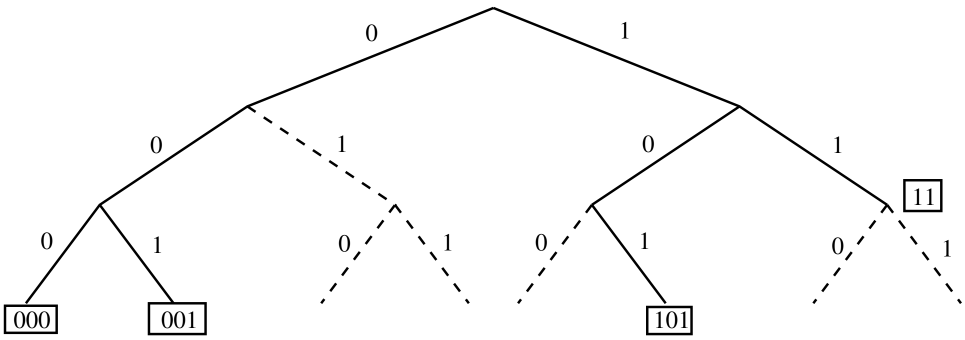

Fig. 1.2 An instantaneous source code: each codeword is assigned to a node in a binary tree in such a way that none of them is the ancestor of another one. Here, the four codewords are framed.

Example 1.15 A useful graphical representation of a source code can be obtained by drawing a binary tree and associating each codeword with the corresponding node in the tree. In Fig. 1.2, we represent a source code with $|\mathcal{X}^N| = 4$ in this way. It is quite easy to recognize that the code is indeed instantaneous. The codewords, which are framed, are such that no codeword is the ancestor of any other codeword in the tree. Given a sequence of codewords, parsing is immediate. For instance, the sequence `00111000101001` can be parsed only into `001`, `11`, `000`, `101`, `001`.

1.5.2 Optimal compression and entropy

Suppose that you have a 'complete probabilistic characterization' of the source you want to compress. What is the 'best code' w for this source?

This problem was solved (to a large extent) by Shannon in his celebrated 1948 paper, by connecting the best achievable average length to the entropy of the source. Following Shannon, we assume that we know the probability distribution of the source $p(\underline{x})$. Moreover, we interpret 'best code' as 'code with the shortest average length'.

Theorem 1.16 *Let L_N^* be the shortest average length achievable by an instantaneous code for the variable $\underline{X} = \{X_1, \dots, X_N\}$, which has entropy $H_{\underline{X}}$. Then:*

1. *For any $N \geq 1$*
$$H_{\underline{X}} \leq L_N^* \leq H_{\underline{X}} + 1\,. \tag{1.32}$$
2. *If the source has a finite entropy rate $h = \lim_{N\to\infty} H_{\underline{X}}/N$, then*
$$\lim_{N\to\infty} \frac{1}{N} L_N^* = h\,. \tag{1.33}$$

Proof The basic idea of the proof of eqn (1.32) is that if the codewords were too short, the code would not be instantaneous. **Kraft's inequality** makes this simple remark more precise. For any instantaneous code w, the lengths $l_w(\underline{x})$ satisfy

$$\sum_{\underline{x}\in\mathcal{X}^N} 2^{-l_w(\underline{x})} \leq 1\,. \tag{1.34}$$

This fact is easily proved by representing the set of codewords as a set of leaves on a binary tree (see Fig. 1.2). Let L_M be the length of the longest codeword . Consider the set of all the 2^{L_M} possible vertices in the binary tree at the generation L_M; let us call them the 'descendants'. If the information $\underline{x}$ is associated with a codeword at generation l (i.e. $l_w(\underline{x}) = l$), there can be no other codewords in the branch of the tree rooted at this codeword, because the code is instantaneous. We 'erase' the corresponding $2^{L_M - l}$ descendants, which cannot be codewords. The subsets of erased descendants associated with each codeword are not overlapping. Therefore the total number of erased descendants, $\sum_{\underline{x}} 2^{L_M - l_w(\underline{x})}$, must be less than or equal to the total number of descendants, 2^{L_M}. This establishes Kraft's inequality.

