1
Channels, codes and capacity

In this chapter we introduce our task: communicating a digital message without error (or with as few errors as possible) despite an imperfect communications medium. Figure 1.1 shows a typical communications system. In this text we will assume that our source is producing binary data, but it could equally be an analog source followed by analog-to-digital conversion.

Through the early 1940s, engineers designing the first digital communications systems, based on pulse code modulation, worked on the assumption that information could be transmitted usefully in digital form over noise-corrupted communication channels but only in such a way that the transmission was unavoidably compromised. The effects of noise could be managed, it was believed, only by increasing the transmitted signal power enough to ensure that the received signal-to-noise ratio was sufficiently high.

Shannon's revolutionary 1948 work changed this view in a fundamental way, showing that it is possible to transmit digital data with arbitrarily high reliability, over noise-corrupted channels, by encoding the digital message with an error correction code prior to transmission and subsequently decoding it at the receiver. The error correction encoder maps each vector of K digits representing the message to longer vectors of N digits known as codewords. The redundancy implicit in the transmission of codewords, rather than the raw data alone, is the quid pro quo for achieving reliable communication over intrinsically unreliable channels. The code rate $r = K/N$ defines the amount of redundancy added by the error correction code. The transmitted bits may be corrupted in some way by the channel, and it is the function of the error correction decoder to use the added redundancy to determine the corresponding K message bits despite the imperfect reception.

In this chapter we will introduce the basic ideas behind error correction. In Section 1.1 we describe the channels considered in this text and in Section 1.2 the fundamental limits to communicating on those channels. Finally, in Section 1.3 we introduce error correction techniques.

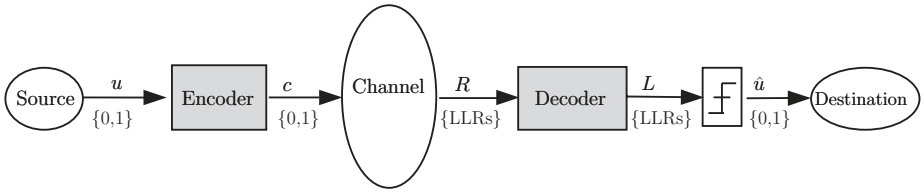

Figure 1.1 A typical communications system.

1.1 Binary input memoryless channels

A *discrete channel* is one that transmits a symbol x from a discrete set $X = \{X_1, X_1, \ldots, X_l\}$, known as the source alphabet, and returns a symbol y from another (possibly different) discrete alphabet, $Y = \{Y_1, Y_1, \ldots, Y_m\}$. A binary input channel transmits two discrete symbols, usually 0, 1, or 1, −1. The channel can also output binary symbols but may equally well output symbols from a larger discrete alphabet or a continuous range of values. Unfortunately, the channels do not always map a given transmitted symbol to the same received symbol (which is why we need error correction).

A communication channel can be modeled as a random process. For a given symbol x_i transmitted at time i, such that x_i is one of the symbols from the set X, i.e. $x_i = X_j \in \{X_1, X_1, \ldots, X_l\}$, the channel transition probability $p(y|x) = p(y = Y_j|x = X_j)$ gives the probability that the returned symbol y_i at time i is the symbol Y_i from the set Y, i.e. $y_i = Y_j \in \{Y_1, Y_1, \ldots, Y_m\}$. A channel is said to be *memoryless* if the channel output at any time instant depends only on the input at that time instant, not on previously transmitted symbols. More precisely, for a sequence of transmitted symbols $\mathbf{x} = [x_1, x_2, \ldots, x_N]$ and received symbols $\mathbf{y} = [y_1, y_2, \ldots, y_N]$:

$$p(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{N} p(y_i|x_i). \tag{1.1}$$

A memoryless channel is therefore completely described by its input and output alphabets and the conditional probability distribution $p(y|x)$ for each input–output symbol pair.

The three channels we consider in this text are the binary symmetric channel (BSC), the binary erasure channel (BEC) and the binary input additive white Gaussian noise (BI-AWGN) channel. They are all binary input memoryless channels.

