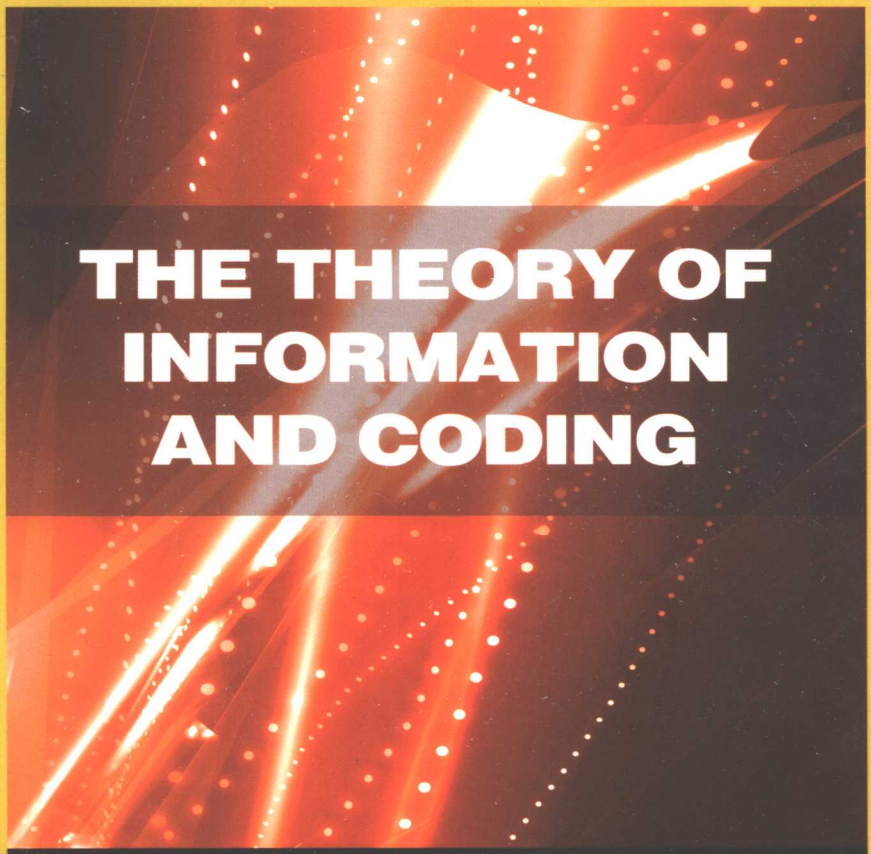
英文版

通信与信息科学教育丛书



信息论与编码理论
（第2版）

ROBERT J. McELIECE 著

THE THEORY OF INFORMATION AND CODING
SECOND EDITION

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

通信与信息科学教育丛书

The Theory of Information and Coding

Second Edition

信息论与编码理论

(第2版)

Robert J. McEliece 著

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

内容简介

本书主要介绍由香农理论发展起来的信息论与编码理论，用于解决通信中的基本问题。首先简要介绍编码的概念；第一部分介绍香农理论、信道与信源编码理论；第二部分详细介绍几种编码方案，可用于信道与信源编码。书中提供了大量实例，每章末均有习题与说明，以便于具有概率论与线性代数知识的读者掌握和理解。

本书可用做信息、通信、电子工程等专业的相关课教材，也可作为有一定英语基础的人员自学使用。

版权贸易合同登记号：01－2002－6469

图书在版编目（CIP）数据

信息论与编码理论 = The Theory of Information and Coding：第 2 版/（美）麦克尔里思（Mceliece，R. J.）著. —北京：电子工业出版社，2003.1
（通信与信息科学教育丛书）
ISBN 7-5053-8382-5

Ⅰ.信… Ⅱ.麦… Ⅲ.①信息论—英文②信源编码—英文 Ⅳ.TN911.2

中国版本图书馆 CIP 数据核字（2002）第 104582 号

责任编辑：段 颖
印 刷：北京天竺颖华印刷厂
出版发行：电子工业出版社 http://www.phei.com.cn
北京市海淀区万寿路 173 信箱 邮编 100036
经 销：各地新华书店
开 本：787×980 1/16 印张：25.75
版 次：2003 年 1 月第 1 版 2003 年 1 月第 1 次印刷
印 数：5 000 册 定价：35.00 元