Conversely, for any set of lengths $\{l(\underline{x})\}_{\underline{x}\in\mathcal{X}^N}$ which satisfy Kraft's inequality (1.34), there exists at least one code whose codewords have lengths $\{l(\underline{x})\}_{\underline{x}\in\mathcal{X}^N}$. A possible construction is obtained as follows. Consider the smallest length $l(\underline{x})$ and take the first allowed binary sequence of length $l(\underline{x})$ to be the codeword for $\underline{x}$. Repeat this operation with the next shortest length and so on, until all the codewords have been exhausted. It is easy to show that this procedure is successful if eqn (1.34) is satisfied.

The problem is therefore reduced to finding the set of codeword lengths $l(\underline{x}) = l^*(\underline{x})$ which minimize the average length $L = \sum_{\underline{x}} p(\underline{x}) l(\underline{x})$ subject to Kraft's inequality (1.34). Supposing first that $l(\underline{x})$ can take arbitrary non-negative real values, this is easily done with Lagrange multipliers, and leads to $l(\underline{x}) = -\log_2 p(\underline{x})$. This set of optimal lengths, which in general cannot be realized because some of the $l(\underline{x})$ are not integers, gives an average length equal to the entropy $H_{\underline{X}}$. It implies the lower bound in eqn (1.32). In order to build a real code with integer lengths, we use

$$l^*(\underline{x}) = \lceil -\log_2 p(\underline{x}) \rceil . \tag{1.35}$$

Such a code satisfies Kraft's inequality, and its average length is less than or equal than $H_{\underline{X}} + 1$, proving the upper bound in eqn (1.32).

The second part of the theorem is a straightforward consequence of the first part. □

The code that we have constructed in the proof is often called a **Shannon code**. For long strings ($N \gg 1$), it is close to optimal. However, it has no reason to be optimal in general. For instance, if only one $p(x)$ is very small, it will assign x to a very long codeword, while shorter codewords are available. It is interesting to know that, for a given source $\{X_1, \ldots, X_N\}$, there exists an explicit construction of the optimal code, called Huffman's code.

At first sight, it may appear that Theorem 1.16, together with the construction of Shannon codes, completely solves the source coding problem. Unhappily, this is far from true, as the following arguments show.

From a computational point of view, the encoding procedure described above is unpractical when N is large. One can build the code once for all, and store it somewhere, but this requires $\Theta(|\mathcal{X}|^N)$ memory. On the other hand, one could reconstruct the code every time a string required to be encoded, but this takes $\Theta(|\mathcal{X}|^N)$ operations. One can use the same code and be a little smarter in the encoding procedure, but this does not yield a big improvement. (The symbol Θ means 'of the order of'; the precise definition is given in Appendix A.)

From a practical point of view, the construction of a Shannon code requires an accurate knowledge of the probabilistic law of the source. Suppose now that you want to compress the complete works of Shakespeare. It is exceedingly difficult to construct a good model for the source 'Shakespeare'. Even worse, when you will finally have such a model, it will be of little use for compressing Dante or Racine.

Happily, source coding has made tremendous progresses in both directions in the last half-century. However, in this book, we shall focus on another crucial aspect of information theory, the transmission of information.

1.6 Data transmission

We have just seen how to encode information in a string of symbols (we used bits, but any finite alphabet is equally good). Suppose now that we want to communicate this string. When the string is transmitted, it may be corrupted by noise, which depends on the physical device used for the transmission. One can reduce this problem by adding redundancy to the string. This redundancy is to be used to correct some of the transmission errors, in the same way as redundancy in the English language could be used to correct some of the typos in this book. This is the domain of **channel coding**. A central result in information theory, again due to Shannon's pioneering work of 1948, relates the level of redundancy to the maximal level of noise that can be tolerated for error-free transmission. As in source coding, entropy again plays a key role in this result. This is not surprising, in view of the duality between the two problems. In data compression, one wants to reduce the redundancy of the data, and the entropy gives a measure of the ultimate possible reduction. In data transmission, one wants to add some well-tailored redundancy to the data.

