Chapter 1

Introduction

In this chapter, we introduce the model of a communication system, as originally proposed by Claude E. Shannon in 1948. We will then focus on the channel portion of the system and define the concept of a probabilistic channel, along with models of an encoder and a decoder for the channel. As our primary example of a probabilistic channel—here, as well as in subsequent chapters—we will introduce the memoryless q-ary symmetric channel, with the binary case as the prevailing instance used in many practical applications. For $q = 2$ (the binary case), we quote two key results in information theory. The first result is a coding theorem, which states that information through the channel can be transmitted with an arbitrarily small probability of decoding error, as long as the transmission rate is below a quantity referred to as the capacity of the channel. The second result is a converse coding theorem, which states that operating at rates above the capacity necessarily implies unreliable transmission.

In the remaining part of the chapter, we shift to a combinatorial setting and characterize error events that can occur in channels such as the q-ary symmetric channel, and can always be corrected by suitably selected encoders and decoders. We exhibit the trade-off between error correction and error detection: while an error-detecting decoder provides less information to the receiver, it allows us to handle twice as many errors. In this context, we will become acquainted with the erasure channel, in which the decoder has access to partial information about the error events, namely, the location of the symbols that might be in error. We demonstrate that—here as well—such information allows us to double the number of correctable errors.

1.1 Communication systems

Figure 1.1 shows a communication system for transmitting information from a *source* to a *destination* through a *channel*. The communication can be

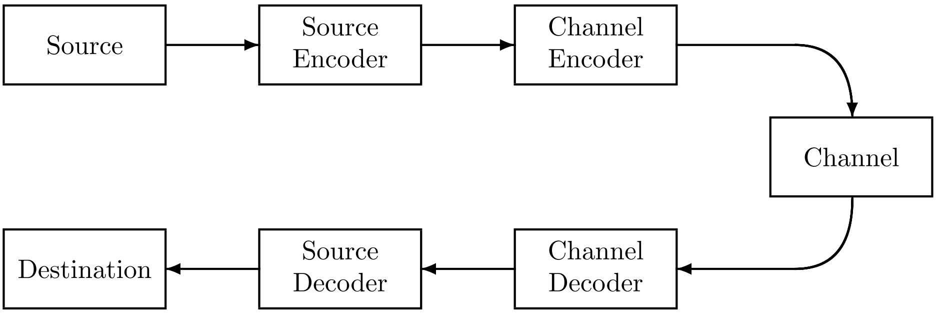

Figure 1.1. Communication system.

either in the space domain (i.e., from one location to another) or in the time domain (i.e., by storing data at one point in time and retrieving it some time later).

The role of source coding is twofold. First, it serves as a translator between the output of the source and the input to the channel. For example, the information that is transmitted from the source to the destination may consist of analog signals, while the channel may expect to receive digital input; in such a case, an analog-to-digital conversion will be required at the encoding stage, and then a back conversion is required at the decoding stage. Secondly, the source encoder may *compress* the output of the source for the purpose of economizing on the length of the transmission; at the other end, the source decoder decompresses the received signal or sequence. Some applications require that the decoder restore the data so that it is identical to the original, in which case we say that the compression is *lossless*. Other applications, such as most audio and image transmissions, allow some (controlled) difference—or distortion—between the original and the restored data, and this flexibility is exploited to achieve higher compression; the compression is then called *lossy*.

Due to physical and engineering limitations, channels are not perfect: their output may differ from their input because of noise or manufacturing defects. Furthermore, sometimes the design requires that the format of the data at the output of the channel (e.g., the set of signals that can be read at the output) should differ from the input format. In addition, there are applications, such as magnetic and optical mass storage media, where certain patterns are not allowed to appear in the recorded (i.e., transmitted) bit stream. The main role of channel coding is to overcome such limitations and to make the channel as transparent as possible from the source and destination points of view. The task of signal translation, which was mentioned earlier in the context of source coding, may be undertaken partially (or wholly) also by the channel encoder and decoder.

1.2 Channel coding

We will concentrate on the channel coding part of Figure 1.1, as shown in Figure 1.2.

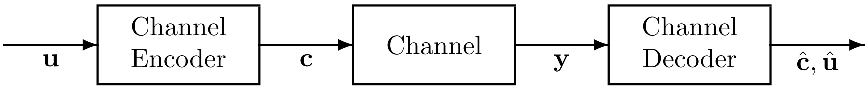

Figure 1.2. Channel coding.