出 版 说 明

近年来，通信与信息科技发展之快和应用之广，大大超出了人们的预料和专家的预测。从国民经济到社会生活的日益信息化，标志着通信与信息科技的空前发展。

为了满足高等院校师生教改和教学的需求以及广大技术人员学习通信与信息新技术的需要，电子工业出版社约请北京地区的清华大学、北京大学、北京航空航天大学、北京邮电大学、北方交通大学、北京理工大学，南京地区的东南大学、解放军理工大学、南京邮电学院，上海地区的上海交通大学，成都地区的西南交通大学、电子科技大学，西安地区的西安电子科技大学、西安交通大学，天津地区的南开大学，深圳地区的深圳大学，东北地区的哈尔滨工业大学等全国知名高等院校教学第一线上的教授和信息产业部有关科研院所的专家，请他们推荐和反复论证，从国外优秀的英文版图书中精选出版了这套《通信与信息科学教育丛书》（英文版）。

本套丛书可作为高等院校通信、计算机、电子信息等专业的高年级本科生、研究生的教材或教学参考书，也适合广大信息产业技术人员参考。

本套丛书所选取的均是国际上通信与信息科学领域具有代表性的经典著作，它们在全世界许多大学被用做教材或教学参考书。其主要特点是具有较强的先进性、实用性和权威性。丛书内容丰富，深入浅出，层次清楚，理论与应用并重，能够较好地引导读者将现代通信信息与信息科学的原理、技术与应用有机结合。我们希望本套丛书能够进一步推动国内高等院校教学与国际接轨，同时满足广大技术人员及时学习通信与信息科学领域中新知识的需求。

恳请广大读者提出宝贵意见和建议（E-mail：davidzhu@ phei. com. cn），以使我们奉献更多、更好的英文原版精品图书。

电子工业出版社
通信与电子技术图书事业部

Foreword

Transmission of information is at the heart of what we call communication. As an area of concern it is so vast as to touch upon the preoccupations of philosophers and to give rise to a thriving technology.

We owe to the genius of Claude Shannon* the recognition that a large class of problems related to encoding, transmitting, and decoding information can be approached in a systematic and disciplined way: his classic paper of 1948 marks the birth of a new chapter of Mathematics.

In the past thirty years there has grown a staggering literature in this fledgling field, and some of its terminology even has become part of our daily language.

The present monograph (actually two monographs in one) is an excellent introduction to the two aspects of communication: coding and transmission.

The first (which is the subject of Part two) is an elegant illustration of the power and beauty of Algebra; the second belongs to Probability Theory which the chapter begun by Shannon enriched in novel and unexpected ways.

MARK KAC
General Editor, Section on Probability

* C. E. Shannon, A Mathematical Theory of Communication, *Bell System Tech. J.* **27** (1948), Introduction: 379–382; Part one: Discrete Noiseless Systems, 382–405; Part two: The Discrete Channel with Noise (and Appendixes), 406–423; Part III: Mathematical Preliminaries, 623–636; Part IV: The Continuous Channel (and Appendixes), 637–656).

Preface to the first edition

This book is meant to be a self-contained introduction to the basic results in the theory of information and coding. It was written during 1972–1976, when I taught this subject at Caltech. About half my students were electrical engineering graduate students; the others were majoring in all sorts of other fields (mathematics, physics, biology, even one English major!). As a result the course was aimed at nonspecialists as well as specialists, and so is this book.

The book is in three parts: Introduction, Part one (Information Theory), and Part two (Coding Theory). It is essential to read the introduction first, because it gives an overview of the whole subject. In Part one, Chapter 1 is fundamental, but it is probably a mistake to read it first, since it is really just a collection of technical results about entropy, mutual information, and so forth. It is better regarded as a reference section, and should be consulted as necessary to understand Chapters 2–5. Chapter 6 is a survey of advanced results, and can be read independently. In Part two, Chapter 7 is basic and must be read before Chapters 8 and 9; but Chapter 10 is almost, and Chapter 11 is completely, independent from Chapter 7. Chapter 12 is another survey chapter independent of everything else.