1.6.1 Communication channels

A typical flowchart of a communication system is shown in Fig. 1.3. It applies to situations as diverse as communication between the earth and a satellite, cellular phones, and storage within the hard disk of a computer. Alice wants to send a message m to Bob. Let us assume that m is an M-bit sequence. This message is first encoded into a longer one, an N-bit message denoted by $\underline{x}$, with $N > M$, where the added bits will provide the redundancy used to correct transmission errors. The encoder is a map from $\{0,1\}^M$ to $\{0,1\}^N$. The encoded message is sent through a communication channel. The output of the channel is a message $\underline{y}$. In the case of a noiseless channel, one would simply have $\underline{y} = \underline{x}$. In the case of a realistic channel, $\underline{y}$ is in general a string of symbols different from $\underline{x}$. Note that $\underline{y}$ is not necessarily a string of bits. The **channel** is described by a transition probability $Q(\underline{y}|\underline{x})$. This is the probability that the received signal is $\underline{y}$, conditional on the transmitted signal being $\underline{x}$. Different physical channels are described by different functions $Q(\underline{y}|\underline{x})$. The decoder takes the message $\underline{y}$ and deduces from it an estimate m' of the sent message.

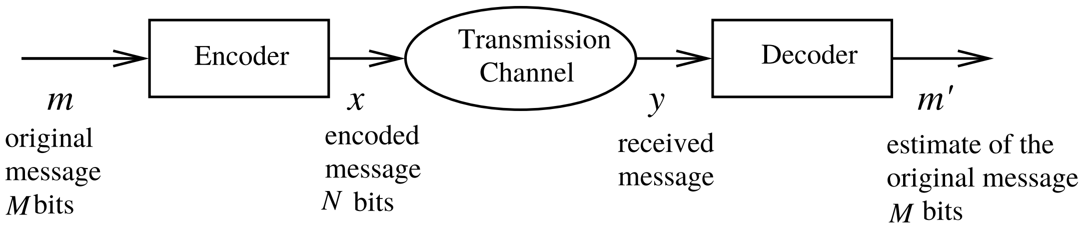

Fig. 1.3 Typical flowchart of a communication device.

Exercise 1.7 Consider the following example of a channel with **insertions**. When a bit `x` is fed into the channel, either `x` or `x0` is received with equal probability 1/2. Suppose that you send the string `111110`. The string `1111100` will be received with probability $2 \cdot 1/64$ (the same output can be produced by an error in either the fifth or the sixth digit). Notice that the output of this channel is a bit string which is always longer than or equal in length to the transmitted string.

A simple code for this channel is easily constructed: use the string `100` for each `0` in the original message and `1100` for each `1`. Then, for instance, we have the encoding

$$\texttt{01101} \mapsto \texttt{100110011001001100}\,. \tag{1.36}$$

The reader is invited to define a decoding algorithm and verify its effectiveness.

Hereafter, we shall consider **memoryless** channels. In this case, for any input $\underline{x} = (x_1, ..., x_N)$, the output message is a string of N letters $\underline{y} = (y_1, ..., y_N)$ from an alphabet $\mathcal{Y} \ni y_i$ (not necessarily binary). In a memoryless channel, the noise acts independently on each bit of the input. This means that the conditional probability $Q(\underline{y}|\underline{x})$ factorizes, i.e.

$$Q(\underline{y}|\underline{x}) = \prod_{i=1}^{N} Q(y_i|x_i)\ , \tag{1.37}$$

and the transition probability $Q(y_i|x_i)$ is independent og i.

Example 1.17 Binary symmetric channel (BSC). The input x_i and the output y_i are both in $\{0, 1\}$. The channel is characterized by one number, the probability p that the channel output is different from the input, called the **crossover** (or **flip**) probability. It is customary to represent this type of channel by the diagram on the left of Fig. 1.4.

