21 世纪高等院校信息与通信工程规划教材

信息论基础与编码

王军选　田小平　曹红梅　编著

人 民 邮 电 出 版 社

北 京

图书在版编目（CIP）数据

信息论基础与编码 / 王军选，田小平，曹红梅编著
. -- 北京 : 人民邮电出版社，2011.9
21世纪高等院校信息与通信工程规划教材
ISBN 978-7-115-25860-1

Ⅰ. ①信… Ⅱ. ①王… ②田… ③曹… Ⅲ. ①信息论
－高等学校－教材②信源编码－高等学校－教材 Ⅳ.
①TN911.2

中国版本图书馆CIP数据核字(2011)第154386号

内 容 提 要

本书系统地介绍了信息论基础与编码的主要内容，以及在无线通信系统中的主要应用。全书共 9 章，在介绍了有关信息度量的基础上，重点讨论了无失真信源编码、限失真信源编码、信道容量、信道编码和密码学的理论知识。全书从简单的理论入手，结合大量的例题描述信息论与编码的原理和应用，其中，原理的叙述力求突出概念和思路，尽量免去深奥的纯数学推导，与具体的应用相结合。在各章还附有相应的习题，便于学生加深理解。

本书可作为高等院校信息工程、通信工程以及电子信息类学生的教材，也可供低年级研究生或工程技术人员阅读参考。

21 世纪高等院校信息与通信工程规划教材

信息论基础与编码

◆ 编　　著　王军选　田小平　曹红梅
责任编辑　贾　楠

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号
邮编　100061　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京艺辉印刷有限公司印刷

◆ 开本：787×1092　1/16
印张：16.75　　2011 年 9 月第 1 版
字数：406 千字　　2011 年 9 月北京第 1 次印刷

ISBN 978-7-115-25860-1

定价：32.80 元

读者服务热线：(010)67170985　印装质量热线：(010)67129223
反盗版热线：(010)67171154
广告经营许可证：京崇工商广字第 0021 号

前　言

“信息论基础与编码”是通信工程、电子信息工程类专业的基础课程。信息论与编码在光通信、无线通信领域中应用广泛。本书系统地介绍了信息论的信源编码、信道编码、密码编码、信道容量等方面的知识和内容。

信息论基础和编码的理论性内容较多，目前相关的教材其内容都有不同的偏重。根据读者不同的需求，信息论基础和编码也可以从不同的层次上去学习和理解：理论分析、技术层次和工程应用。目前全国几乎所有的工科、综合类高校都开设了通信工程或电子信息类专业，由于学生基础、培养侧重点以及办学层次的差异，对信息论基础与编码的学习要求各不相同。另一方面，对于工程技术人员而言，工作性质不尽相同，对信息论基础与编码知识要求的深度与广度必然也是不同的。正是这种需求的多样性，产生了对教材多层次、多样性的要求，这也是编写这本书的原因。

很多有关信息论基础、信息编码、通信原理、数字系统等教材中均有一些信息论以及编码部分的内容，这些内容对于大多数非专业研究人员来说显得过于深奥。因此，本书从简单的理论入手，结合大量的例题，从工程角度来描述信息论与编码的原理与应用，其中，原理的叙述力求突出概念和思路，尽量免去深奥的纯数学推导；设计与应用则尽量具体化，采用实例分析。

本书适合作为高等学校电子信息类高年级学生的教材，也可供低年级研究生和工程技术人员阅读参考。

本书编写分工如下：王军选编写第 1 章、第 3 章、第 6 章、第 7 章、第 8 章，田小平编写第 2 章、第 4 章、第 5 章，曹红梅编写第 9 章。张普等参加了书中部分内容的录入。全书由王军选统稿。在本书编写过程中，得到了很多同事的帮助和鼓励，在此表示感谢。

限于作者的水平，书中难免有不妥和谬误之处，敬请读者指正。

作　者

2011 年 5 月

目　录

第 1 章 概 论

内容简介

本章主要介绍信息论的基本概念、研究对象和内容，以及信息论的发展等。通过本章的学习可以掌握信息论的基本概念，为后面的学习打下坚实的基础。

1.1 信息的基本概念

研究信息论，首先要明白什么是信息，本节的内容主要是几个常用的关于信息的定义以及信息的基本性质。

1.1.1 信息的定义

现代社会已经进入一个新的时代——信息时代。人类的社会生活离不开信息，社会实践活动不仅需要对周围世界的情况能做出正确的反应，而且还要与周围的人群进行沟通。因此，人类不仅时刻需要从外界获取信息，而且还要和其他人交流信息。

什么是信息？至今无确切定义，但它是一种人人皆知的抽象概念，是一种不言自明的概念。信息在日常生活中常被认为是“消息”、“情报”、“信号”等，然而信息和它们既有联系也有明显的区别。消息一般是形式，没有具体的数学含义；情报的含义较窄，不像信息那么广泛；信号则是通信系统中消息的载体。

就狭义而言，在通信中对信息的表达分为 3 个层次：信号、消息、信息。

信号：是信息的物理表达层，是 3 个层次中最具体的层次。它是一个电参量、物理量，是一个载荷信息的运载工具，可测量、可描述，如正弦信号、脉冲信号等。

消息：（或称为符号），是信息的数学表达层，是信息的载体，它虽不是一个物理量，但是可以定量地加以描述，它是具体物理信号的进一步数学抽象，可将具体物理信号抽象为两大类型。

① 离散（数字）消息，是一组未知量，可用随机序列来描述：$\boldsymbol{U}=(U_1\cdots U_l\cdots U_L)$。

② 连续（模拟）消息，也是未知量，它可用随机过程来描述：$U(t,\omega)$。

信息：它是更高层次上的抽象，是信号与消息的更高表达层次。

3 个层次中，信号最具体，信息最抽象。它们三者之间的关系是哲学上的内涵与外延的关系。

这就是说，信息可以认为是具体的物理信号、数学描述的消息的内涵。

而信号则是抽象信息在物理层表达的外延；消息则是抽象信息在数学层表达的外延。同一信息，可以采用不同的信号形式（如文字、语言、图像等）来载荷；同一信息，也可以采用不同的数学表达形式（如离散或连续）来定量描述。同样，同一信号形式，如“0”与“1”，可以表达不同形式的信息，比如无与有、断与通、低与高（电平）等。

在信息的具体定义上，从不同的侧面、角度、层次，有不同的定义。1928 年，哈特莱（R. V. L Hartley）提出，“发信者所发出的信息，就是他在通信符号表中选择符号的具体方式”，该定义只考虑通信符号的具体选择方式，没有涉及信息的价值和具体内容，也没有考虑各种不同选择方式的概率统计特性；维纳（N. Wiener）也曾指出，“信息是信息，不是物质，也不是能量”，将“信息”上升到最基本的概念的位置，后来他又指出“信息是人们在适应外部世界和控制外部世界的过程中，同外部世界进行交换的内容的名称”；1948 年，香农（C.E.Shannon）从研究通信系统传输的实质出发，对信息做了科学的定义，并进行了定性和定量的描述：**信息是事物运动状态或存在方式的不确定性的描述**。

1.1.2 信息的性质

信息虽然没有一个明确的定义，但是却具有两个明显的特征：广泛性与抽象性。

广泛性可从以下 3 个方面来理解：①信息的客观性，动物、大树、沙粒、水滴甚至大海等所有客观存在的事物都具有自己的信息；②人类的生存离不开对信息的处理，从原始社会的狩猎、耕种等生活开始，人类不停地感知、接收信息，人的神经系统在不停地传递信息，人的大脑则在不停地处理与决策信息，人与人之间又在不停地交流信息，人活在世上的百分之百时间都在自觉与不自觉地与信息打交道；③信息的积累，信息可以通过书本、技能等知识进行积累，人类依靠知识改造自然、适应自然，靠知识促进社会的发展与进步。

信息的抽象性体现在信息是组成客观世界并促进社会发展的最基本的要素之一。一般来讲，物质世界主要由物质、能量、信息等要素构成，其中，物质是基础；能量是物质运动的形式，是改造客观世界的主要动力；信息依附于物质和能量，但又不同于物质和能量。没有信息就不能更好地利用物质和能量，人类利用信息和知识改造物质，创造新物质，提高能量利用效率。

根据上面的叙述，信息应该具有如下性质：

① 信息是可以识别的；
② 信息的载体是可以转换的；
③ 信息是可以存储的；
④ 信息是可以传递的；
⑤ 信息是可以加工的；
⑥ 信息是可以共享的；
⑦ 信息是可以测量的。

1.2 信息论研究的对象和内容

信息论经过几十年的发展，已经有系统的理论和体系，研究的领域包括通信、电子、控制、管理等行业。特别是在通信领域，受益于信息论的指导，各种通信新技术层出不穷，带

来了人类社会的巨大变化。

1.2.1 信息论研究的对象

信息论是关于信息的本质和传输规律的科学理论。信息论应用概率论、随机过程、数理统计以及近世代数、矩阵理论等方法来研究信息的计量、发送、传递、交换、接收和储存的一般规律。下面就以图 1-1 所示的通信系统为例，说明香农信息论研究的对象。

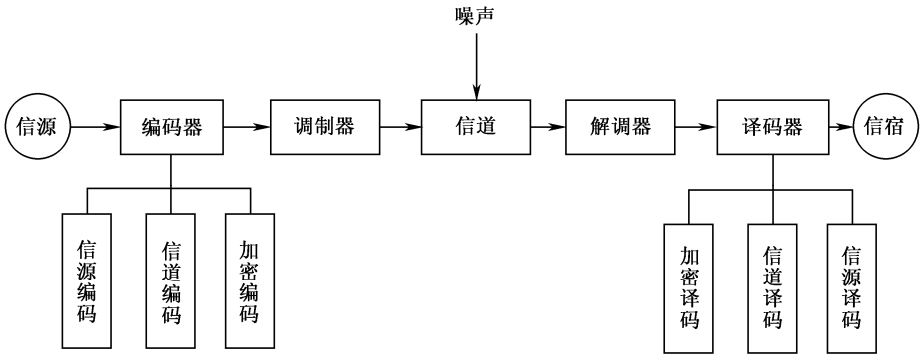


图 1-1 通信系统框图

在图 1-1 所示的通信系统中，形式上传输的是消息，但实质上传输的是信息。消息只是表达信息的载体。因此，在通信中被利用的（亦即携带信息的）载体是不重要的，而重要的是信息。通信系统主要分成 5 个部分。**信源**是产生消息和消息序列的来源，消息可以是离散的，也可以是连续的（数据、文字、语言、图像），通常信源的消息序列是随机发生的，因此要用随机变量来描述。**编码器**包括信源编码器、信道编码器和密码编码器，主要是提高通信系统的有效性、可靠性和安全性。**调制器**是将编码器输出的数字序列变换为振幅、频率或相位受到调制控制的信号，以适合在信道中进行较长距离的传输。**信道**是信号由发送端传输到接收端的媒介，常用的信道包括明线、电缆、高频无线信道、微波通道、光纤通道等，甚至包括磁盘等存储介质。**噪声**（干扰）是对传输信道或存储媒介构成影响的来源的总称，包括热噪声、电磁干扰等。噪声和干扰往往具有随机性，所以信道的特征也可以用概率空间来描述，根据统计特性，分为两类：①加性干扰（噪声），它是由外界原因产生的随机干扰，它与信道中传送的信号的统计特性无关，因而信道的输出是输入和干扰的叠加；②乘性干扰，信道的输出信号可看成是输入信号和一个时变参量相乘的结果。**解调器**是从载波中提取信号，是调制的逆过程，将调制的信号转换为携带信息的消息序列。**译码器**包括进行加密译码、信道译码以及信源译码，是编码器的逆过程，将消息序列转换成适合接收者接收的信息形式。**信宿**是信息的接受者。

通信的目的就是消除或部分消除不确定性，从而获得信息。信息论就是通过对系统中消息的传输和处理的研究来找出信息传输和处理的共同规律，具体到通信系统，就是研究通信系统的有效性、安全性、可靠性等环节。

1.2.2 信息论的基本定义

一般概率空间用$[X, P]$来表示。在离散情况下，X的样本空间可写成$[a_1, a_2, \cdots, a_q]$，样本空间的大小为 q。样本空间中选择任意元素 a_i 的概率表示为 $p(a_i)$。在离散情况下，概率空间

可描述为

$$\begin{bmatrix} X \\ p(x) \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & \cdots & a_q \\ p(a_1) & p(a_2) & \cdots & p(a_q) \end{bmatrix} \tag{1-1}$$

其中，$p(a_i)$就是选择符号 a_i 作为消息的概率，称为**先验概率**，并且满足$\sum_{i=1}^{q} p(a_i) = 1$。在接收端，是否选择这个消息（符号）$a_i$的不确定性是与 a_i的先验概率成反比的，即对 a_i的不确定性可表示为先验概率 $p(a_i)$的倒数的某一函数。

自信息定义：对概率空间 X，其消息（符号）a_i的自信息定义为

$$I(a_i) = \log \frac{1}{p(a_i)} = -\log p(a_i) \tag{1-2}$$

在接收端，收到的消息集合为 Y。由于信道中存在干扰，假设接收端收到的消息（符号）为 b_j（$j = 1,2,\cdots,r$），其中 r 可能与 q 相同，也可能不同，于是把条件概率 $p(a_i \mid b_j)$ 称为**后验概率**，它表示接收端收到消息（符号）b_j后，发送端发的是 a_i的概率。接收端收到 b_j后，发送端发送的是否是 a_i 尚存在的不确定性应是后验概率的函数，即 $\log \frac{1}{p(a_i \mid b_j)}$。于是，收信者在收到消息（符号）$b_j$ 后，已经消除的不确定性为：先验的不确定性减去尚存在的不确定性，这就是收信者**获得的信息量**。

互信息定义：对概率空间 X 和 Y，X 的消息（符号）a_i 和 Y 的消息（符号）b_j之间的互信息定义为

$$I(a_i;b_j) = \log \frac{1}{p(a_i)} - \log \frac{1}{p(a_i \mid b_j)} \tag{1-3}$$

当对数的底取 2 时，$I(a_i)$的单位为比特（bit）；当对数的底取 e 时，$I(a_i)$的单位为奈特（nat）；当对数的底取 10 时，$I(a_i)$的单位为哈特（hart）。

1.2.3 信息论研究的内容

目前，对信息论的研究内容一般有 3 种理解，如图 1-2 所示。

狭义信息论（又称香农信息论）：主要通过数学描述与定量分析，研究通信系统从信源到信宿的全过程，包括信息的测度、信道容量、信源和信道编码理论等问题，强调通过编码和译码使收、发两端联合最优化，并且以定理的形式证明极限的存在，这部分内容是信息论的基础理论。狭义信息论是以存在性研究为主体，又称它为数学信息论。

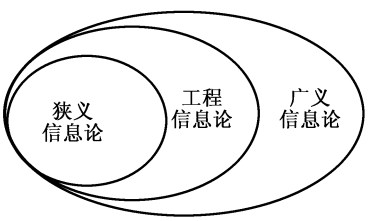


图 1-2 信息论的 3 种形式

工程信息论（又称一般信息论、通信理论）：主要是研究信息传输和处理问题，除了香农理论外，还包括噪声理论、信号滤波和预测、统计检测和估计理论、调制理论、信息处理理论等，比如信源编、译码理论及其设计构造方法，信道编、译码理论及其设计构造方法，最佳调制与解调理论与实现，最佳检测、估值与最佳接收理论与实现等。

广义信息论：广义信息论不仅包括上述两方面的内容，而且包括所有与信息有关的领域，

如模式识别、计算机翻译、心理学、遗传学、语言学等。

1.3 信息论的发展

自从 19 世纪 20～30 年代法拉第发现电磁感应以来，人类社会就开始了无线通信的探索。1832 年莫尔斯建立了电报系统，1895 马可尼发明了无线通信，1922 年卡松提出边带理论，指明信号在调制（编码）与传送过程中与频谱宽度的关系，1922 年哈特莱发表《信息传输》的文章，首先提出消息是代码、符号而不是信息内容本身，使信息与消息区分开来，并提出用消息可能数目的对数来度量消息中所含有的信息量，为信息论的创立提供了思路。美国统计学家费希尔从古典统计理论角度研究了信息理论，苏联数学家哥尔莫戈洛夫也对信息论做过研究。控制论创始人维纳建立了维纳滤波理论和信号预测理论，也提出了信息量的统计数学公式。这些研究均为信息论的建立打下了坚实的基础。

1948 年贝尔研究所的香农在题为《通信的数学理论》的论文中系统地提出了关于信息的论述，创立了较为系统的信息论学科。维纳提出的关于度量信息量的数学公式开辟了信息论的广泛应用前景。1951 年信息论学科获得美国无线电工程学会承认，为以后的迅速发展打下基础。20 世纪 50～60 年代是信息论发展的关键时期，信源编码和信道编码得到了巨大的发展，产生了大量的研究成果，比如霍夫曼编码、费诺编码以及信道编码中的卷积码、LDPC 编码等，同时，该时期也是信息论向其他各门学科渗透的时期。到 20 世纪 70 年代，由于数字计算机的广泛应用，通信系统的能力也有了很大提高。信息的概念和方法已广泛渗透到各个科学领域，它迫切要求突破香农信息论（狭义信息论）的范围，以便使它能成为人类各种活动中所碰到的信息问题的基础理论，从而推动其他许多新兴学科进一步发展。20 世纪 90 年代以来，Turbo 码编码理论、多天线理论等技术的提出，使得无线通信迅速发展，带来了人类社会的巨大变化。

信息论从诞生到今天已有 60 多年了，现已成为一门独立的学科。香农理论的思想、方法，甚至某些结论已渗透到统计数学、计算机科学、物理学、哲学、科学方法论等其他学科中。

同时，信息论还和其他学科结合产生了许多交叉学科，比如在经济、生物等方面已产生了“信息经济学”、“信息生物学”等边缘学科；在和量子力学理论结合后，产生了量子信息论、量子编码理论、量子计算理论等。

第2章 信源与信息熵

内容简介

信源是输出信息的源，信源产生的信息是可以度量的。本章介绍了如何来描述信源的数学模型、信源的分类、互信息、信息熵的定义以及性质等；同时介绍了离散信源、马尔可夫信源以及连续信源的信息熵及性质。

2.1 信源的数学模型及分类

作为 Shannon 信息论研究的对象——信息，被假设为由一系列的随机变量所代表。它们往往用随机出现的符号来表示，我们称输出这些符号集的源为信息源，或者说信息源是发出消息的源，简称信源。

由信号源输出的随机符号，如果其取值于某一连续区间，则称此信息源为连续信源，如语言、图像、图形等；如果其取值于某一离散集合，则称此信息源为离散信源，如文字、数字、数据等符号。

离散信源可进一步分类如下。

① 离散无记忆信源，包括发出单个符号的无记忆信源和发出符号序列的无记忆信源两种。

发出单个符号的无记忆信源每次只发出一个符号代表一个消息，而发出符号序列的无记忆信源每次发出一组含两个以上符号的符号序列代表一个消息。

离散无记忆信源所发出的各个符号是相互独立的，发出的符号序列中的各个符号之间没有统计关联性。

② 离散有记忆信源，包括发出符号序列的有记忆信源和发出符号序列的马尔可夫信源两种。

和离散无记忆信源不同，离散有记忆信源所发出的各个符号的概率是有关联的，这种概率关联性可用两种方式来表示：一种是用信源发出的一个符号序列的联合概率来反映有记忆信源的特征，即发出符号序列的有记忆信源；一般情况下，当记忆长度很长甚至无限长时，表述有记忆信源要比无记忆信源困难得多，实际问题中，往往限制记忆长度，也就是说，某一符号出现的概率只与前面一个或有限个符号有关，与更前面的符号无关，这类信源可以用信源发出符号序列内各个符号之间的条件概率来反映有记忆信源的特征，即发出符号序列的

马尔可夫信源。

假定某一离散信源发出的各个符号消息的集合为

$$X=\{x_1,x_2,\cdots,x_n\}$$

各个符号的先验概率为

$$P=\{p(x_1),p(x_2),\cdots,p(x_n)\}$$

通常将它们写在一起

$$\begin{bmatrix}X\\P\end{bmatrix}=\begin{bmatrix}x_1 & x_2 & \cdots & x_n\\ p(x_1) & p(x_2) & \cdots & p(x_n)\end{bmatrix} \tag{2-1}$$

称为概率空间，其中 $p(x_i)\geqslant 0$，$\sum_{i=1}^{n}p(x_i)=1$。

最简单的有记忆信源是 N=2 的情况，此时信源为：$X=X_1X_2$，其概率空间为

$$\begin{bmatrix}X\\P\end{bmatrix}=\begin{bmatrix}x_1x_1 & x_1x_2 & \cdots & x_nx_n\\ p(x_1x_1) & p(x_1x_2) & \cdots & p(x_nx_n)\end{bmatrix} \tag{2-2}$$

对于无记忆信源来说，其联合概率为

$$p(x_1x_2\cdots x_n)=p(x_1)p(x_2)\cdots p(x_n)$$

若其满足平稳性

$$p(x_1x_2\cdots x_n)=p(x_1)p(x_2)\cdots p(x_n)=p^n$$

2.2　离散信源熵和互信息

离散信源的自信息和互信息是理解信息论的基础，本节主要介绍离散信源的信息熵以及互信息的概念和性质，并通过例子分析信源熵的计算。

2.2.1　信息量

在一切有意义的通信中，虽然消息的传递意味着信息的传递，但对于接收者来说，某些消息比另外一些消息却含有更多的信息。例如，若一方告诉另一方一件非常可能发生的事件“今年冬天的气候要比去年冬天的更冷一些”，比起告诉另一方一件很不可能发生的事件“今年冬天的气候将与去年夏天的一样热”来说，前一消息包含的信息量显然要比后者少些。因为前一事件很可能发生，不足为奇，但后一事件却极难发生，听后使人惊奇！这表明：消息确实有量值的意义。

可以看出：对接收者来说，事件越不可能发生，越是使人感到意外和惊奇，信息量就越大；事件越是有可能发生，信息量就越小。

概率论告诉我们，事件的不确定程度，可用其出现的概率来描述。事件出现的可能性愈小，则其概率就越小；反之，则其概率就越大。

由此我们可以得到：消息中的信息量与消息发生的概率紧密相关，消息出现的概率越小，则消息中包含的信息量就越大。

如果消息是必然的（概率为 1），则它传递的信息量应为 0。如果事件是不可能的（概率

为 0），则它将有无穷的信息量。

如果我们得到的不是由一个事件构成，而是由若干个独立事件构成的消息，那么，这时我们得到的总的信息量就是若干个独立事件的信息量的总和。

综上可以看出，为了计算信息量，消息中所含信息量 I 与消息出现的概率 $p(x)$间的关系式应当反映如下规律。

① 消息中所含信息量 I 是出现该消息的概率 $p(x)$的函数，即

$$I = I[p(x)] \tag{2-3}$$

② 消息出现的概率越小，它所含的信息量越大；消息出现的概率越大，它所含的信息量越小；且当 $p(x)=1$ 时，$I=0$。

③ 若干个互相独立的事件构成的消息，其所含信息量等于各独立事件的信息量之和，即

$$I[p(x_1)p(x_2)\cdots] = I[p(x_1)] + I[p(x_2)] + \cdots \tag{2-4}$$

不难看出，若 I 与 $p(x)$之间的关系式为

$$I = \log_a \frac{1}{p(x)} = -\log_a p(x) \tag{2-5}$$

就可满足上述要求。

上式中的 I 称为随机事件的自信息量。它的单位与所采用的对数的底有关。若取 $a=2$，则自信息量的单位为 bit；若取 $a=e$，则其单位为 nat；若取 $a=10$，则其单位为 det。

对于一个以等概率出现的二进制码元(0, 1)，它所包含的信息量为

$$I(0) = I(1) = -\log_2 \frac{1}{2}\text{bit} = \log_2 2\text{bit} = 1\text{bit}$$

若有一个 m 位的二进制数，其自信息量为

$$I = -\log_2 \frac{1}{2^m}\text{bit} = \log_2 2^m \text{bit} = m\ \text{bit}$$

即需要 m bit 的信息来指明这样的二进制数。

若有两个消息 x_i，y_j 同时出现，则其自信息量定义为

$$I(x_i y_j) = -\log_a p(x_i y_j)$$

$p(x_iy_j)$为联合概率。

若 x_i 与 y_j 相互独立，即 $p(x_i y_j) = p(x_i)p(y_j)$，则有

$$I(x_i y_j) = I(x_i) + I(y_j)$$

若 x_i 与 y_j 的出现不是相互独立的，而是有联系的，此时，要用条件概率 $p(x_i \mid y_j)$ 来表示，即在事件 y_j 出现的条件下，事件 x_i 发生的条件概率，其条件自信息量定义为

$$I(x_i \mid y_j) = -\log_a p(x_i \mid y_j)$$

例题 2-1 英文字母中“a”出现的概率为 0.063，“c”出现的概率为 0.023，“e”出现的概率为 0.105，分别计算它们的自信息量。

解 由自信息量的定义式（2-5），有

$$I(a) = -\log_2 0.063\text{bit} = 3.96\text{bit}$$

$$I(c) = -\log_2 0.023\text{bit} = 5.44\text{bit}$$

$$I(e) = -\log_2 0.105\text{bit} = 3.25\text{bit}$$

例题 2-2　将二信息分别编码为 A 和 B 进行传送，在接收端，A 被误收作 B 的概率为 0.02；而 B 被误收作 A 的概率为 0.01，A 与 B 传送的频繁程度为 2∶1。若接收端收到的是 A，计算原发信息是 A 的条件自信息量。

解　设 U_0 表示发送 A，U_1 表示发送 B；V_0 表示接收 A，V_1 表示接收 B。

由题意知：$p(U_0)=\frac{2}{3}$，$p(U_1)=\frac{1}{3}$，$p(V_1|U_0)=0.02$，$p(V_0|U_0)=0.98$，$p(V_0|U_1)=0.01$，$p(V_1|U_1)=0.99$。则接收到 A 时，原发信息是 A 的条件概率为

$$\begin{aligned} p(U_0|V_0) &= \frac{p(U_0V_0)}{p(V_0)} = \frac{p(V_0|U_0)p(U_0)}{p(V_0)} \\ &= \frac{p(V_0|U_0)p(U_0)}{p(V_0|U_0)p(U_0)+p(V_0|U_1)p(U_1)} \\ &= \frac{0.98\times\frac{2}{3}}{0.98\times\frac{2}{3}+0.01\times\frac{1}{3}} \\ &= \frac{196}{197} \end{aligned}$$

相应的条件自信息量为

$$I[p(U_0|V_0)] = -\log_2\frac{196}{197} = 0.0734\text{ bit}$$

2.2.2　离散信源熵

假设离散信息源是一个由 n 个符号组成的集合，称为符号集。符号集中每一个符号 x_i 在消息中是按一定的概率 $p(x_i)$独立出现，其概率空间为

$$\begin{bmatrix} X \\ P \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ p(x_1) & p(x_2) & \cdots & p(x_n) \end{bmatrix}$$

且有 $\sum_{i=1}^{n} p(x_i)=1$，则 $x_1, x_2, \cdots, x_n$ 所包含的信息量分别为

$$-\log_2 p(x_1), -\log_2 p(x_2), \cdots, -\log_2 p(x_n)$$

于是，每个符号所含信息量的统计平均值，即平均信息量为

$$\begin{aligned} H(X) &= p(x_1)[-\log_2 p(x_1)] + p(x_2)[-\log_2 p(x_2)] + \cdots + p(x_n)[-\log_2 p(x_n)] \\ &= -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i) \end{aligned}$$

由于 H 同热力学中的熵形式相似，故通常又称它为信息源的熵，简称信源熵，其单位为 bit/符号。

前面定义的自信息量 $I(x_i)$是表征信源中各个符号的不确定性。由于信源中各个符号的概率分布（先验概率）不同，因而各个符号的自信息量就不相同，所以，自信息量不能作为信

源总体的信息量。而平均信息量 $H(X)$或信源熵 $H(X)$是从平均意义上来表征信源的总体特征，因此可以表征信源的平均不确定性。

定义信源的平均不确定性 $H(X)$为信源中各个符号不确定性的数学期望，即

$$H(X)=E[I(x)]=\sum_i p(x_i)I(x_i)=-\sum_i p(x_i)\log_2 p(x_i) \tag{2-6}$$

称作信息熵，简称熵。

由于 $I(X)$是非负值的量，所以熵 $H(X)$也是非负值的，只有在 $p(x)=0$ 和 $p(x)=1$ 时，熵 $H(X)$才为零[$p(x)=0$ 时，规定 $0\log 0=0$，其合理性由极限 $\lim\limits_{x\to 0^+} x\log x=0$ 得证]。

例题 2-3 一幅 500×600 的图像，每个像素的灰度等级为 10，若为均匀分布，计算平均每幅图像能提供的信息量。

解 能够组成的图像有 $n=10^{3\times 10^5}$ 个，有

$$H(X)=-\sum_{i=1}^{n} p(x_i)\log_2 p(x_i)=-\log_2 10^{-(3\times 10^5)}=9.96\times 10^5 \text{ （bit/图像）}$$

例题 2-4 某信息源由 4 个符号 0,1,2,3 组成，它们出现的概率分别为：1/8、1/2、1/4、1/8，且每个符号的出现都是相互独立的。试计算某条消息“2010201302130012032101003210100231020020103120321001202 10”的信息量。

解 在此条消息中，符号 0 出现了 23 次，符号 1 出现了 14 次，符号 2 出现了 13 次，符号 3 出现了 7 次，消息共有 57 个符号。

其中出现符号 0 的信息量为：$23\log_2 8=69$bit。

其中出现符号 1 的信息量为：$14\log_2 2=14$bit。

其中出现符号 2 的信息量为：$13\log_2 4=26$bit。

其中出现符号 3 的信息量为：$7\log_2 8=21$bit。

因此，该消息的信息量为：I=69+14+26+21=130bit。平均（算术平均）每个符号的信息量应为：$\bar{I}=\dfrac{I}{57}=2.28\text{bit/符号}$。

若用信源熵的概念进行计算，有

$$H=\left(-\frac{1}{8}\log_2\frac{1}{8}\log_2-\frac{1}{2}\log_2\frac{1}{2}-\frac{1}{4}\log_2\frac{1}{4}-\frac{1}{8}\log_2\frac{1}{8}\right)\text{bit/符号}=1.75\text{bit/符号}$$

那么，该条消息所含的信息量为：$I=57\times 2.28\text{bit}\approx 129.96\text{bit}$。

可以看到，用算术平均和信源熵两种计算所得的结果略有差别，其根本原因在于它们平均处理的方法不同。按算术平均的方法，结果可能存在误差，这种误差将随着消息中符号数的增加而减小。

例题 2-5 一个由字母 A、B、C、D 组成的字，对于传输的每一个字母用二进制脉冲编码，A:00，B:01，C:10，D:11，每个脉冲的宽度为 5ms。

（1）若每个字母是等概率出现时，计算传输的平均信息速率。

（2）若每个字母出现的概率分别为：$p_A=0.2$，$p_B=0.25$，$p_C=0.25$，$p_D=0.3$，计算传输的平均信息速率。

解 信息传输速率也称为信息速率或传信率，它被定义为每秒传递的信息量，单位是比特/秒，记为 bit/s，它是衡量通信系统的一个主要性能指标。

（1）各字母等概率出现时，平均每个符号所含的信息量为

$$H(X) = -4 \times \frac{1}{4}\log_2 \frac{1}{4} = 2\text{bit}/\text{符号}$$

传输一个符号需要 10ms，因此，每秒钟可传 100 个符号。所以，信息速率为

$$R_b = 2 \times 100 = 200\text{bit}/\text{s}$$

（2）同理，各字母不等概率出现时，平均每个符号所含的信息量为

$$H(X) = -\sum_{i=1}^{4} p_i \log_2 p_i = 1.985\text{bit}/\text{符号}$$

信息速率为：$R_b = 198.5\text{bit}/\text{s}$。

对于二元离散信源 X，其输出的符号只有 0 和 1 两个，发生的概率分别为 p 和 q，且 $p+q=1$，该信源的概率空间为

$$\begin{bmatrix} X \\ P \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ p & q \end{bmatrix}$$

该二元信源的熵为

$$\begin{aligned} H(X) &= -p\log p - q\log q \\ &= -p\log p - (1-p)\log(1-p) \\ &= H(p) \end{aligned}$$

由于 p 的取值区间为 $0 \leqslant p \leqslant 1$，给定 p 的值，做出 $H(p) \sim p$ 的变化曲线，如图 2-1 所示。

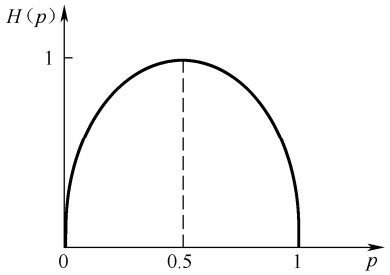


图 2-1 $H(p) \sim p$ 曲线

从图 2-1 可以得出，如果二元信源的输出符号是确定的，即 $p = 1$ 或 $q = 1$ 时，则 $H(p) = 0$，说明该信源不提供任何信息量。若二元信源符号 0 和 1 以等概率发生，即 $p=q=0.5$ 时，则该信源熵达到最大值，$H(X)$=1bit/符号。

对任一有限离散值的集合而言，其熵是一有界函数，即

$$0 \leqslant H(X) \leqslant \log M \qquad (2\text{-}7)$$

M 是可能的离散取值的个数。

熵 $H(X)$是随机变量 x 的不确定性的测度，如果增加一个随机变量 y，那么如何度量在观测到 y 之后 x 的不确定性呢？这需要定义给定 y 情况下 x 的条件熵。

在给定 y_j 条件下，x_i 的条件自信息量为 $I(x_i \mid y_j)$，X 集合的条件信息熵为

$$H(X \mid y_j) = \sum_i p(x_i \mid y_j) I(x_i \mid y_j)$$

进一步，在给定各个 y_j，即 Y 的条件下，X 集合的条件信息熵为

$$\begin{aligned} H(X|Y) &= \sum_j p(y_j) H(X|y_j) \\ &= \sum_{i,j} p(y_j) p(x_i|y_j) I(x_i|y_j) \\ &= \sum_{i,j} p(x_i y_j) I(x_i|y_j) \end{aligned}$$

条件熵 $H(X/Y)$表示已知 Y 后，X 的不确定性。

相应地，在给定 X 的条件下，Y 集合的条件信息熵定义为

$$H(Y|X) = \sum_{i,j} p(x_i y_j) I(y_j|x_i)$$

定义联合信息熵为

$$\begin{aligned} H(XY) &= \sum_{i,j} p(x_i y_j) I(x_i y_j) \\ &= -\sum_{i,j} p(x_i y_j) \log p(x_i y_j) \end{aligned}$$

$H(XY)$表示了联合概率分布所具有的信息量的概率平均值，即 X 和 Y 同时发生的不确定性。联合信息熵与各条件信息熵的关系式为

$$\begin{aligned} H(XY) &= H(X) + H(Y|X) \\ &= H(Y) + H(X|Y) \end{aligned} \tag{2-8}$$

2.2.3 互信息量

假设 X 为信源发出的离散消息符号的集合，X 的各消息符号的先验概率已知为 $p(x_i)$；Y 为信宿收到的离散消息符号的集合。信宿收到一个消息符号 y_j 后计算出信源各消息符号的条件概率为 $p(x_i|y_j)$（即后验概率），则互信息量定义为

$$I(x_i;y_j) = \log \frac{p(x_i|y_j)}{p(x_i)} \tag{2-9}$$

由于无法确定 $p(x_i|y_j)$ 和 $p(x_i)$的大小关系，所以 $I(x_i;y_j)$ 不一定大于或等于 0，有可能小于 0。

对互信息量 $I(x_i;y_j)$ 在 X 集合上取统计平均值，得

$$\begin{aligned} I(X;y_j) &= \sum_i p(x_i|y_j) I(x_i;y_j) \\ &= \sum_i p(x_i|y_j) \log \frac{p(x_i|y_j)}{p(x_i)} \end{aligned}$$

再对 $I(X;y_j)$ 在 Y 集合上的概率加权统计平均，有

$$\begin{aligned} I(X;Y) &= \sum_j p(y_j) I(X;y_j) \\ &= \sum_{i,j} p(y_j) p(x_i|y_j) \log \frac{p(x_i|y_j)}{p(x_i)} \\ &= \sum_{i,j} p(x_i y_j) \log \frac{p(x_i|y_j)}{p(x_i)} \end{aligned} \tag{2-10}$$

此 $I(X;Y)$称为平均互信息量（熵）。

在通信系统中，若发送端的消息符号为 X，接收端的消息符号为 Y，则平均互信息量 $I(X;Y)$就是在接收端收到消息符号 Y 后所能获得的关于发送的消息符号 X 的平均信息量，反之亦然。且 $I(X;Y)$是非负的。注意平均互信息量 $I(X;Y)$与联合信息熵 $H(XY)$的区别。

由以上的结果，可得出下面的性质

$$\begin{aligned} I(X;Y) &= H(X) - H(X \mid Y) \\ &= H(Y) - H(Y \mid X) \\ &= I(Y;X) \\ &= H(X) + H(Y) - H(XY) \end{aligned} \tag{2-11}$$

这几个关系式说明：平均互信息量对 X 与 Y 呈对称性；联合信息熵与平均互信息量之和等于 X 与 Y 的熵之和。这些关系可以用图来表示，如图 2-2 所示。两个圆分别为 $H(X)$和 $H(Y)$，交叠的部分为 $I(X;Y)$。

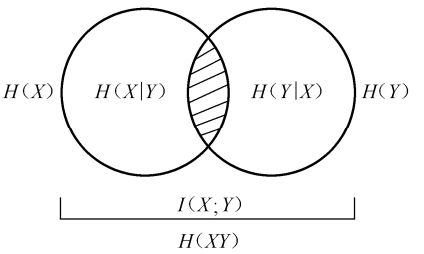


图 2-2　联合熵与条件熵的关系

关系式 $I(X;Y)=H(X)-H(X|Y)$说明了接收信号所提供的有关传输消息的平均信息量等于确定消息 X 所需的平均信息量减去接收 Y 信号后要达到此目的尚需要的平均信息量。

可以把熵当作发出的平均信息量，把 $I(X;Y)$作为接收 Y 后所提供的有关发出消息 X 的平均信息量，把条件熵 $H(X|Y)$看做是由于信道噪声而损失的平均信息量。条件熵 $H(X|Y)$也称为可疑度或疑义度。

而关系式 $I(X;Y)=H(Y)-H(Y|X)$也可看做所接收的平均信息量，它是确定接收到信号需要的平均信息量减去当发出的消息为已知时，欲确定同一接收到信号所需的平均信息量。

因此，$H(Y|X)$是确定信道中出现某种干扰时需要的平均信息量，称为信道噪声的熵；或者说，$H(Y|X)$代表 Y 集合中由于噪声所产生的那部分熵。噪声熵也称为散布度。