Our model of the channel will be that of the *(discrete) probabilistic channel*: a probabilistic channel S is defined as a triple (F, Φ, Prob), where F is a finite *input alphabet*, Φ is a finite *output alphabet*, and Prob is a conditional probability distribution

$$\mathsf{Prob}\{\,\mathbf{y} \text{ received} \mid \mathbf{x} \text{ transmitted}\,\}$$

defined for every pair $(\mathbf{x}, \mathbf{y}) \in F^m \times \Phi^m$, where m ranges over all positive integers and F^m (respectively, Φ^m) denotes the set of all words of length m over F (respectively, over Φ). (We assume here that the channel neither deletes nor inserts symbols; that is, the length of an output word $\mathbf{y}$ always equals the length of the respective input word $\mathbf{x}$.)

The input to the channel encoder is an *information word* (or *message*) $\mathbf{u}$ out of M possible information words (see Figure 1.2). The channel encoder generates a *codeword* $\mathbf{c} \in F^n$ that is input to the channel. The resulting output of the channel is a *received word* $\mathbf{y} \in \Phi^n$, which is fed into the channel decoder. The decoder, in turn, produces a *decoded codeword* $\hat{\mathbf{c}}$ and a *decoded information word* $\hat{\mathbf{u}}$, with the aim of having $\mathbf{c} = \hat{\mathbf{c}}$ and $\mathbf{u} = \hat{\mathbf{u}}$. This implies that the channel encoder needs to be such that the mapping $\mathbf{u} \mapsto \mathbf{c}$ is one-to-one.

The *rate* of the channel encoder is defined as

$$R = \frac{\log_{|F|} M}{n} \,.$$

If all information words have the same length over F, then this length is given by the numerator, $\log_{|F|} M$, in the expression for R (strictly speaking, we need to round up the numerator in order to obtain that length; however, this integer effect phases out once we aggregate over a sequence of $\ell \to \infty$ transmissions, in which case the number of possible information words becomes M^ℓ and the codeword length is $\ell \cdot n$). Since the mapping of the encoder is one-to-one, we have $R \le 1$.

The encoder and decoder parts in Figure 1.2 will be the subject of Sections 1.3 and 1.4, respectively. We next present two (related) examples of probabilistic channels, which are very frequently found in practical applications.

Example 1.1 The memoryless *binary symmetric channel* (in short, BSC) is defined as follows. The input and output alphabets are $F = \Phi = \{0, 1\}$, and for every two binary words $\mathbf{x} = x_1 x_2 \ldots x_m$ and $\mathbf{y} = y_1 y_2 \ldots y_m$ of a given length m,

$$\begin{aligned} &\mathsf{Prob}\{\, \mathbf{y} \text{ received} \mid \mathbf{x} \text{ transmitted} \,\} \\ &\qquad = \prod_{j=1}^{m} \mathsf{Prob}\{\, y_j \text{ received} \mid x_j \text{ transmitted} \,\} , \end{aligned} \tag{1.1}$$

where, for every $x, y \in F$,

$$\mathsf{Prob}\{\, y \text{ was received} \mid x \text{ was transmitted} \,\} = \begin{cases} 1-p & \text{if } y = x \\ p & \text{if } y \neq x \end{cases} .$$

The parameter p is a real number in the range $0 \le p \le 1$ and is called the *crossover probability* of the channel.

The action of the BSC can be described as flipping each input bit with probability p, independently of the past or the future (the adjective "memoryless" reflects this independence). The channel is called "symmetric" since the probability of the flip is the same regardless of whether the input is 0 or 1. The BSC is commonly represented by a diagram as shown in Figure 1.3. The possible input values appear to the left and the possible output values are shown to the right. The label of a given edge from input x to output y is the conditional probability of receiving the output y given that the input is x.

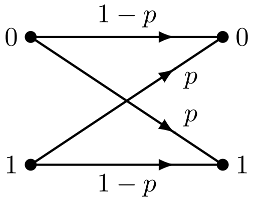

Figure 1.3. Binary symmetric channel.