Example 1.18 Binary erasure channel (BEC). In this case some of the input bits are erased instead of being corrupted: x_i is still in $\{\mathtt{0}, \mathtt{1}\}$, but y_i now belongs to $\{\mathtt{0}, \mathtt{1}, *\}$, where $*$ means that the symbol has been erased. In the symmetric case, this channel is described by a single number, the probability ϵ that a bit is erased, see Fig. 1.4, middle.

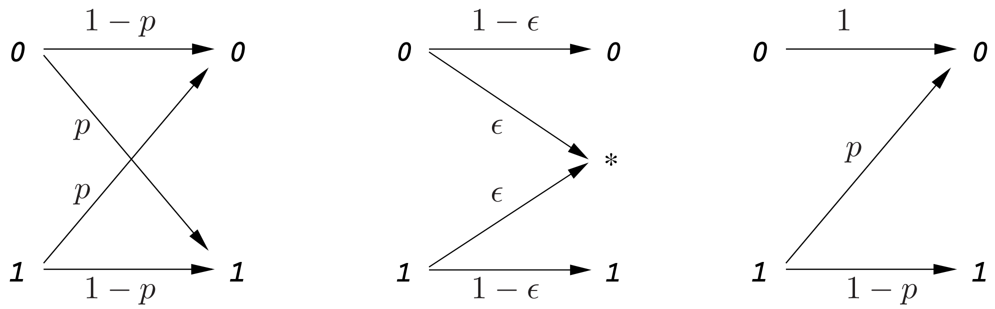

Fig. 1.4 Three communication channels. *Left*: the binary symmetric channel. An error in the transmission, in which the output bit is the opposite of the input one, occurs with probability p. *Middle*: the binary erasure channel. An error in the transmission, signaled by the output $*$, occurs with probability ϵ. *Right*: the Z channel. An error occurs with probability p whenever a $\mathtt{1}$ is transmitted.

Example 1.19 Z channel. In this case the output alphabet is again $\{\mathtt{0}, \mathtt{1}\}$. Now, however, a $\mathtt{0}$ is always transmitted correctly, whereas a $\mathtt{1}$ becomes a $\mathtt{0}$ with probability p. The name of this channel comes from its graphical representation: see Fig. 1.4, right.

A very important characteristic of a channel is the **channel capacity** C. This is defined in terms of the mutual information $I_{X,Y}$ of the variables X (the bit which was sent) and Y (the signal which was received), through

$$C = \max_{p(x)} I_{X,Y} = \max_{p(x)} \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)} . \tag{1.38}$$

We recall that in our case $p(x,y) = p(x)Q(y|x)$, and $I_{X,Y}$ measures the reduction in the uncertainty of x due to the knowledge of y. The capacity C gives a measure of how faithful a channel can be. If the output of the channel is pure noise, x and y are uncorrelated and $C = 0$. At the other extreme if $y = f(x)$ is known for sure, given x, then $C = \max_{\{p(x)\}} H(p) = 1$ bit (for binary inputs). The reason for our interest in the capacity will become clear in Section 1.6.3 with Shannon's coding theorem, which shows that C characterizes the amount of information which can be transmitted faithfully through a channel.

Example 1.20 Consider a binary symmetric channel with flip probability p. Let us call the probability that the source sends $x = \mathtt{0}$ q, and the probability of $x = \mathtt{1}$ $1-q$. It is easy to show that the mutual information in eqn (1.38) is maximized when zeros and ones are transmitted with equal probability (i.e. when $q = 1/2$).

Using eqn (1.38), we get $C = 1 - \mathcal{H}(p)$ bits, where $\mathcal{H}(p)$ is the entropy of a Bernoulli process with parameter p (plotted in Fig. 1.1).

Example 1.21 Consider now a binary erasure channel with error probability ϵ. The same argument as above applies. It is therefore easy to obtain $C = 1 - \epsilon$.