Example 1.1 The *binary symmetric channel (BSC)* shown in Figure 1.2 transmits one of two symbols, the binary digits $x \in \{0, 1\}$, and returns one of two symbols, $y \in \{0, 1\}$. This channel flips the transmitted bit with probability ϵ, i.e. with

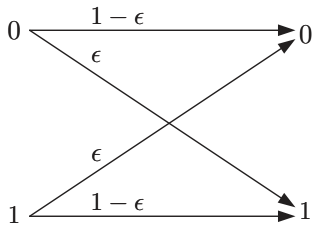

Figure 1.2 The binary symmetric channel (BSC).

probability ϵ the symbol y output by the channel is not the symbol that was sent and with probability $1-\epsilon$ the symbol y is the symbol that was sent. The parameter ϵ is called the *crossover probability* of the channel. So, for the BSC the channel transition probabilities are:

$$\begin{aligned} p(y=0|x=0) &= 1-\epsilon, \\ p(y=0|x=1) &= \epsilon, \\ p(y=1|x=0) &= \epsilon, \\ p(y=1|x=1) &= 1-\epsilon. \end{aligned}$$

A binary input channel is *symmetric* if both input bits are corrupted equally by the channel. The BSC channel is easily seen to be symmetric, as $p(y=0|x=0) = p(y=1|x=1)$ and $p(y=0|x=1) = p(y=1|x=0)$. Indeed all three channels we consider in this text, i.e. the BSC, BEC and BI-AWGN channels, are symmetric.

At the decoder, the symbol y received from the channel is used to decode the symbol x that was sent. In this case we are interested in the probability $p(x|y)$, i.e. given that we have received y, how likely was it that x was sent? We will assume that each bit is equally likely to be transmitted. Thus, for the binary symmetric channel, if $y=1$ the probability that $x=1$ is the probability that no error occurred, i.e. $p(x=1|y=1) = 1-\epsilon$, and the probability that $x=0$ is the probability that the channel flipped the transmitted bit $p(x=0|y=1) = \epsilon$. Similarly, $p(x=1|y=0) = \epsilon$ and $p(x=0|y=0) = 1-\epsilon$.

For a binary variable x it is easy to find $p(x=1)$ given $p(x=0)$, since $p(x=1) = 1-p(x=0)$ and so we only need to store one probability value for x. Log likelihood ratios (LLRs) are used to represent the metrics for a binary variable by a single value: the LLR is given by

$$L(x) = \log \frac{p(x=0)}{p(x=1)}, \tag{1.2}$$

where in this text we will use log to mean log to the base e, or $\log_e$. If $p(x=0) > p(x=1)$ then $L(x)$ is positive and, furthermore, the greater the difference between $p(x=0)$ and $p(x=1)$, i.e. the more sure we are that $p(x)=0$, the

larger the positive value for $L(x)$. Conversely, if $p(x = 1) > p(x = 0)$ then $L(x)$ is negative and, furthermore, the greater the difference between $p(x = 0)$ and $p(x = 1)$, the larger the negative value for $L(x)$. Thus the sign of $L(x)$ provides a hard decision (see the text after (1.9)) on x and the magnitude $|L(x)|$ is the reliability of this decision. Translating from LLRs back to probabilities, we obtain

$$p(x = 1) = \frac{p(x = 1)/p(x = 0)}{1 + p(x = 1)/p(x = 0)} = \frac{e^{-L(x)}}{1 + e^{-L(x)}} \tag{1.3}$$

and

$$p(x = 0) = \frac{p(x = 0)/p(x = 1)}{1 + p(x = 0)/p(x = 1)} = \frac{e^{L(x)}}{1 + e^{L(x)}}. \tag{1.4}$$

A benefit of the logarithmic representation of probabilities is that when probabilities need to be multiplied, log-likelihood ratios need only be added; this can reduce the implementation complexity.

Example 1.2 Given that the probabilities $p(x|y)$ for the BSC are

$$\begin{cases} p(x_i = 1|y_i) = 1 - \epsilon & \text{and} \quad p(x_i = 0|y_i) = \epsilon & \text{if } y_i = 1, \\ p(x_i = 1|y_i) = \epsilon & \text{and} \quad p(x_i = 0|y_i) = 1 - \epsilon & \text{if } y_i = 0, \end{cases}$$

the *received* LLRs for the ith transmitted bit are

$$R_i = L(x_i|y_i) = \log \frac{p(x_i = 0|y_i)}{p(x_i = 1|y_i)} = \begin{cases} \log \epsilon/(1 - \epsilon) & \text{if } y_i = 1, \\ \log(1 - \epsilon)/\epsilon & \text{if } y_i = 0. \end{cases}$$