The problems at the end of the chapters are very important. They contain verification of many omitted details, as well as many important results not mentioned in the text. It is a good idea to at least read the problems.

There are four appendices. Appendix A gives a brief survey of probability theory, essential for Part one. Appendix B discusses convex functions and Jensen's inequality. Appeals to Jensen's inequality are frequent in Part one, and the reader unfamiliar with it should read Appendix B at the first opportunity. Appendix C sketches the main results about finite fields needed in Chapter 9. Appendix D describes an algorithm for counting paths in directed graphs which is needed in Chapter 10.

A word about cross-references is in order: sections, figures, examples, theorems, equations, and problems are numbered consecutively by chapters, using double numeration. Thus "Section 2.3," "Theorem 3.4," and "Prob. 4.17" refer to section 3 of Chapter 2, Theorem 4 of Chapter 3, and Problem 17 of Chapter 4, respectively. The appendices are referred to by letter; thus "Equation (B.4)" refers to the fourth numbered equation in Appendix B.

The following special symbols perhaps need explanation: "□" signals the end of a proof or example; "iff" means *if and only if*; $\lfloor x \rfloor$ denotes the largest integer $\leqslant x$; and $\lceil x \rceil$ denotes the smallest integer $\geqslant x$.

Finally, I am happy to acknowledge my debts: To Gus Solomon, for introducing me to the subject in the first place; to John Pierce, for giving me the opportunity to teach at Caltech; to Gian-Carlo Rota, for encouraging me to write this book; to Len Baumert, Stan Butman, Gene Rodemich, and Howard Rumsey, for letting me pick their brains; to Jim Lesh and Jerry Heller, for supplying data for Figures 6.7 and 12.2; to Bob Hall, for drafting the figures; to my typists, Ruth Stratton, Lillian Johnson, and especially Dian Rapchak; and to Ruth Flohn for copy editing.

ROBERT J. MCELIECE

Preface to the second edition

The main changes in this edition are in Part two. The old Chapter 8 ("BCH, Goppa, and Related Codes") has been revised and expanded into two new chapters, numbered 8 and 9. The old chapters 9, 10, and 11 have then been renumbered 10, 11, and 12. The new Chapter 8 ("Cyclic codes") presents a fairly complete treatment of the mathematical theory of cyclic codes, and their implementation with shift register circuits. It culminates with a discussion of the use of cyclic codes in burst error correction. The new Chapter 9 ("BCH, Reed–Solomon, and Related Codes") is much like the old Chapter 8, except that increased emphasis has been placed on Reed-Solomon codes, reflecting their importance in practice. Both of the new chapters feature dozens of new problems.

Contents

Introduction

In 1948, in the introduction to his classic paper, "A mathematical theory of communication," Claude Shannon[1,*] wrote:

> *"The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point."*

To solve that problem he created, in the pages that followed, a completely new branch of applied mathematics, which is today called *information theory* and/or *coding theory*. This book's object is the presentation of the main results of this theory as they stand 30 years later.

In this introductory chapter we illustrate the central ideas of information theory by means of a specific pair of mathematical models, the *binary symmetric source* and the *binary symmetric channel*.

The binary symmetric source (the source, for short) is an object which emits one of two possible symbols, which we take to be "0" and "1," at a rate of R symbols per unit of time. We shall call these symbols *bits*, an abbreviation of *binary digits*. The bits emitted by the source are random, and a "0" is as likely to be emitted as a "1." We imagine that the source rate R is continuously variable, that is, R can assume any nonnegative value.

The binary symmetric channel (the *BSC*[2] for short) is an object through which it is possible to transmit one bit per unit of time. However, the channel is not completely reliable: there is a fixed probability p (called the *raw bit error probability*[3]), $0 \leq p \leq \frac{1}{2}$, that the output bit will not be the same as the input bit.

