1 Introduction

Information media, such as communication systems and storage devices of data, are not absolutely reliable in practice because of noise or other forms of introduced interference. One of the tasks in coding theory is to detect, or even correct, errors. Usually, coding is defined as *source coding* and *channel coding*. Source coding involves changing the message source to a suitable code to be transmitted through the channel. An example of source coding is the ASCII code, which converts each character to a byte of 8 bits. A simple communication model can be represented by Fig. 1.1.

Example 1.0.1 Consider the source encoding of four fruits, *apple*, *banana*, *cherry*, *grape*, as follows:

$$\text{apple} \to 00, \quad \text{banana} \to 01, \quad \text{cherry} \to 10, \quad \text{grape} \to 11.$$

Suppose the message 'apple', which is encoded as 00, is transmitted over a noisy channel. The message may become distorted and may be received as 01 (see Fig. 1.2). The receiver may not realize that the message was corrupted. This communication fails.

The idea of channel coding is to encode the message again after the source coding by introducing some form of redundancy so that errors can be detected or even corrected. Thus, Fig. 1.1 becomes Fig. 1.3.

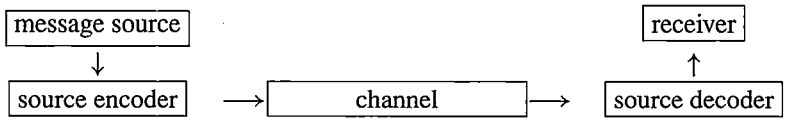

Fig. 1.1.

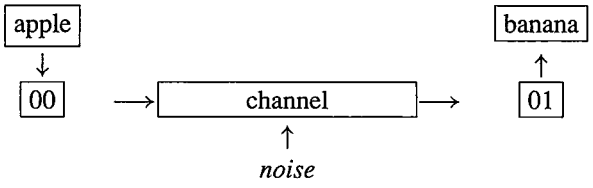

Fig. 1.2.

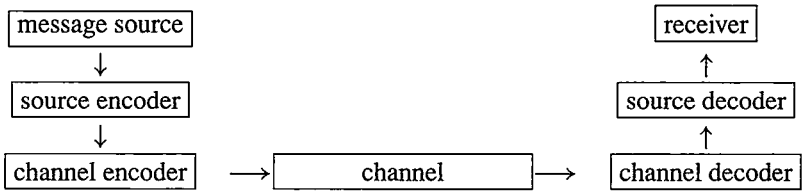

Fig. 1.3.

Example 1.0.2 In Example 1.0.1, we perform the channel encoding by introducing a redundancy of 1 bit as follows:

$$00 \to 000, \quad 01 \to 011, \quad 10 \to 101, \quad 11 \to 110.$$

Suppose that the message 'apple', which is encoded as 000 after the source and channel encoding, is transmitted over a noisy channel, and that there is only one error introduced. Then the received word must be one of the following three: 100, 010 or 001. In this way, we can detect the error, as none of 100, 010 or 001 is among our encoded messages.

Note that the above encoding scheme allows us to detect errors at the cost of reducing transmission speed as we have to transmit 3 bits for a message of 2 bits.

The above channel encoding scheme does not allow us to correct errors. For instance, if 100 is received, then we do not know whether 100 comes from 000, 110 or 101. However, if more redundancy is introduced, we are able to correct errors. For instance, we can design the following channel coding scheme:

$$00 \to 00000, \quad 01 \to 01111, \quad 10 \to 10110, \quad 11 \to 11001.$$

Suppose that the message 'apple' is transmitted over a noisy channel, and that there is only one error introduced. Then the received word must be one of the following five: 10000, 01000, 00100, 00010 or 00001. Assume that 10000 is received. Then we can be sure that 10000 comes from 00000 because there are

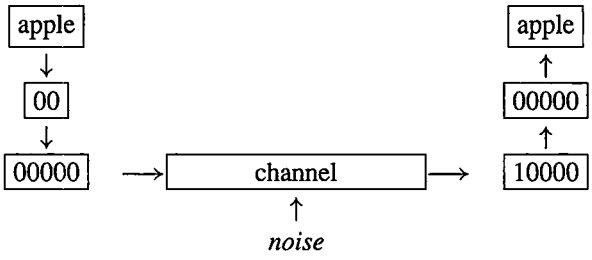

Fig. 1.4.

at least two errors between 10000 and each of the other three encoded messages 01111, 10110 and 11001.