Example 1.3 The *binary erasure channel* (BEC) shown in Figure 1.3 transmits one of two symbols, usually the binary digits $x \in \{0, 1\}$. However, the receiver either receives the bit correctly or it receives a message "e" that the bit was not received (it was *erased*). The BEC erases a bit with probability ε, called the *erasure probability* of the channel. Thus the channel transition probabilities for the BEC are

$$\begin{aligned} p(y = 0|x = 0) &= 1 - \varepsilon, \\ p(y = e|x = 0) &= \varepsilon, \\ p(y = 1|x = 0) &= 0, \\ p(y = 0|x = 1) &= 0, \\ p(y = e|x = 1) &= \varepsilon, \\ p(y = 1|x = 1) &= 1 - \varepsilon. \end{aligned}$$

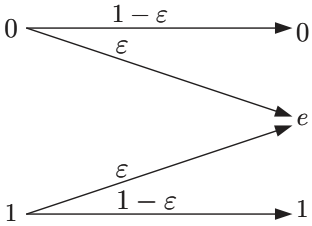

Figure 1.3 The binary erasure channel (BEC).

The BEC does not flip bits, so if y is received as a 1 or a 0 the receiver can be completely certain of the value of x:

$$\begin{aligned} p(x = 0|y = 0) &= 1, \\ p(x = 1|y = 0) &= 0, \\ p(x = 0|y = 1) &= 0, \\ p(x = 1|y = 1) &= 1. \end{aligned}$$

However, if the channel has erased the transmitted bit the receiver has no information about x and can only use the a priori probabilities of the source. If the source is equiprobable (i.e. the bits 1 and 0 are equally likely to have been sent) the receiver can only make a fifty–fifty guess:

$$\begin{aligned} p(x = 0|y = e) &= 0.5, \\ p(x = 1|y = e) &= 0.5. \end{aligned}$$

So, for this channel we have that the received LLRs for the ith transmitted bit are

$$R_i = L(x_i|y_i) = \log \frac{p(x_i = 0|y_i)}{p(x_i = 1|y_i)} = \begin{cases} \log \dfrac{0}{1} = -\infty & \text{if } y_i = 1, \\ \log \dfrac{1}{0} = \infty & \text{if } y_i = 0. \\ \log \dfrac{0.5}{0.5} = 0 & \text{if } y_i = e. \end{cases}$$

The final channel we consider, and the one most commonly used by coding theorists, is a binary input channel with additive noise modeled as samples from a Gaussian probability distribution.

Example 1.4 The *binary-input additive white Gaussian noise (BI-AWGN)* channel can be described by the equation

$$y_i = \mu x_i + z_i, \tag{1.5}$$

where $x_i \in \{-1, +1\}$ is the ith transmitted symbol, y_i is the ith received symbol and z_i is the additive noise sampled from a Gaussian random variable with mean 0 and variance σ^2. This is sometimes written $z_i = AWGN(0, \sigma)$.

The probability density function for z is

$$p(z) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-z^2/2\sigma^2}, \tag{1.6}$$

where $e^x = \exp(x)$ is the exponential function.

When transmitting a binary codeword on the BI-AWGN channel, the codeword bits $c_i \in \{0, 1\}$ can be mapped to the symbols $x_i \in \{-1, +1\}$ in one of two ways: $\{0 \to 1, 1 \to -1\}$ or $\{0 \to -1, 1 \to 1\}$. We will use the traditional convention $\{0 \to 1, 1 \to -1\}$.[1]

The received LLRs for the BI-AWGN channel are then

$$\begin{aligned} R_i = L(x_i|y_i) &= \log \frac{p(c_i = 0|y_i)}{p(c_i = 1|y_i)} \\ &= \log \frac{p(x_i = 1|y_i)}{p(x_i = -1|y_i)} \\ &= \log \frac{p(y_i|x_i = 1)p(x_i = 1)/p(y_i)}{p(y_i|x_i = -1)p(x_i = -1)/p(y_i)} \\ &= \log \frac{p(y_i|x_i = 1)p(x_i = 1)}{p(y_i|x_i = -1)p(x_i = -1)}, \end{aligned}$$

where we have used Bayes' rule

$$p(x_i|y_i) = p(x_i, y_i)/p(y_i) = p(y_i|x_i)p(x_i)/p(y_i)$$

to substitute for $p(x_i = 1|y_i)$ and $p(x_i = -1|y_i)$. If the source is equiprobable then $p(x_i = -1) = p(x_i = 1)$, and we have

$$R_i = L(x_i|y_i) = \log \frac{p(y_i|x_i = 1)}{p(y_i|x_i = -1)}.$$

For the BI-AWGN channel:

$$p(y_i|x_i = 1) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - \mu)^2}{2\sigma^2}\right), \tag{1.7}$$

$$p(y_i|x_i = -1) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i + \mu)^2}{2\sigma^2}\right); \tag{1.8}$$