如果 X 与 Y 是相互独立的，则 Y 已知时，X 的条件概率就等于无条件概率，那么 X 的条件熵就等于 X 的无条件熵。此时，$I(X;Y)=0$，说明由于 X 与 Y 相互独立，无法从 Y 中提取有关 X 的信息量，可以看做信道中的噪声相当大，能传输的平均信息量为零，由信源发出的信息量在信道上全部损失掉了，这种信道称为全损离散信道。

如果输出 Y 是输入 X 的确定的、一一对应的函数，则 Y 已知时 X 的条件概率不是“1”就是“0”，此时：$I(X;Y)=H(X)$，由 Y 所获得的信息量就是 X 的不确定性或熵。这种情况下，信道不损失信息量，噪声熵为 0。这种信道称为无扰离散信道，有

$$I(X;Y)=H(X)=H(Y) \tag{2-12}$$

这是两种特殊的情况。

一般情况下，X 与 Y 既不是相互独立、也不是一一对应的，此时 $0 \leqslant I(X;Y) \leqslant H(X)$，即从 Y 获得 X 的信息量介于 0 与 $H(X)$之间。

符号 x_i 与符号对 $y_j z_k$ 之间的互信息量定义为

$$I(x_i; y_j z_k) = \log \frac{p(x_i \mid y_j z_k)}{p(x_i)}$$

在给定 z_k 条件下，x_i 与 y_j 之间的互信息量定义为条件互信息量，即

$$I(x_i;\ y_j \mid z_k)=\log\frac{p(x_i \mid y_j z_k)}{p(x_i \mid z_k)}$$

由以上两式，可以得出

$$I(x_i;y_j z_k)=I(x_i;z_k)+I(x_i;y_j \mid z_k)$$

该式表明：一个联合事件 $y_j z_k$ 出现后所提供的有关 x_i 的信息量 $I(x_i;y_j z_k)$ 等于 z_k 事件出现后提供的有关 x_i 的信息量 $I(x_i;z_k)$ 加上在给定 z_k 条件下再出现 y_j 事件后所提供的有关 x_i 的信息量 $I(x_i;y_j \mid z_k)$。可以容易地得出三维联合集 XYZ 上的平均互信息量有下列各关系式

$$\begin{aligned}
&I(X;YZ)=I(X;Y)+I(X;Z \mid Y)\\
&I(YZ;X)=I(Y;X)+I(Z;X \mid Y)\\
&I(X;YZ)=I(X;ZY)=I(X;Z)+I(X;Y \mid Z)
\end{aligned}\qquad (2\text{-}13)$$

例题 2-6 有 8 个符号 $u_1,u_2,\cdots,u_8$，它们出现的概率分别为 $\frac{1}{4},\frac{1}{4},\frac{1}{8},\frac{1}{8},\frac{1}{16},\frac{1}{16},\frac{1}{16},\frac{1}{16}$。将其用三位二进制数进行编码，即

$$u_1:000,u_2:001,u_3:010,u_4:011,u_5:100,u_6:101,u_7:110,u_8:111$$

计算编码器输出端出现“0”、“01”、“001”后所提供的有关 u_4 出现的信息量。

解 本题需要求解的是互信息量 $I(u;xyz)$。用 x_0 表示出现“0”，y_1 表示出现“1”，z_1 表示出现“1”。因此，需确定 $I(u_4;x_0)$，$I(u_4;x_0y_1)$，$I(u_4;x_0y_1z_1)$。

编码器输出“0”以后，可能的输入消息就从 8 种缩小到 4 种。从 x_0 的出现来推测输入为 u_1、u_2、u_3、u_4 的概率分别为

$$p(u_1 \mid x_0)=p(u_2 \mid x_0)=\frac{\frac{1}{4}\times\frac{1}{4}}{2\times\frac{1}{4}\times\frac{1}{4}+2\times\frac{1}{4}\times\frac{1}{8}}=\frac{1}{3}$$

$$p(u_3 \mid x_0)=p(u_4 \mid x_0)=\frac{\frac{1}{4}\times\frac{1}{8}}{2\times\frac{1}{4}\times\frac{1}{4}+2\times\frac{1}{4}\times\frac{1}{8}}=\frac{1}{6}$$

$$p(u_5 \mid x_0)=p(u_6 \mid x_0)=p(u_7 \mid x_0)=p(u_8 \mid x_0)=0$$

编码器输出“01”以后，可能的输入消息只有 2 种：u_3 和 u_4，相应的条件概率为

$$p(u_3 \mid x_0y_1)=p(u_4 \mid x_0y_1)=\frac{1}{2},\ \text{其余为0}$$

编码器输出“011”以后，可能的输入消息只有 1 种，即 u_4，相应的条件概率为

$$p(u_4 \mid x_0y_1z_1)=1,\ \text{其余为0}$$

由互信息量的定义，有

$$I(u_4;x_0)=\log_2\frac{p(u_4 \mid x_0)}{p(u_4)}=\log_2\frac{1\mid 6}{1\mid 8}=0.415\text{bit}$$

$$I(u_4;x_0y_1)=\log_2\frac{p(u_4 \mid x_0y_1)}{p(u_4)}=\log_2\frac{1\mid 2}{1\mid 8}=2\text{bit}$$

$$I(u_4;x_0y_1z_1)=\log_2\frac{p(u_4\mid x_0y_1z_1)}{p(u_4)}=\log_2\frac{1}{1\mid 8}=3\text{bit}$$

实际上，因为 $x_0y_1z_1$ (011)就代表 u_4，因此，当此码字出现后所能提供的有关 u_4 的信息量必须等于 u_4 的自信息量，即

$$I(u_4;x_0y_1z_1)=I(u_4)=3\text{bit}$$

2.2.4　数据处理中信息的变化

假定对输入消息的集合 X 进行两级处理，如图 2-3 所示。

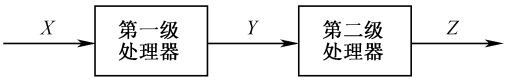


图 2-3　信息的多级处理

第一级处理器的输出消息集合为 Y，第二级处理器的输出消息集合为 Z，若假设在 Y 条件下 X 与 Z 相互独立，那么由式（2-13），可以得出以下各关系式

$$\begin{aligned}
&I(X;Z)=I(X;Y)+I(X;Z\mid Y)-I(X;Y\mid Z)\\
&I(XY;Z)=I(X;Z)+I(Y;Z\mid X)\\
&I(XY;Z)=I(Y;Z)+I(X;Z\mid Y)\\
&I(X;Z)=I(Y;Z)+I(X;Z\mid Y)-I(Y;Z\mid X)
\end{aligned}\qquad(2\text{-}14)$$

由前面的假设，在 Y 条件下 X 与 Z 相互独立，有

$$I(X;Z|Y)=0$$

所以可得

$$I(X;Z)\leqslant I(Y;Z)$$
$$I(X;Z)\leqslant I(X;Y)$$

此不等式说明，随着处理器数目的增多，输入消息与输出消息之间的平均互信息量趋于变小，即在数据处理过程中会失掉一些信息量，但决不会增加信息量。任何数据处理过程都会失掉信息量，最多保持原来的信息量。一旦失掉了信息量，用任何处理手段，都不能再恢复出失掉的信息量。

2.3　信息熵的性质

信源的信息熵有很多重要的性质，其中常用的有如下几个性质。

2.3.1　非负性

信源熵是自信息量的数学期望，而自信息量是非负的，因此信源熵也是非负的，即

$$H(X)=H(x_1,x_2,\cdots,x_n)\geqslant 0$$

2.3.2　确定性

$$H(0,1)=H(1,0,0,\cdots,0)=0$$

即只要信源符号中，有一个符号出现的概率为 1，则信源熵就为零。

2.3.3 对称性

对称性是指熵函数的所有变元可以互换，而函数值保持不变。

$$H(X)=H(x_1,x_2,\cdots,x_n)=H(X)=H(x_2,x_1,\cdots,x_n)$$

例如，有 3 个离散信源

$$\begin{bmatrix}X\\P\end{bmatrix}=\begin{bmatrix}x_1 & x_2 & x_3\\ \frac{1}{3} & \frac{1}{2} & \frac{1}{6}\end{bmatrix}\qquad \begin{bmatrix}Y\\P\end{bmatrix}=\begin{bmatrix}y_1 & y_2 & y_3\\ \frac{1}{2} & \frac{1}{6} & \frac{1}{3}\end{bmatrix}\qquad \begin{bmatrix}Z\\P\end{bmatrix}=\begin{bmatrix}z_1 & z_2 & z_3\\ \frac{1}{2} & \frac{1}{3} & \frac{1}{6}\end{bmatrix}$$

都具有相同的信源熵值。

2.3.4 可加性

$$H(XY)=H(X)+H(Y\mid X)$$
$$H(XY)=H(Y)+H(X\mid Y)$$

因为信息熵具有可加性，可以证明其形式是唯一的，而不会有其他的形式。

2.3.5 极值性

对任意 n 维概率矢量 $P=(p_1,p_2,\cdots,p_n)$ 和 $Q=(q_1,q_2,\cdots,q_n)$，有

$$H(p_1,p_2,\cdots,p_n)=-\sum_{i=1}^{n}p_i\log p_i\leqslant-\sum_{i=1}^{n}p_i\log q_i$$

该定理表明，对任意概率分布 p_i，它对其他概率分布 q_i 的自信息量（$-\log q_i$）取数学期望时，必不小于 p_i 本身的熵，只有 P=Q 时，等号才成立。

2.3.6 最大熵定理

对于由 M 个信息符号组成的离散无记忆信源来说，仅当各个符号出现的概率相等时，其信源熵达到最大，即

$$H(X)\leqslant H\left(\frac{1}{M},\frac{1}{M},\cdots,\frac{1}{M}\right)=\log M$$

2.3.7 条件熵小于无条件熵

$$H(Y\mid X)\leqslant H(Y)$$

当 y 与 x 相互独立时，$p(y|x)$=$p(y)$，上式取等号。

两个条件下的条件熵小于一个条件下的条件熵

$$H(Z\mid XY)\leqslant H(Z\mid Y)$$

联合熵小于信源熵之和

$$H(XY)\leqslant H(X)+H(Y)$$

当两个集合相互独立时，联合熵取最大值

$$H_{\max}(XY)=H(X)+H(Y)$$

例题 2-7　二进制通信系统用符号 0 和 1 传递信息。由于存在失真，传输时会产生误码。用 x_0 表示一个“0”发出，x_1 表示一个“1”发出，y_0 表示一个“0”收到，x_1 表示一个“1”收到。已知概率 $p(x_0)=0.5$，$p(y_0 \mid x_0)=0.75$，$p(y_0 \mid x_1)=0.5$，求解如下问题。

（1）已知发出一个“0”，收到符号后得到的信息量。

（2）已知发出的符号，收到符号后得到的信息量。

（3）知道发出的和收到的符号，能够得到的信息量。

（4）已知收到的符号，被告知发出的符号得到的信息量。

解　（1）由已知条件，可得到

$$p(y_1 \mid x_0)=1-p(y_0 \mid x_0)=1-0.75=0.25$$

因此，有

$$\begin{aligned}H(Y \mid x_0)&=-\sum_{j=0}^{1}p(y_j \mid x_0)\log_2 p(y_j \mid x_0)\\&=-p(y_0 \mid x_0)\log_2 p(y_0 \mid x_0)-p(y_1 \mid x_0)\log_2 p(y_1 \mid x_0)\\&=(-0.75\log_2 0.75-0.25\log_2 0.25)\ \text{bit/符号}\\&=0.82\text{bit}/\text{符号}\end{aligned}$$

（2）先计算联合概率

$$\begin{aligned}p(x_0y_0)&=p(y \mid x_0)p(x_0)=0.75\times 0.5=0.375\\p(x_0y_1)&=p(y_1 \mid x_0)p(x_0)=0.25\times 0.5=0.125\\p(x_1y_0)&=p(y_0 \mid x_1)p(x_1)=0.5\times 0.5=0.25\\p(x_1y_1)&=p(y_1 \mid x_1)p(x_1)=0.5\times 0.5=0.25\end{aligned}$$

因此，有

$$\begin{aligned}H(Y \mid X)&=-\sum_{i=0}^{1}\sum_{j=0}^{1}p(x_iy_j)\log_2(y_j \mid x_i)\\&=(-0.375\log_2 0.375-0.125\log_2 0.125-0.25\log_2 0.25-0.25\log_2 0.25)\ \text{bit/符号}\\&=0.91\text{bit}/\text{符号}\end{aligned}$$

（3）求联合熵

$$\begin{aligned}H(XY)&=-\sum_{i=0}^{1}\sum_{j=0}^{1}p(x_iy_j)\log_2(x_iy_j)\\&=1.91\text{bit}/\text{符号}\end{aligned}$$

（4）求 $H(X \mid Y)$

先计算输出符号的概率分布

$$\begin{aligned}p(y_0)&=\sum_{i=0}^{1}p(x_iy_0)=0.625\\p(y_0)&=1-p(y_1)=0.375\end{aligned}$$

可得

$$H(Y)=-\sum_{j=0}^{1}p(y_j)\log_2 p(y_j)=0.96\text{bit/符号}$$

所以，有

$$H(X|Y)=H(XY)-H(Y)=(1.91-0.96)\text{bit/符号}=0.95\text{bit/符号}$$

2.4 离散序列信源熵

一般情况下，实际信源的输出往往是空间上或时间上的离散随机序列，包括无记忆的离散信源序列和序列中各符号之间存在相关性的有记忆离散信源序列。

2.4.1 离散无记忆信源的序列熵

假定信源输出一长度为 L 的随机序列

$$\boldsymbol{X}=(X_1X_2\cdots X_l\cdots X_L)$$

该序列中的变量 $X_l=\{x_1,x_2,\cdots,x_n\}$，其中 $l=1,2,\cdots,L$，每个随机变量有 n 种可能的取值。此随机序列 $\boldsymbol{X}=x_i$ 的概率为

$$\begin{aligned}p(\boldsymbol{X}=x_i)&=p(X_1=x_{i1},X_2=x_{i2},\cdots,X_L=x_{iL})\\&=p(x_{i1})p(x_{i2}|x_{i1})p(x_{i3}|x_{i1}x_{i2})\cdots p(x_{iL}|x_{i1}x_{i2}\cdots x_{iL})\\&=p(x_{i1})p(x_{i2}|x_{i1})p(x_{i3}|x_{i1}^2)\cdots p(x_{iL}|x_{i1}^{L-1})\end{aligned}$$

上式中：$x_{i1}^{L-1}=x_{i1}x_{i2}\cdots x_{i(L-1)}$。

若信源为无记忆信源，即序列中各符号的取值相互独立时，有

$$p(x_i)=p(x_{i1},x_{i2},\cdots,x_{iL})=\prod_{l=1}^{L}p(x_{il})$$

信源熵为

$$\begin{aligned}H(\boldsymbol{X})&=-\sum_i p(x_i)\log p(x_i)\\&=-\sum_i\prod_{l=1}^{L}p(x_{il})\log\prod_{l=1}^{L}p(x_{il})\\&=\sum_{l=1}^{L}H(X_l)\end{aligned}$$

若与序号 l 无关，即该信源满足平稳性时，有

$$p(x_{i1})=p(x_{i2})=\cdots=p(x_{iL})=p$$

则 $p(x_i)=p^L$，那么信源熵为

$$H(\boldsymbol{X})=LH(X)$$

$H(X)$为单个符号信源的熵，则 L 长的信源序列平均每个符号的熵为

$$H_L(\boldsymbol{X})=\frac{1}{L}H(\boldsymbol{X})=H(X)\qquad(2\text{-}15)$$

即离散无记忆信源平均每个符号的符号熵为单个符号信源的符号熵。

比如：一无记忆信源 $X\in\{0,1\}$，$p(x=0)=p(x=1)=0.5$，若以单个符号出现为一事件，则信源熵为 $H(X)$=1bit/符号；若以两个符号出现为一事件，即 L=2，则信源为 $X\in\{00,01,10,11\}$，

信源序列熵为 $H(\boldsymbol{X})=\log_2 4=2\text{bit}/\text{符号}$，信源符号熵为 $H_2(\boldsymbol{X})=\frac{1}{2}H(\boldsymbol{X})=1\text{bit}/\text{符号}$。

2.4.2　离散有记忆信源的序列熵

对于有记忆信源来说，情况要复杂一些，只能在某些特殊情况下才能得到一些有价值的结论，并且要引入条件熵的概念。

假定有一两个符号组成的联合信源，有以下结论。

1．$H(X_1X_2)=H(X_1)+H(X_2\mid X_1)=H(X_2)+H(X_1\mid X_2)$

说明前后两个符号 X_1X_2 同时发生的不确定性等于信源发出前一个符号 X_1 的信息熵与前一个符号 X_1 已知时信源发出第 2 个符号 X_2 的条件熵。

2．$H(X_1)\geqslant H(X_1\mid X_2)$，$H(X_2)\geqslant H(X_2\mid X_1)$

若前后两个符号无依存关系时，有

$$H(X_1X_2)=H(X_1)+H(X_2)$$

$$H(X_1)=H(X_1\mid X_2)，\quad H(X_1)=H(X_1\mid X_2)$$

一般地，有记忆的信源如文字、数据等，它们输出的不是单个或两个符号，而是有限个符号组成的信源序列，这些输出符号之间存在着相互依存关系。

若信源输出一个 L 长的序列，那么该序列的信源熵为

$$\begin{aligned}H(\boldsymbol{X})&=H(X_1X_2\cdots X_L)\\&=H(X_1)+H(X_2\mid X_1)+\cdots+H(X_L\mid X_1X_2\cdots X_{L-1})\end{aligned}$$

记为

$$H(\boldsymbol{X})=H(X^L)=\sum_{l=1}^{L}H(X_l\mid X^{l-1})$$

那么，平均每个符号的熵为

$$H_L(\boldsymbol{X})=\frac{1}{L}H(\boldsymbol{X})$$

当信源退化为无记忆时

$$H(\boldsymbol{X})=\sum_{l=1}^{L}H(X_l)$$

若进一步又满足平稳性时

$$H(\boldsymbol{X})=LH(X)$$

可见，无记忆信源是有记忆信源的一个特例。

例题 2-8　已知某一离散有记忆信源中各符号的概率空间为

$$\begin{bmatrix}X\\P\end{bmatrix}=\begin{bmatrix}x_0 & x_1 & x_2\\ \frac{1}{36} & \frac{4}{9} & \frac{1}{4}\end{bmatrix}$$

若信源发出两重符号序列消息 (x_i,x_j)，已知两个符号的关联性用条件概率 $p(x_j\mid x_i)$ 表示，分别为

$$p(x_0 \mid x_0)=\frac{9}{11},\quad p(x_1 \mid x_0)=\frac{2}{11},\quad p(x_2 \mid x_0)=0$$

$$p(x_0 \mid x_1)=\frac{1}{8},\quad p(x_1 \mid x_1)=\frac{3}{4},\quad p(x_2 \mid x_1)=\frac{1}{8}$$

$$p(x_0 \mid x_2)=0,\quad p(x_1 \mid x_2)=\frac{2}{9},\quad p(x_2 \mid x_2)=\frac{7}{9}$$

求该信源的序列熵和平均符号熵。

解 先求条件熵

$$\begin{aligned}H(X_2X_1)&=-\sum_{i=0}^{2}\sum_{j=0}^{2}p(x_ix_j)\log_2 p(x_j \mid x_i)\\&=-\sum_{i=0}^{2}\sum_{j=0}^{2}p(x_i)p(x_j \mid x_i)\log_2 p(x_j \mid x_i)\\&=0.872\text{bit/符号}\end{aligned}$$

再求单个符号信源熵

$$H_1(X)=H(X_1)=-\sum_{i=0}^{2}p(x_i)\log_2 p(x_i)=1.543\text{bit}/\text{符号}$$

信源发出二重符号序列的熵为

$$H(\boldsymbol{X})=H(X_1X_2)=H(X_1)+H(X_2 \mid X_1)=0.872+1.543=2.415\text{bit}/\text{序列}$$

平均单个符号的熵为

$$H_2(\boldsymbol{X})=\frac{1}{2}H(\mathbf{X})=\frac{1}{2}H(X^2)=1.21\text{bit}/\text{符号}$$

比较一下，可见 $H_2(\boldsymbol{X})<H_1(\boldsymbol{X})$，即不确定性减小了，这是由于符号间存在相关性所造成的。

下面我们讨论平稳的离散信源。如果离散信源为平稳信源，那么信源发出各符号的联合概率具有时间推移不变性。

$$\begin{aligned}&p\{X_{i1}=x_1,X_{i2}=x_2,\cdots,X_{iL}=x_L\}\\&=p\{X_{i1+h}=x_1,X_{i2+h}=x_2,\cdots,X_{iL+h}=x_L\}\end{aligned}$$

即联合概率分布与时间起点无关，有下述结论。

结论 1：$H(X_L \mid X^{L-1})$ 是长度 L 的单调非增函数。

$$\begin{aligned}&H(X_L \mid X_1X_2\cdots X_{L-1})\leqslant H(X_L \mid X_2X_3\cdots X_{L-1})\\&=H(X_{L-1} \mid X_1X_2\cdots X_{L-2})\text{（由平稳性）}\\&\leqslant H(X_{L-1} \mid X_2X_3\cdots X_{L-2})\\&=H(X_{L-2} \mid X_1X_2\cdots X_{L-3})\cdots\leqslant H(X_2 \mid X_1)\end{aligned}$$

结论 2：$H_L(\boldsymbol{X})\geqslant H(X_L \mid X^{L-1})$。

结论 3：$H_L(\boldsymbol{X})$ 是 L 的单调非增函数，$H_L(\boldsymbol{X})\leqslant H_{L-1}(\boldsymbol{X})$，即平均符号熵随着序列长度 L 的增大而减小。

结论 4：当 $L\to\infty$ 时，有

$$H_\infty(\boldsymbol{X}) = \lim_{L\to\infty} H_L(\boldsymbol{X}) = \lim_{L\to\infty} H(X_L \mid X_1X_2\cdots X_{L-1})$$

$H_\infty(\boldsymbol{X})$ 称为平稳离散有记忆信源的极限熵，也叫做极限信息量。

由结论 3 可以得到

$$H_0(X) \geqslant H_1(X) \geqslant H_2(\boldsymbol{X}) \geqslant \cdots \geqslant H_\infty(\boldsymbol{X}) \tag{2-16}$$

式（2-16）中：$H_0(X)$ 为等概率无记忆信源单个符号的熵；$H_1(X)$ 为不等概率无记忆信源单个符号的熵；$H_2(\boldsymbol{X})$ 为两个符号组成的信源序列的平均单个符号熵；……

如果由结论 4 给出的 $H_\infty(\boldsymbol{X})$ 的计算公式直接进行计算，要得到该极限信息量是非常困难的。在实际应用中，对一般的离散平稳信源来说，常取有限的 L 值的条件熵 $H(X_L \mid X^{L-1})$ 作为 $H_\infty(\boldsymbol{X})$ 的近似值，而该近似值与实际值非常接近。

2.4.3　马尔可夫信源及其极限熵

如果平稳的离散有记忆信源满足马尔可夫过程的性质，即信源在 t 时刻发出的符号只与前面的 m 个符号（$t-1, t-2, \cdots, t-m$ 各时刻的符号）有关，而与更前面的出现的符号（t–m–1 时刻以前）无关，其概率意义可表示为

$$p(x_t \mid x_{t-1}, x_{t-2}, \cdots, x_{t-m}, x_{t-m-1}, \cdots) = p(x_t \mid x_{t-1}, x_{t-2}, \cdots, x_{t-m})$$

这样的平稳离散有记忆信源称为 m 阶马尔可夫信源，其极限熵为

$$\begin{aligned} H_\infty(\boldsymbol{X}) &= \lim_{L\to\infty} H(X_L \mid X_1X_2\cdots X_{L-1}) \\ &= H(X_{m+1} \mid X_1X_2\cdots X_m) \\ &= H_{m+1}(\boldsymbol{X}) \end{aligned} \tag{2-17}$$

假定 m 阶马尔可夫信源为 $X = (x_1, x_2, \cdots, x_q)$，其概率特性为 $p(x_{im+1} \mid x_{i1}, x_{i2}, \cdots, x_{im})$，即在 ($m$+1)时刻，信源符号出现的概率仅与前面已出现的 m 个符号有关，与更前面的符号无关。该信源的符号集中共有 q 个符号，由前面出现的 m 个符号组成的信源序列 $s_i = (x_{i1}, x_{i2}, \cdots, x_{im})$ 总共有 q^m 有种组合，其中 $i_1, i_2, \cdots, i_m \in (1,2,\cdots,q)$。令 $Q = q^m$，可以将这些序列看做是状态集 $S = \{s_1, s_2, \cdots, s_Q\}$ 中的一个，s 称为信源的状态。

那么，信源在某一时刻出现符号 x_j 的概率就与信源此时所处的状态 s_i 有关，可用条件概率表示为

$$p(x_j \mid s_i)，\ i=1,2,\cdots,Q，\ j=1,2,\cdots,q$$

信源符号 x_j 出现后，信源所处的状态将发生变化，信源由状态 s_i 转入一个新的状态 s_j，这种状态的转移用转移概率表示为

$$p_{ij}(m,n) = p(S_n = s_j \mid S_m = s_i) = p(s_j \mid s_i)，\ s_i, s_j \in S$$

状态转移概率 $p_{ij}(m,n)$表示已知在时刻 m 信源处于状态 s_i 的条件下，经过$(n-m)$步后转移到 s_j 的概率。状态转移概率 $p_{ij}(m,n)$具有下列性质。

① $p_{ij}(m,n) \geqslant 0$，$i, j \in S$。

② $\sum_j p_{ij}(m,n) = 1$，$i, j \in S$。

当 $n-m=1$ 时， $p_{ij}(m,m+1)$ 称为一步转移概率或称为基本转移概率，记为

$$p_{ij}(m)=p(S_{m+1}=s_j \mid S_m=s_i)=p_{ij}$$

当 $n-m=k>1$ 时， $p_{ij}(m,m+k)$ 称为时刻 m 的 k 步转移概率，记为

$$p_{ij}^{(k)}(m)=p(S_{m+k}=s_j \mid S_m=s_i)$$

同样，k 步转移概率 $p_{ij}^{(k)}(m)$ 也具有下列性质。

① $p_{ij}^{(k)}(m)\geqslant 0$，$k>1$，$i,j\in S$。

② $\sum p_{ij}^{(k)}(m)=1$，$k>1$，$i,j\in S$。

若 k 步转移概率 $p_{ij}^{(k)}(m)$ 只与 i、j 和 k 有关，而和时刻 m 无关，称这种马尔可夫链为齐次马尔可夫链。

对于一步转移概率 p_{ij} 来说，它总共有 Q^2 个取值，将这所有的 Q^2 个一步转移概率排列成一个矩阵，记为

$$P(m)=\{p_{ij}(m),i,j\in S\}$$

将其展开，有

$$P=\begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1Q} \\ p_{21} & p_{22} & \cdots & p_{2Q} \\ \vdots & \vdots & \ddots & \vdots \\ p_{Q1} & p_{Q2} & \cdots & p_{QQ} \end{bmatrix}$$

该矩阵中的每一个元素都是非负的。第 i 行的元素对应于从某一状态 s_i 经过一步转移到所有状态 s_j 的转移概率，每一行的元素之和为 1。矩阵中第 j 列的元素对应于从所有状态 s_i 经过一步转移到同一个状态 s_j 的转移概率，每一列的元素之和不一定为 1。

那么，k 步转移概率 $p_{ij}^{(k)}(m)$ 与一步转移概率 p_{ij} 之间有没有关系呢？若有，是一种什么样的关系？有一个非常重要的公式，称为 Chapman-Kolmogorov 方程，该方程给出了 k 步转移概率 $p_{ij}^{(k)}(m)$ 与一步转移概率 p_{ij} 之间的关系。

$$p_{ij}^{(k+l)}(m)=\sum_{r\in S} p_{ir}^{(k)}(m)p_{rj}^{(l)}(m+k) \tag{2-18}$$

该方程的直观意义可解释为：在时刻 m，系统处于状态 i，经 $k+l$ 步转移到状态 j，可以从状态 i 经 k 步到达某一状态 r，再由状态 r 经 l 步到达状态 j，而 r 跑遍全状态空间 S。该转移过程如图 2-4 所示。

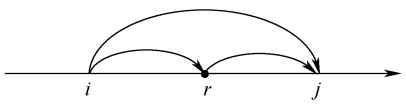


图 2-4 状态转移过程

当 $k=1$ 或 $l=1$ 时，可得

$$p_{ij}^{(l+1)}(m)=\sum_{r\in S}p_{ir}(m)p_{rj}^{(l)}(m+1)$$
$$=\sum_{r\in S}p_{ir}^{(l)}(m)p_{rj}(m+l)$$

当 $k=l=1$ 时，有

$$p_{ij}^{(2)}(m)=\sum_{r\in S}p_{ir}(m)p_{rj}(m+1)$$

表示成矩阵的乘法形式，有

$$P^{(2)}(m)=P(m)P(m+1)$$

用归纳法可得，对任一 $k>1$，k 步转移概率矩阵为

$$P^{(k)}(m)=P(m)P(m+1)\cdots P(m+k-1)$$

对于齐次马尔可夫链来说，转移概率与起始时刻 m 无关，可得

$$P^{(k)}=PP^{(k-1)}=PPP^{(k-2)}=\cdots=P^k \tag{2-19}$$

此即齐次马尔可夫链的一步转移概率与 k 步转移概率之间的关系，说明 k 步转移概率矩阵等于一步转移概率矩阵的 k 次方。

对于马尔可夫链所代表的一个系统进行考察，要讨论两类问题：①瞬态分析，讨论在某一固定时刻 n 系统的概率特性，即求出 k 步转移概率 $p_{ij}^{(k)}$；②稳态分析，讨论当 $k\to\infty$ 时，系统的概率特性，系统是否稳定？当 $k\to\infty$ 时，$p_{ij}^{(k)}$ 的极限是否存在？若存在，是否和状态 i 有关？

若极限 $\lim\limits_{k\to\infty}p_{ij}^{(k)}$ 存在，等于一个与起始状态 i 无关的平稳分布 $W_j=p(S_k=s_j)$，则无论起始状态是什么，该马尔可夫链最后到达稳定，即

$$\lim_{k\to\infty}p_{ij}^{(k)}=W_j \tag{2-20}$$

这个稳定的概率分布 W_j 是计算这种信源的熵所必需的。但有的时候直接计算是非常困难的，实际上只要知道这个极限是存在的，那么可用下面的方程组来求出 W_j，即

$$\sum_i W_i p_{ij}=W_j \tag{2-21}$$

并且 $\sum\limits_i p_{ij}=1$，$\sum\limits_j W_j=1$，W_i、W_j 是稳态分布概率。方程组（2-20）式有唯一解是极限 $\lim\limits_{k\to\infty}p_{ij}^{(k)}$ 存在的必要条件，而不是充分条件。为了使马尔可夫链最后达到稳定，成为遍历的马尔可夫链，还须具备两个性质：不可约性和非周期性。

不可约性：对任意一对 i 和 j，都存在至少一个 k，使 $p_{ij}^{(k)}>0$，即从 i 开始，总能到达 j。说明在该信源的状态 S 中，所有的状态是互通的，这种马尔可夫链就是不可约的马尔可夫链。

非周期性：在所有 $p_{ii}^{(k)}>0$ 的 k 中没有比 1 大的公因子，否则，这种马尔可夫链就具有周期性。

将系统的各种状态用方向（箭头）进行连接，箭头所指方向表示一个状态转移到另一个状态，在箭头的周围注明状态转移概率，这种马尔可夫状态图称为香农线图，如图 2-5 所示。

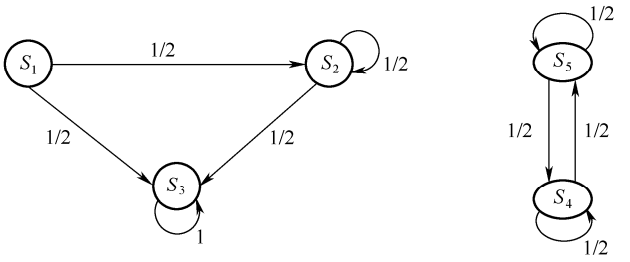


图 2-5 信源的状态转移图

例题 2-9 有一个相对码编码器，输入的码 $X_r(r=1,2,\cdots)$ 是相互独立的，取值为 0 或 1。已知 $p(X=0)=p$，$p(X=1)=q=1-p$，输出的码是 Y_r，编码器原理图如图 2.6 所示。讨论输入、输出码之间的关系并画出其状态转移图。

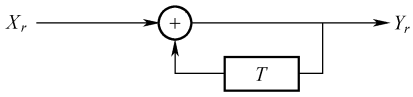


图 2-6 相对码编码器原理

解 从图 2-6 可得

$$Y_1 = X_1\,,\quad Y_2 = X_2 \oplus Y_1\,,\quad \cdots$$

$\oplus$ 表示模 2 加法运算。可以看到，Y_r 是一个马尔可夫链，因为 Y_{r+1} 的概率分布只与 Y_r 有关，而与 Y_{r-1}，Y_{r-2}，…无关。

Y_r 序列的条件概率为

$$p_{00} = p(Y_2 = 0 \mid Y_1 = 0) = p(X = 0) = p$$
$$p_{01} = p(Y_2 = 1 \mid Y_1 = 0) = p(X = 1) = q$$
$$p_{10} = p(Y_2 = 0 \mid Y_1 = 1) = p(X = 1) = q$$
$$p_{11} = p(Y_2 = 1 \mid Y_1 = 1) = p(X = 0) = p$$

转移概率为

$$\begin{bmatrix} p & q \\ q & p \end{bmatrix}$$

与 r 无关，因而属于齐次马尔可夫链，其状态转移图如图 2-7 所示。

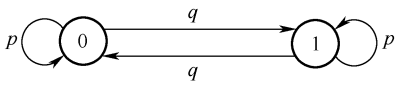


图 2-7 状态转移图

从状态图可以看出，该马尔可夫链具有不可约性和非周期性。其稳态概率分布由下列方程组给出。

① $\sum_i W_i p_{ij} = W_j$

② $\sum_j W_i = 1$

③ $\sum_i p_{ij} = 1$

求出：$W_0 = \frac{1}{2}$，$W_1 = \frac{1}{2}$。该马尔可夫链是遍历的。

遍历性是指不论质点从哪一个状态 S_i 出发，当转移步数 k 足够大时，转移到状态 S_j 的概率 $p_{ij}^{(k)}$ 都近似等于某一个常数 W_j。这说明马尔可夫信源在初始时刻可以处在任意状态，而信源状态之间可以转移。经过足够长的时间之后，信源处于什么状态已与初始状态无关。这时，每一种状态出现的概率已达到一种稳定的分布。

例题 2-10　有一 2 阶马尔可夫链 $X \in [0,1]$，其一步转移概率如表 2-1 所示。

表 2-1　　**不同初始状态下输出符号的概率**

初始状态	符号 0	符号 1
00	1/2	1/2
01	1/3	2/3
10	1/4	3/4
11	1/5	4/5

状态变量 S=(00,01,10,11)。

若在状态 01 时，出现符号 0，状态变为 010，将前面的 0 挤出变为 10，即状态转移到 10，转移概率为 $\frac{1}{3}$，用同样的方法可得其他任一状态的转移概率。

起始状态	终止状态			
	S_1(00)	S_2(01)	S_3(10)	S_4(11)
00	1/2	1/2	0	0
01	0	0	1/3	2/3
10	1/4	3/4	0	0
11	0	0	1/5	4/5

求各状态的平稳分布。利用 $\sum_i W_i p_{ij} = W_j$，可得方程组如下。

$$W_1 = \frac{1}{2}W_1 + \frac{1}{4}W_3, \quad W_2 = \frac{1}{2}W_1 + \frac{3}{4}W_3$$

$$W_3 = \frac{1}{3}W_2 + \frac{1}{5}W_4, \quad W_4 = \frac{2}{3}W_2 + \frac{4}{5}W_4$$

$$W_1 + W_2 + W_3 + W_4 = 1$$

解出平稳分布的概率分别为：$W_1 = \frac{3}{35}$，$W_2 = \frac{6}{35}$，$W_3 = \frac{6}{35}$，$W_4 = \frac{4}{7}$。其状态转移图如图 2-8 所示。

通过本例看到，状态转移概率与符号条件概率是一样的。大多数情况下，两者都是一样

的，而有些情况就不一样了。

例如，某三状态的马尔可夫信源的状态转移图如图 2-9 所示。

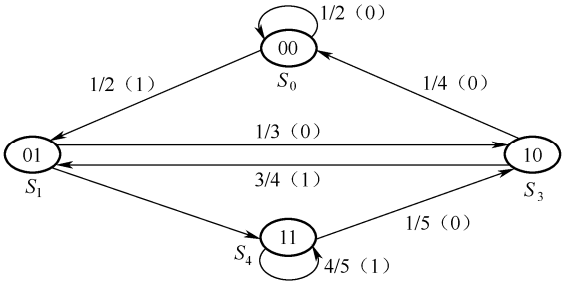


图 2-8　状态转移图

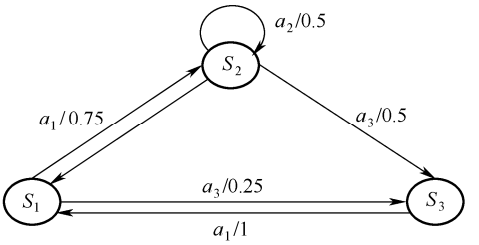


图 2-9　状态转移图

该信源的状态转移概率矩阵为

$$\boldsymbol{P}=p(S_j\mid S_i)=\begin{bmatrix}0 & \frac{3}{4} & \frac{1}{4}\\ 0 & \frac{1}{2} & \frac{1}{2}\\ 1 & 0 & 0\end{bmatrix}$$

而符号条件概率矩阵为

$$\boldsymbol{P}=p(a_j\mid S_i)=\begin{bmatrix}\frac{1}{2} & \frac{1}{4} & \frac{1}{4}\\ 0 & \frac{1}{2} & \frac{1}{2}\\ 1 & 0 & 0\end{bmatrix}$$

可见，这两者是不一样的。

该信源稳定后的状态概率分布为：$p(S_1)=\frac{2}{7}$，$p(S_2)=\frac{3}{7}$，$p(S_3)=\frac{2}{7}$。

符号概率分布为：$p(a_1)=\frac{3}{7}$，$p(a_2)=\frac{2}{7}$，$p(a_3)=\frac{2}{7}$。

现在，我们来计算遍历的 m 阶马尔可夫信源所能够提供的平均信息量，即信源的极限熵 H_∞。

对于齐次、遍历的马尔可夫信源，其状态 s_i 由 $(x_{i1},x_{i2},\cdots,x_{im})$ 唯一确定，因此有

$$p(x_{i(m+1)}\mid x_{im},x_{i(m-1)},\cdots,x_{i2},x_{i1})=p(x_{i(m+1)}\mid s_i)$$