The cases $p = 0$ and $p = 1$ correspond to reliable communication, whereas $p = \frac{1}{2}$ stands for the case where the output of the channel is statistically independent of its input. □

Example 1.2 The memoryless *q-ary symmetric channel* with crossover probability p is a generalization of the BSC to alphabets $F = \Phi$ of size q. The

Cambridge University Press
978-0-521-84504-5 - Introduction to Coding Theory
Ron M. Roth
Excerpt
More information

conditional probability (1.1) now holds for every two words $\mathbf{x} = x_1x_2\ldots x_m$ and $\mathbf{y} = y_1y_2\ldots y_m$ over F, where

$$\mathsf{Prob}\{\, y \text{ was received} \mid x \text{ was transmitted} \,\} = \begin{cases} 1-p & \text{if } y = x \\ p/(q-1) & \text{if } y \neq x \end{cases} .$$

(While the term "crossover" is fully justified only in the binary case, we will nevertheless use it for the general q-ary case as well.) □

In the case where the input alphabet F has the same (finite) size as the output alphabet Φ, it will be convenient to assume that $F = \Phi$ and that the elements of F form a finite Abelian group (indeed, for every positive integer q there is an Abelian group of size q, e.g., the ring $\mathbb{Z}_q$ of integer residues modulo q; see Problem A.21 in the Appendix). We then say that the channel is an *additive channel*. Given an additive channel, let $\mathbf{x}$ and $\mathbf{y}$ be input and output words, respectively, both in F^m. The *error word* is defined as the difference $\mathbf{y}-\mathbf{x}$, where the subtraction is taken component by component. The action of the channel can be described as adding (component by component) an error word $\mathbf{e} \in F^m$ to the input word $\mathbf{x}$ to produce the output word $\mathbf{y} = \mathbf{x} + \mathbf{e}$, as shown in Figure 1.4. In general, the distribution of the error word $\mathbf{e}$ may depend on the input $\mathbf{x}$. The q-ary symmetric channel is an example of a channel where $\mathbf{e}$ is statistically independent of $\mathbf{x}$ (in such cases, the term *additive noise* is sometimes used for the error word $\mathbf{e}$).

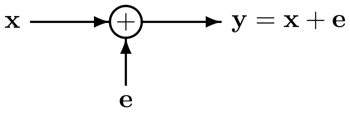

Figure 1.4. Additive channel.

When F is an Abelian group, it contains the zero (or unit) element. The *error locations* are the indexes of the nonzero entries in the error word $\mathbf{e}$. Those entries are referred to as the *error values*.

1.3 Block codes

An (n, M) *(block) code* over a finite alphabet F is a nonempty subset $\mathcal{C}$ of size M of F^n. The parameter n is called the *code length* and M is the *code size*. The *dimension* (or *information length*) of $\mathcal{C}$ is defined by $k = \log_{|F|} M$, and the *rate* of $\mathcal{C}$ is $R = k/n$. The range of the mapping defined by the channel encoder in Figure 1.2 forms an (n, M) code, and this is the context in which the term (n, M) code will be used. The elements of a code are called *codewords*.

In addition to the length and the size of a code, we will be interested in the sequel also in quantifying how much the codewords in the code differ from one another. To this end, we will make use of the following definitions.

Let F be an alphabet. The *Hamming distance* between two words $\mathbf{x}, \mathbf{y} \in F^n$ is the number of coordinates on which $\mathbf{x}$ and $\mathbf{y}$ differ. We denote the Hamming distance by $\mathsf{d}(\mathbf{x}, \mathbf{y})$.

It is easy to verify that the Hamming distance satisfies the following properties of a metric for every three words $\mathbf{x}, \mathbf{y}, \mathbf{z} \in F^n$:

- $\mathsf{d}(\mathbf{x}, \mathbf{y}) \geq 0$, with equality if and only if $\mathbf{x} = \mathbf{y}$.
- Symmetry: $\mathsf{d}(\mathbf{x}, \mathbf{y}) = \mathsf{d}(\mathbf{y}, \mathbf{x})$.
- The triangle inequality: $\mathsf{d}(\mathbf{x}, \mathbf{y}) \leq \mathsf{d}(\mathbf{x}, \mathbf{z}) + \mathsf{d}(\mathbf{z}, \mathbf{y})$.

Let F be an Abelian group. The *Hamming weight* of $\mathbf{e} \in F^n$ is the number of nonzero entries in $\mathbf{e}$. We denote the Hamming weight by $\mathsf{w}(\mathbf{e})$. Notice that for every two words $\mathbf{x}, \mathbf{y} \in F^n$,

$$\mathsf{d}(\mathbf{x}, \mathbf{y}) = \mathsf{w}(\mathbf{y} - \mathbf{x}) .$$

Turning now back to block codes, let $\mathcal{C}$ be an (n, M) code over F with $M > 1$. The *minimum distance* of $\mathcal{C}$ is the minimum Hamming distance between any two distinct codewords of $\mathcal{C}$; that is, the minimum distance d is given by

$$d = \min_{\mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C} \,:\, \mathbf{c}_1 \neq \mathbf{c}_2} \mathsf{d}(\mathbf{c}_1, \mathbf{c}_2) .$$

An (n, M) code with minimum distance d is called an (n, M, d) *code* (when we specify the minimum distance d of an (n, M) code, we implicitly indicate that $M > 1$). We will sometimes use the notation $\mathsf{d}(\mathcal{C})$ for the minimum distance of a given code $\mathcal{C}$.