[1] The mapping $\{0 \to 1, 1 \to -1\}$ is used because the modulo-2 arithmetic on $\{0, 1\}$ maps directly to multiplication on $\{-1, +1\}$ when this mapping is used.

thus

$$\begin{aligned} R_i = L(x_i|y_i) &= \log \frac{\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i-\mu)^2}{2\sigma^2}\right)}{\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i+\mu)^2}{2\sigma^2}\right)} \\ &= \log \exp\left(-\frac{(y_i-\mu)^2}{2\sigma^2} + \frac{(y_i+\mu)^2}{2\sigma^2}\right) \\ &= \frac{1}{2\sigma^2}(-(y_i^2 - 2\mu y_i + \mu^2) + (y_i^2 + 2\mu y_i + \mu^2)) \\ &= \frac{2\mu}{\sigma^2} y_i. \end{aligned} \tag{1.9}$$

The LLR value for a bit c_i is sometimes called a *soft decision* for c_i. A *hard decision* for c_i will return $c_i = 0$, equivalently $x_i = 1$, if R_i is positive and $c_i = 1$, equivalently $x_i = -1$, if R_i is negative.

When considering the relative noise level of a BI-AWGN channel, it is convenient to assume that $\mu = 1$ and adjust σ to reflect the noise quality of the channel. In this case R_i can be written as

$$R_i = \frac{2}{\sigma^2} y_i.$$

Often the noise level is expressed as the ratio of the energy per transmitted symbol, E_s, and the noise power spectral density N_0:

$$\frac{E_s}{N_0} = \frac{\mu^2}{2\sigma^2},$$

and (1.5) is sometimes written in the form

$$y_i = \sqrt{E_s} x_i + z_i.$$

When using error correction coding on a BI-AWGN channel, a fraction r of the transmitted bits correspond to bits in the message and the remainder are extra, redundant, bits added by the code. For channels using error correction the noise level is often expressed as the ratio of energy per message bit, E_b, and N_0, the *signal-to-noise ratio (SNR)*:

$$\frac{E_b}{N_0} = \frac{1}{r}\frac{E_s}{N_0} = \frac{1}{r}\frac{\mu^2}{2\sigma^2},$$

and the received LLR is often given as

$$R_i = L(x_i|y_i) = 4\frac{\sqrt{E_s}}{N_0} y_i = 4\frac{\sqrt{r E_b}}{N_0} y_i,$$

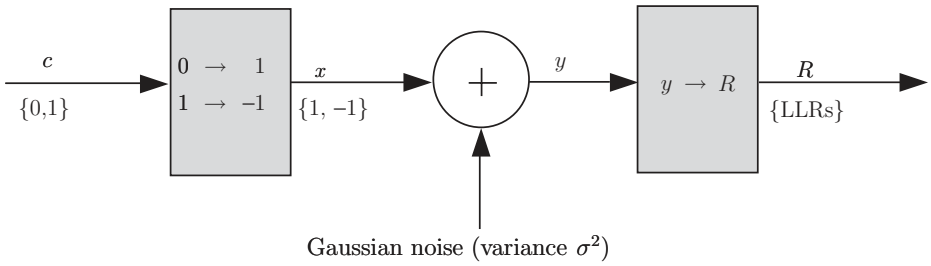

Figure 1.4 The BI-AWGN channel; $R = 4y/\sigma^2$.

or, when μ is assumed to be 1,

$$R_i = \frac{4}{N_0} y_i.$$

The signal-to-noise ratio can be also expressed in dB:

$$\frac{E_b}{N_0}(\text{dB}) = 10 \log_{10} \frac{E_b}{N_0} = 10 \log_{10} \frac{\mu^2}{2r\sigma^2}.$$

Figure 1.4 shows a block diagram for the BI-AWGN channel that we will consider in this text.

1.2 Entropy, mutual information and capacity

In the previous section we mentioned that a communications channel can be modeled as a random process and we described three common such models. In this section we will define some useful properties of random variables and use them to define limits on how well we can communicate over our three channels.

A discrete random variable X has a symbol alphabet $\{X_1, X_2, \ldots, X_q\}$ and probability distribution $p = \{p_1, p_2, \ldots, p_q\}$, where $p_j = p(x = X_j)$ gives the probability that a random sample x from X will return the symbol X_i.