对该等式两边同取对数，并对 $x_{i1},x_{i2},\cdots,x_{im},x_{i(m+1)}$ 和 s_i 取统计平均，再取负值，有

$$\begin{aligned}\text{左边}&=-\sum_{i(m+1),\cdots i1;i}p(x_{i(m+1)},x_{im},\cdots,x_{i2},x_{i1};s_i)\log p(x_{i(m+1)}\mid x_{im},\cdots,x_{i2},x_{i1})\\&=-\sum_{im+1,\cdots i1;i}p(x_{i(m+1)},x_{im},\cdots,x_{i2},x_{i1})\log p(x_{i(m+1)}\mid x_{im},\cdots,x_{i2},x_{i1})\\&=H(x_{i(m+1)}\mid x_{im},\cdots,x_{i2},x_{i1})\\&=H_{m+1}\end{aligned}$$

$$
\begin{aligned}
\text{右边} &= -\sum_{i(m+1),\cdots i1;i} p(x_{i(m+1)}, x_{im}, \cdots, x_{i2}, x_{i1}; s_i) \log p(x_{i(m+1)} \mid s_i) \\
&= -\sum_{i(m+1),\cdots i1;i} p(x_{im}, \cdots, x_{i2}, x_{i1}) \log p(x_{i(m+1)} \mid s_i) \\
&= -\sum_{i(m+1)} \sum_{i} p(s_i) p(x_{i(m+1)} \mid s_i) \log p(x_{i(m+1)} \mid s_i) \\
&= \sum_{i} p(s_i) H(X \mid s_i)
\end{aligned}
$$

左右两边相等，即

$$H_{m+1} = \sum_{i} p(s_i) H(X \mid s_i) \tag{2-22}$$

$p(s_i)$：马尔可夫信源的稳态分布概率，即 W_i。

$H(X \mid s_i)$：信源处于某一状态 s_i 时平均每发出一个符号的信息量。

$$H(X \mid s_i) = -\sum_{j} p(x_j \mid s_i) \log p(x_j \mid s_i) \tag{2-23}$$

对状态 s_i 的全部可能性进行统计平均，就可以得到马尔可夫信源的平均符号熵 H_{m+1}，即极限信息量 H_∞。

例题 2-11　某三状态马尔可夫信源，状态转移概率矩阵为

$$\boldsymbol{P} = \begin{bmatrix} 0.1 & 0 & 0.9 \\ 0.5 & 0 & 0.5 \\ 0 & 0.2 & 0.8 \end{bmatrix}$$

计算该信源的极限信息量 H_∞。

解　设稳态分布的概率为 W_1、W_2、W_3，由

$$\sum_{i=1}^{3} W_i p_{ij} = W_j \ , \quad \sum_{i=1}^{3} W_j = 1$$

可求得：$W_1 = \frac{5}{59}$，$W_1 = \frac{9}{59}$，$W_1 = \frac{45}{59}$。

在 S_1 状态下，每输出一个符号的平均信息量为

$$H(X \mid S_1) = 0.1 \log \frac{1}{0.1} + 0.9 \log \frac{1}{0.9} = 0.469 \text{bit/符号}$$

$$H(X \mid S_2) = 0.5 \log \frac{1}{0.5} + 0.5 \log \frac{1}{0.5} = 1 \text{bit/符号}$$

$$H(X \mid S_3) = 0.2 \log \frac{1}{0.2} + 0.8 \log \frac{1}{0.8} = 0.722 \text{bit/符号}$$

对三个状态取统计平均，得到信源每输出一个符号的极限信息量为

$$H_\infty = \sum_{i=1}^{3} W_i H(X \mid S_i) = 0.743 \text{bit/符号}$$

即马尔可夫信源的熵。

2.5 连续信源熵与互信息

连续信源也是一类信息系统中的常见信源，本节在前面离散信源熵的基础上，介绍连续信源的信息熵以及互信息。

2.5.1 连续信源的信源熵

离散随机变量的互信息定义可以直接推广到连续随机变量。

假设一连续随机变量 $x \in [a,b]$，令 $\Delta x = \dfrac{b-a}{n}$，$x_i \in [a+(i-1)\Delta x, a+i\Delta x]$，$p_x(x)$ 为连续变量 x 的概率密度函数，则 x 取 x_i 的概率为

$$p(x_i) = \int_{a+(i-1)\Delta x}^{a+i\Delta x} p_x(x)\mathrm{d}x = p_x(x_i)\Delta x$$

由离散信源熵的定义，有

$$H_n(X) = -\sum_{i=1}^{n} p(x_i)\log p(x_i) = -\sum_{i=1}^{n} p_x(x_i)\Delta x \log p_x(x_i)\Delta x$$

当 $n \to \infty$ 时，$\Delta x \to 0$，根据积分的定义，得

$$\begin{aligned} H(X) &= \lim_{n\to\infty} H_n(x) \\ &= -\int_a^b p_x(x_i)\log p_x(x_i)\mathrm{d}x - \lim_{\Delta x\to 0}\log\Delta x\int_a^b p_x(x_i)\mathrm{d}x \\ &= -\int_a^b p_x(x_i)\log p_x(x_i)\mathrm{d}x - \lim_{\Delta x\to 0}\log\Delta x \end{aligned} \tag{2-24}$$

上式中的第 1 项具有离散信源熵的形式，第 2 项为无穷大。丢掉无穷大项，定义连续信源熵为

$$H_c(X) = -\int_{-\infty}^{\infty} p_x(x)\log p_x(x)\mathrm{d}x \tag{2-25}$$

该连续信源熵也称为微分熵。

可以看出，连续信源熵与离散信源熵具有相同的形式，但意义不同。对连续信源而言，连续的随机变量需要用无穷多个二进制数字才能精确表示，因此，它的自信息量是无穷的，它的熵也是无穷的。

同样，可以定义两个随机变量 X、Y 的情况。

联合熵

$$H_c(XY) = -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(xy)\log p(xy)\mathrm{d}x\mathrm{d}y \tag{2-26}$$

条件熵

$$H_c(Y/X) = -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x)p(y\mid x)\log p(y\mid x)\mathrm{d}x\mathrm{d}y \tag{2-27}$$

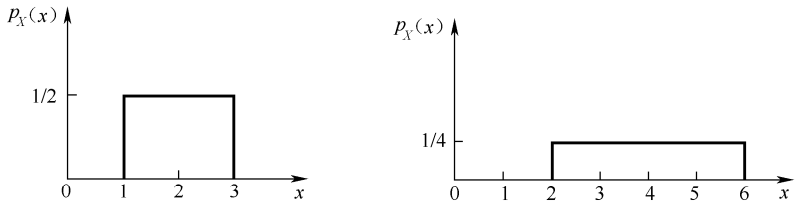
X 与 Y 之间的平均互信息量为

$$\begin{aligned} I(X;Y) &= H_c(X) - H_c(X \mid Y) \\ &= H_c(Y) - H_c(Y \mid X) \\ &= H_c(X) + H_c(Y) - H_c(XY) \\ &= I(Y;X) \end{aligned} \tag{2-28}$$

由联合熵与条件熵的定义，可得

$$H_c(XY) = H_c(X) + H_c(Y \mid X) = H_c(Y) + H_c(X \mid Y) \tag{2-29}$$

例题 2-12　某一连续信源的概率密度如图 2-10 所示。



（a）输出信号概率密度　　（b）输出信号放大 2 倍的概率密度

图 2-10　信源的概率密度

计算其相对熵。

解　该信源的熵为

$$\begin{aligned} H_c(X) &= -\int_{-\infty}^{\infty} p_x(x) \log_2 p_x(x) \mathrm{d}x \\ &= -\int_1^3 \frac{1}{2} \log_2 \frac{1}{2} \mathrm{d}x = 1\text{bit} \end{aligned}$$

以及

$$H_c(X) = -\int_2^6 \frac{1}{4} \log_2 \frac{1}{4} \mathrm{d}x = 2\text{bit}$$

(*b*)是(*a*)的放大，计算结果表示信息量增大了一倍。这是不正确的。之所以出现这种情况，是由于无穷大项造成的。所以，$H_c(X)$ 给出的熵具有相对意义，不是绝对熵值。因此，连续信源的熵也称为相对熵。

2.5.2　最大熵定理

我们已经知道，离散信源在等概率时，其信源熵取最大值。那么对于连续信源，情况如何呢？由于连续信源的熵只有相对的意义，而没有绝对的值，只有在一定的条件下，连续信源才具有最大值。

1．限峰功率最大熵定理

限峰功率就是信源输出幅度受限的情况。对于定义域为有限的随机矢量 X，当它为均匀分布时，具有最大熵。

若连续随机变量 X 的幅度取值限制在闭区间$[a,b]$，则 $\int_a^b p(x)\mathrm{d}x = 1$，若任意 $p_x(x)$ 符合平

均分布条件，即

$$p_x(x)=\begin{cases}\dfrac{1}{b-a}, & a\leqslant x\leqslant b\\ 0, & \text{其他}\end{cases}$$

此时，信源达到最大熵值。

2．限平均功率最大熵定理

限平均功率即信源输出的平均功率受限。对于相关矩阵一定的随机矢量 X，当它是正态分布时具有最大熵。

设随机变量 X 的概率密度为

$$p_x(x)=\frac{1}{\sqrt{2\pi\sigma}}\exp[-(x-m)^2/2\sigma^2]$$

其中，m 为均值，σ^2 为方差，则该信源的熵为

$$\begin{aligned}H_c(X)&=-\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi\sigma}}\exp[-(x-m)^2/2\sigma^2]\ln\frac{1}{\sqrt{2\pi\sigma}}\exp[-(x-m)^2/2\sigma^2]\mathrm{d}x\\&=E_X\{-\ln\frac{1}{\sqrt{2\pi\sigma}}\exp[-(x-m)^2/2\sigma^2]\}\\&=E_X[\frac{1}{2}\ln 2\pi\sigma^2+\frac{1}{2\sigma^2}(x-m)^2]\\&=\frac{1}{2}\ln 2\pi\sigma^2+\frac{1}{2\sigma^2}\sigma^2\\&=\frac{1}{2}\ln 2\pi\sigma^2\mathrm{e}\end{aligned}$$

可以看到，$H_c(X)$ 只与随机变量 X 的方差 σ^2 有关，而方差在物理意义上往往表示信号的交流功率，即 $P=\sigma^2$。因此，在限制信号的平均功率条件下，具有正态分布的信源可输出最大的相对熵 $H_c(X)=\frac{1}{2}\ln(2\pi\sigma^2\mathrm{e})$。

由定理 2 可知，如果噪声是正态分布，则噪声熵最大。因此，高斯白噪声获得最大的噪声熵。在通信系统中，各种设计都以高斯白噪声作为标准，就是根据最坏的条件设计出可靠的系统。

由这两个定理说明，连续信源在不同限制的条件下，其最大熵是不同的。若没有限制条件，则连续信源的最大熵不存在。

2.6 信源的冗余度

冗余度也称为多余度或剩余度，它表示给定信源在实际发出消息时所包含的多余的信息。如果某一消息所包含的符号比表达这个消息所需要的符号多的话，这样的消息就存在冗余度。

冗余度来自两个方面。一是信源符号间的相关性。相关程度越大，信源的实际熵越小，

趋于极限熵 $H_\infty(\boldsymbol{X})$；反之，相关程度越小，信源实际熵就越大。二是信源符号分布的不均匀性。信源符号等概率分布时为最大，当信源输出的符号之间彼此不相关且为等概率分布时，信源的实际熵趋于最大值 $H_0(X)$。

对于极限熵为 $H_\infty(\boldsymbol{X})$ 的平稳信源来说，需要传送这一信源的信息，理论上只需要有传送 $H_\infty(\boldsymbol{X})$ 的手段即可。但是对它的概率分布未能完全掌握，只能算出 $H_m(\boldsymbol{X})$ 的时候，若用能够传送 $H_m(\boldsymbol{X})$ 的手段去传送具有 $H_\infty(\boldsymbol{X})$ 的信源信息，这是很不经济的。

为了定量地描述信源的有效性，定义信息效率为

$$\eta = \frac{H_\infty(\boldsymbol{X})}{H_m(\boldsymbol{X})} \tag{2-30}$$

$\boldsymbol{\eta}$ 表示信源不肯定性的程度，即实际信源比理想信源的缩减程度，由定义可知，$0 \leqslant \boldsymbol{\eta} \leqslant 1$。而$(1-\boldsymbol{\eta})$表示信源肯定性的程度，肯定性不包含有信息量，因此是冗余的，定义冗余度为

$$\gamma = 1-\eta = 1-\frac{H_\infty(\boldsymbol{X})}{H_m(\boldsymbol{X})} \tag{2-31}$$

实际上，当仅仅知道信源符号有 q 种可能的取值，而对每个信源符号的概论特性一无所知时，合理的假设是这 q 种取值是等可能性的，此时信源熵取最大值 $H_0(X)=\log q$。在统计学上认为，最大熵是最合理、最自然、最无主观性的假设。一旦测得其一维分布，就能计算出 $H_1(X)$，显然，$H_0-H_1 \geqslant 0$ 是测定一维分布后获得的信息。测定 m 维分布后获得的信息就是 H_0-H_m。如果所有维分布都能测定，就可以得到 H_0-H_∞。因此，压缩传送信息的手段，有赖于已预先从测量中获得的信息，这一部分就没必要传送了。

以英文字母作为信源符号为例。英文字母共有 26 个，另外加上空格共有 27 个符号，那么该信源的最大熵可用下式计算。

$$H_0(X)=\log_2\frac{1}{27}=4.76\text{bit/符号}$$

对在英文书中各个符号的出现情况加以统计，可得各个符号出现的概率，由大到小依次为：空格→E→T→O→A→N→I→R→…→Z，每个符号出现的概率如表 2-2 所示。

表 2-2　　英文字母出现的概率

符号	概率	符号	概率	符号	概率	符号	概率
空格	0.2	I	0.055	C	0.023	B	0.010 5
E	0.105	R	0.054	F,U	0.022 5	V	0.008
T	0.072	S	0.052	M	0.021	K	0.003
O	0.065 4	H	0.047	P	0.017 5	X	0.002
A	0.063	D	0.035	Y,W	0.012	J,Q	0.001
N	0.059	L	0.029	G	0.011	Z	0.001

若认为各英文字母间是独立的，可求得

$$H_1(X)=-\sum_{i=1}^{27}p(x_i)\log_2 p(x_i)=4.03\text{bit/符号}$$

若考虑前后 2 个、3 个、4 个、……若干个字母之间存在相关性，根据各字母出现的条件概率，可分别求得

$$H_2(\boldsymbol{X}) = 3.32\text{bit/符号}$$
$$H_3(\boldsymbol{X}) = 3.10\text{bit/符号}$$
$$\vdots$$
$$H_\infty(\boldsymbol{X}) = 1.40\text{bit/符号}$$

若用一般的传送方式，即采用等概率假设下的信源符号熵 $H_0(X)$，则信息效率和冗余度分别计算如下。

$$\eta = \frac{1.40}{4.76} = 0.29$$
$$\gamma = 1 - \eta = 0.71$$

因此，对于英文字母构成的文字信源来说，如果仅从理论上分析，信源信息中有 71%是冗余的。100 页的英文书，从理论上看，只有 29 页是有效的，其余 71 页是多余的，因此，可对该信源进行压缩。

正是因为信源符号中存在着统计上的不均匀性和相关性，才使得信源存在冗余度。在通信系统中，为了提高信息传输的效率，往往需要把信源中的大量冗余信息进行压缩，即进行信源编码。考虑到通信系统中存在着干扰和噪声，为了保证信息传输的可靠性，则又需要信源具有一定的冗余度，在信息传输之前，加入某些特殊的冗余度，即进行信道编码。这样，就可以使通信系统的有效性和可靠性都能够得以保证。

本 章 小 结

信源是信息系统的主要研究对象，本章主要介绍了信源的自信息、互信息、信息熵等概念。对离散信源、马尔可夫信源以及连续信源的信息熵进行了分析和介绍，并以相应的例题为基础分析了这些信源信息熵的计算和性质。

习　　题

2.1　试问四进制、八进制脉冲所含信息量是二进制脉冲的多少倍？

2.2　设某班学生在一次考试中获优（A）、良（B）、中（C）、及格（D）和不及格（E）的人数相等。当教师通知某甲：“你没有不及格”，甲获得了多少比特信息？为确定自己的成绩，甲还需要多少信息？

2.3　中国国家标准局所规定的二级汉字共 6 763 个。设每字使用的频度相等，求一个汉字所含的信息量。设每个汉字用一个16×16的二元点阵显示，试计算显示方阵所能表示的最大信息。显示方阵的利用率是多少？

2.4　两个信源 S_1 和 S_2 均有两种输出：$X = 0, 1$ 和 $Y = 0, 1$，概率分别为 $P(X=0) = P(X=1) = 1/2$，$P(Y=0) = 1/4$，$P(Y=1) = 3/4$。试计算 $H(X)$ 和 $H(Y)$。设 S_1 发出序列 0101，S_2 发出 0111，如传输过程无误，第一个字符传送结束后，相应的两个信宿分别收到多少信息量？当整个序列传送结束后，收到的总信息量及平均每次发送的信息量又各是多少（设信源先后发出的数字相互独立）？

2.5　从普通的 52 张扑克牌中随机地抽出一张，解答如下问题。

（1）当告知你抽到的那张牌是：红桃；人头；红桃人头时，你所得到的信息量各是多少？

（2）如果已知那张牌是红人头，为确切地知道是哪张牌，还需要多少信息量？

2.6　同时掷出两个正常的骰子，也就是各面呈现的概率都为 1/6，求解以下问题。

（1）“3 和 5 同时出现”这个事件的自信息。

（2）“两个 1 同时出现”这个事件的自信息。

（3）两个点数的各种组合（无序对）的熵。

（4）两个点数之和（即 2, 3, … , 12 构成的子集）的熵。

（5）两个点数中至少有一个是 1 的自信息量。

2.7　某一无记忆信源的符号集为{0, 1}，已知 P(0) = 1/4，P(1) = 3/4。

（1）求信源熵。

（2）有 100 个符号构成的序列，求某一特定序列［例如有 m 个“0”和（100 − m）个“1”］的自信息量的表达式。

（3）计算（2）中序列的熵。

2.8　某地区的女孩中有 25%是大学生，在女大学生中有 75%是身高 1.6 米以上的，而女孩中身高 1.6 米以上的占总数一半。假如我们得知“身高 1.6 米以上的某女孩是大学生”的消息，问获得多少信息量？

2.9　设离散无记忆信源 $\begin{bmatrix} X \\ P(x) \end{bmatrix}=\begin{bmatrix} 0 & 1 & 2 & 3 \\ \frac{3}{8} & \frac{1}{4} & \frac{1}{4} & \frac{1}{8} \end{bmatrix}$，其发出的消息为（202120130201300120321001103210100210302011220321 0），求解以下问题。

（1）此消息的自信息是多少？

（2）在此消息中平均每个符号携带的信息量是多少？

2.10　从大量统计资料知道，男性中红绿色盲的发病率为 7%，女性发病率为 0.5%，如果你问一位男同志：“你是否是红绿色盲？”他的回答可能是“是”，可能是“否”，问这二个答案中各含多少信息量？平均每个回答中含有多少信息量？如果你问一位女同志，则答案中含有的平均自信息量是多少？

2.11　设信源 $\begin{bmatrix} X \\ P(x) \end{bmatrix}=\begin{bmatrix} x_1, & x_2, & x_3, & x_4, & x_5, & x_6 \\ 0.2, & 0.19, & 0.18, & 0.17, & 0.16, & 0.17 \end{bmatrix}$，求该信源的熵，并解释为什么 $H(X) > \log 6$ 不满足信源熵的极值性。

2.12　设离散无记忆信源 S，其符号集为 $\{s_1, s_2, \cdots, s_q\}$，已知其相应的概率分布为 $(p_1, p_2, \cdots, p_q)$。设另一离散无记忆信源 S'，其符号数为 S 信源符号数的两倍，$S' = \{s'_1, s'_2, \cdots, s'_{2q}\}$，并且各符号的概率分布满足

$$p'_i = \begin{cases} (1-\varepsilon)p_i, & i = 1, 2, \cdots, q \\ \varepsilon p_{i-q}, & i = q+1, q+2, \cdots, 2q \end{cases}$$

试求信源 S' 的信息熵与信源 S 的信息熵的关系式。

2.13　设有一概率空间，其概率分布为 $\{p_1, p_2, \cdots, p_q\}$，并有 $p_1 > p_2$。若取 $p'_1 = p_1 - \varepsilon$，

$p_2' = p_2 + \varepsilon$，其中 $0 < 2\varepsilon \leqslant p_1 - p_2$，其他概率不变。试证明由此所得新的概率空间的熵是增加的，并用熵的物理意义加以解释。

2.14 （1）为了使电视图像获得良好的清晰度和规定的适当的对比度，需要用 5×10^5 个像素和 10 个不同的亮度电平，求传递此图像所需的信息率（bit/s）。并设每秒要传送 30 帧图像，所有像素独立变化，且所有亮度电平等概率出现。

（2）设某彩电系统，除了满足对于黑白电视系统的上述要求外，还必须有 30 个不同的色彩度，试证明传输这彩色系统的信息率约是黑白系统的信息率的 2.5 倍。

2.15 每帧电视图像可以认为是由 5×10^5 个像素组成，所有像素均是独立变化，且每一像素又取 128 个不同的亮度电平，并设亮度电平等概率出现。问每帧图像含有多少信息量? 现有一广播员在约 10 000 个汉字的字汇中选 1 000 个字来口述此电视图像，试问广播员描述此图像所广播的信息量是多少（假设汉字字汇是等概率分布，并彼此无依赖）？若要恰当地描述此图像，广播员在口述中至少需要多少汉字?

2.16 为了传输一个由字母 A、B、C、D 组成的符号集，把每个字母编码成两个二元码脉冲序列，以 00 代表 A，01 代表 B，10 代表 C，11 代表 D。每个二元脉冲宽度为 5ms。

（1）不同字母等概率出现时，计算传输的平均信息速率。

（2）若每个字母出现的概率分别为 $p_{\mathrm{A}} = \frac{1}{2}$，$p_{\mathrm{B}} = \frac{1}{4}$，$p_{\mathrm{C}} = \frac{1}{8}$，$p_{\mathrm{D}} = \frac{1}{8}$，试计算传输的平均信息速率。

2.17 证明：$H(X_1X_2\cdots X_N) \leqslant H(X_1) + H(X_2) + \cdots + H(X_N)$。

2.18 设有一个信源，它产生 0，1 序列的消息。它在任意时间而且不论以前发生过什么符号，均按 $P(0)=0.4$，$P(1)=0.6$ 的概率发出符号。

（1）试问这个信源是否是平稳的?

（2）试计算 $H(X^2)$、$H(X_3|\, X_1X_2)$ 及 H_∞。

（3）试计算 $H(X^4)$ 并写出 X^4 信源中可能有的所有符号。

2.19 有一个二元无记忆信源，其发 0 的概率为 p，而 p 约等于 1，所以在发出的二元序列中经常出现的是那些一串为 0 的序列（称为高概率序列）。对于这样的信源我们可以用另一新信源来代替，新信源中只包含这些高概率序列。这时新信源 $S_n = \{s_1, s_2, s_3, \cdots, s_n, s_{n+1}\}$，共有 n+1 个符号，它与高概率的二元序列的对应关系如下。

二元序列：1,01,001, 0001,…,00…01（n 位），00…000（n 位）

新信源符号：$s_1, s_2, s_3, s_4, \cdots, s_n, s_{n+1}$

（1）求 $H(S_n)$。

（2）当 $n\to\infty$ 时求信源的熵 $H(S) = \lim\limits_{n\to\infty} H(S_n)$。

2.20 有一信源，它在开始时以 $P(a) = 0.6$，$P(b) = 0.3$，$P(c) = 0.1$ 的概率发出 X_1。如果 X_1 为 a 时，则 X_2 为 a、b、c 的概率为 1/3；如果 X_1 为 b，X_2 为 a、b、c 的概率为 1/3；如果 X_1 为 c，X_2 为 a、b 的概率为 1/2，为 c 的概率为 0，而且后面发出 X_i 的概率只与 X_{i-1} 有关，又 $P(X_i|\, X_{i-1}) = P(X_2|\, X_1)$，$i \geqslant 3$。试利用马尔可夫信源的图示法画出状态转移图，并计算信源熵 H_∞。

2.21 一阶马尔可夫信源的状态图如题图 2-21 所示，信源 X 的符号集为 $\{0,1,2\}$，并定义

$\overline{p}=1-p$。

（1）求信源平稳后的概率分布 $P(0)$、$P(1)$和$P(2)$。

（2）求此信源的熵。

（3）近似认为此信源为无记忆时，符号的概率分布等于平稳分布。求近似信源的熵 $H(X)$ 并与 H_∞进行比较。

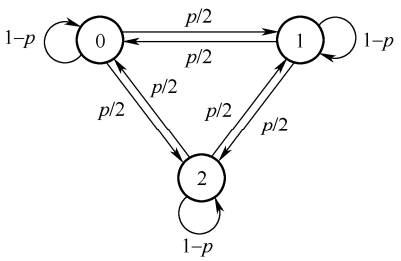
（4）对一阶马尔可夫信源，p 取何值时 H_∞取最大值？当 $p=0$和$p=1$ 时，结果如何？

2.22　一阶马尔可夫信源的状态图如题图 2-22 所示，信源 X 的符号集为{0,1,2}。

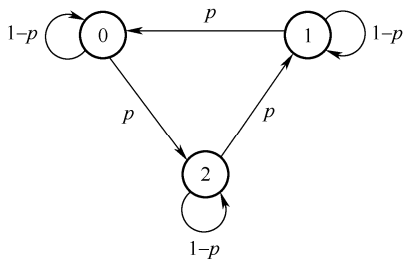
（1）求平稳后信源的概率分布。

（2）求信源的熵 H_∞。

（3）求当 p=0 和 p=1 时信源的熵，并说明其理由。

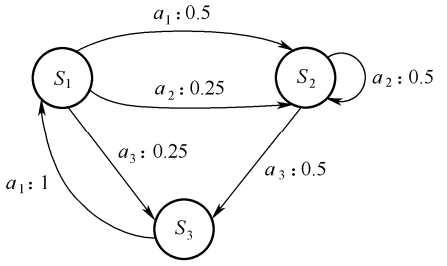


题图 2−21



题图 2−22

2.23　设有一个马尔可夫信源，它的状态集为$\{s_1,s_2,s_3\}$，符号集为$\{a_1,a_2,a_3\}$，其在某状态下发出符号的概率为 $P(a_k\mid s_i)$，$i,k=1,2,3$，如题图 2-23 所示。



题图 2−23

（1）求出图中马尔可夫信源的状态极限概率并找出符号的极限概率。

（2）计算信源处在某一状态下输出符号的条件熵 $H(X\mid s_j)$，$j=1,2,3$。

（3）求出马尔可夫信源熵 H_∞。

2.24　黑白气象传真图的消息只有黑色和白色两种，即信源 X= {黑，白}，设黑色出现的概率为 P（黑）= 0.3，白色出现的概率为 P（白）= 0.7。

（1）假设图上黑白消息出现前后没有关联，求熵 $H(X)$。

（2）假设消息前后有关联，其依赖关系为 P（白|白）= 0.9，P（黑|白）= 0.1，P（白|黑）= 0.2，P（黑|黑）= 0.8，求此一阶马尔可夫信源的熵 H_2。

（3）分别求出上述两种信源的剩余度，比较 $H(X)$和 H_2 的大小，并说明其物理意义。

2.25 给定语音信号样值 X 的概率密度为拉普拉斯分布 $p(x)=\frac{1}{2}\lambda \mathrm{e}^{-\lambda|x|},-\infty<x<+\infty$，求 $H_{\mathrm{c}}(X)$，并证明它小于同样方差的正态变量的连续熵。

2.26 连续随机变量 X 和 Y 的联合概率密度为：$p(x,y)=\begin{cases}\frac{1}{\pi r^2}, & x^2+y^2\leqslant r^2\\ 0, & \text{其他}\end{cases}$，求 $H(X)$、$H(Y)$、$H(XY)$和 $I(X;Y)$。（提示：$\int_0^{\frac{\pi}{2}}\log_2\sin x\mathrm{d}x=-\frac{\pi}{2}\log_2 2$）

2.27 设 $X=X_1X_2\cdots X_N$ 是离散平稳有记忆信源，试证明：

$H(X_1X_2\cdots X_N)=H(X_1)+H(X_2\mid X_1)+H(X_3\mid X_1X_2)+\cdots+H(X_N\mid X_1X_2\cdots X_{N-1})$。

2.28 设 $X=X_1X_2\cdots X_N$ 是 N 维高斯分布的连续信源，且 $X_1,X_2,\cdots,X_N$ 的方差分别是 $\sigma_1^2,\sigma_2^2,\cdots,\sigma_N^2$，它们之间的相关系数 $\rho(X_iX_j)=0(i,j=1,2\cdots,N,i\neq j)$。试证明：$N$ 维高斯分布的连续信源熵为

$$H_{\mathrm{c}}(X)=H_{\mathrm{c}}(X_1X_2\cdots X_N)=\frac{1}{2}\sum_{i=1}^{N}\log 2\pi\mathrm{e}\sigma_i^2$$

2.29 设有一连续信源 X，其概率密度函数为

$$p(x)=\begin{cases}bx^2, & 0\leqslant x\leqslant a\\ 0, & \text{其他}\end{cases}$$

（1）试求信源 X 的熵 $H_{\mathrm{c}}(X)$。

（2）试求 $Y=X+A$（$A>0$）的熵 $H_{\mathrm{c}}(Y)$。

（3）试求 $Y=2X$ 的熵 $H_{\mathrm{c}}(Y)$。

第3章 信道与信道容量

3.1 信道的基本概念

信道是传送信息的载体（信号所通过的通道）。信息是抽象的，信道则是具体的。比如，二人对话，二人间的空气就是信道；打电话，电话线就是信道；看电视、听收音机，收、发间的空间就是信道。信道的作用在信息系统中用于传输与存储信息，而在通信系统中则主要用于传输。

在通信系统中研究信道，主要是为了描述、度量、分析不同类型信道，计算其容量，即极限传输能力，并分析其特性。

3.1.1 信道的数学模型与分类

实际的通信系统中，信道的种类很多，包含的设备也各式各样。根据载荷消息的媒体的不同，可有邮递信道、电信道、光信道、声信道等。从信道传输的角度来考虑，信道可以根据输入和输出信号的形式、信道的统计特性及信道的用户多少等方法来进行分类。

1. 信道分类

信道可以从不同角度加以分类，常用的有下面几种分类。

从工程物理背景——传输介质类型。

从数学描述方式——信号与干扰描述方式。

从信道本身的参数类型——恒参与变参。

从用户类型——单用户与多用户。

- （1）**传输介质类型**
  - 固体介质
    - 明线
    - 电缆
      - 对称平衡电缆（市内）
      - 小同轴（长途）
      - 中同轴（长途）
  - 空气介质
    - 长波
    - 中波
    - 短波
    - 超短波
    - 微波
      - 移动
      - 视距接力
      - 散射
        - 对流层
        - 电离层
      - 卫星
    - 光波
  - 混合介质
    - 波导
    - 光缆
- （2）**信号与干扰类型**
  - 信号类型
    - 离散
    - 连续
    - 半离散
    - 半连续
  - 干扰类型
    - 无干扰：干扰少到可忽略
    - 有干扰
      - 线性叠加干扰
        - 无源热噪声
        - 有源散弹噪声
        - 脉冲噪声
      - 乘性干扰
        - 交调
        - 衰落
        - 码间干扰
- （3）**信道参量类型**
  - 恒参信道（时不变信道）
  - 变参信道（时变信道）
- （4）**用户类型**
  - 二用户信道（点对点通信）
  - 多用户信道（通信网）

信道划分是人为的，如图 3-1 所示。

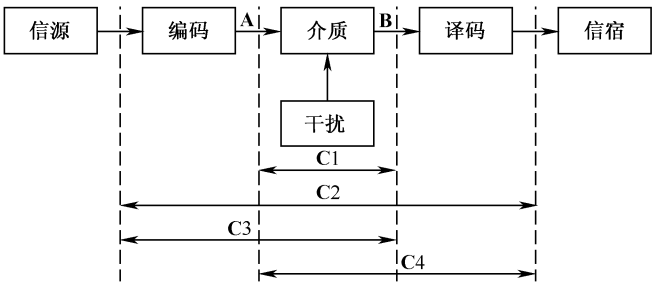


图 3-1　信道分类举例

其中，C1 为连续信道，调制信道；

C2 为离散信道，编码信道；

C3 为半离散、半连续信道；

C4 为半连续、半离散信道。

2．信道模型

信道的数学模型用来表征实际物理信道的特性，它对通信系统的分析和设计是十分方便的。下面我们简要描述调制信道和编码信道这两种广义信道的数学模型。

（1）调制信道模型。调制信道是为研究调制与解调问题所建立的一种广义信道，它所关心的是调制信道输入信号形式和已调信号通过调制信道后的最终结果，对于调制信道内部的变换过程并不关心。因此，调制信道可以用具有一定输入、输出关系的方框来表示。通过对调制信道进行大量的分析研究，发现它具有如下共性。

① 有一对（或多对）输入端和一对（或多对）输出端。

② 绝大多数的信道都是线性的，即满足线性叠加原理。

③ 信号通过信道具有一定的延迟时间而且它还会受到（固定的或时变的）损耗。

④ 即使没有信号输入，在信道的输出端仍可能有一定的功率输出（噪声）。

根据以上几条性质，调制信道可以用一个二端口（或多端口）线性时变网络来表示，这个网络便称为调制信道模型，如图 3-2 所示。

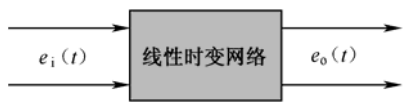


图 3-2　调制信道模型

对于二对端的信道模型，其输出与输入的关系有

$$e_o(t) = f[e_i(t)] + n(t) \tag{3-1}$$

其中，$e_i(t)$ 为输入的已调信号；$e_o(t)$ 为信道总输出波形；$n(t)$ 为加性噪声，$n(t)$ 与 $e_o(t)$ 相互独立，无依赖关系。$f[e_i(t)]$ 表示已调信号通过网络所发生的（时变）线形变换。现在，我们假定能把 $f[e_i(t)]$ 写为 $k(t)\times e_i(t)$，其中，$k(t)$ 依赖于网络的特性，$k(t)$ 乘 $e_i(t)$ 反映了网络特性对 $e_i(t)$ 的作用。$k(t)$ 的存在，对 $e_i(t)$ 来说是一种干扰，通常称其为乘性干扰。于是式（3-1）可表示为

$$e_o(t) = k(t)\times e_i(t) + n(t) \tag{3-2}$$

式（3-2）即为二对端信道的数学模型。

由以上分析可知，信道对信号的影响可归结为两点：一是乘性干扰 $k(t)$，二是加性干扰 $n(t)$。对于信号来说，如果我们了解 $k(t)$ 与 $n(t)$ 的特性，就能知道信道对信号的具体影响。通常信道特性 $k(t)$ 是一个复杂的函数，它可能包括各种线性失真、非线性失真、交调失真、衰落等。同

时由于信道的迟延特性和损耗特性随时间作随机变化，故 $k(t)$ 往往只能用随机过程来描述。

在我们实际使用的物理信道中，根据信道传输函数 $k(t)$ 的时变特性的不同可以分为两大类：一类是 $k(t)$ 基本不随时间变化，即信道对信号的影响是固定的或变化极为缓慢的，这类信道称为恒定参量信道，简称恒参信道；另一类信道是传输函数 $k(t)$ 随时间随机快速变化，这类信道称为随机参量信道，简称随参信道。

（2）编码信道模型。编码信道包括调制信道、调制器和解调器，它与调制信道模型有明显的不同，是一种数字信道或离散信道。编码信道的输入是离散的时间信号，输出也是离散的时间信号，对信号的影响则是将输入数字序列变成另一种输出数字序列。由于信道噪声或其他因素的影响，将导致输出数字序列发生错误，因此输入、输出数字序列之间的关系可以用一组转移概率来表示。

二进制数字传输系统的一种简单的编码信道模型如图 3-3 所示。图中 $p(0)$ 和 $p(1)$ 分别是发送“0”符号和“1”符号的先验概率，$p(0|0)$ 与 $p(1|1)$ 是正确转移的概率，而 $p(1|0)$ 与 $p(0|1)$ 是错误转移概率。信道噪声越大将导致输出数字序列发生错误越多，错误转移概率 $p(1|0)$ 与 $p(0|1)$ 也就越大；反之，错误转移概率 $p(1|0)$ 与 $p(0|1)$ 就越小。输出的总的错误概率为

$$p_e = p(0)p(1|0) + p(1)p(0|1)$$

在图 3-3 所示的编码信道模型中，由于信道噪声或其他因素影响导致输出数字序列发生错误是统计独立的，因此这种信道是无记忆编码信道。根据无记忆编码信道的性质可以得到下式。

$$p(0|0) = 1 - p(1|0)$$

$$p(1|1) = 1 - p(0|1)$$

转移概率完全由编码信道的特性所决定。一个特定的编码信道，有确定的转移概率。由无记忆二进制编码信道模型，容易推出无记忆多进制的模型。图 3-4 给出了一个无记忆四进制编码信道模型。如果编码信道是有记忆的，即信道噪声或其他因素影响导致输出数字序列发生错误是不独立的，则编码信道模型要比图 3-3 或图 3-4 所示的模型复杂得多，信道转移概率表示式也将变得很复杂。

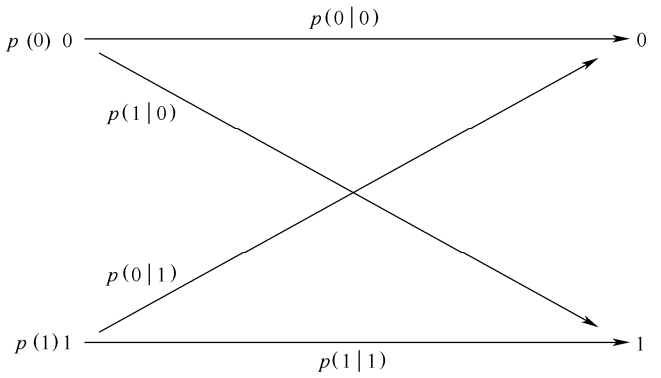


图 3-3　二进制编码信道模型

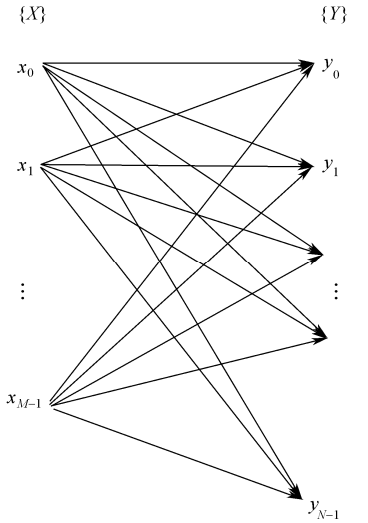


图 3-4　多进制无记忆编码信道模型

3.1.2 信道参数

在阐明了离散单符号信道的数学模型，即给出了信道输入与输出的统计依赖关系后，接下来研究信道参数。

1. 信道疑义度

根据熵的概念，可以计算信道输入信源 X 的熵。

$$H(X)=\sum_{i=1}^{r}p(a_i)\log\frac{1}{p(a_i)}=-\sum_{X}p(x)\log p(x) \tag{3-3}$$