Example 1.3 The binary $(3, 2, 3)$ *repetition code* is the code

$$\{000, 111\}$$

over $F = \{0, 1\}$. The dimension of the code is $\log_2 2 = 1$ and its rate is $1/3$. □

Example 1.4 The binary $(3, 4, 2)$ *parity code* is the code

$$\{000, 011, 101, 110\}$$

over $F = \{0, 1\}$. The dimension is $\log_2 4 = 2$ and the code rate is $2/3$. □

1.4 Decoding

1.4.1 Definition of decoders

Let $\mathcal{C}$ be an (n, M, d) code over an alphabet F and let S be a channel defined by the triple (F, Φ, Prob). A *decoder* for the code $\mathcal{C}$ with respect to the channel S is a function

$$\mathcal{D} : \Phi^n \rightarrow \mathcal{C} .$$

The *decoding error probability* $\mathrm{P}_{\mathrm{err}}$ of $\mathcal{D}$ is defined by

$$\mathrm{P}_{\mathrm{err}} = \max_{\mathbf{c} \in \mathcal{C}} \mathrm{P}_{\mathrm{err}}(\mathbf{c}) ,$$

where

$$\mathrm{P}_{\mathrm{err}}(\mathbf{c}) = \sum_{\mathbf{y} \,:\, \mathcal{D}(\mathbf{y}) \neq \mathbf{c}} \mathsf{Prob}\{ \mathbf{y} \text{ received} \mid \mathbf{c} \text{ transmitted} \} .$$

Note that $\mathrm{P}_{\mathrm{err}}(\mathbf{c})$ is the probability that the codeword $\mathbf{c}$ will be decoded erroneously, given that $\mathbf{c}$ was transmitted.

Our goal is to have decoders with small $\mathrm{P}_{\mathrm{err}}$.

Example 1.5 Let $\mathcal{C}$ be the binary $(3, 2, 3)$ repetition code and let S be the BSC with crossover probability p.

Define a decoder $\mathcal{D} : \{0, 1\}^3 \rightarrow \mathcal{C}$ as follows:

$$\mathcal{D}(000) = \mathcal{D}(001) = \mathcal{D}(010) = \mathcal{D}(100) = 000$$

and

$$\mathcal{D}(011) = \mathcal{D}(101) = \mathcal{D}(110) = \mathcal{D}(111) = 111 .$$

The probability $\mathrm{P}_{\mathrm{err}}$ equals the probability of having two or more errors:

$$\begin{aligned} \mathrm{P}_{\mathrm{err}} = \mathrm{P}_{\mathrm{err}}(000) = \mathrm{P}_{\mathrm{err}}(111) &= \tbinom{3}{2} p^2 (1-p) + \tbinom{3}{3} p^3 \\ &= 3p^2 - 3p^3 + p^3 \\ &= p(2p-1)(1-p) + p . \end{aligned}$$

So, $\mathrm{P}_{\mathrm{err}}$ is smaller than p when $p < 1/2$, which means that coding has improved the probability of error per message, compared to uncoded transmission. The price, however, is reflected in the rate: three bits are transmitted for every information bit (a rate of $(\log_2 M)/n = 1/3$). □

1.4.2 Maximum-likelihood decoding

We next consider particular decoding strategies for codes and channels. Given an (n, M, d) code $\mathcal{C}$ over F and a channel $S = (F, \Phi, \mathsf{Prob})$, a *maximum-likelihood decoder* (MLD) for $\mathcal{C}$ with respect to S is the function $\mathcal{D}_{\mathrm{MLD}} : \Phi^n \to \mathcal{C}$ defined as follows: for every $\mathbf{y} \in \Phi^n$, the value $\mathcal{D}_{\mathrm{MLD}}(\mathbf{y})$ equals the codeword $\mathbf{c} \in \mathcal{C}$ that maximizes the probability

$$\mathsf{Prob}\{\, \mathbf{y} \text{ received} \mid \mathbf{c} \text{ transmitted} \,\} \; .$$

In the case of a tie between two (or more) codewords, we choose one of the tying codewords arbitrarily (say, the first according to some lexicographic ordering on $\mathcal{C}$). Hence, $\mathcal{D}_{\mathrm{MLD}}$ is well-defined for the code $\mathcal{C}$ and the channel S.