Note that we lose even more in terms of information transmission speed in this case.

See Fig. 1.4 for this example.

Example 1.0.3 Here is a simple and general method of adding redundancy for the purpose of error correction. Assume that source coding has already been done and that the information consists of bit strings of fixed length k. Encoding is carried out by taking a bit string and repeating it $2r + 1$ times, where $r \geq 1$ is a fixed integer. For instance,

$$01 \longrightarrow 0101010101$$

if $k = 2$ and $r = 2$. In this special case, decoding is done by first considering the positions 1, 3, 5, 7, 9 of the received string and taking the first decoded bit as the one which appears more frequently at these positions; we deal similarly with the positions 2, 4, 6, 8, 10 to obtain the second decoded bit. For instance, the received string

$$1100100010$$

is decoded to 10. It is clear that, in this special case, we can decode up to two errors correctly. In the general case, we can decode up to r errors correctly. Since r is arbitrary, there are thus encoders which allow us to correct as many errors as we want. For obvious reasons, this method is called a *repetition code*. The only problem with this method is that it involves a serious loss of information transmission speed. Thus, we will look for more efficient methods.

The goal of channel coding is to construct encoders and decoders in such a way as to effect:

(1) fast encoding of messages;

(2) easy transmission of encoded messages;
(3) fast decoding of received messages;
(4) maximum transfer of information per unit time;
(5) maximal detection or correction capability.

From the mathematical point of view, the primary goals are (4) and (5). However, (5) is, in general, not compatible with (4), as we will see in Chapter 5. Therefore, any solution is necessarily a trade-off among the five objectives.

Throughout this book, we are primarily concerned with channel coding. Channel coding is also called *algebraic coding* as algebraic tools are extensively involved in the study of channel coding.

Exercises

1.1 Design a channel coding scheme to detect two or less errors for the message source {00, 10, 01, 11}. Can you find one of the best schemes in terms of information transmission speed?
1.2 Design a channel coding scheme to correct two or less errors for the message source {00, 10, 01, 11}. Can you find one of the best schemes in terms of information transmission speed?
1.3 Design a channel coding scheme to detect one error for the message source

$$\{000, 100, 010, 001, 110, 101, 011, 111\}.$$

Can you find one of the best schemes in terms of information transmission speed?
1.4 Design a channel coding scheme to correct one error for the message source

$$\{000, 100, 010, 001, 110, 101, 011, 111\}.$$

Can you find one of the best schemes in terms of information transmission speed?

2 Error detection, correction and decoding

We saw in Chapter 1 that the purpose of channel coding is to introduce redundancy to information messages so that errors that occur in the transmission can be detected or even corrected. In this chapter, we formalize and discuss the notions of error-detection and error-correction. We also introduce some well known decoding rules, i.e., methods that retrieve the original message sent by detecting and correcting the errors that have occurred in the transmission.

2.1 Communication channels

We begin with some basic definitions.

Definition 2.1.1 Let $A = \{a_1, a_2, \ldots, a_q\}$ be a set of size q, which we refer to as a *code alphabet* and whose elements are called *code symbols*.

(i) A *q-ary word* of length n over A is a sequence $\mathbf{w} = w_1 w_2 \cdots w_n$ with each $w_i \in A$ for all i. Equivalently, $\mathbf{w}$ may also be regarded as the vector $(w_1, \ldots, w_n)$.
(ii) A *q-ary block code* of length n over A is a nonempty set C of q-ary words having the same length n.
(iii) An element of C is called a *codeword* in C.
(iv) The number of codewords in C, denoted by $|C|$, is called the *size* of C.
(v) The (*information*) *rate* of a code C of length n is defined to be $(\log_q |C|)/n$.
(vi) A code of length n and size M is called an (n, M)-*code*.

Remark 2.1.2 In practice, and especially in this book, the code alphabet is often taken to be a finite field $\mathbf{F}_q$ of order q (cf. Chapter 3).

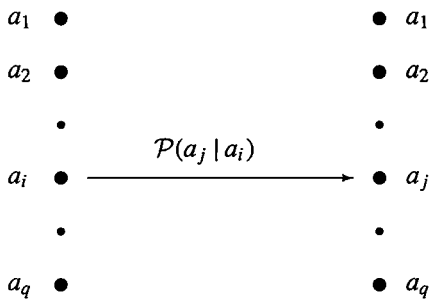

Fig. 2.1.