第二个等式是简明写法，求和号是对变量 X 的所有取值求和，而 $p(x)$ 表示随机变量 X 取任意一个取值的概率。

$H(X)$ 是在接收到输出 Y 以前，关于输入变量 X 的先验不确定性的度量，所以称为先验熵。如果信道中无干扰（噪声），信道输出符号与输入符号一一对应，那么，接收到传送过来的符号后就消除了对发送符号的先验不确定性。但一般信道中有干扰（噪声）的存在，接收到输出 Y 后对发送是什么符号仍有不确定性。那么，怎样来度量接收到 Y 后关于 X 的不确定性呢？当没有接收到输出 Y 时，已知输入变量 X 的概率分布为 $p(x)$；而当接收到输出符号 $y=b_j$ 后，输入符号的概率分布发生了变化，变成后验概率分布 $p(x|b_j)$。那么，接收到输出符号 $y=b_j$ 后，关于 X 的平均不确定性为

$$H(X|b_j)=\sum_{i=1}^{r}p(a_i|b_j)\log\frac{1}{p(a_i|b_j)}=\sum_{X}p(x|b_j)\log\frac{1}{p(x|b_j)} \tag{3-4}$$

这是接收到输出符号为 b_j 后关于 X 的后验熵。可见，接收到输出符号 b_j，先验熵变成后验熵。所以后验熵是当信道接收端接收到输出符号 b_j 后，关于输入符号的信息测度。

后验熵在输出 y 的取值范围内是个随机量，将后验熵对随机变量 Y 求期望，得条件熵为

$$\begin{aligned}H(X|Y)&=E\left[H(X|b_j)\right]=\sum_{j=1}^{s}p(b_j)H(X|b_j)=\sum_{j=1}^{s}p(b_j)\sum_{i=1}^{r}p(a_i|b_j)\log\frac{1}{p(a_i|b_j)}\\&=\sum_{i=1}^{r}\sum_{j=1}^{s}p(a_i|b_j)\log\frac{1}{p(a_i|b_j)}=\sum_{X,Y}p(xy)\log\frac{1}{p(x|y)}\end{aligned} \tag{3-5}$$

这个条件熵称为信道疑义度。它表示在输出端收到输出变量 Y 的符号后，对于输入端的变量 X 尚存的平均不确定性（存在疑义）。这个对 X 尚存的不确定性是由于干扰（噪声）引起的。如果是一一对应信道，那么接收到输出 Y 后，对 X 的不确定性将完全消除，则信道疑义度 $H(X|Y)=0$。由于一般情况下条件熵小于无条件熵，即有 $H(X|Y)<H(X)$。这正说明接收到变量 Y 的所有符号后，关于输入变量 X 的平均不确定性将减少，即总能消除一些关于输入端 X 的不确定性，从而获得了一些信息。

2. 平均互信息

根据上述，我们已知 $H(X)$ 代表接收到输出符号以前关于输入变量 X 的平均不确定性，而 $H(X|Y)$ 代表接收到输出符号后关于输入变量 X 的平均不确定性。可见，通过信道传输消

除了一些不确定性，获得了一定的信息，所以定义

$$I(X;Y)=H(X)-H(X\mid Y) \tag{3-6}$$

$I(X;Y)$称为X和Y之间的平均互信息。它代表接收到输出符号后平均每个符号获得的关于X的信息量。它也表明，输入与输出两个随机变量之间的统计约束程度。

根据式（3-3）和式（3-5）得

$$\begin{aligned}I(X;Y)&=\sum_{X}P(x)\log\frac{1}{p(x)}-\sum_{X,Y}p(xy)\log\frac{1}{p(x\mid y)}\\&=\sum_{X,Y}p(xy)\log\frac{1}{p(x)}-\sum_{X,Y}p(xy)\log\frac{1}{p(x\mid y)}\\&=\sum_{X,Y}p(xy)\log\frac{p(x\mid y)}{p(x)}\\&=\sum_{X,Y}p(xy)\log\frac{p(xy)}{p(x)p(y)}\\&=\sum_{X,Y}p(xy)\log\frac{p(y\mid x)}{p(y)}\end{aligned} \tag{3-7}$$

式中，X是输入随机变量，$x\in X$；Y是输入随机变量，$y\in Y$。

平均互信息$I(X;Y)$就是互信息$I(x;y)$在两个概率空间X和Y中求统计平均的结果。互信息$I(x;y)$是代表收到某消息y后获得关于某事件x的信息量，即

$$I(x;y)=\log\frac{p(x\mid y)}{p(x)}=\log\frac{p(xy)}{p(x)p(y)}=\log\frac{p(y\mid x)}{p(y)} \tag{3-8}$$

互信息可取正值，也可取负值，如果互信息$I(x;y)$取负值，说明收信者在未收到消息y以前对消息x是否出现的猜测的难易程度较小。但由于噪声的存在，接收到消息y后，反而使收信者对消息x是否出现的猜测难易程度增加了。也就是收信者接收到消息y后对x出现的不确定性反而增加，所以获得的信息量为负值。

对于平均互信息

$$\begin{aligned}I(X;Y)&=\underset{XY}{E}\left[I(x;y)\right]\\&=\sum_{XY}p(xy)I(x;y)\\&=\sum_{XY}p(xy)\log\frac{p(x\mid y)}{p(x)}\end{aligned} \tag{3-9}$$

它是互信息$I(x;y)$的统计平均值，所以平均互信息$I(X;Y)$永远不会取负值。最差情况是平均互信息$I(X;Y)=0$，也就是在信道输出端接收到输出符号Y后不获得任何关于输入符号X的信息量。

3．平均条件互信息

互信息$I(x;y)$是两个概率空间中两事件之间的互信息。平均互信息$I(X;Y)$是两个概率空间X、Y之间的平均互信息。我们可以将这概念推广到三个概率空间中求事件之间的互信息。设三个离散概率空间X，Y，Z，$x\in X$，$y\in Y$，$z\in Z$，且有概率关系式

$$\sum_X \sum_Y \sum_Z p(xyz) = 1 \tag{3-10}$$

$$\left.\begin{aligned} p(xy) &= \sum_Z p(xyz) \qquad x \in X, y \in Y \\ p(xz) &= \sum_Y p(xyz) \qquad x \in X, z \in Z \\ p(yz) &= \sum_X p(xyz) \qquad y \in Y, z \in Z \end{aligned}\right\} \tag{3-11}$$

$$\left.\begin{aligned} p(x) &= \sum_Y p(xy) = \sum_Z p(xz) \qquad x \in X \\ p(y) &= \sum_X p(xy) = \sum_Z p(yz) \qquad y \in Y \\ p(z) &= \sum_X p(xy) = \sum_Y p(yz) \qquad z \in Z \end{aligned}\right\} \tag{3-12}$$

这三个概率空间可以看做两个串接系统 1 和系统 2 的输入和输出空间，如图 3-5（a）所示，也可以考虑为如图 3-5（b）和图 3-5（c）所示，将 X 作为系统 1 的输入空间，而 Y 和 Z 作为系统 1 的输出空间，其中 Y、Z 可为并行输出或按时间前后的串行输出。

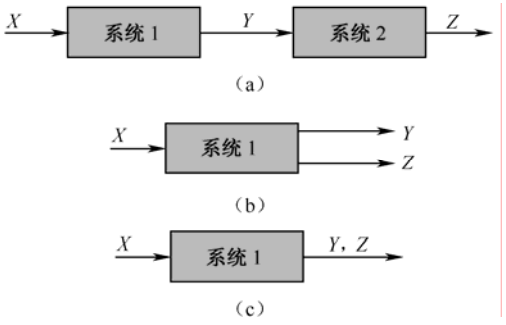


图 3-5 三个概率空间连接关系图

我们定义在已知事件 $z \in Z$ 的条件下，接收到 y 后获得关于某事件 x 的条件互信息，即

$$\begin{aligned} I(x;y|z) &= \log \frac{p(x \mid yz)}{p(x \mid z)} \\ &= \log \frac{p(y \mid xz)}{p(y \mid z)} = \log \frac{p(xy|z)}{p(x|z)p(y|z)} \end{aligned} \tag{3-13}$$

将式（3-13）和式（3-8）比较可以得到下面结论：条件互信息与互信息的区别仅在于先验概率和后验概率都是在某一特定条件下的取值。

另外，也可以从互信息的定义得出，当已知 $y \in Y$ ， $z \in Z$ 后，总共获得关于 $x \in X$ 的互信息，即

$$\begin{aligned} I(x;yz) &= \log \frac{p(x \mid yz)}{p(x)} \\ &= \log \frac{p(x \mid y)p(x \mid yz)}{p(x)p(x \mid y)} = \log \frac{p(x \mid y)}{p(x)} + \log \frac{p(x \mid yz)}{p(x \mid y)} \\ &= I(x;y) + I(x;z \mid y) \end{aligned} \tag{3-14}$$

此关系式表明，yz 联合给出关于 x 的互信息量等于 y 给出关于 x 的互信息量与 y 已知条件下 z 给出关于 x 的互信息量之和。

同理可得

$$I(x;yz)=I(x;z)+I(x;y\mid z) \tag{3-15}$$

类似地，将条件互信息 $I(x;y\mid z)$ 在概率空间 XYZ 中求统计平均，得到平均条件互信息为

$$\begin{aligned}I(X;Y\mid Z)&=E\left[I(x;y\mid z)\right]=\sum_X\sum_Y\sum_Z p(xyz)\log\frac{p(x\mid yz)}{p(x\mid z)}\\&=H(X\mid Z)-H(X\mid ZY)\\&=\sum_X\sum_Y\sum_Z p(xyz)\log\frac{p(xy\mid z)}{p(x\mid z)p(y\mid z)}\\&=H(X\mid Z)+H(Y\mid Z)-H(XY\mid Z)\end{aligned} \tag{3-16}$$

而平均互信息为

$$I(X;YZ)=E\left[I(x;yz)\right]=\sum_X\sum_Y\sum_Z p(xyz)\log\frac{p(x\mid yz)}{p(x)} \tag{3-17}$$

所以，可以推导出下面的关系式。

$$\begin{aligned}I(X;YZ)&=I(X;Y)+I(X;Z\mid Y)\\&=I(X;Z)+I(X;Y\mid Z)\end{aligned} \tag{3-18}$$

式（3-18）表明，联合变量 YZ 和变量 X 之间相互可能提供的平均互信息，等于变量 X 和变量 Y（或 Z）的平均互信息，加上在此变量 Y（或 Z）已知条件下变量 X 和另一变量 Z（或 Y）的平均互信息。平均互信息 $I(X;YZ)$ 是随机变量 X 与联合变量 YZ 之间统计依赖程度的信息测度。

例 3-1 设信源$\begin{pmatrix}X\\p(x)\end{pmatrix}=\begin{bmatrix}x_1 & x_2\\0.6 & 0.4\end{bmatrix}$通过一干扰信道，接收符号为 $y=[y_1,y_2]$，信道传递概率如图 3-6 所示。

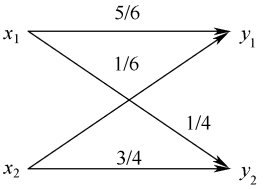


图 3-6 二元信道传递概率

求：（1）信源 X 中事件 x_1 和 x_2 分别含有的自信息。

（2）收到消息 y_j (j=1, 2)后，获得的关于 x_i (i=1,2)的信息量。

（3）信源 X 和信源 Y 的信息熵。

（4）信道疑义度 $H(X\mid Y)$ 和噪声熵 $H(Y\mid X)$。

（5）接收到信息 Y 后获得的平均互信息。

解：（1）信源

$$\begin{pmatrix}X\\p(x)\end{pmatrix}=\begin{bmatrix}x_1 & x_2\\0.6 & 0.4\end{bmatrix}$$

所以事件 x_1 含有的自信息为

$$I(x_1)=-\log p(x_2)=-\log_2 0.6\approx 0.737\text{bit}$$

事件 x_2 含有的自信息为

$$I(x_2)=-\log p(x_2)=-\log_2 0.4\approx 1.32\text{bit}$$

由此可见，概率越小的事件含有的自信息越大。

（2）根据题意是计算 $I(x_i;y_j)$　i=1,2；j=1,2。

由图 3-6 可得

$$p(y_1 \mid x_1)=\frac{5}{6} \qquad p(y_2 \mid x_1)=\frac{1}{6}$$

$$p(y_1 \mid x_2)=\frac{3}{4} \qquad p(y_2 \mid x_2)=\frac{1}{4}$$

互信息公式为

$$I(x_i;y_j)=\log\frac{p(x_i \mid y_j)}{p(x_i)}=\log\frac{p(y_j \mid x_i)}{p(y_j)} \qquad i=1,2；\ j=1,2$$

而

$$p(y_j)=\sum_X p(x_i)p(y_j \mid x_i) \quad j=1,2$$

$$p(x_i \mid y_j)=\frac{p(x_i)p(y_j \mid x_i)}{p(y_j)} \qquad i=1,2；\ j=1,2$$

由以上两式可知，要计算出 $p(x_i \mid y_j)$ 必须先算出 $p(y_j)$，所以，计算互信息时一般选用第二个等式进行计算，可得

$$p(y_1)=\sum_X p(x_i)p(y_1 \mid x_i)=0.6\times\frac{5}{6}+0.4\times\frac{3}{4}=\frac{4}{5}$$

同理 $p(y_2)=0.6\times\frac{1}{6}+0.4\times\frac{1}{4}=\frac{1}{5}$，满足 $p(y_1)+p(y_2)=1$，由此可计算得

$$I(x_1;y_1)=\log\frac{p(y_1 \mid x_1)}{p(y_1)}=\log\frac{5/6}{4/5}=\log\frac{25}{24}\approx 0.059\text{bit}$$

$$I(x_1;y_2)=\log\frac{p(y_2 \mid x_1)}{p(y_2)}=\log\frac{1/6}{1/5}=\log\frac{5}{6}\approx -0.263\text{bit}$$

$$I(x_2;y_1)=\log\frac{p(y_1 \mid x_2)}{p(y_1)}=\log\frac{3/4}{4/5}=\log\frac{15}{16}\approx -0.093\text{bit}$$

$$I(x_2;y_2)=\log\frac{p(y_2 \mid x_2)}{p(y_2)}=\log\frac{1/4}{1/5}=\log\frac{5}{4}\approx 0.322\text{bit}$$

从计算所得结果可知，互信息 $p(x_i \mid y_j)$（i=1,2；j=1,2）可取正值，也可取负值。其中 $I(x_1;y_2)$ 和 $I(x_2;y_1)$ 为负值，它表明由于噪声的存在，接收到消息 y_2 (y_1)后，对消息 x_1 (x_2)是否出现的不确定性反而增加了。

（3）信源 X 的信息熵

$$H(X)=-\sum_{i=1}^{2}p(x_i)\log p(x_i)=-0.6\log 0.6-0.4\log 0.4\approx 0.971\,\text{bit/符号}$$

$$H(Y)=-\sum_{j=1}^{2}p(y_i)\log p(y_i)=-0.8\log 0.8-0.2\log 0.2\approx 0.722\,\text{bit/符号}$$

（4）信道疑义度

$$H(X|Y)=-\sum_{i=1}^{2}\sum_{j=1}^{2}p(x_i)p(y_j|x_i)\log p(x_i|y_j)$$

因为

$$p(x_1|y_1)=\frac{p(x_1)p(y_1|x_1)}{p(y_1)}=\frac{\frac{6}{10}\times\frac{5}{6}}{\frac{4}{5}}=\frac{5}{8}$$

$$p(x_2|y_1)=\frac{p(x_2)p(y_1|x_2)}{p(y_1)}=\frac{\frac{4}{10}\times\frac{3}{4}}{\frac{4}{5}}=\frac{3}{8}$$

$$p(x_1|y_2)=\frac{p(x_1)p(y_2|x_1)}{p(y_2)}=\frac{1}{2}$$

$$p(x_2|y_2)=\frac{p(x_2)p(y_2|x_2)}{p(y_2)}=\frac{1}{2}$$

所以

$$\begin{aligned}H(X|Y)&=-p(x_1)p(y_1|x_1)\log p(x_1|y_1)-p(x_2)p(y_1|x_2)\log p(x_2|y_1)\\&\quad-p(x_1)p(y_2|x_1)\log p(x_1|y_2)-p(x_2)p(y_2|x_2)\log p(x_2|y_2)\\&=-\frac{6}{10}\times\frac{5}{6}\log\frac{5}{8}-\frac{4}{10}\times\frac{3}{4}\log\frac{3}{8}-\frac{6}{10}\times\frac{1}{6}\log\frac{1}{2}-\frac{4}{10}\times\frac{1}{4}\log\frac{1}{2}\\&\approx 0.339\,0+0.424\,5+0.1+0.1\\&=0.963\,5\,\text{bit/符号}\end{aligned}$$

噪声熵

$$\begin{aligned}H(Y|X)&=-\sum_{i=1}^{2}\sum_{j=1}^{2}p(x_i)p(y_j|x_i)\log p(y_j|x_i)\\&=-p(x_1)p(y_1|x_1)\log p(y_1|x_1)-p(x_1)p(y_2|x_1)\log p(y_2|x_1)\\&\quad-p(x_2)p(y_1|x_2)\log p(y_1|x_2)-p(x_2)p(y_2|x_2)\log p(y_2|x_2)\\&=-\frac{6}{10}\times\frac{5}{6}\log\frac{5}{6}-\frac{6}{10}\times\frac{1}{6}\log\frac{1}{6}-\frac{4}{10}\times\frac{3}{4}\log\frac{3}{4}-\frac{4}{10}\times\frac{1}{4}\log\frac{1}{4}\\&\approx 0.131\,5+0.258\,5+0.124\,5+0.2\\&=0.714\,5\,\text{bit/符号}\end{aligned}$$

（5）接收到信息 Y 后获得的平均互信息 $I(X;Y)$

$$\begin{aligned}I(X;Y)&=\sum_{i=1}^{2}\sum_{j=1}^{2}p(x_i)p(y_j|x_i)\log\frac{p(x_i|y_j)}{p(x_i)}\\&=H(X)-H(X|Y)\\&=H(Y)-H(Y|X)\\&=\sum_{i=1}^{2}\sum_{j=1}^{2}p(x_i)p(y_j|x_i)I(x_i;y_j)\\&\approx 0.007\,5\,\text{bit/符号}\end{aligned}$$

3.1.3 信道容量的定义

我们研究信道的目的主要是讨论信道中平均每个符号所能传送的信息量，平均每个符号所能传送的信息量，即信道的信息传输率 R。由前已知，平均互信息 $I(X;Y)$ 就是接收到符号 Y 后平均每个符号获得的关于 X 的信息量。因此信道的信息传输率就是平均互信息，即

$$R = I(X;Y) = H(X) - H(X|Y) \text{（比特/符号）} \tag{3-19}$$

有时我们所关心的是信道在单位时间内（一般以秒为单位）平均传输的信息量。若平均传输一个符号需要 t 秒钟，则信道每秒钟平均传输的信息量为

$$R_t = \frac{1}{t} I(X;Y) = \frac{1}{t} H(X) - \frac{1}{t} H(X|Y) \text{（bit/s）} \tag{3-20}$$

一般称此为信息传输速率。为了便于区别，增加一个下标 t 来表示。它的单位是比特/秒（bit/s）或奈特/秒（nat/s）。

定理 3-1 平均互信息 $I(X;Y)$ 是输入信源概率分布 $p(x)$ 的∩型上凸函数。

证明： 现在选择输入信源的两种已知概率分布 $p_1(x)$ 和 $p_2(x)$，其对应的联合概率分布 $p_1(xy) = p_1(x)p(y|x)$ 和 $p_2(xy) = p_2(x)p(y|x)$，因而信道输出端的平均互信息量分别为 $I[p_1(x)]$ 和 $I[p_2(x)]$。再选择输入变量 X 的另一种概率分布 $p(x)$，令 $0<\lambda<1$，则 $p(x) = \lambda p_1(x) + (1-\lambda)p_2(x)$，其对应的平均互信息量为 $I[p(x)]$。

根据平均互信息的定义得

$$\begin{aligned}
&\lambda I[p_1(x)] + (1-\lambda) I[p_2(x)] - I[p(x)] \\
&= \sum_{X,Y} \lambda p_1(xy) \log_2 \frac{p(y|x)}{p_1(y)} + \sum_{X,Y} (1-\lambda) p_2(xy) \log_2 \frac{p(y|x)}{p_2(y)} - \sum_{X,Y} p(xy) \log_2 \frac{p(y|x)}{p(y)} \\
&= \sum_{X,Y} \lambda p_1(xy) \log_2 \frac{p(y|x)}{p_1(y)} + \sum_{X,Y} (1-\lambda) p_2(xy) \log_2 \frac{p(y|x)}{p_2(y)} - \sum_{X,Y} [\lambda p_1(xy) + (1-\lambda) p_2(xy)] \log_2 \frac{p(y|x)}{p(y)}
\end{aligned}$$

式中是根据概率关系

$$\begin{aligned}
p(xy) &= p(x)p(y|x) = [\lambda p_1(x) + (1-\lambda) p_2(x)] p(y|x) \\
&= \lambda p_1(xy) + (1-\lambda) p_2(xy)
\end{aligned}$$

所以得

$$\begin{aligned}
&\lambda I[p_1(x)] + (1-\lambda) I[p_2(x)] - I[p(x)] \\
&= \sum_{X,Y} \lambda p_1(xy) \log_2 p_1(y|x) - \sum_{X,Y} \lambda p_1(xy) \log_2 p_1(y) + \sum_{X,Y} (1-\lambda) p_2(xy) \log_2 p(y|x) \\
&\quad - \sum_{X,Y} (1-\lambda) p_2(xy) \log_2 p_2(y) - \sum_{X,Y} [\lambda p_1(xy) + (1-\lambda) p_2(xy)] \log_2 p(y|x) + \sum_{X,Y} [\lambda p_1(xy) + \\
&\quad (1-\lambda) p_2(xy)] \log_2 p(y) \\
&= \sum_{X,Y} [\lambda p_1(xy) + (1-\lambda) p_2(xy)] \log_2 p(y) - \sum_{X,Y} \lambda P_1(xy) \log_2 p_1(y) - \sum_{X,Y} (1-\lambda) p_2(xy) \log_2 p_2(y) \\
&= \lambda \sum_{X,Y} p_1(xy) \log_2 \frac{p(y)}{p_1(y)} + (1-\lambda) \sum_{X,Y} p_2(xy) \log_2 \frac{p(y)}{p_2(y)}
\end{aligned}$$

由于 $f(x) = \log x$ 是一个∩形凸函数，故有 $E[f(x)] \leqslant f[E(x)]$，如图 3-7 所示。

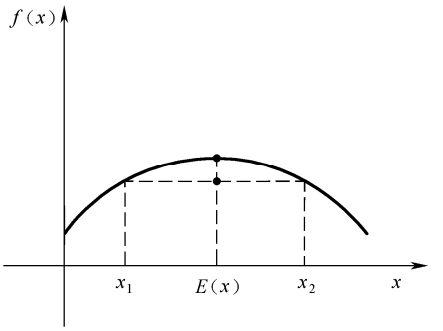


图 3-7

$$\begin{aligned}\sum_{X,Y} p_1(xy)\log_2\frac{p(y)}{p_1(y)} &\leqslant \log_2 p(xy)\frac{p(y)}{p_1(y)}\\ &= \log_2\sum_{Y}\frac{p(y)}{p_1(y)}\sum_{X}p_1(xy)\\ &= \log_2\sum_{Y}\frac{p(y)}{p_1(y)}p(y)\\ &= \log_2\sum_{Y}p(y)\\ &= 0\end{aligned}$$

所以，同理 $\sum_{X,Y} p_2(xy)\log_2\frac{p(y)}{p_2(y)}\leqslant 0$ 。

又因为 $0<\lambda<1$，所以

$$\lambda I\left[p_1(x)\right]+(1-\lambda)I\left[p_2(x)\right]-I\left[p(x)\right]\leqslant 0$$

$$I\left[\lambda p_1(x)+(1-\lambda)p_2(x)\right]\geqslant \lambda I\left[p_1(x)\right]+(1-\lambda)I\left[p_2(x)\right]$$

根据上凸函数定义可知，$I(X;Y)$ 是输入信源概率分布 $p(x)$ 的 $\cap$ 型上凸函数。

由前可知，$I(X;Y)$ 是输入随机变量 X 的概率分布 $p(x)$ 的 $\cap$ 形凸函数。因此对于一个固定的信道，总存在一种信源（某种概率分布 $p(x)$），使传输每个符号平均获得的信息量最大。也就是每个固定信道都有一个最大的信息传输率。定义这个最大的信息传输速率为信道容量 C，即

$$C=\max_{P(x)}\left\{I(X;Y)\right\} \tag{3-21}$$

其单位是比特/符号或奈特/符号，而相应的输入概率分布称为最佳输入分布。若平均传输一个符号需要 t 秒钟，则信道单位时间为平均传输的最大信息量为

$$C_t=\frac{1}{t}\max_{P(x)}\left\{I(X;Y)\right\}\ \text{（bit/s）} \tag{3-22}$$

一般仍称 C_t 为信道容量，增加一个下标 t 以示区别。

信道容量 C 与输入信源的概率分布无关，它只是信道传输概率的函数，只与信道的统计特性有关。因此，信道容量是完全描述信道特性的参量，是信道能够传输的最大信息量。

3.2 离散信道的容量及其计算

本节将分析无干扰离散信道、对称离散信道、准对称离散信道以及一般离散信道的信道容量及其计算方法。

3.2.1 无干扰离散信道

考虑一个离散无干扰信道，其输入字符集是 $X=\{x_0,x_1,\cdots,x_{q-1}\}$，输出字符集是 $Y=\{y_0,y_1,\cdots,y_{Q-1}\}$，转移概率 $p(y_j\mid x_i)$ 如（3-23）式的定义，它由信道特征决定。若给定信道，即信道的转移概率已定，则对应于输入符号的概率分布 $p(x_i)$ 可以求出相应的信道传输信息 $I(X;Y)$，即

$$p(Y=y_j\mid X=x_i)\equiv p(y_j\mid x_i) \tag{3-23}$$

$$I(X;Y)=\sum_{i=0}^{q-1}\sum_{j=0}^{Q-1}p(x_i)p(y_j\mid x_i)\log\frac{p(y_j\mid x_i)}{p(y_j)} \tag{3-24}$$

式中 $p(y_j)$ 可以利用下式计算得到，即

$$p(y_j)\equiv p(Y=y_j)=\sum_{i=0}^{q-1}p(x_i)p(y_j\mid x_i) \tag{3-25}$$

所以信道传输信息 $I(X;Y)$ 的大小由输入符号的概率分布 $p(x_i)$ 决定，其中最大值就定义为信道容量，用符号 C 来表示，即

$$C=\max I(X;Y)=\max\sum_{i=0}^{q-1}\sum_{j=0}^{Q-1}p(x_i)p(y_j\mid x_i)\log\frac{p(y_j\mid x_i)}{p(y_j)} \tag{3-26}$$

C 的单位是信道上每传送一个符号（每使用一次信道）所能携带的比特数，即比特/符号（bit/符号或 bit/信道）。当然以上 $I(X;Y)$ 值的最大化是在下列限制条件下进行的。

$$p(x_i)\geqslant 0$$

$$\sum_{i=0}^{q-1}p(x_i)=1 \tag{3-27}$$

如不是以 2 为底而以 e 为底取自然对数时，信道容量的单位变为奈特/符号（nat/符号）。如果已知符号传送周期是 T（秒），也可以“秒”为单位来计算信道容量，此时 $C_s=C/T$，以比特/秒（bit/s）或奈特/秒（nat/s）为信道容量单位。

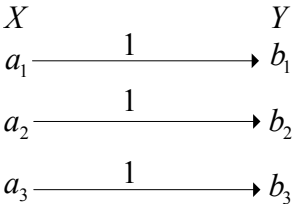
转移概率矩阵 $\boldsymbol{P}$ 已知后，由式（3-26）计算离散无干扰信道容量的关键是能找出使 $I(X;Y)$ 最大的 $p(x_i)(i=0,\cdots,q-1)$ 的概率分布。若将 $\boldsymbol{P}_x=\left[p(x_0),p(x_1),\cdots,p(x_{q-1})\right]$ 定义为输入符号的概率矢量 $\boldsymbol{P}_x$，由式（3-23）及关系式

$$I(X;Y)=H(X)-H(X\mid Y)=H(Y)-H(Y\mid X) \tag{3-28}$$

可得

$$C=\max_{\boldsymbol{P}_x}I(X;Y)=\max_{\boldsymbol{P}_x}\left[H(X)-H(X\mid Y)\right]=\max_{\boldsymbol{P}_x}\left[H(Y)-H(Y\mid X)\right] \tag{3-29}$$

例 3-2 一个离散无噪信道如下所示。



其信道矩阵是单位矩阵，即

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

因为对于离散无噪信道有

$$P(y_j \mid x_i) = P(y_i \mid x_j) = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} (i, j = 1,2,3)$$

因此满足

$$I(X;Y) = I(X) = I(Y)$$

信道容量为

$$C = \max_{p(x)} H(X) = \max_{p(y)} H(Y)$$

3.2.2 对称离散无记忆信道容量

如果转移概率矩阵 $\boldsymbol{P}$ 的每一行都是第一行的置换（包含同样元素），称该矩阵是输入对称的。如果转移概率矩阵 $\boldsymbol{P}$ 的每一列都是第一列的置换（包含同样元素），称该矩阵是输出对称的；如果输入、输出都对称，则称该离散无记忆信道（DMC）为对称的 DMC 信道。

例如：$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$ 和 $\begin{bmatrix} \frac{1}{2} & \frac{1}{3} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{2} \end{bmatrix}$ 都是对称的。

可以证明，有扰的对称 DMC 信道具有如下性质。

（1）对称信道的条件熵 $H(Y \mid X)$ 与信道输入符号的概率分布无关，且有 $H(Y \mid X) = H(Y \mid x_i), i = 0,1,\cdots,q-1$。

$$\begin{aligned} H(Y \mid X) &= -\sum_i p(x_i) \sum_j p(y_j \mid x_i) \log p(y_j \mid x_i) \\ &= -\sum_j p(y_j \mid x_i) \log p(y_j \mid x_i) \\ &= H(Y \mid x_i) \end{aligned}$$

（2）当信道输入符号等概分布时，信道输出符号也等概分布；反之，若信道输出符号等概分布，则输入符号必定也是等概分布。

（3）当信道输入符号等概分布时，对称 DMC 信道达到其信道容量，为

$$C = \log Q - H(Y \mid x_i) = \log Q + \sum_{j=1}^{Q} p_{ij} \log p_{ij} \tag{3-30}$$

由于对称信道的条件熵 $H(Y \mid X)$ 与信道输入符号的概率分布无关，式（3-29）转化成

$$\max_{\boldsymbol{P}_x}\left[H(Y)-H(Y\mid X)\right]=\max_{\boldsymbol{P}_x}\left[H(Y)-H(Y\mid x_i)\right]$$
$$=\max_{\boldsymbol{P}_x}\left[H(Y)\right]-H(Y\mid x_i)$$

于是问题就简化为 $\max\limits_{\boldsymbol{P}_x}\left[H(Y)\right]$。由信息论原理，当输出符号集的各符号等概出现时可得到最大信源熵，即

$$H(Y)\leqslant\log Q \quad \text{或者} \quad \max\left[H(Y)\right]=\log Q \tag{3-31}$$

这就是（3-30）式的来历。

例 3-3　信道转移概率矩阵为 $\boldsymbol{P}=\begin{bmatrix}\frac{1}{3}&\frac{1}{3}&\frac{1}{6}&\frac{1}{6}\\ \frac{1}{6}&\frac{1}{6}&\frac{1}{3}&\frac{1}{3}\end{bmatrix}$，代入式（3-30）求得信道容量为

$$C=\log 4-H(\frac{1}{3},\frac{1}{3},\frac{1}{6},\frac{1}{6})=0.082\text{bit/符号}$$

3.2.3　准对称离散无记忆信道容量

若信道矩阵 $\boldsymbol{M}$ 的列可以划分成若干个互不相交的子集 B_k，即 $B_1\cap B_2\cap\cdots\cap B_n=\varnothing$；$B_1\cup B_2\cup\cdots\cup B_n=Y$，由 B_k 为列组成的矩阵 $\boldsymbol{M}_k$ 是对称矩阵，则称信道矩阵 $\boldsymbol{M}$ 所对应的信道为准对称信道。

例如，信道矩阵

$$\boldsymbol{P}_1=\begin{bmatrix}\frac{1}{2}&\frac{1}{2}&\frac{1}{4}&\frac{1}{4}\\ \frac{1}{4}&\frac{1}{2}&\frac{1}{4}&\frac{1}{2}\end{bmatrix}\qquad \boldsymbol{P}_2=\begin{bmatrix}0.6&0.3&0.1\\ 0.3&0.1&0.6\end{bmatrix}$$

都是准对称信道。在信道矩阵 $\boldsymbol{P}_1$ 中 Y 可以划分成三个子集，由子集的列组成的矩阵为

$$\begin{bmatrix}\frac{1}{2}&\frac{1}{6}\\ \frac{1}{6}&\frac{1}{2}\end{bmatrix},\quad\begin{bmatrix}\frac{1}{2}\\ \frac{1}{2}\end{bmatrix},\quad\begin{bmatrix}\frac{1}{4}\\ \frac{1}{4}\end{bmatrix}$$

它们满足对称性，所以 $\boldsymbol{P}_1$ 所对应的信道为准对称信道。同理 $\boldsymbol{P}_2$ 可划分成

$$\begin{bmatrix}0.6&0.1\\ 0.1&0.6\end{bmatrix}\quad\text{和}\quad\begin{bmatrix}0.1\\ 0.1\end{bmatrix}$$

这两个子矩阵也满足对称性。

我们可以证明达到准对称离散信道容量的输入分布（最佳输入分布）是等概率分布，也可以计算准对称离散信道的信道容量为

$$C=\log r-H(p'_1,p'_2,\cdots,p'_s)-\sum_{k=1}^{n}N_k\log M_k \tag{3-32}$$

式中，r 是输入符号集的个数，$(p'_1,p'_2,\cdots,p'_s)$ 为准对称信道矩阵中的行元素。设矩阵可划分成 n 个互不相交的子集。N_k 是第 k 个子矩阵 $\boldsymbol{M}_k$ 中行元素之和，$\boldsymbol{S}_k$ 是第 k 个子矩阵 $\boldsymbol{M}_k$ 中列元素之和，即

$$\boldsymbol{N}_k=\sum_{y\in Y_k}p(y\mid x_i) \tag{3-33}$$

$$\boldsymbol{S}_k=\sum_X p(y\mid x_i)\quad y\in\boldsymbol{Y}_k\quad (k=1,2,\cdots,n) \tag{3-34}$$

当输入等概率分布时，式（3-33）和式（3-34）都与 x 无关。

例 3-4 已知一个信道的信道转移矩阵为 $\boldsymbol{P}=\begin{bmatrix}0.5 & 0.3 & 0.2\\ 0.3 & 0.5 & 0.2\end{bmatrix}$，求该信道的容量。

解：由 $\boldsymbol{P}$ 可看出信道的输入符号有两个，可设 $p(x_1)=\alpha$，$p(x_2)=1-\alpha$。信道的输出符号有三个，用 y_1，y_2，y_3 表示。由 $p(x_iy_j)=p(x_i)p(y_j\mid x_i)$ 得联合概率的矩阵为

$$\begin{bmatrix}0.5\alpha & 0.3\alpha & 0.2\alpha\\ 0.3(1-\alpha) & 0.5(1-\alpha) & 0.2(1-\alpha)\end{bmatrix}$$

由 $p(y_j)=\sum_i p(x_iy_j)$ 得

$$p(y_1)=0.5\alpha+0.3(1-\alpha)=0.3+0.2\alpha$$
$$p(y_2)=0.3\alpha+0.5(1-\alpha)=0.5-0.2\alpha$$
$$p(y_3)=0.2\alpha+0.2(1-\alpha)=0.2$$

其中，$p(y_3)$ 恒定，与 x_i 分布无关。

$$\begin{aligned}I(X;Y)&=H(Y)-H(Y\mid X)\\&=-\sum_j p(y_j)\ln p(y_j)+\sum_i p(x_i)\sum_j p(y_j\mid x_i)\ln p(y_j\mid x_i)\\&=-(0.3+0.2\alpha)\ln(0.3+0.2\alpha)-(0.5-0.2\alpha)\ln(0.5-0.2\alpha)+0.5\ln 0.5+0.3\ln 0.3\end{aligned}$$

由 $\dfrac{\partial I(X;Y)}{\partial\alpha}=0$ 得 $0.2\ln(0.3+0.2\alpha)-0.2+0.2\ln(0.5-0.2\alpha)+0.2=0$。

解得 $\alpha=1/2$，即输入符号等概分布时，$I(X;Y)$ 达到极大值。所以信道容量为

$$C=\max I(X;Y)=0.036\text{bit/符号}$$

此时输出符号的概率为 $p(y_1)=p(y_2)=0.4$，$p(y_3)=0.2$。

3.2.4 一般离散无记忆信道容量

根据信道容量的定义，在固定信道的条件下，对所有可能的输入概率分布 $p(x)$ 求平均互信息的极大值。由前已知，$I(X;Y)$ 是输入概率分布 $p(x)$ 的∩形凸函数，所以极大值一定存在。而 $I(X;Y)$ 是 r 个变量 $\{p(a_1),p(a_2),\cdots,p(a_r)\}$ 的多元函数，并满足 $\sum_{i=1}^{r}p(a_i)=1$，所以可以运用拉格朗日乘子法来计算这个条件极值。

引进一个新函数

$$\phi=I(X;Y)-\lambda\sum_X p(a_s) \tag{3-35}$$

式中，λ 为拉格朗日乘子（待定常数）。解方程组

$$\begin{cases}\dfrac{\partial\phi}{\partial p(a_i)}=\dfrac{\partial\left[I(X;Y)-\lambda\sum_X p(a_s)\right]}{\partial p(a_i)}=0\\ \sum_X p(a_s)=1\end{cases} \tag{3-36}$$

可先求解出达到极值的概率分布和拉格朗日乘子 λ 的值，然后再求解出信道容量 C。

因为

$$I(X;Y)=\sum_{i=1}^{r}\sum_{j=1}^{s}p(a_i)p(b_j\mid a_i)\log\frac{p(b_j\mid a_i)}{p(b_i)}$$