A *maximum a posteriori decoder* for $\mathcal{C}$ with respect to a channel $S = (F, \Phi, \mathsf{Prob})$ is defined similarly, except that now the codeword $\mathbf{c}$ maximizes the probability

$$\mathsf{Prob}\{\, \mathbf{c} \text{ transmitted} \mid \mathbf{y} \text{ received} \,\} \; .$$

In order to compute such a probability, however, we also need to know the *a priori* probability of transmitting $\mathbf{c}$. So, unlike an MLD, a maximum *a posteriori* decoder assumes some distribution on the codewords of $\mathcal{C}$. Since

$$\begin{aligned} &\mathsf{Prob}\{\, \mathbf{c} \text{ transmitted} \mid \mathbf{y} \text{ received} \,\} \\ &\qquad = \mathsf{Prob}\{\, \mathbf{y} \text{ received} \mid \mathbf{c} \text{ transmitted} \,\} \cdot \frac{\mathsf{Prob}\{\, \mathbf{c} \text{ transmitted} \,\}}{\mathsf{Prob}\{\, \mathbf{y} \text{ received} \,\}} \; , \end{aligned}$$

the terms maximum *a posteriori* decoder and MLD coincide when the *a priori* probabilities $\mathsf{Prob}\{\, \mathbf{c} \text{ transmitted} \,\}$ are the same for all $\mathbf{c} \in \mathcal{C}$; namely, they are all equal to $1/M$.

Example 1.6 We compute an MLD for an (n, M, d) code $\mathcal{C}$ with respect to the BSC with crossover probability $p < 1$. Let $\mathbf{c} = c_1 c_2 \ldots c_n$ be a codeword in $\mathcal{C}$ and $\mathbf{y} = y_1 y_2 \ldots y_n$ be a word in $\{0, 1\}^n$. Then

$$\begin{aligned} &\mathsf{Prob}\{\, \mathbf{y} \text{ received} \mid \mathbf{c} \text{ transmitted} \,\} \\ &\qquad\qquad = \prod_{j=1}^{n} \mathsf{Prob}\{\, y_j \text{ received} \mid c_j \text{ transmitted} \,\} \; , \end{aligned}$$

where

$$\mathsf{Prob}\{\, y_j \text{ received} \mid c_j \text{ transmitted} \,\} = \begin{cases} 1-p & \text{if } y_j = c_j \\ p & \text{otherwise} \end{cases} \; .$$

Therefore,

$$\begin{aligned} \mathsf{Prob}\{\, \mathbf{y} \text{ received} \mid \mathbf{c} \text{ transmitted} \,\} &= p^{\mathsf{d}(\mathbf{y},\mathbf{c})} (1-p)^{n-\mathsf{d}(\mathbf{y},\mathbf{c})} \\ &= (1-p)^n \cdot \left(\frac{p}{1-p} \right)^{\mathsf{d}(\mathbf{y},\mathbf{c})} \; , \end{aligned}$$

where $\mathsf{d}(\mathbf{y}, \mathbf{c})$ is the Hamming distance between $\mathbf{y}$ and $\mathbf{c}$. Observing that $p/(1-p) < 1$ when $p < 1/2$, it follows that—with respect to the BSC with crossover probability $p < 1/2$—for every (n, M, d) code $\mathcal{C}$ and every word $\mathbf{y} \in \{0,1\}^n$, the value $\mathcal{D}_{\mathrm{MLD}}(\mathbf{y})$ is a closest codeword in $\mathcal{C}$ to $\mathbf{y}$. In fact, this holds also for the q-ary symmetric channel whenever the crossover probability is less than $1-(1/q)$ (Problem 1.7). □

A *nearest-codeword decoder* for an (n, M) code $\mathcal{C}$ over F is a function $F^n \to \mathcal{C}$ whose value for every word $\mathbf{y} \in F^n$ is a closest codeword in $\mathcal{C}$ to $\mathbf{y}$, where the term "closest" is with respect to the Hamming distance. A nearest-codeword decoder for $\mathcal{C}$ is a decoder for $\mathcal{C}$ with respect to any additive channel whose input and output alphabets are F. From Example 1.6 we get that with respect to the BSC with crossover probability $p < 1/2$, the terms MLD and nearest-codeword decoder coincide.