Example 2.1.3 A code over the code alphabet $\mathbf{F}_2 = \{0, 1\}$ is called a *binary code*; i.e., the code symbols for a binary code are 0 and 1. Some examples of binary codes are:

(i) $C_1 = \{00, 01, 10, 11\}$ is a (2,4)-code;
(ii) $C_2 = \{000, 011, 101, 110\}$ is a (3,4)-code;
(iii) $C_3 = \{0011, 0101, 1010, 1100, 1001, 0110\}$ is a (4,6)-code.

A code over the code alphabet $\mathbf{F}_3 = \{0, 1, 2\}$ is called a *ternary code*, while the term *quaternary code* is sometimes used for a code over the code alphabet $\mathbf{F}_4$. However, a code over the code alphabet $\mathbf{Z}_4 = \{0, 1, 2, 3\}$ is also sometimes referred to as a quaternary code (cf. Chapter 3 for the definitions of $\mathbf{F}_3$, $\mathbf{F}_4$ and $\mathbf{Z}_4$).

Definition 2.1.4 A *communication channel* consists of a finite *channel alphabet* $A = \{a_1, \ldots, a_q\}$ as well as a set of *forward channel probabilities* $\mathcal{P}(a_j \text{ received} \mid a_i \text{ sent})$, satisfying

$$\sum_{j=1}^{q} \mathcal{P}(a_j \text{ received} \mid a_i \text{ sent}) = 1$$

for all i (see Fig. 2.1). (Here, $\mathcal{P}(a_j \text{ received} \mid a_i \text{ sent})$ is the conditional probability that a_j is received, given that a_i is sent.)

Definition 2.1.5 A communication channel is said to be *memoryless* if the outcome of any one transmission is independent of the outcome of the previous transmissions; i.e., if $\mathbf{c} = c_1 c_2 \cdots c_n$ and $\mathbf{x} = x_1 x_2 \cdots x_n$ are words of length n, then

$$\mathcal{P}(\mathbf{x} \text{ received} \mid \mathbf{c} \text{ sent}) = \prod_{i=1}^{n} \mathcal{P}(x_i \text{ received} \mid c_i \text{ sent}).$$

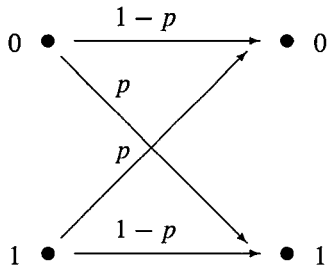

Fig. 2.2. Binary symmetric channel.

Definition 2.1.6 A *q-ary symmetric channel* is a memoryless channel which has a channel alphabet of size q such that

(i) each symbol transmitted has the same probability p ($<1/2$) of being received in error;
(ii) if a symbol is received in error, then each of the $q-1$ possible errors is equally likely.

In particular, the *binary symmetric channel* (*BSC*) is a memoryless channel which has channel alphabet $\{0, 1\}$ and channel probabilities

$$\mathcal{P}(1 \text{ received} \mid 0 \text{ sent}) = \mathcal{P}(0 \text{ received} \mid 1 \text{ sent}) = p,$$
$$\mathcal{P}(0 \text{ received} \mid 0 \text{ sent}) = \mathcal{P}(1 \text{ received} \mid 1 \text{ sent}) = 1 - p.$$

Thus, the probability of a bit error in a BSC is p. This is called the *crossover probability* of the BSC (see Fig. 2.2).

Example 2.1.7 Suppose that codewords from the code $\{000, 111\}$ are being sent over a BSC with crossover probability $p = 0.05$. Suppose that the word 110 is received. We can try to find the more likely codeword sent by computing the forward channel probabilities:

$$\begin{aligned}\mathcal{P}(110 \text{ received} \mid 000 \text{ sent}) &= \mathcal{P}(1 \text{ received} \mid 0 \text{ sent})^2 \times \mathcal{P}(0 \text{ received} \mid 0 \text{ sent}) \\ &= (0.05)^2(0.95) = 0.002375, \\ \mathcal{P}(110 \text{ received} \mid 111 \text{ sent}) &= \mathcal{P}(1 \text{ received} \mid 1 \text{ sent})^2 \times \mathcal{P}(0 \text{ received} \mid 1 \text{ sent}) \\ &= (0.95)^2(0.05) = 0.045125.\end{aligned}$$

Since the second probability is larger than the first, we can conclude that 111 is more likely to be the codeword sent.