Exercise 1.8 Compute the capacity of a Z channel.

1.6.2 Error-correcting codes

We need one last ingredient in order to have a complete definition of the channel coding problem: the behaviour of the information source. We shall assume that the source produces a sequence of uncorrelated, unbiased bits. This may seem at first a very crude model for any real information source. Surprisingly, Shannon's source–channel separation theorem ensures that there is indeed no loss of generality in treating this case.

The sequence of bits produced by the source is divided into blocks $m_1, m_2, m_3, \dots$ of length M. The **encoding** is a mapping from $\{0,1\}^M \ni m$ to $\{0,1\}^N$, with $N \geq M$. Each possible M-bit message m is mapped to a **codeword** $\underline{x}(m)$, which can be seen as a point in the N-dimensional unit hypercube. The codeword length N is also called the **block length**. There are 2^M codewords, and the set of all possible codewords is called the **codebook**. When a message is transmitted, the corresponding codeword $\underline{x}$ is corrupted to $\underline{y} \in \mathcal{Y}^N$ with probability $Q(\underline{y}|\underline{x}) = \prod_{i=1}^N Q(y_i|x_i)$. The output alphabet $\mathcal{Y}$ depends on the channel. The **decoder** is a mapping from $\mathcal{Y}^N$ to $\{0,1\}^M$ which takes the received message $\underline{y} \in \mathcal{Y}^N$ and maps it to one of the possible original messages $m' = d(\underline{y}) \in \{0,1\}^M$.

An **error-correcting code** is defined by a pair of functions, the encoding $\underline{x}(m)$ and the decoding $d(\underline{y})$. The ratio

$$R = \frac{M}{N} \tag{1.39}$$

of the original number of bits to the transmitted number of bits is called the **rate** of the code. The rate is a measure of the redundancy of the code. The smaller the rate, the more redundancy is added to the code, and the more errors one should be able to correct.

The **block error probability** of a code on an input message m, denoted by $\mathrm{P_B}(m)$, is the probability that the decoded message differs from the message which was sent:

$$\mathrm{P_B}(m) = \sum_{\underline{y}} Q(\underline{y}|\underline{x}(m)) \;\; \mathbb{I}(d(\underline{y}) \neq m) \; . \tag{1.40}$$

Knowing the error probability for each possible transmitted message amounts to an exceedingly detailed characterization of the performance of the code. One can therefore introduce a **maximal block error probability** as

$$\mathrm{P_B^{max}} \equiv \max_{m \in \{0,1\}^M} \mathrm{P_B}(m) \, . \tag{1.41}$$

This corresponds to characterizing the code by its 'worst case' performances. A more optimistic point of view corresponds to averaging over the input messages. Since we have assumed all of them to be equiprobable, we introduce the **average block error probability**, defined as

$$\mathrm{P}_{\mathrm{B}}^{\mathrm{av}} \equiv \frac{1}{2^M} \sum_{m \in \{0,1\}^M} \mathrm{P}_{\mathrm{B}}(m) \,. \tag{1.42}$$

Since this is a very common figure of merit for error-correcting codes, we shall call it simply the block error probability, and use the symbol P_{B} without further specification hereafter.

Example 1.22 Repetition code. Consider a BSC which transmits a wrong bit with probability p. A simple code consists in repeating each bit k times, with k odd. Formally, we have $M = 1$, $N = k$, and

$$\underline{x}(0) = \underbrace{\mathtt{000\ldots00}}_{k} \,, \tag{1.43}$$

$$\underline{x}(1) = \underbrace{\mathtt{111\ldots11}}_{k} \,. \tag{1.44}$$

This code has rate $R = M/N = 1/k$. For instance, with $k = 3$, the original stream `0110001` is encoded as `000111111000000000111`. A possible decoder consists in parsing the received sequence into groups of k bits, and finding the message m' using a majority rule among the k bits. In our example with $k = 3$, if the received group of three bits is `111` or `110` or any permutation, the corresponding input bit is assigned to `1`, otherwise it is assigned to `0`. For instance, if the channel output is `000101111011000010111`, this decoder returns `0111001`.