而

$$p(b_j)=\sum_{i=1}^{r}p(a_i)p(b_j\mid a_i)$$

所以

$$\frac{\partial}{\partial P(a_i)}\log p(b_j)=\left[\frac{\partial}{\partial p(a_i)}\ln p(b_j)\right]\log\mathrm{e}=\frac{p(b_j\mid a_i)}{p(b_j)}\log\mathrm{e}$$

解方程式（3-36）中第一个方程式，得

$$\sum_{j=1}^{s}p(b_j\mid a_i)\log\frac{p(b_j\mid a_i)}{p(b_j)}-\sum_{k=1}^{r}\sum_{j=1}^{s}p(a_k)p(b_j\mid a_i)\frac{p(b_j\mid a_i)}{p(b_j)}\log\mathrm{e}-\lambda=0$$

（对$i=1,2,\cdots,r$都成立） （3-37）

注意，式（3-37）中的对数是取任意大于 1 的数为底。

又因为

$$\sum_{k=1}^{r}p(a_k)p(b_j\mid a_k)=p(b_j)$$

$$\sum_{j=1}^{s}p(b_j\mid a_i)=1\qquad i=1,2,\cdots,r$$

所以方程组式（3-36）变换成

$$\begin{cases}\displaystyle\sum_{j=1}^{s}p(b_j\mid a_i)\log\frac{p(b_j\mid a_i)}{p(b_j)}=\lambda+\log\mathrm{e}\\ \displaystyle\sum_{i=1}^{r}p(a_i)=1\end{cases}\qquad(3\text{-}38)$$

假设解得使平均互信息 $I(X;Y)$ 达到极值的输入概率分布是 $\{p_1,p_2,\cdots,p_r\}$（简写成 $\{p_i\}$ 或 $\boldsymbol{P}$）。然后把（3-38）中前 r 个方程式两边分别乘以达到极值的输入概率 p_i，并求和得

$$\sum_{i=1}^{r}\sum_{j=1}^{s}p_ip(b_j\mid a_i)\log\frac{p(b_j\mid a_i)}{p(b_j)}=\lambda+\log\mathrm{e}$$

上式左边即是信道容量，所以得

$$C=\lambda+\log\mathrm{e}\qquad(3\text{-}39)$$

现令

$$I(x_i;Y)=\sum_{j=1}^{s}P(b_j\mid a_i)\log\frac{P(b_j\mid a_i)}{P(b_j)}\qquad(3\text{-}40)$$

式中，$I(x_i;Y)$ 是输出端接收到 Y 后获得关于 $x=a_i$ 的信息量，即是信源符号 $x=a_i$ 对输出端 Y

平均提供的互信息。上式中对数取不同的底就得到相应不同的单位。一般来讲，$I(x_i;Y)$ 的值与 x_i 有关。根据式（3-38）和式（3-39）得

$$I(x_i;Y)=C \qquad (i=1,2,\cdots,r)$$

所以，对于一般离散信道有下述的定理。

定理 3-2 一般离散信道的平均互信息 $I(X;Y)$ 达到极大值（即等于信道容量）的充要条件是输入概率分布 $\{p_i\}$ 满足

$$\begin{cases}(a)\ I(x_i;Y)=C & \text{对所有}x_i\text{其中}p_i\neq 0\\ (b)\ I(x_i;Y)\leqslant C & \text{对所有}x_i\text{其中}p_i=0\end{cases} \tag{3-41}$$

这时 C 就是所求的信道容量。

因为

$$\frac{\partial I(X;Y)}{\partial p_i}=I(x_i;Y)-\log e \quad (i=1,2,\cdots,r)$$

又根据式（3-39），所以定理 3-2 中的充要条件(a)与(b)可改写成

$$\begin{cases}(a)\ \dfrac{\partial I(X;Y)}{\partial p_i}=\lambda & \text{对所有}x_i\text{其中}p_i\neq 0\\ (b)\ \dfrac{\partial I(X;Y)}{\partial p_i}\leqslant\lambda & \text{对所有}x_i\text{其中}p_i=0\end{cases} \tag{3-42}$$

从定理 3-2 可以得出这样一个结论：当信道平均互信息达到信道容量时，输入信源符号集中每一个信源符号 x 对输出端 Y 提供相同的互信息，只是概率为零的符号除外。这个结论和直观概念是一致的。在某给定的输入分布下，若有一个输入符号 $x=a_i$ 对输出 Y 所提供的平均互信息比其他输入符号所提供的平均互信息大，那么，我们就可以更多地使用这一符号来增大平均互信息 $I(X;Y)$。但是，这就会改变输入符号的概率分布，必然使这个符号的平均互信息 $I(x_i;Y)$ 减小，而其他符号对应的平均互信息增加。所以，经过不断调整输入符号的概率分布，就可使每个概率不为零的输入符号对输出 Y 提供相同的平均互信息。

定理 3-2 只给出了达到信道容量时，最佳输入概率分布应满足的条件。并没有给出输入符号的最佳概率分布值，因而也没有给出信道容量的数值。另外，定理本身也隐含着，达到信道容量的最佳分布并不一定是唯一的。在一些特殊情况下，我们常常可以利用这一定理来找出所求的输入概率分布和信道容量。

例 3-5 设离散无记忆信道输入序列 $X=\{x_1,x_2,x_3\}$，输出序列 $Y=\{y_1,y_2\}$，其信道转移矩阵为

$$\boldsymbol{P}=\begin{bmatrix}1 & 0\\ 0.5 & 0.5\\ 0 & 1\end{bmatrix}$$

求信道容量和最佳分布。

解：根据矩阵中元素 $p(y|x)$ 的分布，可以设想最佳分布为 $p(x_1)=0.5$，$p(x_2)=0$，$p(x_3)=0.5$，这样信道的输入和输出就构成了一一对应的关系，如图 3-8 所示，接收方收到输出符号集 Y 中的某个元素后，对发送方发送的是输入符号集 X 中的哪个元素是完全确定的，即 $H(X|Y)=0$。

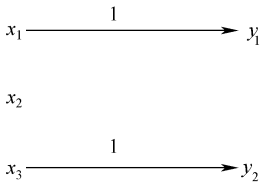


图 3-8　取 $p(x_2)=0$ 时信道输入和输出的对应关系

若 $p_2 \neq 0$，则接收方收到 $Y=\{y_1,y_2\}$ 中的某个符号后，关于发送的是 $X=\{x_1,x_2,x_3\}$ 中的哪个符号就会增加不确定性，即 $H(X|Y)\neq 0$，从平均互信息量的表达式 $I(X;Y)=H(X)-H(X|Y)$ 可看出，由于 $H(X|Y)$ 的非负性，这样得到的平均互信息量就会减少，所以直观地看，按设想的分布得到 $I(X;Y)$ 应取最大值。

下面还要验证一下上述分布是否满足定理 3-2 所要求的充要条件，先算出

$$\begin{cases} p(y_1)=\sum\limits_i p(x_i)p(y_1|x_i)=0.5\times 1=0.5 \\ p(y_2)=\sum\limits_i p(x_i)p(y_2|x_i)=0.5\times 1=0.5 \end{cases}$$

$$\begin{cases} I(x_1;Y)=\sum\limits_j P(y_j|x_1)\log\dfrac{p(y_j|x_1)}{p(y_j)}=1\times\log\dfrac{1}{0.5}+0\times\log\dfrac{0}{0.5}=\log 2 \\ I(x_2;Y)=\sum\limits_j P(y_j|x_2)\log\dfrac{p(y_j|x_2)}{p(y_j)}=0.5\times\log\dfrac{0.5}{0.5}+0.5\times\log\dfrac{0.5}{0.5}=0 \\ I(x_3;Y)=\sum\limits_j P(y_j|x_3)\log\dfrac{p(y_j|x_3)}{p(y_j)}=0\times\log\dfrac{0}{0.5}+1\times\log\dfrac{1}{0.5}=\log 2 \end{cases}$$

显然满足定理 3-2 的充要条件为 $\begin{cases} I(x_1;Y)=I(x_3;Y)=\log 2 & p(x_1)\neq 0 \ \ p(x_3)\neq 0 \\ I(x_2;Y)=0 & p(x_2)=0 \end{cases}$，故信道容量 $C=\log 2=1$ bit/符号。

3.3　离散序列信道及其容量

前几节讨论了最简单的离散信道，即信道的输入和输出只是单个随机变量的信道。然而一般离散信道的输入和输出却是一系列时间（或空间）离散的随机变量，即随机序列。若输入或输出随机序列中每一个随机变量都取值于同一输入或输出的符号集，这种离散信道的数学模型如图 3-9 所示。

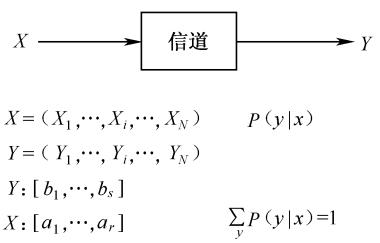


图 3-9　离散信道模型

现在，我们来着重研究离散序列信道，这种信道的传递概率满足

$$P(\boldsymbol{y}|\boldsymbol{x})=P(y_1y_2\cdots y_N|x_1x_2\cdots x_N)=\prod_{i=1}^{N}P(y_i|x_i) \tag{3-43}$$

因此，离散无记忆信道的数学模型基本与单符号离散信道的数学模型相同，仍用$[X,P(y|x),Y]$概率空间来描述。而不同的只是当信道传输消息序列时，输入随机序列与输出随机序列之间的传递概率等于对应时刻的随机变量的传递概率的乘积。

在离散无记忆信道中，为了便于研究传递消息序列所能获得的信息量，我们从离散无记忆信道的N次扩展信道入手。

设离散无记忆信道的输入符号集$A=\{a_1,\cdots,a_r\}$，输出符号集$B=\{b_1,\cdots,b_s\}$，信道矩阵为

$$\boldsymbol{P}=\begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1s} \\ p_{21} & p_{22} & \cdots & p_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ p_{r1} & p_{r2} & \cdots & p_{rs} \end{bmatrix}$$

且满足

$$\sum_{j=1}^{s} p_{ij}=1 \qquad (i=1,2,\cdots,r)$$

则此无记忆信道的N次扩展信道的数学模型如图 3-10 所示。

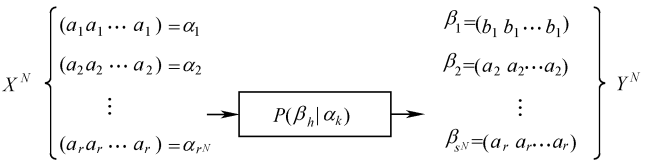


图 3-10 N次扩展信道

因为在输入随机序列$\boldsymbol{X}=(X_1,X_2,\cdots,X_N)$中，每一个随机变量$X_i$ $(i=1,2,\cdots,N)$各取值于同一输入符号集A，而符号集A共有r个符号，所以随机矢量$\boldsymbol{X}$的可能取值r^N个。同理，随机矢量$\boldsymbol{Y}$的可能取值有s^N个。根据信道无记忆的特性，可得N次扩展信道的信道矩阵

$$\boldsymbol{\pi}=\begin{bmatrix} \pi_{11} & \pi_{12} & \cdots & \pi_{1s^N} \\ \pi_{21} & \pi_{22} & \cdots & \pi_{2s^N} \\ \vdots & \vdots & \ddots & \vdots \\ \pi_{r^N1} & \pi_{r^N2} & \cdots & \pi_{r^Ns^N} \end{bmatrix}$$

式中

$$\pi_{kh}=P(\beta_h|\alpha_k) \qquad (k=1,2,\cdots,r^N,\ h=1,2,\cdots,s^N)$$

并满足

$$\sum_{h=1}^{s^N}\pi_{kh}=1 \qquad (k=1,2,\cdots,r^N)$$

而

$$\alpha_k=\left(a_{k_1}a_{k_2}\cdots a_{k_N}\right) \qquad a_{k_i}\in\{a_1,\cdots,a_r\} \qquad (i=1,\cdots,N)$$

$$\beta_h=\left(b_{h_1}b_{h_2}\cdots b_{h_N}\right) \qquad b_{h_i}\in\{b_1,\cdots,b_r\} \qquad (i=1,\cdots,N)$$

所以得

$$\begin{aligned}\pi_{kh} &= P(\beta_h \mid \alpha_k) = P(b_{h_1} b_{h_2} \cdots b_{h_N} \mid a_{k_1} a_{k_2} \cdots a_{k_N}) \\ &= \prod_{i=1}^{N} P(b_{h_i} \mid a_{k_i}) \qquad (k = 1,2,\cdots,r^N \quad h = 1,2,\cdots,s^N)\end{aligned} \tag{3-44}$$

根据平均互信息的定义，可得无记忆信道的 N 次扩展信道的平均互信息，即

$$\begin{aligned}I(\boldsymbol{X};\boldsymbol{Y}) &= I(X^N;Y^N) = H(X^N) - H(X^N \mid Y^N) = H(Y^N) - H(Y^N \mid X^N) \\ &= \sum_{X^N,Y^N} P(\alpha_k \beta_h) \log \frac{P(\alpha_k \mid \beta_h)}{P(\alpha_k)} = \sum_{X^N,Y^N} P(\alpha_k \beta_h) \log \frac{P(\beta_h \mid \alpha_k)}{P(\beta_h)}\end{aligned} \tag{3-45}$$

在一般离散信道中，信道输入和输出两个离散随机序列之间的平均互信息 $I(\boldsymbol{X};\boldsymbol{Y})$ 与两序列中对应的离散随机变量之间的平均互信息 $I(X_i;Y_i)$ 存在着以下关系。

在图 3-9 所示的一般离散信道模型中，设 $\boldsymbol{X} = (X_1,\cdots,X_i,\cdots,X_N)$，$\boldsymbol{Y} = (Y_1,\cdots,Y_i,\cdots,Y_N)$，其中 $X_i \in A = \{a_1,a_2,\cdots,a_r\}, Y_i \in B = \{b_1,b_2,\cdots,b_s\}$，且有

$$\boldsymbol{x} \in \boldsymbol{X}, \boldsymbol{y} \in \boldsymbol{Y}, x_i \in X_i, y_i \in Y_i$$

（1）当信道是无记忆的，即信道传递概率满足 $P(\boldsymbol{y} \mid \boldsymbol{x}) = \prod_{i=1}^{N} P(y_i \mid x_i)$ 有

$$I(\boldsymbol{X};\boldsymbol{Y}) \leqslant \sum_{i=1}^{N} I(X_i;Y_i) \tag{3-46}$$

（2）当信道的输入信源是无记忆的，即满足 $P(\boldsymbol{x}) = \prod_{i=1}^{N} P(x_i)$ 有 （3-47）

$$I(\boldsymbol{X};\boldsymbol{Y}) \geqslant \sum_{i=1}^{N} I(X_i;Y_i) \tag{3-48}$$

（3）当信道和信源都是无记忆的时，即式（3-43）和式（3-47）两条件都满足，有

$$I(\boldsymbol{X};\boldsymbol{Y}) = \sum_{i=1}^{N} I(X_i;Y_i) \tag{3-49}$$

若信道的输入序列 $\boldsymbol{X} = (X_1X_2\cdots X_N)$ 中随机变量 $X_i (i = 1,2,\cdots,N)$ 不但取自同一信源符号集，并且具有同一种概率分布，而且通过相同的信道传送到输出端（即信道传递概率分布不随 i 而改变，为时不变信道）因此有

$$I(X_1;Y_1) = I(X_2;Y_2) = \cdots = I(X_N;Y_N) = I(X;Y)$$

得

$$\sum_{i=1}^{N} I(X_i;Y_i) = NI(X;Y) \tag{3-50}$$

式中，N 是序列的长度。所以，对于此离散无记忆信道的 N 次扩展信道来说，则有

$$I(\boldsymbol{X};\boldsymbol{Y}) \leqslant NI(X;Y) \tag{3-51}$$

若输入信源也是无记忆的话，则有

$$I(\boldsymbol{X};\boldsymbol{Y}) = NI(X;Y)$$

此式说明当信源是无记忆时，无记忆的 N 次扩展信道的平均互信息等于原来信道的平均互信息的 N 倍。

对于一般的离散无记忆信道的 N 次扩展信道，因为有

$$I(\boldsymbol{X};\boldsymbol{Y}) \leqslant \sum_{i=1}^{N} I(X_i;Y_i)$$

所以其信道容量

$$\begin{aligned} C^N &= \max_{P(\boldsymbol{x})} I(\boldsymbol{X};\boldsymbol{Y}) \\ &= \max_{P(\boldsymbol{x})} \sum_{i=1}^{N} I(X_i;Y_i) = \sum_{i=1}^{N} \max_{P(x_i)} I(X_i;Y_i) \\ &= \sum_{i=1}^{N} C_i \end{aligned} \tag{3-52}$$

式中，令 $C_i = \max\limits_{P(x_i)} I(X_i;Y_i)$，这是某时刻 i 通过离散无记忆信道传输的最大信息量。根据前节所述可求出离散无记忆信道的信道容量 C。因为现在输入随机序列 $\boldsymbol{X} = (X_1 X_2 \cdots X_N)$ 在同一信道中传输（也可以认为在时不变信道中传输），所以得 $C_i = C$ $(i = 1,2,\cdots,N)$。即任何时刻通过离散无记忆信道传输的最大信息量都相同。代入式（3-52）得

$$C^N = NC \tag{3-53}$$

此式说明离散无记忆的 N 次扩展信道的信道容量等于原单符号离散信道的信道容量的 N 倍。且只有当输入信源是无记忆的及每一输入变量 X_i 的分布各自达到最佳分布 $P(x)$ 时，才能达到这个信道容量 NC。

一般情况下，消息序列在离散无记忆的 N 次扩展信道中传输的信息量为

$$I(\boldsymbol{X};\boldsymbol{Y}) \leqslant NC \tag{3-54}$$

3.4 独立并联信道及其容量

一般独立并联如图 3-11 所示。

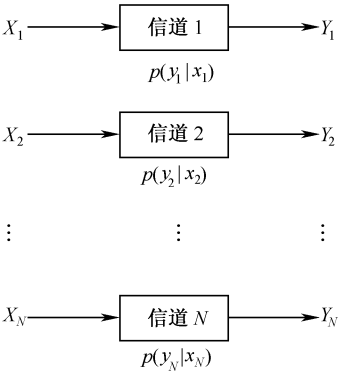


图 3-11　N个独立并联信道

独立并联信道又称并用信道，也等价于时变的 N 次无记忆扩展信道。设有 N 个信道，它们的输入分别是 $X_1, X_2, \cdots, X_N$，它们的输出分别是 $Y_1, Y_2, \cdots, Y_N$，它们的传递概率分别是 $p(y_1|x_1), p(y_2|x_2), \cdots, p(y_N|x_N)$。在这 N 个独立并联信道中，每一个信道的输出 Y_i 只与本信道的输入 X_i 有关，与其他信道输入、输出都无关。那么，这 N 个信道的联合传递概率满足

$$p(y_1y_2\cdots y_N \mid x_1x_2\cdots x_N)=p(y_1\mid x_1)p(y_2\mid x_2)\cdots p(y_N\mid x_N) \tag{3-55}$$

相当于信道是无记忆时应满足的条件。因此，对于 N 个独立并联信道有

$$I(X_1X_2\cdots X_N;Y_1Y_2\cdots Y_N)\leqslant\sum_{i=1}^{N}I(X_i;Y_i) \tag{3-56}$$

即联合平均互信息不大于各信道的平均互信息之和。

因此得独立并联信道的信道容量

$$C_{1,2,\cdots,N}=\max_{P(x_1\cdots x_N)}I(X_1\cdots X_N;Y_1\cdots Y_N)\leqslant\sum_{i=1}^{N}C_i \tag{3-57}$$

式中，C_i 是各个独立信道的信道容量，即

$$C_i=\max_{P(x_i)}I(X_i;Y_i)$$

所以，独立并联信道的信道容量不大于各个信道的信道容量之和。只有当输入符号 X_i 相互独立，且输入符号 X_i 的概率分布达到各信道容量的最佳输入分布时，独立并联信道的信道容量才等于各信道容量之和，即

$$C_{1,2,\cdots,N}=\sum_{i=1}^{N}C_i \tag{3-58}$$

3.5　串联信道容量及数据处理定理

在一些实际通信系统中常常出现串联信道的情况，如微波中继接力通信就是一种串联信道。本节将研究串联信道的容量。

假设有以离散单符号信道 I，其输入变量为 X，取值 $\{a_1,a_2,\cdots,a_r\}$，输出变量 Y，取值 $\{b_1,b_2,\cdots,b_s\}$；并设另一离散单符号信道 II，其输入变量为 Y，输出变量 Z，取值 $\{c_1,c_2,\cdots,c_t\}$。这两信道串联起来，如图 3-12 所示。这两个信道的输入和输出符号集都是完备集。信道 I 的传递概率是 $p(y\mid x)=p(b_j\mid a_i)$，而信道 II 的传递概率一般与前面的符号 X 和 Y 都有关，所以记为 $p(z\mid xy)=p(c_k\mid a_ib_j)$。

若信道 II 的传递概率使其输出 Z 只与输入 Y 有关，与前面的输入 X 无关，即满足

$$p(z\mid yx)=p(z\mid y)\qquad(\text{对所有}x,y,z) \tag{3-59}$$

称这两信道的输入和输出 X，Y，Z 序列构成马尔可夫链。

这两个串联信道可以等价成一个总的离散信道如图 3-13 所示，其输入为 X，取值 $\{a_1,a_2,\cdots,a_r\}$，输出为 Z，取值 $\{c_1,c_2,\cdots,c_t\}$，此信道的传递概率为

$$p(z\mid x)=\sum_{Y}p(y\mid x)\,p(z\mid xy)\qquad(x\in X,y\in Y,z\in Z)$$

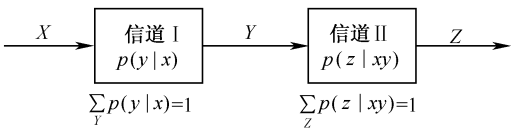


图 3-12　串联信道

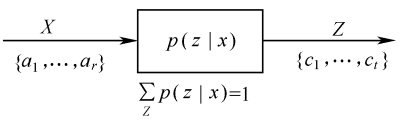


图 3-13　等价的总信道

则总信道的传递矩阵

$$\underset{r\times t}{[p(z|x)]}=\underset{r\times s}{[p(y|x)]}\cdot\underset{s\times t}{[p(z|xy)]} \tag{3-60}$$

若 X，Y，Z 满足马尔可夫链，得总信道的传递概率

$$p(z|x)=\sum_{Y}p(y|x)\cdot p(z|y)\qquad (x\in X,y\in Y,z\in Z) \tag{3-61}$$

信道矩阵为

$$\underset{r\times t}{[p(z|x)]}=\underset{r\times s}{[p(y|x)]}\cdot\underset{s\times t}{[p(z|y)]} \tag{3-62}$$

下面我们先讨论串联信道中平均互信息 $I(X;Y)$、$I(X;Z)$ 和 $I(Y;Z)$ 之间的关系。

定理 3-3 离散串联信道中，平均互信息满足

$$I(XY;Z)\geqslant I(Y;Z) \tag{3-63}$$

$$I(XY;Z)\geqslant I(X;Z) \tag{3-64}$$

当且仅当对所有 x，y，z，满足

$$p(z|xy)=p(z|y) \tag{3-65}$$

$$\text{或 } p(z|xy)=p(z|x) \tag{3-66}$$

时，式（3-63）或式（3-64）中等号成立。

上式 $I(XY;Z)$ 表示联合变量 XY 与变量 Z 之间的平均互信息，也就是接收到 Z 后获得关于联合变量 X 和 Y 的信息量。而 $I(Y;Z)$ 是接收到 Z 后获得关于变量 Y 的信息量。$I(X;Z)$ 是接收到 Z 后获得关于变量 X 的信息量。

证明 根据平均互信息的定义得

$$I(XY;Z)=\sum_{X,Y,Z}p(xyz)\log\frac{p(z|xy)}{p(z)}=E\left[\log\frac{p(z|xy)}{p(z)}\right] \tag{3-67}$$

$$I(Y;Z)=\sum_{Y,Z}p(yz)\log\frac{p(z|y)}{p(z)}=\sum_{X,Y,Z}p(xyz)\log\frac{p(z|y)}{p(z)}=E\left[\log\frac{p(z|y)}{p(z)}\right] \tag{3-68}$$

在式（3-67）和式（3-68）中，$E[\cdot]$ 都是对 X，Y，Z 三个概率空间求均值。所以得

$$\begin{aligned}I(Y,Z)-I(XY,Z)&=E\left[\log\frac{p(z|y)}{p(z)}-\log\frac{p(z|xy)}{p(z)}\right]\\&=E\left[\log\frac{p(z|y)}{p(z|xy)}\right]\end{aligned} \tag{3-69}$$

由于 $\log x$ 为 $\cap$ 形上凸函数，得

$$\begin{aligned}E\left[\log\frac{p(z|y)}{p(z|xy)}\right]&\leqslant\log E\left[\frac{p(z|y)}{p(z|xy)}\right]\\&=\log\sum_{X,Y,Z}p(xyz)\frac{p(z|y)}{p(z|xy)}=\log\sum_{X,Y,Z}p(xy)p(z|y)\\&=\log\sum_{X,Y}p(xy)\sum_{Z}p(z|y)=\log\sum_{X,Y}p(xy)=\log 1=0\end{aligned}$$

其中，因为已知

$$\sum_{Z} p(z \mid xy) = 1$$

$$\sum_{Y} p(y \mid x) = 1 \text{ 及 } \sum_{X,Y} p(xy) = 1 \quad (x \in X \quad y \in Y \quad z \in Z)$$

所以可得

$$\sum_{X} p(x \mid y) = 1$$

$$\sum_{Z} p(z \mid y) = \sum_{Z}\sum_{X} p(xz \mid y) = \sum_{X}\sum_{Z} p(z \mid xy)p(x \mid y) = 1 \qquad (y \in Y)$$

因此证得

$$I(XY;Z) \geqslant I(Y;Z) \tag{3-70}$$

当且仅当 $p(z \mid xy) = p(z \mid y)$（对所有 x，y，z）时，式（3-69）才等于零，因而式（3-70）的等号才成立。

同理，可以证得

$$I(XY;Z) \geqslant I(X;Z) \tag{3-71}$$

当且仅当 $p(z \mid xy) = p(z \mid x)$（对所有 x，y，z 都成立）时，等式成立。 [证毕]

定理 3-3 中等式成立的条件 $p(z \mid xy) = p(z \mid y)$ 表示随机变量 Z 只依赖于变量 Y，与前面的变量 X 无直接关系。也就是说，随机变量 X、Y、Z 组成一个马尔可夫链。在一般串联信道中，随机变量 Z 往往只依赖于信道 II 的输入 Y，不直接与变量 X 发生关系，即随机变量 Z 仅仅通过变量 Y 而依赖于 X。所以串联信道的输入和输出变量之间组成一个马尔可夫链，并存在下述定理。

定理 3-4 若 X、Y、Z 组成一个马尔可夫链，则有

$$I(X;Z) \leqslant I(X;Y) \tag{3-72}$$

$$I(X;Z) \leqslant I(Y;Z) \tag{3-73}$$

证明 首先证明式（3-71）。

因为 X、Y、Z 是马尔可夫链，所以满足 $p(z \mid xy) = p(z \mid y)$（对所有 x，y，z）；则定理 3-3 中等式成立，得

$$I(XY;Z) \geqslant I(Y;Z)$$

而又因式（3-71）成立，所以得

$$I(X;Z) \leqslant I(Y;Z)$$

其中等式成立的条件是

$$p(z \mid xy) = p(z \mid x) \quad \text{（对所有 } x, y, z\text{）} \tag{3-74}$$

再证式（3-72）。

因为 X、Y、Z 是马尔可夫链，可证得 Z、Y、X 也是马尔可夫链，所以有

$$p(x \mid yz) = p(x \mid y) \quad \text{（对所有 } x, y, z\text{）} \tag{3-75}$$

运用与定理 3-4 同样的证明方法，可得

$$I(ZY;X) \geqslant I(Y;X) \tag{3-76}$$

当且仅当 $P(x|yz)=P(x|y)$（对所有 x，y，z 都成立）时，式（3-76）的等号成立。另又可证得

$$I(ZY;X)\geqslant I(Z;X) \tag{3-77}$$

当且仅当 $P(x|yz)=P(x|z)$（对所有 x，y，z 都成立）时，式（3-77）的等号成立。

现因为式（3-75）成立，所以得

$$I(ZY;X)=I(Y;X)$$

再与式（3-77）联立，得

$$I(Y;X)\geqslant I(Z;X)$$

根据平均互信息的交互性，有

$$I(X;Y)\geqslant I(X;Z)$$

当且仅当 $P(x|yz)=P(x|y)=P(x|z)$（对所有 x，y，z 都成立）时，等式成立。由此得证式（3-72）。 [证毕]

由式（3-72）可以推得，在串联信道中一般有

$$H(X|Z)\geqslant H(X|Y) \tag{3-78}$$

定理 3-4 是很重要的。式（3-72）和式（3-78）表明通过串联信道的传输只会丢失更多的信息。如果满足

$$p(x|y)=p(x|z)\quad（对所有\ x，y，z\ 都成立） \tag{3-79}$$

即串联信道的总的信道矩阵等于第一级信道矩阵时，这个条件显然是满足的。如果第二个信道是数据处理系统，定理 3-4 就表明通过数据处理后，一般只会增加信息的损失，最多保持原来获得的信息，不可能比原来获得的信息有所增加。也就是说，对接收到的数据 Y 进行处理后，无论变量 Z 是 Y 的确定对应关系还是概率关系，绝不会减少关于 X 的不确定性。若要使数据处理后获得的关于 X 的平均互信息保持不变，必须满足式（3-79）。故定理 3-4 称为数据处理定理。

因此，对于一系列不涉及原始信源的数据处理，即对于一系列串联信道，如图 3-4 所示，有

$$H(X)\geqslant I(X;Y)\geqslant I(X;Z)\geqslant I(X;W)\geqslant\cdots \tag{3-80}$$

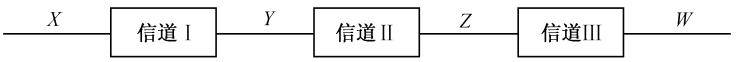


图 3-14 一系列串联信道

式（3-80）是定理 3-4 的推广，就是数据处理定理。它说明，在任何信息传输系统中，最后获得的信息至多是信源所提供的信息。如果一旦在某一过程中丢失一些信息，以后的系统不管如何处理，如不触及到丢失信息过程的输入端，就不能再恢复已丢失的信息。这就是信息不增性原理，它与热熵不减原理正好对应。这一点对于理解信息的实质具有深刻的物理意义。

根据信道容量的定义，串联信道的信道容量为

$$\left.\begin{aligned} C_{串}(\mathrm{I,II}) &= \max_{P(x)} I(X;Z) \\ C_{串}(\mathrm{I,II,III}) &= \max_{P(x)} I(X;W) \\ &\cdots \end{aligned}\right\} \tag{3-81}$$

由式（3-80）可知：串联的无源数据处理信道越多，其信道容量（最大信息传输）可能会越小，当串联信道数无限大时，信道容量就有可能趋于零。

3.6 连续信道及其容量

正如上一章所述连续信源情况下，在取两个相对熵之差时具有与离散信源一样的信息特征。互信息即是两熵之差，互信息的最大值就是信道容量，因而连续信道具有与离散信道类似的信息传输率和信道容量表达式。下面介绍加性噪声信道。

3.6.1 连续单符号加性信道

最简单、最常见的就是幅度连续的单符号信道，如图 3-15 所示。信道输入和输出都是取值连续的一维随机变量，加入信道的噪声是均值为零、方差为 σ^2 的加性高斯噪声，概率密度函数记作 $p_n(n)=N(0,\sigma^2)$。根据第 2 章所述，该噪声的连续熵为 $H_c(n)=\frac{1}{2}\log 2\pi e\sigma^2$。

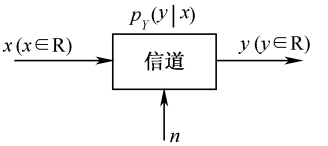


图 3-15 连续单符号信道

单符号连续信道的平均互信息为

$I(X;Y)=H_c(X)-H_c(X|Y)=H_c(Y)-H_c(Y|X)=H_c(X)+H_c(Y)-H_c(XY)$，信息传输率为 $R=I(X;Y)$ bit/符号。信道容量为

$$C=\max_{p(x)} I(X;Y)=\max_{p(x)}\left[H_c(Y)-H_c(Y|X)\right] \tag{3-82}$$

对于加性噪声信道，容易证明 $H_c(Y|X)=H_c(n)$，则有

$$C=\max_{p(x)} H_c(Y)-H_c(n)=\max_{p(x)} H_c(Y)-\frac{1}{2}\log 2\pi e\sigma^2 \tag{3-83}$$

要求式（3-83）第一项最大，由限平均功率最大熵定理，只有当信道输出 Y 正态分布时熵最大，其概率密度函数 $p_Y(y)=N(0,P)$，其中 P 为 Y 的平均功率限制值。由于信道输入 X 与噪声统计独立，且 $y=x+n$，所以其功率可以相加，即 $P=S+\sigma^2$，S 为信道输入 X 的平均功率值。

因为 $p_Y(y)=N(0,P)$，$p_n(n)=N(0,\sigma^2)$，$y=x+n$，所以 $p_X(x)=N(0,S)$，即当信道输入 X 是均值为 0、方差为 S 的高斯分布随机变量时，信息传输率达到最大值

$$C=\frac{1}{2}\log 2\pi eP-\frac{1}{2}\log 2\pi e\sigma^2=\frac{1}{2}\log\frac{P}{\sigma^2}=\frac{1}{2}\log\left(1+\frac{S}{\sigma^2}\right) \tag{3-84}$$

式中，S/σ^2 是信号功率与噪声功率之比，常称为信噪比，用 SNR 表示，即 $C=\frac{1}{2}\log(1+SNR)$。可见信道容量取决于信道的信噪比。

值得注意的是，这里研究的信道只存在加性噪声，而对输入功率没有损耗。但在实际通信系统中，几乎都存在大小不等的功率损耗，所以计算时输入信号的功率 S 应是经过损耗后的功率。例如信道损耗为 $|H(e^{jw})|^2$，输入功率为 S，则式（3-84）中的信号功率应为 $S|H(e^{jw})|^2$。

另外，在很多实际系统中，噪声并不是高斯型的，但若是加性的，可以求出信道容量的上下界。若是乘性噪声，则很难分析。对于加性均值为 0，平均功率为 σ^2 的非高斯噪声信道，其信道容量有下列上下界

$$\frac{1}{2}\log\left(1+\frac{S}{\sigma^2}\right)\leqslant C\leqslant\frac{1}{2}\log 2\pi eP-H_c(n) \tag{3-85}$$

式中，$H_c(n)$ 是噪声熵，P 为输出信号的功率，$P=S+\sigma^2$。这里不作证明，仅说明物理意义。首先看右边，第一项 $\frac{1}{2}\log 2\pi eP$ 是均值为 0、方差为 P 的高斯信号的熵，由于噪声 n 是非高斯的，如果输入信号 X 的分布能使 $x+n=y$ 呈高斯分布的，则 $H_c(Y)$ 达到最大值，此时信道容量达到上限值 $\frac{1}{2}\log 2\pi eP-H_c(n)$，而一般情况下，信道容量必小于该上限值；再看式（3-85）的左边，可写成 $\frac{1}{2}\log 2\pi eP-\frac{1}{2}\log 2\pi e\sigma^2$，第二项 $\frac{1}{2}\log 2\pi e\sigma^2$ 是均值为 0、方差为 σ^2 的高斯噪声的熵，此为平均功率受限 σ^2 时的最大值，即噪声熵考虑的是最坏情况，所以是信道容量的下限值。

式（3-85）说明在同样平均功率受限情况下，非高斯噪声信道的容量要大于高斯噪声信道容量，所以在处理实际问题时，通常采用计算高斯噪声信道容量的方法，保守地估计容量，且高斯噪声信道容量容易计算。

3.6.2 多维无记忆加性连续信道

信道输入随机序列 $\boldsymbol{x}=(x_1,x_2,\cdots,x_L)$，输出随机序列 $\boldsymbol{y}=(y_1,y_2,\cdots,y_L)$，加性信道有 $\boldsymbol{y}=\boldsymbol{x}+\boldsymbol{n}$，其中 $\boldsymbol{n}=(n_1,n_2,\cdots,n_L)$ 是均值为零的高斯噪声，表示各单元时刻 1，2，…，L 上的噪声，如图 3-11 所示。

由于信道无记忆，所以有 $p(\boldsymbol{y}|\boldsymbol{x})=\prod_{l=1}^{L}p(y_l|x_l)$。加性信道中噪声随机序列的各时刻分量是统计独立的，即 $p_n(\boldsymbol{n})=p_Y(\boldsymbol{y}|\boldsymbol{x})=\prod_{l=1}^{L}p_n(n_l)$，各分量都是均值为 0、方差为 σ^2 的高斯变量，所以多维无记忆高斯加性信道就可等价成 L 个独立的并联高斯加性信道。

由式（3-56）可得

$$I(\boldsymbol{X};\boldsymbol{Y})\leqslant\sum_{l=1}^{L}I(X_L;Y_L)=\sum_{l=1}^{L}\frac{1}{2}\log\left(1+\frac{P_l}{\sigma_l^2}\right)$$

则

$$C=\max_{p(x)}I(\boldsymbol{X};\boldsymbol{Y})=\sum_{l=1}^{L}\frac{1}{2}\log\left(1+\frac{P_l}{\sigma_l^2}\right)\text{bit/}L\text{ 维自由度} \tag{3-86}$$

式中，σ_l^2 是第 l 个单元时刻高斯噪声的方差，均值为 0。因此当且仅当输入随机矢量 $\boldsymbol{X}$ 中各

分量统计独立，且是均值为 0、方差为 P_l 的高斯变量时，才能达到此信道容量。式（3-86）既是多维无记忆高斯加性连续信道的信道容量，也是 L 个独立并联高斯加性信道的信道容量。下面进行讨论。

（1）当每个单元时刻上的噪声都是均值为 0、方差相同的 σ^2 高斯噪声时，由式（3-86）得

$$C=\frac{L}{2}\log\left(1+\frac{S}{\sigma^2}\right)\text{bit/}L\text{ 维自由度} \tag{3-87}$$

当且仅当输入矢量 $\boldsymbol{X}$ 的各分量统计独立，且各分量是均值为 0、方差为 S 的高斯变量时，信道中传输的信息率可达到最大。

（2）当各单元时刻 L 个高斯噪声均值为 0、但方差不同且为 σ_l^2 时，若输入信号的总平均功率受限，约束条件为

$$E\left[\sum_{l=1}^{L}X_l^2\right]=\sum_{l=1}^{L}E\left[X_l^2\right]=\sum_{l=1}^{L}P_l=P \tag{3-88}$$