Decoding rule

In a communication channel with coding, only codewords are transmitted. Suppose that a word $\mathbf{w}$ is received. If $\mathbf{w}$ is a valid codeword, we may conclude that there is no error in the transmission. Otherwise, we know that some errors have occurred. In this case, we need a rule for finding the most likely codeword sent. Such a rule is known as a *decoding rule*. We discuss two such general rules in this chapter. Some other decoding rules, which may apply to certain specific families of codes, will also be introduced in subsequent chapters.

2.2 Maximum likelihood decoding

Suppose that codewords from a code C are being sent over a communication channel. If a word $\mathbf{x}$ is received, we can compute the forward channel probabilities

$$\mathcal{P}(\mathbf{x} \text{ received} \mid \mathbf{c} \text{ sent})$$

for all the codewords $\mathbf{c} \in C$. The *maximum likelihood decoding* (*MLD*) *rule* will conclude that $\mathbf{c}_\mathbf{x}$ is the most likely codeword transmitted if $\mathbf{c}_\mathbf{x}$ maximizes the forward channel probabilities; i.e.,

$$\mathcal{P}(\mathbf{x} \text{ received} \mid \mathbf{c}_\mathbf{x} \text{ sent}) = \max_{\mathbf{c} \in C} \mathcal{P}(\mathbf{x} \text{ received} \mid \mathbf{c} \text{ sent}).$$

There are two kinds of MLD:

(i) *Complete maximum likelihood decoding* (*CMLD*). If a word $\mathbf{x}$ is received, find the most likely codeword transmitted. If there are more than one such codewords, select one of them arbitrarily.

(ii) *Incomplete maximum likelihood decoding* (*IMLD*). If a word $\mathbf{x}$ is received, find the most likely codeword transmitted. If there are more than one such codewords, request a retransmission.

2.3 Hamming distance

Suppose that codewords from a code C are being sent over a BSC with crossover probability $p < 1/2$ (in practice, p should be much smaller than $1/2$). If a word $\mathbf{x}$ is received, then for any codeword $\mathbf{c} \in C$ the forward channel probability is given by

$$\mathcal{P}(\mathbf{x} \text{ received} \mid \mathbf{c} \text{ sent}) = p^e(1-p)^{n-e},$$

where n is the length of $\mathbf{x}$ and e is the number of places at which $\mathbf{x}$ and $\mathbf{c}$ differ. Since $p < 1/2$, it follows that $1 - p > p$, so this probability is larger for

larger values of $n - e$, i.e., for smaller values of e. Hence, this probability is maximized by choosing a codeword $\mathbf{c}$ for which e is as small as possible. This value e leads us to introduce the following fundamental notion of Hamming distance.

Definition 2.3.1 Let $\mathbf{x}$ and $\mathbf{y}$ be words of length n over an alphabet A. The (*Hamming*) *distance* from $\mathbf{x}$ to $\mathbf{y}$, denoted by $d(\mathbf{x}, \mathbf{y})$, is defined to be the number of places at which $\mathbf{x}$ and $\mathbf{y}$ differ. If $\mathbf{x} = x_1 \cdots x_n$ and $\mathbf{y} = y_1 \cdots y_n$, then

$$d(\mathbf{x}, \mathbf{y}) = d(x_1, y_1) + \cdots + d(x_n, y_n), \tag{2.1}$$

where x_i and y_i are regarded as words of length 1, and

$$d(x_i, y_i) = \begin{cases} 1 & \text{if } x_i \neq y_i \\ 0 & \text{if } x_i = y_i. \end{cases}$$

Example 2.3.2 (i) Let $A = \{0, 1\}$ and let $\mathbf{x} = 01010, \mathbf{y} = 01101, \mathbf{z} = 11101$. Then

$$\begin{aligned} d(\mathbf{x}, \mathbf{y}) &= 3, \\ d(\mathbf{y}, \mathbf{z}) &= 1, \\ d(\mathbf{z}, \mathbf{x}) &= 4. \end{aligned}$$