A continuous random variable X can take any value in an uncountable set A_x. For example, A_x could define all real numbers between 0 and 1. A continuous random variable has a probability density function $p(x)$.

1.2.1 A measure of information

In order to motivate the concept of information we will consider a simple contest. There are two competitors and each has a "black box" that emits symbols x

from a random variable X with symbol alphabet $\{X_1, X_2\}$ and probabilities $p_1 = p(X_1) = \frac{99}{100}$, $p_2 = p(X_2) = \frac{1}{100}$. Each competitor receives only symbols from their own black box. The winner of the contest is the first to correctly name both symbols, that is, the first to have complete information about the symbol set of X. Now, suppose on the first round the first contestant receives the symbol X_1 while the second receives X_2. At this point it is clear that the second contestant is more likely to win (as in order to know both X_1 and X_2 the second contestant now only needs to receive the symbol X_1, which is much more likely to occur). In a sense the second contestant has received more *information* about X than the first, and this is reflected in the way information is measured. The lower the probability that a symbol occurs, the more information that is obtained from an occurrence of that symbol.

We denote as $I(p_j)$ the information obtained from receiving a symbol X_j, because the information is not a function of the symbol itself but of the symbol's probability of occurrence. There are three properties that we may expect the function $I(p)$ to have:

I1 $I(p) \geq 0$, that is, the information we gain by receiving a symbol cannot be negative (i.e. even though we may learn nothing from receiving a symbol we cannot lose information we already have).
I2 $I(p)$ is continuous in p.
I3 $I(p_1 p_2) = I(p_1) + I(p_2)$, that is, the information obtained from the knowledge that both X_1 and X_2 occurred is equal to the information obtained from the knowledge that X_1 occurred plus the information obtained from the knowledge that X_2 occurred.

The only function that satisfies all three assumptions is a logarithm:

$$I(p) = A \log_2 \frac{1}{p} \qquad \text{for some constant } A > 0,$$

and that is why this function was proposed by Hartley in 1928 as the measure of the information produced when a symbol with probability of occurrence p is received. Although any logarithm would work, base 2 logarithms are used most commonly.

The unit of measurement of information is the binary unit, and the constant A is chosen so as to equate one binary unit to the information received from one symbol of a binary source when both symbols are equally probable:

$$I(p) = A \log_2 \frac{1}{p} = A \log_2 \frac{1}{0.5} = A,$$

so $I(p) = 1$ when $A = 1$. Binary units (bits) are the measure of information regardless of whether X is binary, and it is important not to confuse them with binary digits (also shortened to bits).

1.2.2 Entropy

In the previous section we saw how to measure the information of a sample from a binary random variable; now we look at how to measure the amount of information in the variable itself. The information content of a random variable X is the average information over all its symbols and is called its *entropy* $H(X)$:

$$H(X) = \mathrm{E}[I(p(x))] = \sum_{j=1}^{q} p_j I(p_j) = \sum_{j=1}^{q} p_j \log_2 \frac{1}{p_j} = -\sum_{j=1}^{q} p_j \log_2 p_j.$$

We have assumed that the emission of a symbol is independent of time, i.e. the fact that a given symbol is emitted at one instant has no effect on which source symbol will be emitted at any other instant. Since the entropy is the average information per symbol its units are bits per symbol.

Example 1.5 Consider a discrete random variable X for which each symbol is equally likely to occur, that is $p_j = 1/q$ for all $j = 1, \ldots, q$. Then

$$H(X) = \sum_{j=1}^{q} p_j \log_2 \frac{1}{p_j} = \sum_{i=1}^{q} \frac{1}{q} \log_2 q = \log_2 q.$$

If $q = 3$ then

$$H(X) = \log_2 3 = 1.585 \text{ bits per symbol.}$$

However, for a discrete random variable $X = \{X_1, X_2, X_3\}$ with $p = \{\frac{1}{4}, \frac{1}{4}, \frac{1}{2}\}$, we have

$$H(X) = \sum_{j=1}^{q} p_j \log_2 \frac{1}{p_j} = \tfrac{1}{4} \log_2 4 + \tfrac{1}{4} \log_2 4 + \tfrac{1}{2} \log_2 2 = 1.5 \text{ bits per symbol.}$$

Thus the equiprobable random variable has a higher entropy.

To obtain an intuitive feel for the above result we return to our contest but this time each competitor has a different random variable. The winner of the contest is still the first to correctly name both symbols, that is, the first to have complete information about their random variable.