We now imagine two individuals, the sender and the receiver. The sender must try to convey to the receiver as accurately as possible the source output,

---

* Notes, denoted by superior numerals, appear at the end of most chapters.

and the only communication link allowed between the two is the BSC described above. (However, we will allow the sender and receiver to get together before the source is turned on, so that each will know the nature of the data-processing strategies the other will be using.) We assume that both the sender and receiver have access to unlimited amounts of computing power, storage capacity, government funds, and other resources.

We now ask, For a given source rate R, how accurately can the sender communicate with the receiver over the BSC? We shall eventually give a very precise general answer to this question, but let's begin by considering some special cases.

Suppose $R = 1/3$. This means that the channel can transmit bits three times as fast as the source produces them, so the source output can be *encoded* before transmission by repeating each bit three times. For example, if the source's first five bits were 10100, the encoded stream would be 111000111000000. The receiver will get three versions of each source bit, but because of the channel "noise" these versions may not all be the same. If the channel garbled the second, fifth, sixth, twelfth, and thirteenth transmitted bits, the receiver would receive 101011111001100. A little thought should convince you that in this situation the receiver's best strategy for *decoding* a given source bit is to take the majority vote of the three versions of it. In our example he would decode the received message as 11100, and would make an error in the second bit. In general, a source bit will be received in error if either two or three of its three copies are garbled by the channel. Thus, if P_e denotes the *bit error probability*,

$$\begin{aligned} P_e &= P\{2 \text{ channel errors}\} + P\{3 \text{ channel errors}\} \\ &= 3p^2(1-p) \qquad + p^3 \\ &= 3p^2 - 2p^3. \end{aligned} \tag{0.1}$$

Since $p \leq \frac{1}{2}$, this is less than the raw bit error probability p; our simple coding scheme has improved the channel's reliability, and for very small p the relative improvement is dramatic.

It is now easy to see that even higher reliability can be achieved by repeating each bit more times. Thus, if $R = 1/(2n+1)$ for some integer n, we could repeat each bit $2n+1$ times before transmission (see Prob. 0.2) and use majority-vote decoding as before. It is simple to obtain a formula for the resulting bit error probability $P_e^{(2n+1)}$:

$$P_e^{(2n+1)} = \sum_{k=n+1}^{2n+1} P\{k \text{ channel errors out of } 2n+1 \text{ transmitted bits}\}$$

$$= \sum_{k=n+1}^{2n+1} \binom{2n+1}{k} p^k (1-p)^{2n+1-k}$$

$$= \binom{2n+1}{n+1} p^{n+1} + \text{terms of higher degree in } p. \tag{0.2}$$

If $n > 1$, this approaches 0 much more rapidly as $p \to 0$ than the special case $n = 1$ considered above.[4] So in this rather weak sense the longer repetition schemes are more powerful than the shorter ones. However, we would like to make the stronger assertion that, for a fixed BSC with a fixed raw error probability $p < \frac{1}{2}$, $P_e^{(2n+1)} \to 0$ as $n \to \infty$, that is, by means of these repetition schemes the channel can be made as reliable as desired. It is possible but not easy to do this by studying formula (0.2) for $P_e^{(2n+1)}$. We shall use another approach and invoke the *weak law of large numbers*,* which implies that, if N bits are transmitted over the channel, then for any $\varepsilon > 0$

$$\lim_{N\to\infty} P\left\{\left|\frac{\text{number of channel errors}}{N} - p\right| > \varepsilon\right\} = 0. \tag{0.3}$$

In other words, for large N, the fraction of bits received in error is unlikely to differ substantially from p. Thus we can make the following estimate of $P_e^{(2n+1)}$:

$$P_e^{(2n+1)} = P\left\{\text{fraction of transmitted bits received in error} \geqslant \frac{n+1}{2n+1} = \frac{1}{2} + \frac{1}{4n+2}\right\}$$

$$\leqslant P\left\{\text{fraction} > \tfrac{1}{2}\right\}$$

$$\leqslant P\left\{|\text{fraction} - p| > \tfrac{1}{2} - p\right\},$$

and so by (0.3) $P_e^{(2n+1)}$ does approach 0 as $n \to \infty$. We have thus reached the conclusion that if R is very small, it is possible to make the overall error probability very small as well, even though the channel itself is quite noisy. This is of course not particularly surprising.