(ii) Let $A = \{0, 1, 2, 3, 4\}$ and let $\mathbf{x} = 1234, \mathbf{y} = 1423, \mathbf{z} = 3214$. Then

$$\begin{aligned} d(\mathbf{x}, \mathbf{y}) &= 3, \\ d(\mathbf{y}, \mathbf{z}) &= 4, \\ d(\mathbf{z}, \mathbf{x}) &= 2. \end{aligned}$$

Proposition 2.3.3 *Let* $\mathbf{x}, \mathbf{y}, \mathbf{z}$ *be words of length* n *over* A. *Then we have*

(i) $0 \leq d(\mathbf{x}, \mathbf{y}) \leq n$,
(ii) $d(\mathbf{x}, \mathbf{y}) = 0$ *if and only if* $\mathbf{x} = \mathbf{y}$,
(iii) $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$,
(iv) (Triangle inequality.) $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$.

Proof. (i), (ii) and (iii) are obvious from the definition of the Hamming distance. By (2.1), it is enough to prove (iv) when $n = 1$, which we now assume.

If $\mathbf{x} = \mathbf{z}$, then (iv) is obviously true since $d(\mathbf{x}, \mathbf{z}) = 0$.

If $\mathbf{x} \neq \mathbf{z}$, then either $\mathbf{y} \neq \mathbf{x}$ or $\mathbf{y} \neq \mathbf{z}$, so (iv) is again true. □

2.4 Nearest neighbour/minimum distance decoding

Suppose that codewords from a code C are being sent over a communication channel. If a word $\mathbf{x}$ is received, the *nearest neighbour decoding rule* (or *minimum distance decoding rule*) will decode $\mathbf{x}$ to $\mathbf{c}_\mathbf{x}$ if $d(\mathbf{x}, \mathbf{c}_\mathbf{x})$ is minimal among all the codewords in C, i.e.,

$$d(\mathbf{x}, \mathbf{c}_\mathbf{x}) = \min_{\mathbf{c} \in C} d(\mathbf{x}, \mathbf{c}). \tag{2.2}$$

Just as for the case of maximum likelihood decoding, we can distinguish between complete and incomplete decoding for the nearest neighbour decoding rule. For a given received word $\mathbf{x}$, if two or more codewords $\mathbf{c}_\mathbf{x}$ satisfy (2.2), then the complete decoding rule arbitrarily selects one of them to be the most likely word sent, while the incomplete decoding rule requests for a retransmission.

Theorem 2.4.1 *For a BSC with crossover probability $p < 1/2$, the maximum likelihood decoding rule is the same as the nearest neighbour decoding rule.*

Proof. Let C denote the code in use and let $\mathbf{x}$ denote the received word (of length n). For any vector $\mathbf{c}$ of length n, and for any $0 \leq i \leq n$,

$$d(\mathbf{x}, \mathbf{c}) = i \Leftrightarrow \mathcal{P}(\mathbf{x} \text{ received} \mid \mathbf{c} \text{ sent}) = p^i(1-p)^{n-i}.$$

Since $p < 1/2$, it follows that

$$p^0(1-p)^n > p^1(1-p)^{n-1} > p^2(1-p)^{n-2} > \cdots > p^n(1-p)^0.$$

By definition, the maximum likelihood decoding rule decodes $\mathbf{x}$ to $\mathbf{c} \in C$ such that $\mathcal{P}(\mathbf{x} \text{ received} \mid \mathbf{c} \text{ sent})$ is the largest, i.e., such that $d(\mathbf{x}, \mathbf{c})$ is the smallest (or seeks retransmission if incomplete decoding is in use and $\mathbf{c}$ is not unique). Hence, it is the same as the nearest neighbour decoding rule. □

Remark 2.4.2 From now on, we will assume that all BSCs have crossover probabilities $p < 1/2$. Consequently, we can use the minimum distance decoding rule to perform MLD.

Example 2.4.3 Suppose codewords from the binary code

$$C = \{0000, 0011, 1000, 1100, 0001, 1001\}$$

are being sent over a BSC. Assuming $\mathbf{x} = 0111$ is received, then

$$\begin{aligned} d(0111, 0000) &= 3, \\ d(0111, 0011) &= 1, \\ d(0111, 1000) &= 4, \end{aligned}$$