则此时各单元时刻的信号平均功率应合理分配，才能使信道容量最大。也就是需要在式（3-88）的约束条件下，求式（3-86）中 P_l 的分布。这是一个标准的求极大值的问题，采用拉格朗日乘子法来计算。

作辅助函数

$$f(P_1,P_2,\cdots,P_L)=\sum_{l=1}^{L}\frac{1}{2}\log\left(1+\frac{P_l}{\sigma_l^2}\right)+\lambda\sum_{l=1}^{L}P_l$$

令

$$\frac{\partial f(P_1,P_2,\cdots,P_L)}{\partial P_l}=0,\quad l=1,2,\cdots,L$$

解得

$$\frac{1}{2}\frac{1}{P_l+\sigma_l^2}+\lambda=0,\quad l=1,2,\cdots,L$$

$$P_l+\sigma_l^2=-\frac{1}{2\lambda},\quad l=1,2,\cdots,L \tag{3-89}$$

上式表示各单元时刻上信号平均功率与噪声功率之和，即各个时刻的信道输出功率相等，设为常数 v，则

$$v=\frac{P+\sum_{l}\sigma_l^2}{L}$$

则各单元时刻信号平均功率为

$$P_l=v-\sigma_l^2=\frac{P+\sum_{i=1}^{L}\sigma_i^2}{L}-\sigma_l^2\qquad l=1,2,\cdots,L \tag{3-90}$$

此时信道容量

$$C=\frac{1}{2}\sum_{l=1}^{L}\log\frac{P+\sum_{i=1}^{L}\sigma_i^2}{L\sigma_l^2}$$

但是，如果某些单元时刻的噪声 σ_l^2 太大，大于常数 v，使式（3-90）中 P_l 出现负数值，说明这些时刻的信道质量太差，无法使用，必须置 $P_l=0$，不分配功率，予以关闭。然后重新调整信道功率分配，直至 P_l 不出现负值。这就是著名的“注水法”原理（Water-Filling），示意如图 3-16 所示。将各单元时刻或并联信道看成用来盛水的容器，将信号功率看成水，向容器中倒水，最后的水平面是平的，每个子信道中装的水量即是分配的信号功率。这时信道容量为

$$C=\frac{1}{2}\sum_l \log\left(1+\frac{P_l}{\sigma_l^2}\right),\quad \sum_l P_l=P,\quad P_l\geqslant 0$$

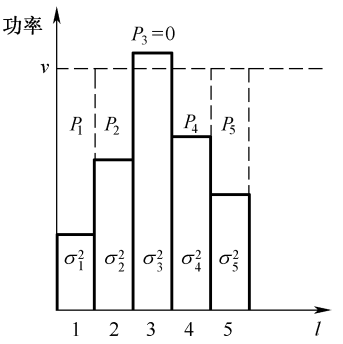


图 3-16 “注水法”原理示意图

3.6.3 加性高斯白噪声信道的信道容量

波形信道中，在限时 t_B，限频 f_m 条件下可转化成多维连续信道，将输入随机过程 $\{x(t)\}$、输出随机过程 $\{y(t)\}$ 转化成 L 维随机序列 $\boldsymbol{x}=(x_1,x_2,\cdots,x_L)$ 和 $\boldsymbol{y}=(y_1,y_2,\cdots,y_L)$，因而可得波形信道的平均互信息为

$$\begin{aligned}I\left[x(t);y(t)\right]&=\lim_{L\to\infty} I(\boldsymbol{X};\boldsymbol{Y})=\lim_{L\to\infty}\left[H_c(\boldsymbol{X})-H_c(\boldsymbol{X}\mid\boldsymbol{Y})\right]\\&=\lim_{L\to\infty}\left[H_c(\boldsymbol{Y})-H_c(\boldsymbol{Y}\mid\boldsymbol{X})\right]=\lim_{L\to\infty}\left[H_c(\boldsymbol{X})+H_c(\boldsymbol{Y})-H_c(\boldsymbol{XY})\right]\end{aligned}$$

一般情况下，波形信道都是研究单位时间内的信息传输率 R_t，即

$$R_t=\lim_{t_B\to\infty}\frac{1}{t_B}I(\boldsymbol{X};\boldsymbol{Y})\ \text{bit/s}$$

信道容量为

$$C=\max_{p(x)}\left[\lim_{t_B\to\infty}\frac{1}{t_B}I(\boldsymbol{X};\boldsymbol{Y})\right]\text{bit/s}$$

高斯白噪声加性波形信道是经常假设的一种信道，加入信道的噪声是限带的加性高斯白噪声 $\{n(t)\}$，其均值为 0，功率谱密度为 $N_0/2$。因为一般信道的频带宽度总是受限的，设其为 W（即 $|f|\leqslant W$），而低频限带高斯白噪声的各样本值彼此统计独立，所以限频高斯白噪声过程可分解成 L 维统计独立的随机序列，在 $[0,t_B]$ 时刻内，$L=2Wt_B$。这是多维无记忆高斯加性信道，根据式（3-86），信道容量为

$$C=\frac{1}{2}\sum_{l=1}^{L}\log\left(1+\frac{P_l}{\sigma_l^2}\right)$$

式中，σ_l^2 是每个噪声分量的功率，$\sigma_1^2=P_n=\dfrac{N_0}{2}\cdot 2W\cdot t_B/2Wt_B=\dfrac{N_0}{2}$。$P_l$ 是每个信号样本值的平均功率，设信号的平均功率受限于 P_S，则 $P_l=P_S t_B/2Wt_B=\dfrac{P_S}{2W}$。信道的容量为

$$C=\frac{L}{2}\log\left(1+\frac{P_S}{2W}\Big/\frac{N_0}{2}\right)=\frac{L}{2}\left(1+\frac{P_S}{N_0W}\right)=Wt_B\log\left(1+\frac{P_S}{N_0W}\right)\text{bit/}L\text{ 维}\qquad(3\text{-}91)$$

要使信道传送的消息达到信道容量，必须使输入信号 $\{x(t)\}$ 具有均值为 0、平均功率为 P_S 的高斯白噪声的特性。不然，传送的信息率低于信道容量，信道得不到充分利用。

高斯白噪声加性信道单位时间的信道容量

$$C_t = \lim_{t_B \to \infty} \frac{C}{t_B} = W \log\left(1 + \frac{P_S}{N_0 W}\right) \text{bit/s} \tag{3-92}$$

式中，P_S 是信号的平均功率；$N_0 W$ 为高斯白噪声在带宽 W 内的平均功率（功率谱密度为 $N_0/2$）。可见信道容量与信噪功率比和带宽有关。

这就是重要的香农公式。当信道的频带受限于 W（单位 Hz），信道噪声为加性高斯白噪声，功率谱密度为 $N_0/2$，噪声功率为 $N_0 W$，输入信号的平均功率受限于 P_S，信道的信噪功率比 $SNR = P_S/N_0 W$，则当信道输入信号是平均功率受限的高斯白噪声信号时，信道中的信息传输率可以达到式（3-92）的信道容量。此为在噪声信道中可靠通信，信道传输率的上限值。

而常用的实际信道一般为非高斯噪声波形信道，类似 3.6.1 小节所述，其噪声熵比高斯噪声的小，信道容量是以高斯加性信道的信道容量为下限值。所以香农公式也适用于其他一般非高斯波形信道，由香农公式得到的值是其信道容量的下限值。

下面对式（3-92）的香农公式做深入讨论，以说明增加信道容量的途径。

（1）当带宽 W 一定时，信噪比 SNR 与信道容量 C_t 呈对数关系，如图 3-17 所示。若 SNR 增大，C_t 就增大，但增大到一定程度后就会趋于缓慢。这说明增加输入信号功率有助于容量的增大，但该方法是有限的；另一方面降低噪声功率也是有用的，当 $N_0 \to 0$ 时，$C_t \to \infty$，即无噪声信道的容量为无限大。

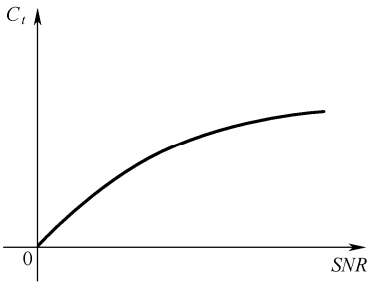


图 3-17 信道容量与信噪比的关系

（2）当输入信号功率 P_S 一定，增加信道带宽，容量可以增加，但到一定阶段后增加变得缓慢，因为当噪声为加性高斯白噪声时，随着 W 的增加，噪声功率 $N_0 W$ 也随之增加。当 $W \to \infty$ 时，$C_t \to \infty$，利用关系式 $\ln(1+x) \approx x$（x 很小时）可求出 C_∞ 值，即

$$C_\infty = \lim_{W \to \infty} \frac{P_S}{N_0} \frac{WN_0}{P_S} \log\left(1 + \frac{P_S}{N_0 W}\right) = \lim_{x \to \infty} \frac{P_S}{N_0} \log(1+x)^{1/x}$$

$$= \lim_{x \to \infty} \frac{P_S}{N_0 \ln 2} \ln(1+x)^{1/x} = \lim_{x \to \infty} \frac{P_S}{N_0 \ln 2} \text{bit/s}$$

该式说明即使带宽无限，信道容量仍是有限的。当 $C_\infty = 1\text{bit/s}$，$P_S/N_0 = \ln 2 = -1.6\text{dB}$，即当带宽不受限制时，传送 1bit 信息，信噪比最低只需–1.6dB，这就是香农限，是加性高斯噪声信道信息传输率的极限值，是一切编码方式所能达到的理论极限。要获得可靠通信的实际值往往都比这个值大得多。

$C_t/W = \log(1+SNR)\,\text{bit/(s·Hz)}$，单位频带的信息传输率，也叫频带利用率。该值越大，信道利用就越充分。当 $C_t/W = 1\text{bit/(s·Hz)}$ 时，$SNR = 1(0\text{dB})$；当 $C_t/W \to 0$ 时，$SNR = -1.6\text{dB}$，

此时信道完全丧失通信能力，如图 3-18 所示。

（3）C_t 一定时，带宽 W 越大，信噪比 SNR 可降低，即两者是可以互换的。若有较大的传输带宽，则在保持信号功率不变的情况下，可允许较大的噪声，即系统的抗噪声能力提高。无线通信中的扩频系统就是利用了这个原理，将所需传送的信号扩频，使之远远大于原始信号带宽，以增强抗干扰能力。

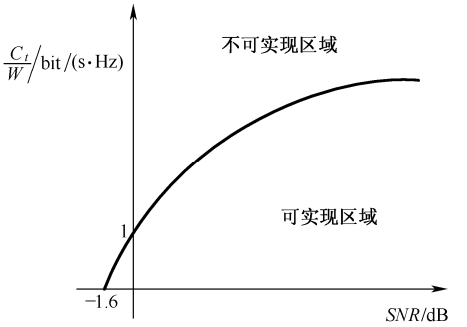


图 3-18 频带利用率与信噪比的关系

例 3-6 电话信道的带宽 3.3kHz，若信噪功率比为 20dB，$SNR=100\text{dB}$，则运用香农公式，该信道的容量为 $C_t=W\log(1+SNR)=3.3\log(1+100)=22\text{kbit/s}$。而实际信道达到的最大信道传输率 19.2kbit/s，那是考虑了串音、回波等干扰因素，所以比理论计算值要小。

3.7 信源与信道的匹配

信源发出的消息（符号）一定要通过信道来传输。对于某一信道，其信道容量是一定的，而且只有当输入符号的概率分布 $p(x)$ 满足一定条件时才能达到信道容量 C。这就是说，只有一定的信源才能使某一信道的信息传输速率达到最大。一般信源与信道连接时，其信息传输率 $R=I(X;Y)$ 并未达到最大。这样，信道的信息传输率还有提高的可能，即信道没有得到充分利用。当信源与信道连接时，若信息传输率达到了信道容量，我们则称此信源与信道达到匹配。否则认为信道有冗余。信道冗余度定义为

$$\text{信道冗余度}=C-I(X;Y) \tag{3-93}$$

式中，C 是该信道的信道容量，$I(X;Y)$ 是信源通过该信道实际传输的平均信息量。

$$\text{信道相对冗余度}=\frac{C-I(X;Y)}{C}=1-\frac{I(X;Y)}{C} \tag{3-94}$$

在无损信道中，信道容量 $C=\log r$（r 是信道输入符号的个数），而 $I(X;Y)=H(X)$，这里 $H(X)$ 是输入信道的信源熵，因而

$$\text{无损信道的相对冗余度}=1-\frac{H(X)}{\log r} \tag{3-95}$$

与第 2 章定义的信源冗余度相比，可知上式也是信源的冗余度。所以，提高无损信道信息传输率的研究就等于减少信源冗余度的研究。对于无损信道，可以通过信源编码，减少信源的冗余度，使信息传输率达到信道容量。

一般通信系统中，信源发出的消息（符号）必须转换成适合信道传输的符号（信号）来传输。对于离散无损信道，如何进行转换，才能使信道的信息传输率达到信道容量，达到信源与信道匹配呢？例如，某离散无记忆信源为

$$\begin{bmatrix} S \\ p(s) \end{bmatrix}=\begin{bmatrix} s_1, & s_2, & s_3, & s_4, & s_5, & s_6 \\ \frac{1}{2}, & \frac{1}{4}, & \frac{1}{8}, & \frac{1}{16}, & \frac{1}{32}, & \frac{1}{32} \end{bmatrix}$$

通过一无噪、无损二元离散信道进行传输。我们可求得，二元离散信道的信道容量为 $C=1$（bit/信道符号）。此信源的信息熵为 $H(S)=1.937$ 比特/信源符号。我们必须对信源 S 进行二元编码，才能使信源符号在此二元信道中传输。进行二元编码的结果可有许多种，若有

$$\begin{array}{cccccccc} & s_1, & s_2, & s_3, & s_4, & s_5, & s_6 \\ C_1: & 000, & 001, & 010, & 011, & 100, & 101 \\ C_2: & 0000 & 0001 & 0010 & 0011 & 0100 & 0101 \end{array}$$

可见，码 C_1 中每个信源符号需用 3 个二元符号，而码 C_2 中需用四个二元符号。对于码 C_1，可得信道的信息传输率 $R_1=\dfrac{H(S)}{3}=0.646$ bit/信道符号；对于码 C_2，$R_2=\dfrac{H(S)}{4}=0.484$ bit/信道符号。这时，$R_2<R_1<C$，信道有冗余。那么，是否存在一种信源编码，使信道的信息传输率 R 接近或等于信道容量呢？也就是，是否存在一种编码，使每个信源符号所需的二元符号最少呢？这就是香农无失真信源编码理论，也就是无失真数据压缩理论。

无失真信源编码就是将信源输出的消息变换成适合信道传输的新信源的消息（符号）来传输，而使新信源的符号接近等概分布，新信源的熵接近最大熵 $\log r$，这样，信道传输的信息量达到最大，信道冗余度接近于零，使信源和信道达到匹配，信道得到充分利用。

习　　题

3.1　设有一离散无记忆信源，其概率空间为 $\begin{bmatrix} X \\ P(x) \end{bmatrix}=\begin{bmatrix} x_1 & x_2 \\ 0.6 & 0.4 \end{bmatrix}$，信源发出符号通过一干扰信道，接收符号为 $Y=\{y_1,y_2\}$，信道传递矩阵为 $P=\begin{bmatrix} \frac{5}{6} & \frac{1}{6} \\ \frac{1}{4} & \frac{3}{4} \end{bmatrix}$，求解下述问题。

（1）信源 X 中事件 x_1 和 x_2 分别含有的自信息量。

（2）收到消息 y_j (j=1,2)后，获得的关于 x_i (i=1,2)的信息量。

（3）信源 X 和信宿 Y 的信息熵。

（4）信道疑义度 $H(X|Y)$ 和噪声熵 $H(Y|X)$。

（5）接收到消息 Y 后获得的平均互信息量 $I(X;Y)$。

3.2　设有扰离散信道的输入端是以等概率出现的 A、B、C、D 四个字母。该信道的正确传输概率为 0.5，错误传输概率平均分布在其他三个字母上。验证在该信道上每个字母传输的平均信息量为 0.21bit。

3.3　已知信源 X 包含两种消息：x_1,x_2，且 $P(x_1)=P(x_2)=1/2$，信道是有扰的，信宿收到的消息集合 Y 包含 y_1,y_2。给定信道矩阵为 $P=\begin{bmatrix} 0.98 & 0.02 \\ 0.2 & 0.8 \end{bmatrix}$，求平均互信息 $I(X;Y)$。

3.4　设二元对称信道的传递矩阵为 $\begin{bmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{2}{3} \end{bmatrix}$。

（1）若 $P(0)=\frac{3}{4}$，$P(1)=\frac{1}{4}$，求 $H(X)$，$H(X|Y)$、$H(Y|X)$ 和 $I(X;Y)$。

（2）求该信道的信道容量及其达到信道容量时的输入概率分布。

3.5　求下列两个信道的信道容量，并加以比较。

（1）$\begin{bmatrix} \overline{p}-\varepsilon & p-\varepsilon & 2\varepsilon \\ p-\varepsilon & \overline{p}-\varepsilon & 2\varepsilon \end{bmatrix}$　　（2）$\begin{bmatrix} \overline{p}-\varepsilon & p-\varepsilon & 2\varepsilon & 0 \\ p-\varepsilon & \overline{p}-\varepsilon & 0 & 2\varepsilon \end{bmatrix}$

其中 $p+\overline{p}=1$。

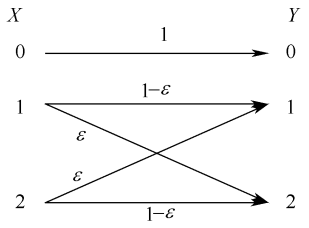
3.6　求题图 3-6 中信道的信道容量及最佳的输入概率分布。并求当 $\varepsilon=0$ 和 $\frac{1}{2}$ 时的信道容量 C。

3.7　有一个二元对称信道，其信道矩阵为 $\begin{bmatrix} 0.98 & 0.02 \\ 0.02 & 0.98 \end{bmatrix}$。设该信道以 1 500 个二元符号每秒的速率传输输入符号。现有一消息序列共有 14 000 个二元符号，并设在这个消息中，$P(0)=P(1)=1/2$。问从信息传输的角度来考虑，10s 内能否将这消息序列无失真地传送完？

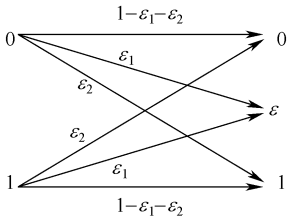
3.8　有一离散信道，其信道转移概率如题图 3-8 所示，试求解如下问题。

（1）信道容量 C。

（2）若 ε_2=0，求信道容量。



题图 3-6



题图 3-8

3.9　设离散信道矩阵为

$$\boldsymbol{P}=\begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{3} & \frac{1}{6} & \frac{1}{3} \end{bmatrix}$$

求信道容量 C。

3.10　若有一离散非对称信道，其信道转移概率如题图 3-10 所示。试求解如下问题。

（1）信道容量 C_1。

（2）若将两个同样信道串接，求串接后的转移概率。

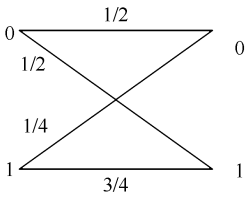
（3）求串接后信道的信道容量 C_2。

3.11　设有一离散级联信道如题图 3-11 所示。试求解如下问题。

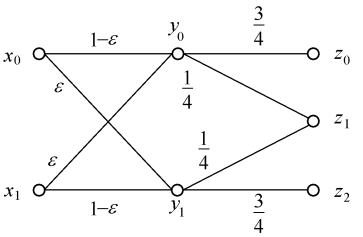
（1）X 与 Y 间的信道容量 C_1。

（2）Y 与 Z 间的信道容量 C_2。

（3）X 与 Z 间的信道容量 C_3 及其输入分布 $P(x)$。



题图 3-10



题图 3-11

3.12　若有两个串接的离散信道，它们的信道矩阵为

$$\boldsymbol{P}=\begin{bmatrix}0 & 0 & 0 & 1\\ 0 & 0 & 0 & 1\\ \frac{1}{2} & \frac{1}{2} & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}$$

并设第一个信道的输入符号 $X\in\{a_1,a_2,a_3,a_4\}$ 是等概率分布，求 $I(X;Z)$ 和 $I(X;Y)$ 并加以比较。

3.13　若 X 、 Y 、 Z 是 3 个随机变量，试证明如下命题。

（1）$I\big(X;YZ\big)=I(X;Y)+I(X;Z\,|\,Y)=I(X;Z)+I(X;Y\,|\,Z)$。

（2）$I\big(X;Y\,|\,Z\big)=I(Y;X\,|\,Z)=H(X\,|\,Z)-H(X\,|\,YZ)$。

（3）$I\big(X;Y\,|\,Z\big)\geqslant 0$，当且仅当（$X$，$Y$，$Z$）是马氏链时等式成立。

3.14　若三个离散随机变量有如下关系：$X+Y=Z$，其中 X 和 Y 相互独立，试证明如下命题。

（1）$I(X;Z)=H(Z)-H(Y)$。

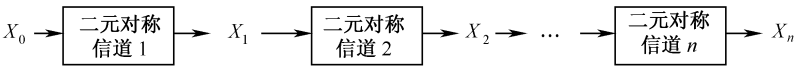
（2）$I(XY;Z)=H(Z)$。

（3）$I\big(X;YZ\big)=H(X)$。

（4）$I\big(Y;Z\,|\,X\big)=H(Y)$。

（5）$I\big(X;Y\,|\,Z\big)=H(X\,|\,Z)=H(Y\,|\,Z)$。

3.15　把 n 个二元对称信道串接起来，每个二元对称信道的错误传递概率为 p。证明这 n 个串接信道可以等效于一个二元对称信道，其错误传递概率为 $\frac{1}{2}[1-(1-2p)^n]$。并证明：$\lim\limits_{n\to\infty}I(X_0;X_n)=0$，设 $p\neq 0$或1，信道的串接如题图 3-15 所示。



题图 3-15

3.16　有一信源发出恒定宽度，但不同幅度的脉冲，幅度值 x 处在 a_1 和 a_2 之间。此信源连至某信道，信道接收端接收脉冲的幅度 y 处在 b_1 和 b_2 之间。已知随机变量 X 和 Y 的联合概率密度函数

$$p(xy)=\frac{1}{(a_2-a_1)(b_2-b_1)}$$

试计算 $H(X)$、$H(Y)$、$H(XY)$ 和 $I(X;Y)$。

3.17 设某连续信道，其特性为

$$p(y|x)=\frac{1}{\alpha\sqrt{3\pi}}\exp[-\frac{\left(y-\frac{x}{2}\right)^2}{3\alpha^2}]\ (-\infty<x,y<\infty)$$

而信道输入变量 X 的概率密度函数为

$$p(x)=\frac{1}{2\alpha\sqrt{\pi}}\exp\left[-\frac{x^2}{4\alpha^2}\right]$$

试计算：（1）信源的熵 $H(X)$；（2）平均互信息 $I(X;Y)$。

3.18 设加性高斯白噪声信道中，信道带宽为 3kHz，又设{（信号功率＋噪声功率）/噪声功率}=10dB。

（1）试计算该信道传送的最大信息率（单位时间）。

（2）若功率信噪比降为 5dB，要达到相同的最大信息传输率，信道带宽应是多少？

3.19 设电话信号的信息率为 56kbit/s，在一个噪声功率谱为 $N_0=5\times10^{-6}\text{mW/Hz}$、限频 F、限输入功率 P 的高斯信道中传送，若 F=4kHz，问无差错传输所需的最小功率 P 是多少？若 $F\to\infty$，则 P 是多少？

3.20 在图片传输中，每帧约有 2.25×10^6 像素，为了能很好地重现图像，需分 16 个亮度电平，并假设亮度电平等概率分布。试计算每秒钟传送 30 帧图片所需信道的带宽（信噪功率比为 30dB）。

第4章 信息率失真函数

内容简介

信息率失真函数，简称率失真函数，是香农信息论的一大分支，是研究信源编码问题的基础。

在实际的通信与信息处理过程中，信源发送的信息传输到接收端，一般是不可能保持与发送端完全一样，或多或少地总会产生一些失真。在大多数情况下，这些失真是可以容许的。

因为实际上信息在信道传输的过程中，有用的信息总会受到干扰和噪声的影响。而且在接收端，不管接收者是人还是机器，他（它）总是具有一定的灵敏度和分辨力的，超过接收者灵敏度和分辨力所传送的信息是毫无意义的。

比如，人的视觉和听觉都允许有一定的失真，电影和电视就是利用了人眼的视觉残留，没有发觉影片是由一帧一帧画面快速连接起来的，感觉好像是连续的。

因此，只要不超过给定的失真指标要求，对信息的传输和处理就不会造成很大的影响。我们将这类失真指标称为允许失真。由香农建立的信息率失真函数理论，就是研究在不同类型的信源和信宿的各种允许失真的最大限度值 D 的条件下，信源所必须具有的最小信息率 $R(D)$，即信息率失真函数。

4.1 平均失真和信息率失真函数

关于信息的传输或处理，首先想到的是希望不失真，也就是能原原本本地、精确地恢复出原来的信息。但是，实际上并非要求如此严格，信息有一定的失真是可以容忍的。

但是，当失真大于某一限度后，信息质量将被严重损伤，甚至丧失其实用价值。因此，要规定一定的失真限度。那么用什么标准呢？如何定量地进行失真的测度呢？

4.1.1 失真函数

假如信源 X 输出一个随机序列 $X=x_1,x_2,\cdots,x_n$，经过信道传输后变成了 $Y=y_1,y_2,\cdots,y_m$，如果输入与输出是一一对应的，即

$$x_i=y_j,\quad i=1,2,\cdots,n,\quad j=1,2,\cdots,m \tag{4-1}$$

就认为无失真。若 $x_i\neq y_j$，就产生了失真。

失真的大小，用一个量来表示，即失真函数 $d(x_i,y_j)$，来衡量用 y_j 代替 x_i 所引起的失真程度。

定义

$$d(x_i,y_j)=\begin{cases}0, & x_i=y_j\\ a>0, & x_i\neq y_j\end{cases} \tag{4-2}$$

将所有的 $d(x_i,y_j)$，$i=1,2,\cdots,n$，$j=1,2,\cdots,m$ 排列起来，写成一个矩阵。

$$\boldsymbol{d}=\begin{bmatrix}d(x_1,y_1) & d(x_1,y_2) & \cdots & d(x_1,y_m)\\ d(x_2,y_1) & d(x_2,y_2) & \cdots & d(x_n,y_m)\\ \vdots & \vdots & \ddots & \vdots\\ d(x_n,y_1) & d(x_n,y_2) & \cdots & d(x_n,y_m)\end{bmatrix} \tag{4-3}$$

称该矩阵为失真矩阵。

例题 4-1 设信源符号集为 X={0,1}，接收端收到的符号集为 Y={0,1,2}，规定失真函数为

$$d(0,0)=d(1,1)=0$$
$$d(0,1)=d(1,0)=1$$
$$d(0,2)=d(1,2)=2$$

求失真矩阵 d。

解 根据式(4-3)，可得失真矩阵为

$$d=\begin{bmatrix}0 & 1 & 2\\ 1 & 0 & 2\end{bmatrix}$$

比较常用的失真函数有以下几种。

均方失真：$d(x_i,y_j)=(x_i-y_j)^2$。

绝对失真：$d(x_i,y_j)=\left|x_i-y_j\right|$。

相对失真：$d(x_i,y_j)=\dfrac{\left|x_i-y_j\right|}{\left|x_i\right|}$。

误码失真：$d(x_i,y_j)=\delta(x_i-y_j)=\begin{cases}0, & x_i=y_j\\ 1, & x_i\neq y_j\end{cases}$。

前三种失真函数适用于连续信源，最后一种失真函数适用于离散信源。

失真函数的定义可以推广到矢量传输情况。若离散矢量信源符号为矢量序列 $\boldsymbol{X}=[X_1X_2\cdots X_i\cdots X_n]$，其中的 $X_i=(x_{i1}x_{i2}\cdots x_{iN})$ 为 N 长的符号序列，经信道传输后，接收端的矢量序列为 $\boldsymbol{Y}=[Y_1Y_2\cdots Y_j\cdots Y_m]$，$Y_j=(y_{j1}y_{j2}\cdots y_{jN})$，则失真函数定义为

$$d(x_i,y_j)=\frac{1}{N}\sum_{k=1}^{N}d(x_{ik},y_{jk})$$

其中的 $d(x_{ik},y_{jk})$ 是信源输出第 i 个 N 长符号序列 X_i 中的第 k 个符号与接收端收到的第 j 个 N

长符号序列 Y_j 中的第 k 个符号之间的失真函数。

4.1.2 平均失真

将失真函数的数学期望称为平均失真。

$$\begin{aligned} D &= \sum_{i=1}^{n}\sum_{j=1}^{m} p(x_i y_j) d(x_i, y_j) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m} p(x_i) p(y_j \mid x_i) d(x_i, y_j) \end{aligned} \tag{4-4}$$

式中，$p(x_i)$ 为信源符号的概率分布，$p(x_i y_j)$ 为信源符号 x_i 与接收符号 y_j 的联合概率分布，$p(y_j \mid x_i)$ 为已知信源符号为 x_i 的条件下，接收到的符号为 y_j 的概率，也称转移概率分布，$d(x_i, y_j)$ 为信源符号与接收符号之间的失真函数。

平均失真 D 是对给定信源分布 $p(x_i)$，在给定转移概率分布 $p(y_j \mid x_i)$ 的信道中传输信息时，失真的总体量度。

若信源为连续信源时，平均失真定义为

$$D = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_{xy}(xy) d(x, y) \mathrm{d}x\mathrm{d}y \tag{4-5}$$

若为矢量传输时，平均失真定义为

$$D = \frac{1}{N}\sum_{k=1}^{N} E[d(x_{ik}, y_{jk})] = \frac{1}{N}\sum_{k=1}^{N} D_k \tag{4-6}$$

D_k 为第 k 个符号的平均失真。

4.1.3 信息率失真函数

信源的信息率失真函数也称为率失真函数。

假定传输信道的容量为 C，用该信道来传输信息传输率为 R 的信源，如果 $R>C$，就必须对信源进行压缩，压缩后的信息传输率 R' 小于信道容量 C。同时，要保证对信息的压缩所引入的失真不超过预先规定的失真限度 D'。对信源进行压缩，就是要求在满足平均失真 $D \leqslant D'$ 的前提下，使信息率尽可能小。

满足平均失真 $D \leqslant D'$ 的所有转移概率分布 p_{ij} 构成了一个假想的信道，称为 D' 允许信道，记为

$$P_{D'} = \{p(y \mid x):\ D \leqslant D'\} \tag{4-7}$$

对于离散无记忆信道，有

$$P_{D'} = \{p(y_j \mid x_i):\ D \leqslant D',\ i = 1, 2, \cdots, n,\ j = 1, 2, \cdots, m\}$$

在这个允许信道 $P_{D'}$ 中，总能找到一个信道 $p(Y \mid X)$，使得给定的信源经过此信道传输后，平均互信息量 $I(X;Y)$ 达到最小，这个最小的平均互信息量 $I(X;Y)$ 就称为信息率失真函数，用 $R(D)$ 表示，即

$$R(D) = \min_{P_{D'}} I(X;Y) \tag{4-8}$$

若信源为离散无记忆信源，则 $R(D)$ 函数为

$$R(D)=\min_{p_{ij}\in P_{D'}}\sum_{i=1}^{n}\sum_{j=1}^{m}p(x_i)p(y_j\mid x_i)\log\frac{p(y_j\mid x_i)}{p(y_j)} \tag{4-9}$$

式中，$p(x_i)$ 为信源符号的概率分布，$p(y_j\mid x_i)$ 为转移概率分布，$p(y_j)$ 为接收端收到符号的概率分布。

例题 4-2 已知某编码器的输入符号的概率分布为 $p(x)=\{0.5,0.5\}$，两个信道的转移概率分别为

$$[p'_{ij}]=\begin{bmatrix}0.6 & 0.4\\ 0.2 & 0.8\end{bmatrix},\quad [p''_{ij}]=\begin{bmatrix}0.9 & 0.1\\ 0.2 & 0.8\end{bmatrix}$$

计算平均互信息量。

解 由 $p(x_iy_j)=p(x_i)p(y_j\mid x_i)$，可得输入符号与输出符号的联合概率

$$p'(x_1y_1)=0.5\times0.6=0.3,\ p'(x_1y_2)=0.5\times0.4=0.2$$
$$p'(x_2y_1)=0.5\times0.2=0.1,\ p'(x_2y_2)=0.5\times0.8=0.4$$

由于 $p(y_j)=\sum_i p(x_iy_j)$，得 $p'(y_1)=0.4,\ p'(y_2)=0.6$，而 $p(x_i\mid y_j)=p(x_iy_j)/p(y_j)$，因此得到

$$p'(x_1\mid y_1)=\frac{3}{4},\ p'(x_1\mid y_2)=\frac{1}{3},\ p'(x_2\mid y_1)=\frac{1}{4},\ p'(x_1\mid y_1)=\frac{2}{3}$$

则平均互信息为

$$I'(X;Y)=\sum_{ij}p'(x_iy_j)\log_2\frac{p(x_i\mid y_j)}{p(x_i)}=0.125\text{bit/符号}$$

同样，可得 p''_{ij} 时的平均互信息为

$$I''(X;Y)=0.397\text{bit/符号}$$

从此例我们可以看到，若固定 $p(x)$ 不变时，平均互信息量随信道的转移概率的变化而变化。这是因为信道受到干扰的作用不同，传递的信息量也不同。可以证明这样一个结论：$p(x)$ 一定时，平均互信息量 $I(X;Y)$ 是关于信道的转移概率的下凸函数，即存在一极小值。

根据平均互信息量的表达式

$$I(X;Y)=H(Y)-H(Y\mid X)=H(X)-H(X\mid Y)$$

可见，互信息为信源发出的信息量与在噪声干扰条件下消失的信息量之差。在这里讨论的是有关信源的问题，一般不考虑噪声问题，而是对信源的原始信息在允许的失真限度内进行了压缩，即去掉冗余部分。由于这种压缩损失了一定的信息，因此造成了一定的失真。将这种失真等效成由噪声而造成的信息损失，可将对信息的压缩看做是信息经过了一个等效的噪声信道。

信息率失真函数的物理意义：对于给定信源，在平均失真 D 不超过失真限度 D' 的条件下，信息率容许压缩的最小值。

例题 4-3　设信源的符号为 $A=\{a_1,a_2,\cdots,a_{2n}\}$，概率分布为 $p(a_i)=\frac{1}{2n}$，i=1,2,…,n，B=A，失真函数为

$$d(a_i,a_j)=\begin{cases}1, & i\neq j\\ 0, & i=j\end{cases}$$

假定失真限度为 $D^{'}=\frac{1}{2}$，即发 $2n$ 个字符，有 n 个正确就可满足要求，求平均失真 D 及信息率 R。

解　由题设知失真限度为 $\frac{1}{2}$，要求有一半正确即可。如果采用如下的编码方案：对原 $2n$ 个字符只用前 n 个，而后 n 个都与第 n 个一样，即

$$a_1\to a_1,\ a_2\to a_2,\cdots,a_n\to a_n$$
$$a_{n+1}\to a_n,\ a_{n+2}\to a_n,\cdots,a_{2n}\to a_n$$

用信道模型图来表示，如图 4-1 所示。

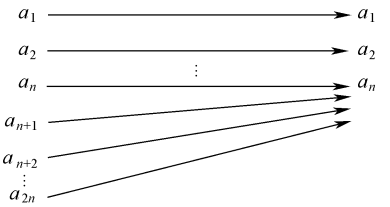


图 4-1　信道模型

这样，符号 $a_1,a_2,\cdots,a_{n-1}$ 发生的概率仍然为 $\frac{1}{2n}$，而符号 a_n 发生的概率却增加了，变为 $\frac{n+1}{2n}$。

因此，信道的输出熵或信息率为

$$R=H(\frac{1}{2n},\frac{1}{2n},\cdots,\frac{1}{2n},\frac{n+1}{2n})=-\sum_{i=1}^{2n}p(a_i)\log p(a_i)$$
$$=(n-1)\cdot\frac{1}{2n}\log 2n+\frac{n+1}{2n}\log\frac{2n}{n+1}$$
$$=\log 2n-\frac{n+1}{2n}\log(n+1)$$

原来 $2n$ 个符号的信源熵或信息率为 $H\left(\frac{1}{2n},\frac{1}{2n},\cdots,\frac{1}{2n}\right)=\log 2n$。可以看出，经过这样的编码后，信息率压缩了 $\frac{n+1}{2n}\log(n+1)$，付出的代价是容忍了 50%的平均失真。这虽然不是最佳的方案，但是给了我们这样一个概念：在允许有一定失真的条件下，信息率是可以降低的，或者说信源数据是可以压缩的。

就本例的结论来说，若 n=1，则共有 2 个符号，即二进制信源，此时 R=0。这就是说，如果允许错一半，可以不发字符，只要作无条件的“猜测”，统计地说，总可以猜对一半，因此，数据压缩极大；但另外一方面说明，允许错一半的要求也太松了一些！

4.2　信息率失真函数的性质

本节介绍了率失真函数的定义域以及它的下凸性、连续性和单调性等内容。

4.2.1 $R(D)$函数的定义域

在$0 \leqslant D \leqslant D_{\max}$域内，$R(D)$为正；在$D \geqslant D_{\max}$域，$R(D)=0$；$D_{\max}$为使$R(D)=0$的$D$的最小值。

下面进行说明，先看一下D域的下限为什么是0？因为D是非负实数$d(x,y)$的数学期望，因此D也是非负的，下限为0。$D=0$的要求意味着必须一一对应编码才行，即无失真情况，或者说对应于无噪声信道。这种情况下，信道传输的信息量就等于信源熵。

$$R(D) = R(0) = H(X)$$

那么，为什么$R(D) \geqslant 0$？根据定义，$R(D)$函数是在约束条件下平均互信息量$I(X;Y)$的最小值，由于$I(X;Y)$是非负的，因此它的最小值也是非负的，其下限值为0。

$R(D)=0$意味着允许的误差较大，以致于要求的信息率可以为0。当然，允许更大的误差，$R(D)$更可以为0。因此，在$R(D)=0$的条件下，D必有一个下界，低于这个下界时，$R(D) \neq 0$。换句话说，在$R(D) \geqslant 0$的条件下，D必有一个上界，大于这个上界时，$R(D)$便可以为0。

取$R(D)=0$的所有D中最小的，定义为$R(D)$定义域的上限$D_{\max}$，即$D_{\max}$是满足$R(D)=0$的所有平均失真D中的最小值。如何来计算$D_{\max}$呢？

$R(D)=0$就是$I(X;Y)=0$，说明随机变量X与Y互相不提供任何信息，彼此相互独立，此时条件概率$p(y_j \mid x_i)$与x_i无关，有

$$p_{ij} = p(y_j \mid x_i) = p(y_j) = p_j$$

此时，平均失真可简化为：$D = \sum_{i=1}^{n}\sum_{j=1}^{m} p_i p_j d_{ij}$，$d_{ij} = d(x_i, y_j)$。

现在需要求出满足$\sum_{j=1}^{m} p_j = 1$条件的D中的最小值，即

$$D_{\max} = \min \sum_{j=1}^{m} p_j \sum_{i=1}^{n} p_i d_{ij} \tag{4-10}$$

从该式可以看到，在$j = 1,2,\cdots,m$中，可找到$\sum_{i=1}^{n} p_i d_{ij}$值最小的$j$，当该$j$的$P_j$为1，而其余$P_j$为0时，上式右端达最小，进一步可简化为

$$D_{\max} = \min_{j} \sum_{i=1}^{n} p_i d_{ij} \tag{4-11}$$

例题 4-4 设输入输出字符表为$X=Y=\{0,1\}$，输入概率分布为$p(x) = \left\{\frac{1}{3}, \frac{2}{3}\right\}$，失真矩阵为

$$\boldsymbol{d} = \begin{bmatrix} d(x_1,y_1) & d(x_1,y_2) \\ d(x_2,y_1) & d(x_2,y_2) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

求$D_{\max}$=?