* Discussed in Appendix A.

So much, temporarily, for rates less than 1. What about rates larger than 1? How accurately can we communicate under those circumstances?

If $R > 1$, we could, for example, merely transmit the fraction $1/R$ of the source bits and require the receiver to guess the rest of the bits, say by flipping an unbiased coin. For this not-very-bright scheme it is easy to calculate that the resulting bit error probability would be

$$\begin{aligned} P_e &= \frac{1}{R} \times p + \frac{R-1}{R} \times \frac{1}{2} \\ &= \tfrac{1}{2} - (\tfrac{1}{2} - p)/R. \end{aligned} \tag{0.4}$$

Another, less uninspired method which works for some values of $R > 1$ will be illustrated for $R = 3$. If $R = 3$ there is time to transmit only one third of the bits emitted by the source over the channel. So the sender divides the source bits into blocks of three and transmits only the majority-vote of the three. For example if the source emits 101110101000101, the sender will transmit 11101 over the channel. The receiver merely triples each received bit. In the present case if the channel garbled the second transmitted bit he would receive 10101, which he would expand to 111000111000111, thereby making five bit errors. In general, the resulting bit error probability turns out to be

$$\begin{aligned} P_e &= \tfrac{1}{4} \times (1 - p) + \tfrac{3}{4} \times p \\ &= \tfrac{1}{4} + p/2. \end{aligned} \tag{0.5}$$

Notice that this is less than $\frac{1}{3} + p/3$, which is what our primitive "coin-flipping" strategy gives for $R = 3$. The generalization of this strategy to other integral values of R is left as an exercise (see Prob. 0.4).

The schemes we have considered so far have been trivial, though perhaps not completely uninteresting. Let us now give an example which is much less trivial and in fact was unknown before 1948.

We assume now that $R = 4/7$, so that for every four bits emitted by the source there is just time to send three extra bits over the channel. We choose these extra bits very carefully: if the four source bits are denoted by x_0, x_1, x_2, x_3, then the extra or *redundant* or *parity-check* bits, labeled x_4, x_5, x_6, are determined by the equations

$$\begin{aligned} x_4 &\equiv x_1 + x_2 + x_3 \pmod 2, \\ x_5 &\equiv x_0 + x_2 + x_3 \pmod 2, \\ x_6 &\equiv x_0 + x_1 + x_3 \pmod 2. \end{aligned} \tag{0.6}$$

Thus, for example, if $(x_0, x_1, x_2, x_3) = (0110)$, then $(x_4, x_5, x_6) = (011)$, and the complete seven-bit *codeword* which would be sent over the channel is 0110011.

To describe how the receiver makes his estimate of the four source bits from a garbled seven-bit codeword, that is, to describe his *decoding algorithm*, let us rewrite the parity-check equations (0.6) in the following way:

$$\begin{aligned} x_1 + x_2 + x_3 + x_4 &= 0, \\ x_0 + x_2 + x_3 + x_5 &= 0, \\ x_0 + x_1 + x_3 + x_6 &= 0. \end{aligned} \tag{0.7}$$

(In (0.7) it is to be understood that the arithmetic is modulo 2.) Stated in a slightly different way, if the binary matrix H is defined by

$$H = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix},$$

we see that each of the 16 possible codewords $\mathbf{x} = (x_0, x_1, x_2, x_3, x_4, x_5, x_6)$ satisfies the matrix-vector equation

$$H\mathbf{x}^T = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{0.8}$$

(In (0.8) the superscript T means "transpose.")

It turns out to be fruitful to imagine that the BSC adds (mod 2) either a 0 or a 1 to each transmitted bit, 0 if the bit is not received in error and 1 if it is. Thus if $\mathbf{x} = (x_0, x_1, \ldots, x_6)$ is transmitted, the received vector is $\mathbf{y} = (x_0 + z_0, x_1 + z_1, \ldots, x_6 + z_6)$, where $z_i = 1$ if the channel caused an error in the ith coordinate and $z_i = 0$ if not. Thus, if $\mathbf{z} = (z_0, \ldots, z_6)$ denotes the *error pattern*, then $\mathbf{y} = \mathbf{x} + \mathbf{z}$.