Exercise 1.9 The k-repetition code corrects up to $\lfloor k/2 \rfloor$ errors per group of k bits. Show that the block error probability for general k is

$$\mathrm{P}_{\mathrm{B}} = \sum_{r=\lceil k/2 \rceil}^{k} \binom{k}{r} (1-p)^{k-r} p^r \,. \tag{1.45}$$

Note that, for any finite k and $p > 0$, P_{B} is strictly positive. In order to have $\mathrm{P}_{\mathrm{B}} \to 0$, we must consider $k \to \infty$. Since the rate is $1/k$, the price to pay for a vanishing block error probability is a vanishing communication rate!

Happily, however, we shall see that much better codes exist.

1.6.3 The channel coding theorem

Consider a communication channel whose capacity (eqn (1.38)) is C. In his seminal 1948 paper, Shannon proved the following theorem.

Theorem 1.23 *For every rate $R < C$, there exists a sequence of codes $\{\mathcal{C}_N\}$, of block length N, rate R_N, and block error probability $\mathrm{P}_{\mathrm{B},N}$, such that $R_N \to R$ and*

$\mathrm{P}_{\mathrm{B},N} \to 0$ *as* $N \to \infty$*. Conversely, if, for a sequence of codes* $\{\mathcal{C}_N\}$*, one has* $R_N \to R$ *and* $\mathrm{P}_{\mathrm{B},N} \to 0$ *as* $N \to \infty$*, then* $R < C$*.*

In practice, for long messages (i.e. large N), reliable communication is possible if and only if the communication rate remains below the channel capacity. The direct part of the proof will be given in Sec. 6.4 using the random code ensemble. We shall not give a full proof of the converse part in general, but only in the case of a BSC, in Sec. 6.5.2. Here we confine ourselves to some qualitative comments and provide the intuitive idea underlying this theorem.

First of all, the result is rather surprising when one meets it for the first time. As we saw in the example of repetition codes above, simple-minded codes typically have a positive error probability for any non-vanishing noise level. Shannon's theorem establishes that it is possible to achieve a vanishing error probability while keeping the communication rate bounded away from zero.

One can get an intuitive understanding of the role of the capacity through a qualitative argument, which uses the fact that a random variable with entropy H 'typically' takes 2^H values. For a given codeword $\underline{x}(m) \in \{0,1\}^N$, the channel output $\underline{y}$ is a random variable with an entropy $H_{\underline{y}|\underline{x}} = NH_{y|x}$. There exist about $2^{NH_{y|x}}$ such outputs. For perfect decoding, one needs a decoding function $d(\underline{y})$ that maps each of them to the original message m. Globally, the typical number of possible outputs is 2^{NH_y}, and therefore one can distinguish at most $2^{N(H_y - H_{y|x})}$ codewords. In order to have a vanishing maximal error probability, one needs to be able to send all of the $2^M = 2^{NR}$ codewords. This is possible only if $R < H_y - H_{y|x} \leq C$.

Notes

There are many textbooks that provide introductions to probability and to information theory. A classic probability textbook is Feller (1968). For a more recent reference see Durrett (1995). The original Shannon paper (Shannon, 1948) is universally recognized as the foundation of information theory. A very nice modern introduction to the subject is the book by Cover and Thomas (1991). The reader may find in there a description of Huffman codes, which we did not treat in the present Chapter, as well as more advanced topics in source coding.

We did not show that the six properties listed in Section 1.2 in fact provide an alternative (axiomatic) definition of entropy. The interested reader is referred to Csiszár abd Körner (1981). An advanced book on information theory with much space devoted to coding theory is Gallager (1968). The recent and very rich book by MacKay (2002) discusses the relations with statistical inference and machine learning.

The information-theoretic definition of entropy has been used in many contexts. It can be taken as a founding concept in statistical mechanics. This approach, pioneered by Jaynes (1957), is discussed by Balian (1992).