解 由$D_{\max}$的计算公式，有

$$
\begin{aligned}
D_{\max} &= \min_j \sum_{i=1}^{2} p_i d_{ij} \\
&= \min_j (p_1 d_{11} + p_2 d_{21},\ p_1 d_{12} + p_2 d_{22}) \\
&= \min\left(\frac{1}{3}\times 0 + \frac{2}{3}\times 1, \frac{1}{3}\times 1 + \frac{2}{3}\times 0\right) \\
&= \min\left(\frac{2}{3}, \frac{1}{3}\right) = \frac{1}{3}
\end{aligned}
$$

例题 4-5 若输入输出字符表及输入概率分布同例题 4-4，失真矩阵为

$$
\boldsymbol{d} = \begin{bmatrix} \frac{1}{2} & 1 \\ 2 & 1 \end{bmatrix}
$$

求 $D_{\max}$。

解　由 $D_{\max}$ 的计算公式，有

$$
\begin{aligned}
D_{\max} &= \min_j \sum_{i=1}^{2} p_i d_{ij} \\
&= \min\left(\frac{1}{3}\times\frac{1}{2} + \frac{2}{3}\times 2, \frac{1}{3}\times 1 + \frac{2}{3}\times 1\right) \\
&= 1
\end{aligned}
$$

4.2.2　$R(D)$函数的下凸性

$R(D)$函数的曲线在其定义域内具有下凸性。对任何一对失真 D' 与 D''，及任何一个数 $\alpha \in [0,1]$，必有不等式

$$
R[\alpha D' + (1-\alpha)D''] \leqslant \alpha R(D') + (1-\alpha)R(D'')
$$

成立。

证明　令 $D^{\alpha} = \alpha D' + (1-\alpha)D''$，该 D^{α} 是(D', D'')内的一个点，因为 α 从 0～1 变化时，D^{α} 从 $D' \sim D''$，所以 $R(D^{\alpha})$ 是一个 $R(D)$ 函数。

由率失真函数的定义有

$$
R(D') = \min_{p_{ij} \in P_{D'}} I(p_{ij}) = I(p'_{ij})
$$

$$
R(D'') = \min_{p_{ij} \in P_{D''}} I(p_{ij}) = I(p''_{ij})
$$

令 $p_{ij}^{\alpha} = \alpha p'_{ij} + (1-\alpha)p''_{ij}$，那么这个 p_{ij}^{α} 是否是 $P_{D^{\alpha}}$ 的元？

由平均失真的定义有

$$
\begin{aligned}
D(p_{ij}^{\alpha}) &= \sum_i \sum_j p_i p_{ij}^{\alpha} d_{ij} \\
&= \sum_i \sum_j p_i [\alpha p'_{ij} + (1-\alpha)p''_{ij}] d_{ij} \\
&= \alpha \sum_i \sum_j p_i p'_{ij} d_{ij} + (1-\alpha) \sum_i \sum_j p_i p''_{ij} d_{ij} \\
&\leqslant \alpha D' + (1-\alpha)D'' = D^{\alpha}
\end{aligned}
$$

这是因为 $p_{ij}^{'}$ 和 $p_{ij}^{''}$ 分别是 $P_{D^{'}}$ 和 $P_{D^{''}}$ 中的元，所造成的失真必小于 $D^{'}$ 和 $D^{''}$，而 p_{ij}^{α} 就是 $P_{D^{\alpha}}$ 中的元，所造成的失真必小于 D^{α}。

利用 $I(p_{ij})$ 的下凸性，有

$$\begin{aligned} R(D^{\alpha}) &= \min_{p_{ij}\in P_{D^{\alpha}}} I(p_{ij}) \leqslant I(p_{ij}^{\alpha}) \\ &= I[\alpha p_{ij}^{'} + (1-\alpha)p_{ij}^{''}] \\ &\leqslant \alpha I(p_{ij}^{'}) + (1-\alpha)p_{ij}^{''} \\ &\leqslant \alpha R(D^{'}) + (1-\alpha)R(D^{''}) \end{aligned}$$

4.2.3 $R(D)$函数的连续性

$R(D)$函数的下凸性意味着 $R(D)$函数在定义域内是连续的。

证明 设 $D^{'} = D + \delta$，当 $\delta \to 0$ 时，$P_{D^{'}} \to P_D$。

因为 $I(p_{ij})$ 是 p_{ij} 的连续函数，当 $\delta p_{ij} \to 0$ 时，有

$$I(p_{ij} + \delta p_{ij}) \to I(p_{ij})$$

$$R(D^{'}) = \min_{p_{ij}\in P_{D^{'}}} I(p_{ij}) \to \min_{p_{ij}\in P_D} I(p_{ij}) = R(D)$$

4.2.4 $R(D)$函数的单调递减性

$R(D)$函数的单调递减性可以理解为：容许的失真越大，所要求的信息率越小；反之，容许的失真越小，所要求的信息率就越大。只要允许的失真不同，所要求的最低信息率就不同，不会是相等的。

证明 令 $D > D^{'}$，应用 $P_D \supset P_{D'}$，于是

$$R(D) = \min_{p_{ij}\in P_D} I(p_{ij}) \leqslant \min_{p_{ij}\in P_{D^{'}}} I(p_{ij}) = R(D^{'})$$

上面的不等式是由于 P_D 包含了 $P_{D^{'}}$，在一个较大范围内求得的极小值必然不会大于其中一个小范围内的极小值，因此 $R(D)$函数是非递增的函数。

进一步证明：$R(D)$函数是严格递减的，即上面的不等式中的等号是不成立的。

设有 $0 < D^{'} < D^{''} < D_{\max}$，令

$$R(D^{'}) = I(p_{ij}^{'}),\quad p_{ij}^{'} \in P_{D^{'}}$$

$$R(D_{\max}) = I(p_{ij}^{''}) = 0,\quad p_{ij}^{''} \in P_{D_{\max}}$$

对于足够小的 α，必有

$$D^{'} < (1-\alpha)D^{'} + \alpha D_{\max} = D^{\alpha} < D^{''}$$

令 $p_{ij}^{\alpha} = (1-\alpha)p_{ij}^{'} + \alpha p_{ij}^{''}$，则

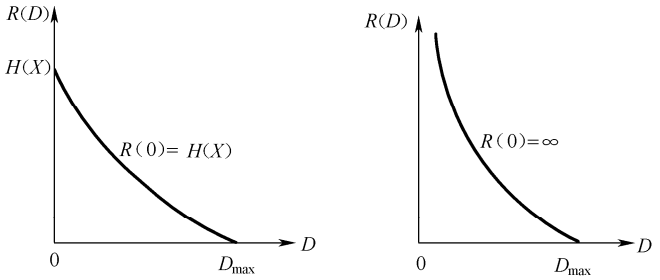
$$\begin{aligned} D(p_{ij}^{\alpha}) &= (1-\alpha)D(p_{ij}^{'}) + \alpha D(p_{ij}^{''}) \\ &= (1-\alpha)D(p_{ij}^{'}) + \alpha D_{\max} = D^{\alpha} \end{aligned}$$

所以 $p_{ij}^{\alpha} \in P_{D^{\alpha}}$，有

$$
\begin{aligned}
R(D^{\alpha}) &= \min_{p_{ij}\in P_{D^{\alpha}}} I(p_{ij}) \leqslant I(p_{ij}^{\alpha}) \\
&\leqslant (1-\alpha) I(p_{ij}^{'}) + \alpha I(p_{ij}^{''}) \\
&= (1-\alpha) I(p_{ij}^{''}) < R(D^{'})
\end{aligned}
$$

即 $R(D^{\alpha}) \neq R(D^{'})$，所以 $R(D)$函数是严格递减的。

根据 $R(D)$函数的定义域和它的性质，可以画出 $R(D)$～D 的函数曲线如图 4-2 所示。



（a）离散信源　　（b）连续信源

图 4-2　信源的 $R(D)$函数

由图 4-2 可见，当规定了允许失真 D，又确定了适当的失真测量 $d(x_i, y_j)$，就可以找到该失真条件下的最小信息率 $R(D)$，这个最小的信息率是一个极限值。采用不同的方法进行数据压缩（即信源编码）时，其压缩的程度如何，$R(D)$函数就是对数据进行压缩的一把尺子。

4.3 离散信源的 $R(D)$函数及其计算

给定信源的概率 p_i 及失真函数 $d(x_i, y_j)$，就可以求出该信源的 $R(D)$函数。求解 $R(D)$函数，实质上就是在保真度准则之下求解互信息的极小值问题。

$$
R(D) = \min_{p_{ij}\in P_D} \sum_{i,j} p_i p_{ij} \log \frac{p_{ij}}{p_j} \tag{4-12}
$$

实际上就是选择一个条件概率 p_{ij}，使 $I(X;Y)$最小，所需的约束条件如下。

① $\sum_{i,j} p_i p_{ij} d_{ij} = D$

② $\sum_{j} p_{ij} = 1$，$i=1,2,\cdots,m$

③ $p_{ij} \geqslant 0$，$i=1,2,\cdots,m$；　$j=1,2,\cdots,n$

求解这类问题的比较简单的方法是拉格朗日乘子法。我们知道，拉格朗日乘子法不能处理不等式的约束关系。因此，先不考虑约束条件③，由条件①和条件②的$(m+1)$个约束方程，可用$(m+1)$个乘子 $\mu_i (i=1,2,\cdots,m)$ 和 S（待定参数）来构造一个辅助方程。

$$
J(p_{ij}) = I(p_{ij}) - \sum_{i} \mu_i \sum_{j} p_{ij} - S \sum_{i,j} p_i p_{ij} d_{ij}
$$

为了求出 $J(p_{ij})$ 的无条件极值，只要对 p_{ij} 求导并令其为 0，就可得到一组以 p_{ij} 为未知数，

S 和 μ_i 为参数的方程组，然后再借助于(m+1)个约束方程来确定(m+1)个参数即可求解。

令 $\dfrac{\mathrm{d}J(p_{ij})}{\mathrm{d}p_{ij}}=0$，有

$$\frac{\mathrm{d}}{\mathrm{d}p_{ij}}[I(p_{ij})-\sum_i \mu_i \sum_j p_{ij}-S\sum_{i,j} p_i p_{ij} d_{ij}]=0$$

得

$$p_i \log\frac{p_{ij}}{p_j}-\mu_i-Sp_i d_{ij}=0$$

或写为

$$\log\frac{p_{ij}}{p_j}=\frac{\mu_i}{p_i}-Sd_{ij}$$

设 $\log\lambda_i=\dfrac{\mu_i}{p_i}, i=1,2,\cdots,m$，有

$$\log\frac{p_{ij}}{p_j}=\log(\lambda_i \mathrm{e}^{Sd_{ij}})$$

得

$$p_{ij}=p_j\lambda_i \mathrm{e}^{Sd_{ij}} \tag{4-13}$$

将该式代入约束条件②，有

$$\sum_j p_{ij}=\sum_j p_j\lambda_i \mathrm{e}^{Sd_{ij}}=1$$

可求得

$$\lambda_i=\frac{1}{\sum_j p_j \mathrm{e}^{Sd_{ij}}},\quad i=1,2,\cdots,m \tag{4-14}$$

对式（4-13）两端同乘以 p_i，并对 i 求和，得

$$\sum_i p_i p_{ij}=p_j=\sum_i p_i p_j \lambda_i \mathrm{e}^{Sd_{ij}}=p_j\sum_i p_i\lambda_i \mathrm{e}^{Sd_{ij}}$$

得到 $\sum_i p_i\lambda_i \mathrm{e}^{Sd_{ij}}=1$。将（4-14）式所得代入上式，有

$$\sum_i p_i \frac{\mathrm{e}^{Sd_{ij}}}{\sum_j p_j \mathrm{e}^{Sd_{ij}}}=1 \tag{4-15}$$

因此，当信源给定时，p_i 为已知，就可由该式的 n 个 p_i 的方程组解出 $p_j(j=1,2,\cdots,n)$。

利用（4-15）式解出的 p_j 中含有一个待定参数 S，为了保证解出的每一个 p_j 均为非负值（$p_j \geqslant 0$），需要对参数 S 加以限制。

从理论上来说，对于给定的平均失真 D，可以解出 S，但人们并不去求解它。这是因为 S 有明确的几何意义和应用，后面将说明 S 的几何意义。

利用式（4-15）求解出这些 p_j 值，就可由式（4-14）求出对应的 m 个 λ_i 值。当 p_j 、 λ_i 已知时，由式（4-13）即可求出相应的 $m \times n$ 个 p_{ij} 的值。

将解出 p_{ij} 的值代入平均失真的公式中，可解出随 S 参数值变化的 D 值，即

$$D(S)=\sum_i\sum_j p_i p_j \lambda_i d_{ij}=\sum_i\sum_j p_i p_{ij} \lambda_i \mathrm{e}^{Sd_{ij}} d_{ij} \tag{4-16}$$

信源的信息率失真函数 $R(D)$为

$$\begin{aligned} R(S)&=\sum_i\sum_j p_i p_j \lambda_i \mathrm{e}^{Sd_{ij}} \log\frac{p_j \lambda_i \mathrm{e}^{Sd_{ij}}}{p_j} \\ &=S\sum_i\sum_j p_i p_j \lambda_i \mathrm{e}^{Sd_{ij}} d_{ij}+\sum_i\sum_j p_i p_{ij} \log\lambda_i \\ &=SD(S)+\sum_i p_i \log\lambda_i \sum_j p_{ij} \\ &=SD(S)+\sum_i p_i \log\lambda_i \end{aligned} \tag{4-17}$$

此即用参数 S 表示的 $R(D)$函数的表达式，相应的 $R(D)$用 $R(S)$代表。给出一个具体的 S 值，就可按照顺序求出对应的 p_j 、 λ_i 、 p_{ij} 、$D(S)$、$R(S)$的值。

参数 S 实际上就是 $R(D)$函数的斜率，即 $R'(D)=S$ 。

证明　对 $R(S)$的表达式取导数，有

$$\begin{aligned} R'(D)&=\frac{\mathrm{d}R}{\mathrm{d}D}=S+D\cdot\frac{\mathrm{d}S}{\mathrm{d}D}+\sum_i\frac{p_i}{\lambda_i}\cdot\frac{\mathrm{d}\lambda_i}{\mathrm{d}D} \\ &=S+\left[D+\sum_i\frac{p_i}{\lambda_i}\cdot\frac{\mathrm{d}\lambda_i}{\mathrm{d}S}\right]\frac{\mathrm{d}S}{\mathrm{d}D} \end{aligned}$$

再由式 $\sum_i p_i \lambda_i \mathrm{e}^{Sd_{ij}}=1$ 对 S 取导数，得

$$\sum_i\left(\lambda_i d_{ij}+\frac{\mathrm{d}\lambda_i}{\mathrm{d}S}\right)p_i \mathrm{e}^{Sd_{ij}}=0$$

两边乘以 p_j ，并对 j 求和，得

$$\sum_i\sum_j p_i p_j \lambda_i \mathrm{e}^{Sd_{ij}} d_{ij}+\sum_i\sum_j p_i \frac{\mathrm{d}\lambda_i}{\mathrm{d}S} p_j \lambda_i \mathrm{e}^{Sd_{ij}}=0$$

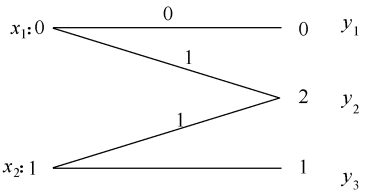
将式（4-16）和式（4-14）代入上式，得

$$D(S)+\sum_i\frac{p_i}{\lambda_i}\cdot\frac{\mathrm{d}\lambda_i}{\mathrm{d}S}=0$$

即 $R'(D)=S$ 。

由 $R(D)$函数的性质可知，$R(D)$是 D 的单调非增函数，表明 $R'(D)=S$ 取负值，当 S 由 $-\infty$ 上升到 0 时，对应的 D 值由 0 上升到 $D_{\max}$ 值。

例题 4-6　设 X 为二元等概率平稳无记忆信源，规定失真函数如下。



失真矩阵为 $\boldsymbol{d}=\begin{bmatrix}0 & 1 & \infty\\ \infty & 1 & 0\end{bmatrix}$，试求其信息率失真函数 $R(D)$。

解 计算平均失真 D

$$D=\sum_i\sum_j p(x_iy_j)d(x_i,y_j)$$
$$=\sum_i\sum_j p_ip_{ij}d_{ij}$$

已知 $\{p_i\}=\left\{\frac{1}{2},\frac{1}{2}\right\}$，若最大允许失真为 D，则条件转移概率 p_{ij} 与失真测度构成一一对应关系。

本例中，d_{ij} 为对称型，则 p_{ij} 也为对称型。由概率的归一性，可进一步假设为

$$[p_{ij}]=\begin{bmatrix}A & 1-A & 0\\ 0 & 1-A & A\end{bmatrix}$$

则平均失真为

$$D=\sum_{i,j}p_ip_{ij}d_{ij}$$
$$=\frac{1}{2}[A\times0+(1-A)\times1+0\times\infty]+\frac{1}{2}[0\times\infty+(1-A)\times1+A\times0]$$
$$=\frac{1}{2}(1-A)+\frac{1}{2}(1-A)$$
$$=1-A$$

所以 A=1−D，则条件概率为

$$[p_{ij}]=\begin{bmatrix}1-D & D & 0\\ 0 & D & 1-D\end{bmatrix}$$

由 $p_j=\sum_i p_ip_{ij}$，求得 $\{p_j\}=\{\frac{1-D}{2},D,\frac{1-D}{2}\}$；得到

$$H(Y)=H[p_j]=H\left[\frac{1-D}{2},D,\frac{1-D}{2}\right]$$
$$H[Y\mid X]=H[p_{ij}]=H[1-D,D]$$

信息率失真函数为

$$R(D)=I(X;Y)$$
$$=H(Y)-H(Y\mid X)$$
$$=H\left(\frac{1-D}{2},D,\frac{1-D}{2}\right)-H(1-D,D)$$

$$=-2\cdot\frac{1-D}{2}\log\frac{1-D}{2}+(1-D)\log(1-D)$$
$$=(1-D)\log 2$$

任给一个 D 值，就可求出相应的 $R(D)$值，作出曲线如图 4-3 所示。

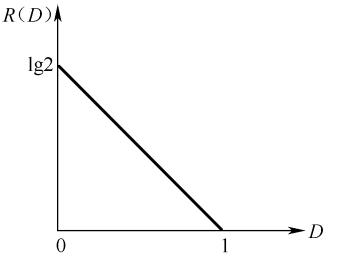


图 4-3　$R(D)\sim D$曲线

可以看到，当 $D=0$ 时，$R(D)=H(X)=\log 2$，$D_{\max}=1$；$D>1$ 时，$R(D)=0$。在这种特殊情况下，可以求出信源的 $R(D)\sim D$ 曲线的显式表达式。

例题 4-7　设 X 为 n 元等概率、平稳无记忆信源，规定失真函数为

$$[d_{ij}]=\begin{bmatrix}0 & & & 1\\ & 0 & & \\ & & \ddots & \\ 1 & & & 0\end{bmatrix}$$

计算该信源的 $R(D)$函数。

解　计算该信源的 $R(D)$函数可采用两种方法进行。

方法一：对于等概率、对称性失真函数的信源来说，仍然像例题 4-6 一样，可以不用求解参数方程，而直接求解。

对应于该失真函数的转移概率矩阵应为

$$[p_{ij}]=\begin{bmatrix}A & \frac{1-A}{n-1} & \cdots & \frac{1-A}{n-1}\\ \frac{1-A}{n-1} & A & \cdots & \frac{1-A}{n-1}\\ \vdots & \vdots & \ddots & \vdots\\ \frac{1-A}{n-1} & \frac{1-A}{n-1} & \cdots & A\end{bmatrix}$$

已知，$p_i=\frac{1}{n}$，$i=1,2,\cdots,n$，则平均失真为

$$D=\sum_{i=1}^{n}\sum_{j=1}^{n}p_ip_{ij}d_{ij}=n\times\frac{1-A}{n-1}\times\frac{1}{n}(n-1)=1-A$$

可得 $A=1-D$。

输出符号的概率为

$$p_j=\sum_{i=1}^{n}p_ip_{ij}=\frac{A}{n}+(n-1)\times\frac{1-A}{n(n-1)}=\frac{1}{n},\ j=1,2,\cdots,n$$

$$H(Y)=H\left(\frac{1}{n},\frac{1}{n},...,\frac{1}{n}\right)=\log n$$

$$H(Y\mid X)=H\left(1-D,\frac{D}{n-1},\cdots,\frac{D}{n-1}\right)=-(1-D)\log(1-D)-D\log\frac{D}{n-1}$$

信息率失真函数为

$$\begin{aligned}R(D)&=I(X;Y)\\&=H(Y)-H(Y\mid X)\\&=\log n+(1-D)\log(1-D)-D\log(n-1)\\&=\log n-H[1-D,D]-D\log(n-1)\end{aligned}$$

给定不同的 n 值，可作出 $R(D)\sim D$ 变化的曲线，如图 4-4 所示。

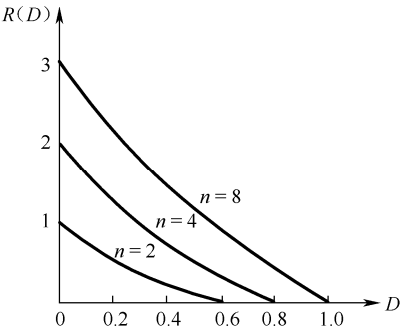


图 4-4　不同 n 值的 $R(D)\sim D$ 曲线

当 n=2,4,8 时，$R(0)$分别为 1bit，2bit，3bit。说明：要求无失真时，不同的进制所要求的信息率是不一样的。

允许一定的失真时，即 D 值一定时，进制数越少，对信源的压缩就越大；允许失真值 D 越大，对信源的压缩也越大。

方法二：利用式 $\sum_i p_i\lambda_i \mathrm{e}^{Sd_{ij}}=1$，确定 λ_i。将 i=1,2,⋯,n 代入，可得方程组为

$$\begin{cases}\lambda_1+\lambda_2\mathrm{e}^{S}+\cdots+\lambda_n\mathrm{e}^{S}=n\\ \lambda_1\mathrm{e}^{S}+\lambda_2+\cdots+\lambda_n\mathrm{e}^{S}=n\\ \cdots\\ \lambda_1\mathrm{e}^{S}+\lambda_2\mathrm{e}^{S}+\cdots+\lambda_n=n\end{cases}$$

解得 $\lambda_i=\dfrac{n}{1+(n-1)\mathrm{e}^{S}}$，　i=1,2,⋯,n。

由式 $\lambda_i=\dfrac{1}{\sum_j p_j\mathrm{e}^{Sd_{ij}}}$，$i=1,2,\cdots,n$ 确定 p_j，即

$$\begin{cases}p_1+p_2\mathrm{e}^{S}+\cdots+p_n\mathrm{e}^{S}=\dfrac{1+(n-1)\mathrm{e}^{S}}{n}\\ p_1\mathrm{e}^{S}+p_2+\cdots+p_n\mathrm{e}^{S}=\dfrac{1+(n-1)\mathrm{e}^{S}}{n}\\ \cdots\\ p_1\mathrm{e}^{S}+p_2\mathrm{e}^{S}+\cdots+p_n=\dfrac{1+(n-1)\mathrm{e}^{S}}{n}\end{cases}$$

解出 $p_j=\dfrac{1}{n}$，$j=1,2,\cdots,n$。

$$D(S)=\frac{(n-1)\mathrm{e}^{S}}{1+(n-1)\mathrm{e}^{S}}\rightarrow S=\log\frac{D}{(n-1)(1-D)}$$

$$\begin{aligned}R(S)&=SD(S)+\sum_{i=1}^{n}p_i\log\lambda_i\\&=D\log\frac{D}{(n-1)(1-D)}+\log n(1-D)\\&=\log n-D\log(n-1)-H[1-D,D]\end{aligned}$$

可以看到，用两种方法所得到的结果是一样的。

例题 4-8 设输入、输出符号表位 $X=Y=\{0,1\}$，输入符号的概率为 $p(x)=\{p,1-p\}$，$0<p\leqslant\frac{1}{2}$，失真函数为 $d_{ij}=\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$，求 $R(D)$。

解 先计算 λ_i，i=1,2，即 λ_1、λ_2。由式（4-11），可得

$$\begin{cases}\lambda_1 p_1 \mathrm{e}^{Sd_{11}}+\lambda_2 p_2 \mathrm{e}^{Sd_{21}}=1\\ \lambda_1 p_1 \mathrm{e}^{Sd_{12}}+\lambda_2 p_2 \mathrm{e}^{Sd_{22}}=1\end{cases}$$

解得 $\lambda_1=\dfrac{1}{p(1+\mathrm{e}^{S})}$，$\lambda_2=\dfrac{n}{(1-p)(1+\mathrm{e}^{S})}$。再计算输出符号的概率分布

$$\begin{cases}p(y_1)\mathrm{e}^{Sd_{11}}+p(y_2)\mathrm{e}^{Sd_{12}}=\dfrac{1}{\lambda_1}\\ p(y_1)\mathrm{e}^{Sd_{21}}+p(y_2)\mathrm{e}^{Sd_{22}}=\dfrac{1}{\lambda_2}\end{cases}$$

解得 $p(y_1)=\dfrac{p-(1-p)\mathrm{e}^{S}}{1-\mathrm{e}^{S}}$，$p(y_2)=\dfrac{1-p-p\mathrm{e}^{S}}{1-\mathrm{e}^{S}}$。

计算平均失真，即

$$D=\lambda_1 pp(y_1)d_{11}\mathrm{e}^{Sd_{11}}+\lambda_1 pp(y_2)d_{12}\mathrm{e}^{Sd_{12}}+\lambda_2(1-p)p(y_1)d_{21}\mathrm{e}^{Sd_{21}}+\lambda_2(1-p)p(y_2)d_{22}\mathrm{e}^{Sd_{22}}$$
$$=\frac{\mathrm{e}^{S}}{1+\mathrm{e}^{S}}$$

则信息率失真函数为

$$\begin{aligned}R(D)&=SD+p\log\lambda_1+(1-p)\log\lambda_2\\&=-[p\log p+(1-p)\log(1-p)]+[D\log D+(1-D)\log(1-D)]\\&=H[p,1-p]-H[D,1-D]\end{aligned}$$

$H[p,1-p]$为信源熵，$H[D,1-D]$为允许一定的失真而可以压缩的信息率，$D_{\max}=p$。对于不同的 p 值，可以作出一组 $R(D)$曲线，如图 4-5 所示。

对于给定的平均失真 D，信源分布越均匀，$R(D)$越大，可压缩性越小；反之，信源分布越不均匀，$R(D)$越小，可压缩性越大。

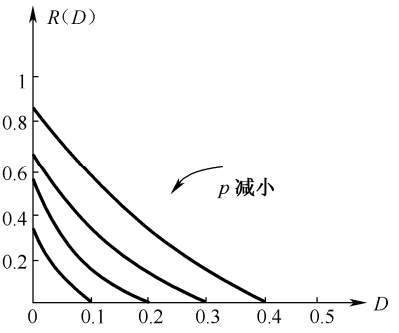


图 4-5 不同 p 值的 $R(D)\sim D$ 曲线

通过以上所举的例子可以看到，要计算实际信源的 $R(D)$函数的显式表达式是相当困难的，一般情况下是通过计算机进行迭代运算而得到结果的。

根据 $R(D)$函数的定义，要寻求迭代算法的关键在于首先寻找一对描述互信息量的、互为因果关系的自变量。通常选 p_j 与 p_{ij}，互信息表示为 $I(p_j;p_{ij})$。当信源给定时，p_i 为已知，那么求互信息的极小值是通过改变 p_{ij} 来实现的。

在含有参数 S 的 $R(D)$函数表达式中，曾经得到

$$p_j=\sum_i p_i p_{ij} \quad 及 \quad p_{ij}=\frac{p_j \mathrm{e}^{Sd_{ij}}}{\sum_j p_j \mathrm{e}^{Sd_{ij}}}$$

这两个式子就是要寻找的含有参数 S 的互为因果关系的迭代表达式。

假设一个 S 值，逐次计算出 p_j 与 p_{ij}，代入互信息 $I(p_j;p_{ij})$的表达式中，就能实现对 $R(S)$的迭代运算。再对不同的 S 值反复进行迭代，最终求得 $R(D)$函数的曲线。

为了运算方便，将 p_j、p_{ij} 改写为

① $p_j^n=\sum_i p_i p_{ij}^n$，② $p_{ij}^{n+1}=\frac{p_j^n \mathrm{e}^{Sd_{ij}}}{\sum_j p_j^n \mathrm{e}^{Sd_{ij}}}$。

计算 $R(D)$的迭代算法和步骤如下。

（1）令 n=1，对给定的信源 $p_i=p_i^0$，取一个初始的条件概率分布，如取等概率分布为 $p_{ij}^1=\frac{1}{m}$，由①式计算 p_j^i。

（2）对任何 S<0，总在∞～0 之间取 n 个点，先去一个最大的负值 $S=S_1$，对给定的 d_{ij} 按照②式计算 p_{ij}^2。

（3）将 p_j^1、p_{ij}^2 的数值代入互信息公式，得

$$I(p_j^1;p_{ij}^2)=R(1,2)$$

（4）取 n=n+1，重复上述运算，可计算出

$$p_{ij}^1=\frac{1}{m}\to p_j^1\to p_{ij}^2\to p_j^2\to\cdots\to p_{ij}^r\to p_j^r$$

求得相应的信息率为

$$R(1,1)\geqslant R(1,2)\geqslant R(2,2)\geqslant\cdots\geqslant R(r-1,r)\geqslant R(r,r)\to R(S_1)$$

（5）若 $R(r,r)-R(r-1,r)>\varepsilon$，回到（4）重新计算，直到其误差小于 ε 时，停止运算。

这里：$R(r-1,r)=\min_{p_{ij}} I(p_j^{r-1};p_{ij}^r)$

$$R(r,r)=\min_{p_j} I(p_j^r;p_{ij}^r)$$

$$R(s)=\lim_{r\to\infty}\min_{p_{ij},p_j} I(p_j^r;p_{ij}^r)$$

（6）重新从 n=1 开始，并取第二个 S_{21} 值，重复上述过程，求出 $R(S_2)$。

（7）至少取 4～5 个不同的 S 值，计算出对应的 $R(S)$。最后将几个点的 $R(S)$值连接起来，就构成了已知信源的 $R(D)$～D 曲线。

4.4 连续信源的 $R(D)$函数及其计算

4.4.1 幅度连续无记忆信源的 $R(D)$函数

在通信中，除了数字式信源和可离散化的数字信源外，还存在着大量的连续的模拟信源，如语音、图像等。这类信源输出的是一个连续的模拟量，并且是不确定的，可将其看做是一

个随机过程。

连续信源中一系列的问题在概念上与离散信源是不同的，但也存在一些共同点。连续随机变量可看做是离散随机变量的极限，可用离散随机变量来逼近。

1．失真函数

设 $X=(x_1,x_2,\cdots,x_n)$ 为时间离散、幅度连续的信源输出，对其进行编码后的输出为 $Y=(y_1,y_2,\cdots,y_n)$，1,2,…,n 为时间标志。由源字 X 到编码输出 Y 所产生的失真为平均值测量。

$$d_n(X,Y)=\frac{1}{n}\sum_{t=1}^{n}d(x_t,y_t) \tag{4-18}$$

将函数族 $F_d=\{d_n,1\leqslant n<\infty\}$ 作为由 d 产生的单字符保真度测量标准。d(x,y)可以有不同的表达形式。

2．$R(D)$函数

对于一个幅度连续的信源，假定条件概率密度函数为 $p(y/x)$，失真函数为 d(x,y)，则平均失真为

$$D=\iint p(x)p(y\mid x)\mathrm{d}(x,y)\mathrm{d}x\mathrm{d}y \tag{4-19}$$

平均互信息量为

$$I[q(y);p(y\mid x)]=\iint p(x)p(y\mid x)\log\frac{p(y\mid x)}{q(y)}\mathrm{d}x\mathrm{d}y \tag{4-20}$$

其中，$q(y)=\int p(x)p(y\mid x)\mathrm{d}x$ 为输出变量的概率密度函数。

对于连续信源的信息率失真函数，要解决的问题仍然是在一定的保真度要求下，使平均误差小于某一个限度的条件下最起码的平均互信息量是多少？由平均失真的定义式可见，实际上是要确定一个合适的 $p(y|x)$。每一个 $p(y|x)$都有一个对应的平均互信息量 $I[q(y);\ p(y|x)]$。因此定义平均互信息量的下确界为 $R(D)$函数，即

$$R(D)=\inf_{p(y|x)\in P_{D'}}I[q(y);p(y\mid x)] \tag{4-21}$$

其中，$P_{D'}=\{p(y\mid x);D\leqslant D'\}$。inf 表示下确界，对应于离散信源的极小值。在连续信源情况下，不一定有极小值，但可以有下确界。

求下确界的问题，仍然是一个极值问题，是求泛函的极值。因此不是求偏导而是求变分了。平均互信息量 $I[q(y);p(y|x)]$是变量 $p(y|x)$的函数，称 $p(y|x)$为宗量，I 为泛函。

因此，求解连续信源的信息率失真函数 $R(D)$，是利用变分法求出 $R(D)$函数的参数表达式，类似于离散信源的求极值。所需的约束条件为：

① $D=\iint p(x)p(y\mid x)d(x,y)\mathrm{d}x\mathrm{d}y$；

② $\int p(y\mid x)\mathrm{d}y=1$。

在这两个约束条件下，求 I 对 $p(y|x)$的下确界。

类似于离散信源求极值，采用拉格朗日乘子法，引入待定参数 S、乘子（函数）$\mu(x)$，将条件极值化为无条件极值，然后对 $p(y|x)$取变分，并令其为 0。求解过程类似于离散信源，

略去数学推导过程，求出的 $R(D)$函数的参数表达式为

$$R(S) = SD(S) + \int p(x)\log\lambda(x)\mathrm{d}x \tag{4-22}$$

$$D(S) = \iint p(x)q(y)\lambda(x)\mathrm{e}^{Sd(x,y)}\,d(x,y)\mathrm{d}x\mathrm{d}y \tag{4-23}$$

其中的 $\lambda(x)$ 为

$$\lambda(x) = \frac{1}{\int q(y)\mathrm{e}^{Sd(x,y)}\mathrm{d}y} \tag{4-24}$$

而参数 S 就是 $R(D)$函数的斜率，有 $S<0$。

从式（4-22）、式（4-23）、式（4-24）中可以看到，其结果非常类似于离散信源的结果。但是在一般情况下，要计算出 $p(y|x)$是很困难的。只是在某些特殊情况下，求解较简单，可得到相应的 $R(D)$函数的表达式。

4.4.2 差值误差测量与香农界

当失真函数为差值的情况下，其 $R(D)$函数的求解比较简单，可得到其显式表达式。

1．差值变量的概率密度函数

失真函数为差值变量即 $\mathrm{d}(x,y)=\mathrm{d}(x-y)$。从 $\lambda(x)$ 的表达式中可以看到，$\lambda(x)$ 中含有参数 S，并且与输入随机变量 x 的概率密度 $p(x)$成反比。因此，可以假设

$$\lambda(x) = \frac{K(S)}{p(x)} \tag{4-25}$$

由式

$$\begin{cases} \int \lambda(x)p(x)\mathrm{e}^{\mathrm{Sd}(x,y)}\mathrm{d}x = 1, & q(y) \neq 0 \\ \int \lambda(x)p(x)\mathrm{e}^{\mathrm{Sd}(x,y)}\mathrm{d}x \leqslant 1, & q(y) = 0 \end{cases}$$

得

$$K(S)\int \mathrm{e}^{\mathrm{Sd}(x,y)}\mathrm{d}x \leqslant 1 \tag{4-26}$$

寻找合适的 $K(S)$使等式成立，此时

$$K(S) = \frac{1}{\int \mathrm{e}^{\mathrm{Sd}(x,y)}\mathrm{d}x} \tag{4-27}$$

设失真函数为 $\mathrm{d}(x,y)=\mathrm{d}(x-y)$，令 $z=x-y$，将 y 作为参变量，则有 $\mathrm{d}z=\mathrm{d}x$，可得

$$K(S) = \frac{1}{\int \mathrm{e}^{\mathrm{Sd}(z)}\mathrm{d}z}$$

或

$$\int \mathrm{e}^{\mathrm{Sd}(z)} \cdot K(S)\mathrm{d}z = 1$$

将 $\mathrm{e}^{\mathrm{Sd}(z)}\,K(S)$ 当作一个以 S 为参数的函数，记为 $g_{\mathrm{S}}(z)$。可以看到，$g_{\mathrm{S}}(z)$ 满足下列条件。

① 非负性：$g_{\mathrm{S}}(z) \geqslant 0$。

② 归一性：$\int g_S(z)\mathrm{d}z=1$。

说明 $g_S(z)$ 具有概率密度函数的特性。

因此，定义差值变量 z 的概率密度函数为

$$g_S(z)=K(S)\mathrm{e}^{Sd(z)}=\frac{\mathrm{e}^{Sd(z)}}{\int \mathrm{e}^{Sd(z)}\mathrm{d}z} \tag{4-28}$$

由式（4-25）可得

$$\begin{aligned} p(x)&=\lambda^{-1}(x)K(S)\\ &=K(S)\int q(y)\mathrm{e}^{Sd(x-y)}\mathrm{d}y\\ &=\int q(y)K(S)\mathrm{e}^{Sd(x-y)}\mathrm{d}y\\ &=\int q(y)g_S(x-y)\mathrm{d}y\\ &=g_S(x)*q(x) \end{aligned} \tag{4-29}$$