1.4.3 Capacity of the binary symmetric channel

We have seen in Example 1.5 that coding allows us to reduce the decoding error probability $\mathrm{P}_{\mathrm{err}}$, at the expense of transmitting at lower rates. We next see that we can, in fact, achieve arbitrarily small values of $\mathrm{P}_{\mathrm{err}}$, while still transmitting at rates that are bounded away from 0.

Define the *binary entropy function* $\mathsf{H} : [0,1] \to [0,1]$ by

$$\mathsf{H}(x) = -x \log_2 x - (1-x)\log_2(1-x) ,$$

where $\mathsf{H}(0) = \mathsf{H}(1) = 0$. The binary entropy function is shown in Figure 1.5. It is symmetric with respect to $x = 1/2$ and takes its maximum at that point ($\mathsf{H}(1/2) = 1$). It is $\cap$-concave and has an infinite derivative at $x = 0$ and $x = 1$ (a real function f is $\cap$-concave over a given interval if for every two points x_1 and x_2 in that interval, the line segment that connects the points $(x_1, f(x_1))$ and $(x_2, f(x_2))$ lies entirely on or below the function curve in the real plane; the function f is called $\cup$-convex if $-f$ is $\cap$-concave).

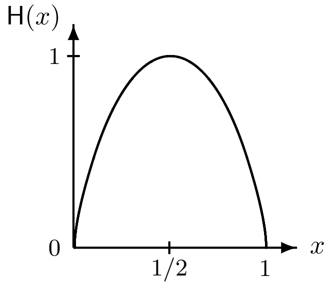

Figure 1.5. Binary entropy function.

Let S be the BSC with crossover probability p. The *capacity* of S is given by

$$\mathsf{cap}(S) = 1 - \mathsf{H}(p) \ .$$

The capacity is shown in Figure 1.6 as a function of p. Notice that $\mathsf{cap}(S) = 1$ when $p \in \{0, 1\}$ and $\mathsf{cap}(S) = 0$ when $p = 1/2$.

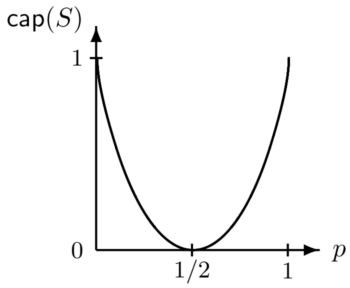

Figure 1.6. Capacity of the BSC.

The next two theorems are special cases of fundamental results in information theory. These results state that the capacity of a channel is the largest rate at which information can be transmitted reliably through that channel.

Theorem 1.1 (Shannon Coding Theorem for the BSC) *Let S be the memoryless binary symmetric channel with crossover probability p and let R be a real in the range $0 \le R < \mathsf{cap}(S)$. There exists an infinite sequence of (n_i, M_i) block codes over $F = \{0, 1\}$, $i = 1, 2, 3, \cdots$, such that $(\log_2 M_i)/n_i \ge R$ and, for maximum-likelihood decoding for those codes (with respect to S), the decoding error probability $\mathrm{P}_{\mathrm{err}}$ approaches 0 as $i \to \infty$.*

Theorem 1.2 (Shannon Converse Coding Theorem for the BSC) *Let S be the memoryless binary symmetric channel with crossover probability p and let R be a real greater than $\mathsf{cap}(S)$. Consider $\underline{\text{any}}$ infinite sequence of (n_i, M_i) block codes over $F = \{0, 1\}$, $i = 1, 2, 3, \cdots$, such that $(\log_2 M_i)/n_i \ge R$ and $n_1 < n_2 < \cdots < n_i < \cdots$. Then, for $\underline{\text{any}}$ decoding scheme for those codes (with respect to S), the decoding error probability $\mathrm{P}_{\mathrm{err}}$ approaches 1 as $i \to \infty$.*

The proofs of these theorems will be given in Chapter 4. In particular, we will show there that $\mathrm{P}_{\mathrm{err}}$ in Theorem 1.1 can be guaranteed to decrease exponentially with the code length n_i. On the other hand, our proof in that chapter will only establish the *existence* of codes with the property that is stated in the theorem, without exhibiting an efficient algorithm for producing them. The constructive part will be filled in later on in Section 12.5. At