The receiver, who knows only $\mathbf{y}$ but wants to know $\mathbf{x}$, now does a very clever thing: he computes the following vector $\mathbf{s} = (s_0, s_1, s_2)$:

$$\begin{aligned} \mathbf{s}^T &= H\mathbf{y}^T \\ &= H(\mathbf{x} + \mathbf{z})^T \\ &= H\mathbf{x}^T + H\mathbf{z}^T \\ &= H\mathbf{z}^T \qquad \text{(see (0.8))}. \end{aligned} \tag{0.9}$$

Here $\mathbf{s}$ is called the *syndrome*[5] of $\mathbf{y}$; a 0 component in the syndrome indicates

that the corresponding parity-check equation is satisfied by $\mathbf{y}$, a 1 indicates that it is not. According to (0.9), the syndrome does not depend on which codeword was sent, but only on the error pattern $\mathbf{z}$. However, since $\mathbf{x} = \mathbf{y} + \mathbf{z}$, if the receiver can find $\mathbf{z}$ he will know $\mathbf{x}$ as well, and so he focuses on the problem of finding $\mathbf{z}$. The equation $\mathbf{s}^T = \mathrm{H}\mathbf{z}^T$ shows that $\mathbf{s}^T$ is the (binary) sum of those columns of H corresponding to 1's in $\mathbf{z}$, that is, corresponding to the bits of the codeword that were garbled by the channel:

$$\mathbf{s}^T = z_0 \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} + z_1 \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} + \cdots + z_6 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{0.10}$$

The receiver's task, once he has computed $\mathbf{s}$, is to "solve" the equation $\mathbf{s}^T = H\mathbf{z}^T$ for $\mathbf{z}$. Unfortunately, this is only three equations in seven unknowns, and for any $\mathbf{s}$ there will always be 16 possibilities for $\mathbf{z}$. This is clearly progress, since there were a priori 128 possibilities for $\mathbf{z}$, but how can the receiver choose among the remaining 16? For example, suppose $\mathbf{y} = (0111001)$ was received. Then $\mathbf{s} = (101)$, and the 16 candidate $\mathbf{z}$'s turn out to be:

0	1	0	0	0	0	0		0	0	1	0	0	1	1
1	1	0	0	0	1	1		0	0	0	1	0	1	0
0	0	0	0	1	0	1		0	1	1	1	0	0	1
0	1	1	0	1	1	0		1	0	1	0	0	0	0
0	1	0	1	1	1	1		1	0	0	1	0	0	1
1	0	0	0	1	1	0		1	1	1	1	0	1	0
1	1	1	0	1	0	1		0	0	1	1	1	0	0
1	1	0	1	1	0	0		1	0	1	1	1	1	1

Faced with this set of possible error patterns, it is fairly obvious what to do: since the raw bit error probability p is $< \frac{1}{2}$, the fewer 1's (errors) in an error pattern, the more likely it is to have been the actual error pattern. In the current example, we're lucky: there is a unique error pattern (0100000) of least weight, the weight being the number of 1's. So in this case the receiver's best estimate of $\mathbf{z}$ (based both on the syndrome and on the channel statistics) is $\mathbf{z} = (0100000)$; the estimate of the transmitted codeword is $\mathbf{x} = \mathbf{y} + \mathbf{z} = (0011001)$; and finally, the estimate of the four source bits is (0011).

Of course we weren't really lucky in the above example, since we can show that for any syndrome $\mathbf{s}$ there will always be a unique solution to $H\mathbf{z}^T = \mathbf{s}^T$ of weight 0 or 1. To see this, notice that if $\mathbf{s} = (000)$, then $\mathbf{z} = (0000000)$ is the desired solution. But if $\mathbf{s} \neq (000)$, then $\mathbf{s}^T$ must occur as one of the columns