说明：信源的概率密度为输出变量 y 的概率密度和差值变量的概率密度的卷积。相当于 $q(x)$经过了一个信道滤波函数 $g_S(x)$，输出为 $p(x)$。

随机变量的概率密度函数的傅里叶变换为其特征函数。因此，对 $p(x)=g_S(x)*q(x)$ 两端进行傅里叶变换，有

$$\int_{-\infty}^{\infty}p(x)\mathrm{e}^{\mathrm{j}tx}\mathrm{d}x=[\int_{-\infty}^{\infty}g_S(x)\mathrm{e}^{\mathrm{j}tx}\mathrm{d}x]\cdot\int_{-\infty}^{\infty}q(x)\mathrm{e}^{\mathrm{j}tx}\mathrm{d}x]$$

即

$$\Phi_p(t)=\Phi_g(t)\cdot\Phi_q(t)\rightarrow\Phi_q(t)=\frac{\Phi_p(t)}{\Phi_g(t)}$$

求其傅里叶反变换，有

$$q(y)=\frac{1}{2\pi}\int_{-\infty}^{\infty}\Phi_g(t)\mathrm{e}^{-\mathrm{j}ty}\mathrm{d}t$$

由 $q(y)$即可解出 $\lambda(x)$、$D(S)$及 $R(S)$。

2．均方误差测量下的 $R(D)$函数

均方误差测量时，$d(x-y)=(x-y)^2=z^2$。差值变量的概率密度函数为

$$g_S(z)=\frac{\mathrm{e}^{Sz^2}}{\int \mathrm{e}^{Sz^2}\mathrm{d}z}$$

由于斜率 S 总是取负值，即 $S<0$，可用$-|S|$表示 S。求积分

$$\int_{-\infty}^{\infty}\mathrm{e}^{-|S|z^2}\mathrm{d}z=2\int_0^{\infty}\mathrm{e}^{-|S|z^2}\mathrm{d}z=\sqrt{\frac{\pi}{|S|}}$$

有

$$g_S(z)=\sqrt{\frac{|S|}{\pi}}\cdot\mathrm{e}^{-|S|z^2}$$

特征函数为

$$\Phi_{\mathrm{g}}(t)=\int_{-\infty}^{\infty}\mathrm{e}^{jtz}\cdot\sqrt{\frac{\pi}{|S|}}\cdot\mathrm{e}^{-|S|z^2}\mathrm{d}z=\mathrm{e}^{-\frac{t^2}{4|S|}}$$

输出 y 的特征函数为

$$\Phi_{\mathrm{q}}(t)=\Phi_{\mathrm{p}}(t)\cdot\Phi_{\mathrm{g}}(t)=\Phi_{\mathrm{p}}(t)\cdot\mathrm{e}^{-\frac{t^2}{4|S|}}$$

求反变换，得

$$q(y)=\frac{1}{2\pi}\int\Phi_{\mathrm{p}}(t)\cdot\mathrm{e}^{-\frac{t^2}{4|S|}-jty}\mathrm{d}t$$

含有参数 S 的平均失真为

$$\begin{aligned}D(S)&=\iint\lambda(x)p(x)q(y)\mathrm{e}^{-|S|z^2}\cdot z^2\mathrm{d}x\mathrm{d}y\\&=\int K(S)\mathrm{e}^{-|S|z^2}\cdot z^2\mathrm{d}z\cdot\int q(y)\mathrm{d}y\\&=\int g_{\mathrm{S}}(z)\cdot z^2\mathrm{d}z\\&=\int\sqrt{\frac{|S|}{\pi}}\cdot\mathrm{e}^{-|S|z^2}\cdot z^2\mathrm{d}z\\&=\frac{1}{2|S|}\end{aligned}$$

即 $S=-\dfrac{1}{2D}$。

含有参数 S 的信息率失真函数 $R(S)$为

$$\begin{aligned}R(S)&=SD+\int p(x)\log\lambda(x)\mathrm{d}x\\&=SD+\int p(x)\log K(S)\mathrm{d}x-\int p(x)\log p(x)\mathrm{d}x\\&=H_{\mathrm{c}}(X)+SD-\log[\int\mathrm{e}^{-|S|z^2}\mathrm{d}z]\\&=H_{\mathrm{c}}(X)-\frac{1}{2}\log(2\pi\mathrm{e}D)\end{aligned}$$

$H_{\mathrm{c}}(X)$ 为信源的微分熵，$\dfrac{1}{2}\log(2\pi\mathrm{e}D)$ 为允许的失真为 D 时而压缩掉的信息量。给定 $p(x)$ 时，即可求得 $R(D)$。

例如，若信源为高斯信源，其概率密度为正态分布时

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

微分熵为

$$\begin{aligned}H_{\mathrm{c}}(X)&=-\int p(x)\log p(x)\mathrm{d}x\\&=\frac{1}{2}\log(2\pi\mathrm{e}\sigma^2)\end{aligned}$$

$R(D)$函数为

$$R(D)=\frac{1}{2}\log(2\pi e\sigma^2)-\frac{1}{2}\log(2\pi eD)=\frac{1}{2}\log\frac{\sigma^2}{D}$$

任给一个 D 值，就可以得到高斯信源的 $R(D)$函数。

在 $0<D\leqslant D_{\max}$ 域内，$R(D)$保持下凸、单调下降，斜率 $S<0$，$D=0$ 时 $R(0)=\infty$。即连续信源的信息量为无限大，要无失真地用数字信号来传送模拟信号，则需要无限大的信息速率。

其实，包含在连续信源中的信息不需要全部传送，有一定的失真也不是完全不允许。正因为如此，数字化才成为可能。

3．绝对误差测量下的 $R(D)$函数

失真函数为绝对失真时，$d(x,y)=d(x-y)=|x-y|=|z|$。差值变量的概率密度函数为

$$g_S(z)=\frac{e^{-|S|\cdot|z|}}{\int e^{-|S|\cdot|z|}dz}=\frac{|S|}{2}e^{-|Sz|}$$

平均失真为

$$D(S)=\int g_S(z)\cdot|z|\,dz=\frac{1}{|S|},\quad S=-\frac{1}{D}$$

信息率失真函数为

$$\begin{aligned}R(D)&=SD+H_c(X)-\log(\int e^{-|Sz|}dz)\\&=H_c(X)-\log(2eD)\end{aligned}$$

该 $R(D)$函数是否有效，还需要检验 $q(y)$是不是符合概率密度函数的定义。利用特征函数，有

$$\Phi_g(t)=\int g_S(z)e^{jtz}dz=\frac{S^2}{S^2+t^2}$$

$$\Phi_q(t)=\frac{\Phi_p(t)}{\Phi_g(t)}=\Phi_p(t)+\frac{t^2}{S^2}\Phi_p(t)$$

求反变换：$q(y)=p(y)-D^2p''(y)$。

对所有 y 来说，$q(y)\geqslant 0$；对给定的 $p(x)$，比如柯西分布

$$p(x)=\frac{2}{\pi}(1+x^2)^{-2}$$

有

$$q(y)=\frac{2}{\pi}(1+y^2)^{-4}[y^4+2(1-10D^2)y^2+4D^2+1]$$

要满足 $q(y)\geqslant 0$，则需要 $[y^4+2(1-10D^2)y^2+4D^2+1]>0$，可得到条件为 $D\leqslant\frac{\sqrt{6}}{5}$。同时，所有 y 应满足 $q(y)\leqslant 1$ 的条件，包括 $y=0$，有

$$q(0)=\frac{2}{\pi}(4D^2+1)\leqslant 1$$

得到 $D\leqslant\sqrt{\frac{\pi-2}{8}}=0.38$。综合两个条件，取 $D\leqslant 0.38$。因此，在 $0\leqslant D\leqslant 0.38$ 的域内，由 $R(D)=H_c(X)-\log(2eD)$ 式所求得的就是信源的信息率失真函数，是满足约束条件的解。式

$R(D)=H_c(X)-\log 2eD$ 称为香农低界。

4.4.3 带记忆的信源的 $R(D)$函数

对于有记忆的信源来说，其分析、计算更加困难。但这一类信源更接近于实际情况，也更适合于进行数据压缩的研究对象。

有记忆的信源存在着统计相关性，这一点可以充分地加以利用。我们仅仅给出相当特殊情况下的一些结论和性能。

当最大的允许失真 D 给定时，和无记忆信源相比，有记忆信源能够得到较大的数据压缩。对于正态分布的信源，有如下结论。

$$R(D)|_{有记忆}\leqslant R(D)|_{无记忆}\leqslant R(D)|_{无记忆白色谱}$$

这一结论往往用 R 的逆函数表示而更加直观。当给定信源的速率 R 时，在小失真范围内，表达式为

$$D(R)|_{有记忆}=r^2D(R)|_{无记忆}\leqslant D(R)|_{无记忆白色谱}$$

其中，$r^2\leqslant 1$ 是 X 序列的谱平坦度。对白色正态噪声，有

$$\Phi(\omega)=\sigma^2=常数，\ r^2=1$$

该式表明，有记忆信源的失真与无记忆信源的失真相比，可以降低 r^2 倍。

例题 4-9 讨论具有相关系数为 ρ 的一阶正态马尔可夫信源的信息率失真函数 $R(D)$。

假设信源正态分布的均值为 0、方差为 σ^2，其相关函数（协方差矩阵）为

$$R_{xx}=\rho^{|k|}\cdot\sigma^2$$

功率谱密度为

$$\Phi(\omega)=\frac{1-\rho^2}{1+\rho^2-2\rho\cos\omega t}\sigma^2$$

当 $\rho\geqslant 0$ 时，功率谱密度在 $\omega t=\pi$ 上有最小值，即

$$\min_{\omega}\{\Phi(\omega)\}=\frac{1-\rho^2}{(1+\rho)^2}\sigma^2=\frac{1-\rho}{1+\rho}\sigma^2$$

在小失真区域内，可求得

$$R(0)=\frac{1}{2}\log\frac{\sigma^2(1-\rho^2)}{D}\ 且\ D\leqslant\frac{1-\rho}{1+\rho}\sigma^2$$

可以看出，当 ρ 增大时，$R(D)$是减小的，即 $\rho\uparrow\rightarrow$ 相关性强 $\rightarrow R(D)\downarrow\rightarrow$ 压缩比大 。

习　　题

4.1　设输入符号为 $X\in\{0,1\}$，输出符号为 $Y\in\{0,1\}$。定义失真函数为

$$d(0,0)=d(1,1)=0$$
$$d(0,1)=d(1,0)=1$$

试求失真矩阵 d。

4.2 某信源有3个消息，概率分布为$\begin{bmatrix} X \\ P(x) \end{bmatrix}=\begin{bmatrix} x_1 & x_2 & x_3 \\ 0.2 & 0.3 & 0.5 \end{bmatrix}$，失真矩阵为$\boldsymbol{d}=\begin{bmatrix} 4 & 2 & 1 \\ 0 & 3 & 2 \\ 2 & 0 & 1 \end{bmatrix}$，求$D_{\min}$和$D_{\max}$。

4.3 某无记忆信源X，概率空间为$\begin{bmatrix} X \\ P(x) \end{bmatrix}=\begin{bmatrix} -1 & 0 & 1 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}$，接收符号集为$Y=\left\{-\frac{1}{2},\frac{1}{2}\right\}$，其失真矩阵为$\begin{bmatrix} 1 & 2 \\ 1 & 1 \\ 2 & 1 \end{bmatrix}$，求$D_{\min}$和$D_{\max}$，并求选择何种信道可达到该$D_{\min}$和$D_{\max}$的失真度。

4.4 一个四元对称信源$\begin{bmatrix} X \\ P(x) \end{bmatrix}=\begin{bmatrix} 0 & 1 & 2 & 3 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix}$，接收符号$Y=\{0,1,2,3\}$，其失真矩阵为$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$，求$D_{\min}$、$D_{\max}$及信源的$R(D)$函数，并画出其曲线（取4至5个点）。

4.5 某二元信源$\begin{bmatrix} X \\ P(x) \end{bmatrix}=\begin{bmatrix} 0 & 1 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}$，其失真矩阵为$\begin{bmatrix} 0 & a \\ a & 0 \end{bmatrix}$，求$D_{\max}$、$D_{\min}$及信源的$R(D)$函数。

4.6 设信源$\begin{bmatrix} X \\ P(x) \end{bmatrix}=\begin{bmatrix} x_1 & x_2 \\ p & 1-p \end{bmatrix}$（$p<0.5$），其失真度为汉明失真度，试问当允许平均失真度$D=0.5p$时，每一信源符号平均最少需要几个二进制符号表示？

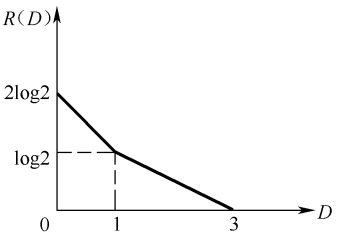
4.7 已知信源$X=\{0,1\}$，信宿$Y=\{0,1,2\}$。设信源符号为等概率分布，而且失真函数为$d=\begin{bmatrix} 0 & \infty & 1 \\ \infty & 0 & 1 \end{bmatrix}$，求信源的率失真函数$R(D)$。

4.8 设信源$X=\{0,1,2\}$，相应的概率分布为$P(0)=P(1)=0.4$，$P(2)=0.2$，且失真函数为$d_{ij}=\begin{cases} 0, i=j \\ 1, i\neq j \end{cases}$ $(i,j=0,1,2)$。求信源的率失真函数$R(D)$。

4.9 设离散无记忆信源$X=\{0,1,2,3\}$，信宿$Y=\{0,1,2,3,4,5,6\}$，且信源为等概率分布。失真函数定义为

$$d(x_i,y_j)=\begin{cases} 0, & i=j \\ 1, i=0,1\text{且}j=4 \\ 1, i=2,3\text{且}j=5 \\ 3, & j=6 \\ \infty, & \text{其他} \end{cases},$$

证明率失真函数$R(D)$如题图4-9所示。



题图 4-9

4.10 若有一个 n 元等概率、平稳无记忆信源 X，且规定失真函数为

$$(d_{ij})=\begin{pmatrix} 0 & \frac{1}{n-1} & \cdots & \frac{1}{n-1} \\ \frac{1}{n-1} & 0 & \cdots & \frac{1}{n-1} \\ \vdots & & & \vdots \\ \frac{1}{n-1} & \frac{1}{n-1} & \cdots & 0 \end{pmatrix}$$

试求率失真函数 $R(D)$。

4.11 用参量表达式方法求解一个二元不等概离散信源的率失真函数 $R(D)$。已知信源的概率分布为：$P(0)=p$，$P(1)=1-p$，且 $d_{ij}=\begin{cases}0, & \text{当}i=j,\\ 1, & \text{当}i\neq j,\end{cases}$，其中 i，$j=1,2$。

4.12 若连续信源的概率密度函数 $p(x)$ 和失真函数 $d(x,y)$ 为

$$\begin{cases} p(x)=\dfrac{1}{\sqrt{2\pi}\sigma}\mathrm{e}^{-\frac{(x-m)^2}{2\sigma^2}}, & \text{正态} \\ d(x,y)=(x-y)^2=\theta^2, & \text{均方准则} \end{cases}$$

试求 $R(D)$。

第5章 信源编码

内容简介

信源编码的目的是为了提高通信系统的有效性。由第 2 章我们已经知道，传送信源信息只需要信源极限熵大小的信息速率。但在实际的信息传输系统或通信系统中，用来传送信源信息的信息速率远远大于信源熵或 $R(D)$函数。

那么，由 $R(D)$函数计算出的信源熵这样的最小信息率的极限是否能够达到或接近达到呢？信源编码定理作出了肯定的回答，即在信源分块的符号数足够大的情况下，总可以找到一种编码方案，使信息速率可以压缩到信源熵的值。反过来说，当信息率小于信源熵时，在反变换或译码时，必然会引起失真或差错。

信源编码分为无失真编码和限失真编码，比如对图像、图形、语音等连续信源来说，压缩后必有失真；而对于数字、数据、文字等离散信源来说，可以做到压缩无失真。

对信源来说，其各个符号之间存在分布的不均匀性和相关性，使得信源存在冗余度。信源编码的主要任务就是减少冗余、提高编码的效率。具体地说，就是针对信源输出符号序列的统计特性，寻找一定的方法，把信源输出符号序列变换为最短的码字序列，即

符号序列→码字序列

信源编码的基本途径有两个：一是使序列中的各个符号尽可能地相互独立，即解除它们之间的相关性；二是使编码中各个符号出现的概率尽可能地相等，即概率均匀化。

信源编码的基础是信息论中的两个编码定理：①无失真编码定理；②限失真编码定理。无失真编码定理是可逆编码的基础，即当信源符号转换成代码后，可以从代码无失真地恢复出原信源符号。编码定理不但证明了必存在一种编码方法，使码字的平均长度可任意接近符号熵（但不能低于符号熵），而且还阐明了达到此目的的途径，就是使概率与码长相匹配。无失真编码只适用于离散信源。

对于连续信源来说，由于其取值可有无限多个，编成代码后就无法无失真地恢复出原来的连续值，即存在一定的失真。对于这个问题，将由限失真编码定理进行限失真编码。

5.1 编码的定义

将信源消息分成若干组，即符号序列 X，$X=(x_1x_2\cdots x_l\cdots x_L)$，序列中的每个符号取自于符号集 A，$x_l=(a_1,a_2,\cdots,a_n)$。

每个符号序列 X 依照固定的码表映射成一个码字 Y，$Y=(y_1y_2\cdots y_k\cdots y_K)$，其中 $y_k=\{b_1,b_2,\cdots,b_m\}$。这样的码称为分组码，或叫块码。分组码有对应的码表，非分组码不存在码表。

从信源序列到码字序列的信源编码器模型如图 5-1 所示。

信源产生 L 长的序列，经编码后输出为 K 长的码字。

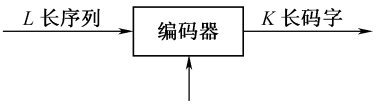


图 5-1 信源编码器模型

若假定信源符号集为 $X\in\{x_1,x_2,\cdots,x_n\}$，其概率分布为 $P=\{p(x_1),p(x_2),\cdots,p(x_n)\}$。将此信源 X 通过一个基本符号集为{0，1}的二元信道进行传输，必须先将信源符号 x_i 变换成由 0 和 1 符号组成的码符号序列，即先进行编码。

比如，对 4 个信源符号进行如下编码

$$x_1:00,\ x_2:01,\ x_3:10,\ x_4:11$$

或

$$x_1:0,\ x_2:01,\ x_3:001,\ x_4:111$$

码可分为两类：一是固定长度的码，码中所有码字的长度都相同，如上面第一种编码所形成的码字；二是可变长度码，码中的码字长度各不相同，如上面第二种编码所形成的码字。

奇异码和非奇异码：若信源符号和码字是一一对应的，则这种码称为非奇异码；若不是一一对应，则称为奇异码。

比如，$x_1:0$，$x_2:10$，$x_3:00$，$x_4:01$，这种编码后，信源符号和码字是一一对应的，就是非奇异码。而 $x_1:0$，$x_2:11$，$x_3:00$，$x_4:11$，这种编码后，信源符号和码字不是一一对应的，就是奇异码。

唯一可译码：任意有限长的码元序列，只能被唯一地分割成一个个的码字，这种码称为唯一可译码。比如

$$x_1:1,\ x_2:10,\ x_3:100,\ x_4:1000$$

就是唯一可译码。

显然，奇异码不是唯一可译码。而非奇异码中有唯一可译码和非唯一可译码。比如非奇异码{0，10，11}是一种唯一可译码，对有限长的码序列 100111000，它只能被分割成 10，0，11，10，0，0。其余任何分割法都会产生非定义的码字。而非奇异码{0，10，00，01}就不是唯一可译码。

唯一可译码又分为非即时码和即时码两种。

非即时码：即接收端收到一个完整的码字后不能立即译码，还需要等下一个码字开始接收后，才能判断是否可以译码。比如

$$x_1:1,\ x_2:10,\ x_3:100,\ x_4:1000$$

就是非即时码。

即时码：即时码也称非延长码或异前缀码，即任意一个码字都不是其他码字的前缀部分。比如

$$x_1:1,\ x_2:01,\ x_3:001,\ x_4:0001$$

就是即时码。

综合起来，可将码进行如下分类。

- 码
  - 非分组码
  - 分组码
    - 奇异码
    - 非奇异码
      - 非唯一可译码
      - 唯一可译码
        - 非即时码
        - 即时码（非延长码）

即时码也称异字头码，它保证了译出的码字是唯一的。那么，如何来构造这样的码字呢？可通过称之为码树或树图来产生各个码字，如图 5-2 所示。

图的顶部称为树根，若采用 r 进制编码，则从树根出发引出 r 条分支，每条分支的端点称为节点；从每一个节点出发，又可引出 r 条分支，…这样就构成了一棵码树，从根部开始，依次称各层节点为 0 级、1 级、2 级、…、n 级节点。

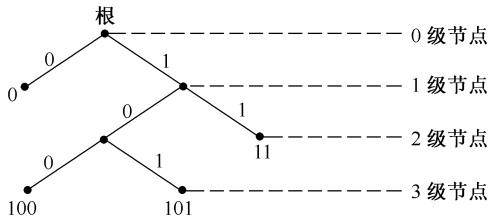
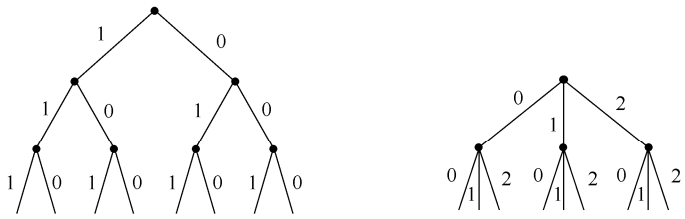


图 5-2　码树的结构（二进制编码）

若有一个码字长度 l，可指定某一级 l 节点（共有 $l \times r$ 个）为端节点，该节点就不再延伸，而从树根开始到端节点的分支标号序列便为 l 长的 r 进制码。

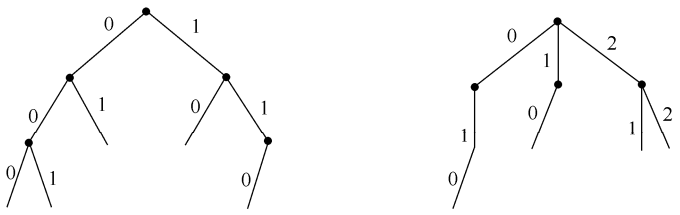
如果共有 k 个消息按 $l_1, l_2, \cdots, l_k$ 长度进行编码，只要在码树上指定 k 个相应级数的端节点，就可以得到 k 个异字头码。图 5-2 中表示的是 4 个异字头码，长度分别为 $l_1 = 1$，$l_2 = 2$，$l_3 = l_4 = 3$，相应的码字分别为 0，11，100，101。

在码树中，若各个分支都延伸到最后一级端点，此时，总共有 r^n 个码字，这样的码树称为满树；否则就称为非满树，如图 5-3 和图 5-4 所示。



（a）二进制满树　　（b）三进制满树

图 5-3　不同进制的满树



（a）二进制非满树　　（b）三进制非满树

图 5-4　不同进制的非满树

唯一可译码存在的充分必要条件为各码字的长度 l_i 应符合克拉夫特（Kraft）不等式

$$\sum_{i=1}^{n} M^{-l_i} \leqslant 1 \tag{5-1}$$

M 为进制数，n 为信源符号数。

例 5-1 设二进制码树中 $X=(x_1,x_2,x_3,x_4)$，$l_1=1$，$l_2=2$，$l_3=2$，$l_4=3$，则

$$\sum_{i=1}^{4} 2^{-l_i} = 2^{-1}+2^{-2}+2^{-2}+2^{-3} = \frac{9}{8} > 1$$

这个结果不满足 Kraft 不等式，因此，不存在满足这种 l_i 的唯一可译码。

如果将码长改为 $l_1=1$，$l_2=2$，$l_3=3$，$l_4=3$，则

$$\sum_{i=1}^{4} 2^{-l_i} = 2^{-1}+2^{-2}+2^{-3}+2^{-3} = 1$$

此时，所构造的码字就存在唯一可译码。

需要注意的是：Kraft 不等式只是用来说明唯一可译码是否存在，并不能作为唯一可译码的判据。例如码字{0,10,010,111}，虽然满足 Kraft 不等式，但它不是唯一可译码。

设信源输出为 $X=(x_1x_2\cdots x_l\cdots x_L)$，其中 $x_l \in \{a_1,a_2,\cdots,a_n\}$，则信源编码的任务就是把信源输出序列 $X=(x_1x_2\cdots x_l\cdots x_L)$ 变换成码字序列 $Y=(y_1y_2\cdots y_k\cdots y_K)$，其中 $y_k=\{b_1,b_2,\cdots,b_m\}$。

变换的要求是能够在无失真或失真小于某一值的条件下，从码字 Y 恢复源字 X，或者说能按一定的质量要求进行反变换译码；另一个要求则是希望传送 Y 时所需要的信息率要比传送源字 X 所需要的信息率小。

一个码字 Y 有 K 个分量，而每一个分量都有 m 种可能的取值，所以一个码字 Y 的信息量为 $K\log_2 m$（bit/码元序列）。

该码字所代表的信源序列 X 有 L 个分量，或 L 个有信源符号。因此，每个信源符号所需要的平均信息率为

$$R=\bar{K}=\frac{K}{L}\log_2 m=\frac{1}{L}\log_2 M \tag{5-2}$$

$M=m^K$ 是 Y 所能编成的码字的个数。

编码的目的就是要找到一种编码方法，使平均信息率 $\bar{K}$ 最小，同时，又能满足对失真的要求，编码定理就是要确定这两者之间的关系。

5.2 无失真信源编码

若信息的接收者要求无失真地精确复制信源输出的消息，这时的信源编码是无失真编码。只有对离散信源可以做到无失真编码。

离散信源的无失真编码，实质上是一种统计匹配编码。信息论指出，信源中的统计冗余度主要取决于两个主要因素：①消息概率分布的不均匀性；②消息间的相关性。对于无记忆信源，主要取决于概率分布的不均匀性。

统计匹配编码是根据信源的不同概率分布而选用与之相匹配的编码，以达到在系统中传输信息的速率最小，且满足译码时无失真或低于某一个允许的失真限度值。

对于信源输出的消息序列 $X=(x_1x_2\cdots x_l\cdots x_L)$，其中 $x_l\in\{a_1,a_2,\cdots,a_n\}$，经信源编码后得到的码字序列 $Y=(y_1y_2\cdots y_k\cdots y_K)$，其中 $y_k=\{b_1,b_2,\cdots,b_m\}$。输入消息总共有 n^L 种可能的组合，而输出的码字总共有 m^K 种可能的组合。要实现无失真的信源编码，必须满足两项最基本的要求。

① 无失真，即要求编成的码字 Y 能够无失真地复制消息 X。

② 有效，即尽可能少地传送信源中最必要的信息，或者说传送的码字要少于信源给出的消息。

显然，在不考虑信源的统计特性时，这两项基本要求是相互矛盾的，是不可能同时达到的。因为若要满足无失真的要求，就必须使每个信源输出的消息都能找到一个对应的码字，即应满足

$$m^K\geqslant n^L \quad 或 \quad \frac{K}{L}\geqslant\frac{\log_2 n}{\log_2 m} \tag{5-3}$$

若 $n=m$，则必有码字长度 $K\geqslant$消息长度 L，这显然不满足第二条有效性的要求；若 $K=L$，则 $m\geqslant n$，也不满足有效性的要求。

因此，要想同时满足上述两个基本要求，唯一的办法是从信源的统计特性上想办法。

不等式（5-3）中，左端为码字长度与消息长度之比；右端为等概率条件下信源熵与码字熵之比。考虑信源的实际统计特性，一般情况下是不等概的，此时的信源熵为 $H(X)=-\sum_i p_i\log p_i$。将其代入式（5-3），有

$$\frac{K}{L}\geqslant\frac{H(X)}{\log_2 m} \tag{5-4}$$

这样，即使 $m=n$，只要满足 $\log_2 m>H(X)$，就有可能实现 $K<L$，有可能同时满足上述的两个基本要求。

在具体实现时，既可以采用码字长度 K 不变的等长码，也可以更加灵活地采用码字长度 K 变化的变长码。

无失真信源编码定理包括定长编码定理和变长编码定理，下面分别讨论。

5.2.1 定长编码定理

定理：由 L 个符号组成的、每个符号的熵为 $H(X)$ 的离散无记忆平稳信源符号序列 $X=(X_1X_2\cdots X_l\cdots X_L)$，可用 K 个符号 $Y=(Y_1Y_2\cdots Y_k\cdots Y_K)$ 进行定长编码，且 $Y_k=\{y_1,y_2,\cdots,y_m\}$。对任意 $\varepsilon>0$，$\delta>0$，只要满足

$$\frac{K}{L}\log_2 m\geqslant H(X)+\varepsilon \tag{5-5}$$

则当 L 足够大时，必可使译码误差小于 δ。

反之，当

$$\frac{K}{L}\log_2 m\leqslant H(X)-2\varepsilon \tag{5-6}$$

时，译码必有错。

该定理的前一部分是正定理，后一部分是逆定理。由此定理我们看到，当编码器容许的输出信息率，即每个信源符号所必须输出的码长是 $\overline{K}=\frac{K}{L}\log_2 m$ 时，只要满足 $\overline{K}>H(X)$，这

种编码器一定可以做到几乎无失真，条件是所取的符号数 L 足够大。

由定理可以得到

$$K\log_2 m > LH(X) = H(\boldsymbol{X})$$

上式左边为 K 长码字所能携带的最大信息量；上式右边为 L 长信源序列携带的信息量。

因此，该定理实际上表明，只要码字所能携带的信息量大于信源序列输出的信息量，则可以使传输几乎无失真，同样 L 需足够大。

如果 $\overline{K} < H(X)$ 时，不能构成无失真的编码，译码时肯定会出现差错。

在正定理和逆定理之间有 3ε 的中间区域，信源熵 $H(X)$就处在此区域中，这个以 $H(X)$ 值为标志的区域称为临界状态。若 $\overline{K} = H(X)$ 时，编码可能有失真，也可能无失真。比如，信源符号出现的概率不相等，一般来说 $H(X)$不会是整数，因此，用 $\overline{K} = H(X)$ 编码总会带来失真。

例如，某信源有 8 个等概率符号，L=1，信源序列熵为 $H(X)$=$\log_2 8$=3bit/符号。也就是说，该信源符号可用 3bit 的信息率进行无失真的编码。

如果信源符号输出的概率不相等，比如，$p(x_i) = \{0.4, 0.18, 0.1, 0.1, 0.07, 0.06, 0.05, 0.04\}$，此时，$H(X)$=2.55bit/符号小于 3bit。按常理来说，8 种符号一定要用 3bit 组成的码字表示才能区别开来。若用 $\overline{K} = H(X)$ =2.55bit/符号来表示，则只有 $2^{2.55} = 5.856$ 种可能，有部分符号没有对应的码字。而这些符号一旦出现，当被传输到接收端时，没有对应的码字进行译码，就会引起译码差错，出现失真。

因此，定长编码一般都存在译码差错，只是差错的大小不同而已。

定义编码效率为

$$\eta = \frac{H(X)}{\overline{K}} \tag{5-7}$$

最佳编码时，编码效率为

$$\eta = \frac{H(X)}{H(X)+\varepsilon} \leqslant 1, \qquad \varepsilon > 0 \tag{5-8}$$

编码定理从理论上阐明了编码效率接近于 1 的理想编码器的存在，即输出符号的信息率与信源熵之比接近于 1。但要用定长编码来实现却是十分困难的，要求 L 相当大。

下面作一般的讨论。很多学者在对离散随机序列信源的统计特性进行深入地研究、证明后发现，这类信源具有渐进等同分割性（Asymptotic Equipatition Property，AEP），其基本思想是：一个总数为 n^L 种消息序列的信源，随着序列长度 L 的增长且足够大时，越来越明显地产生两极分化的现象。其中的一类组成大概率事件的序列集合，记作 A_ε 子集，它具有如下明显的特征。

① $\lim\limits_{L\to\infty} P(A_\varepsilon) = 1$。

② 序列的符号熵收敛于信源输出符号熵。

③ 序列趋于等概率分布。

另一类则组成小概率事件的非典型序列的集合，记作 A_ε^c 子集，它具有如下特性。

$$\lim_{L\to\infty} P(A_\varepsilon^c) = 1$$

A_ε 子集和 A_ε^c 子集为互补的集。

由 AEP 可知，无需对全部信源输出的消息序列进行信源编码，只对其中的典型序列集合 A_ε 子集中的序列进行编码即可，这一部分能够实现无失真地译码。而对于那些非典型序列集合 A_ε^c 子集中序列，由于未进行编码，这些序列一旦出现，将不能被正确译码，译码时必然出错。而出现差错的概率 P_e 就是子集 A_ε^c 中的元素发生的概率 $P(A_\varepsilon^c)$，那么，P_e 到底是多大呢？

当信源符号数为有限时，其方差 $\sigma^2 < \infty$ 也是有限的，由切比雪夫不等式，可得

$$P_e \leqslant \frac{\sigma^2(X)}{L\varepsilon^2} \tag{5-9}$$

其中，$\sigma^2(X) = E[I(x_i) - H(X)]^2$ 为信源的自信息方差，$\varepsilon > 0$。

当 $\sigma^2(X)$ 和 ε^2 均为定值时，只要 L 足够大，则 P_e 可以小于任一正数 δ，即 $\frac{\sigma^2(X)}{L\varepsilon^2} \leqslant \delta$。

若信源序列长度 L 满足 $L \geqslant \frac{\sigma^2(X)}{L\varepsilon^2}$ 时，就能够达到差错率的要求。

例 5-2 设离散无记忆信源有 8 种字符，相应的概率为

$$\begin{bmatrix} X \\ P \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \\ 0.4 & 0.18 & 0.1 & 0.1 & 0.07 & 0.06 & 0.05 & 0.04 \end{bmatrix}$$

信源熵为

$$H(X) = -\sum_{i=1}^{8} p_i \log p_i = 2.55\,\text{bit/符号}$$

自信息的方差为

$$\begin{aligned} \sigma^2(\boldsymbol{X}) &= D[I(x_i)] \\ &= \sum_{i=1}^{8} p_i (\log_2 p_i)^2 - [H(X)]^2 \\ &= 7.82 \end{aligned}$$

若采用定长二元编码，要求编码效率为 $\eta = 90\%$，对于无记忆信源，$H_L(X) = H(X)$，因此，有

$$\eta = \frac{H(X)}{H(X) + \varepsilon} = 0.9$$

可得到 $\varepsilon = 0.28$。

假定要求译码错误概率 $\delta \leqslant 10^{-6}$，有

$$L \geqslant \frac{\sigma^2(X)}{\varepsilon^2 \delta} = \frac{7.82}{0.28^2 \times 10^{-6}} = 9.8 \times 10^7 \approx 10^8$$

可以看出，在编码效率与差错率要求并不十分苛刻的情况下，对该信源就需要 $L = 10^8$ 个信源符号一起进行联合编码才能达到要求，实际上这是不可能的。因此，为了解决这一问题，必须寻找其他的编码方法，而变长度编码方法就是其中的一种。

5.2.2 变长编码定理

先通过一个简单的例子了解一下变长编码的优点。

例 5-3 设有一简单的离散无记忆信源

$$\begin{bmatrix} X \\ P \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} \end{bmatrix}$$

对其进行变长编码（具体的编码方法将在后面详细地讨论）。

解 对该信源的 4 个符号逐位进行如下编码。

X：	x_1	x_2	x_3	x_4
P：	0.5	0.25	0.125	0.125
变长码：	0	10	110	111

信源熵为：$H(X)=-\sum_{i=1}^{4} p_i \log_2 p_i = \frac{7}{4}$（bit/符号）。

平均码长为：$\overline{K}=\sum_{i=1}^{4} p_i K_i = \frac{1}{2}\times 1+\frac{1}{4}\times 2+2\times\frac{1}{8}\times 3=\frac{7}{4}$（二元码符号/信源符号）。

编码效率为：$\eta=\frac{H(X)}{\overline{K}}=\frac{7/4}{7/4}=1$。

可见，采用变长编码，即使是逐位编码（L=1），其编码效率可达 100%。这虽然是一个特例，但在一般情况下，变长编码的效率都是非常高的，大大优于定长编码。

变长编码也叫不等长度编码，码字的长度 K 是变化的。在实际情况中，一般的离散无记忆信源输出的各消息符号的概率是不相等的。若根据信源各个符号的统计特性，对出现概率大的消息用较短的码字，而对出现概率小的消息用较长的码字，从整个信源来看，可以得到较短的平均码长。这样就实现了与信源的统计特性相匹配，这就是变长编码的基本思想。

变长码的优点很多，但其译码要比定长码的复杂。对于定长码来说，只要使不同的消息对应不同的码字，而接收端只要根据约定的码组长度就能正确识别出一个码字的起始位置，从而实现正确译码。但对变长码来说，要正确识别每个码字的起始位置就不是那么容易了。

1．单个符号变长编码定理

定理：设某离散无记忆信源 $X=(x_1,x_2,\cdots,x_n)$，其符号熵为 $H(X)$，每个信源符号用 m 进制码元进行编码，必存在一种无失真编码方法，其码字的平均长度 $\overline{K}$ 应满足下列不等式。

$$\frac{H(X)}{\log_2 m} \leqslant \overline{K} \leqslant \frac{H(X)}{\log_2 m}+1 \tag{5-10}$$

m=2 时，即二进制编码情况，有

$$H(X) \leqslant \overline{K} \leqslant H(X)+1 \tag{5-11}$$

证明 若对某一个 $X=x_i$ 的信源符号，选用一个对应长度 K_i 与其自信息量 $I(x_i)=-\log_m p_i=-\frac{\log_2 p_i}{\log_2 m}$ 相匹配的码字，即

$$1-\frac{\log_2 p_i}{\log_2 m} > K_i \geqslant -\frac{\log_2 p_i}{\log_2 m}$$

满足上式的 K_i 必存在，即选用当 $\frac{-\log_2 p_i}{\log_2 m}$ 为整数时，就等于 K_i，等式成立。若 $\frac{-\log_2 p_i}{\log_2 m}$ 不为

整数时，则 K_i 可取比它大一些的、最接近的整数，即满足不等式。

对上述不等式 p_i 取数学期望，有

$$1-\sum_i p_i\frac{\log_2 p_i}{\log_2 m}>\sum_i p_i K_i\geqslant-\sum_i p_i\frac{\log_2 p_i}{\log_2 m}$$

即

$$1+\frac{H(X)}{\log_2 m}>\overline{K}\geqslant\frac{H(X)}{\log_2 m}$$

证毕。

该编码定理给出了最佳变长码的平均码长 $\overline{K}$ 的上限和下限。在尚未编出码字之前就能知道 K 落在什么范围，当然是非常重要的。定理指出了最佳变长码应该是与信源相匹配的编码，特别是下限更为重要，因为它是信源压缩编码的极限。通常称达到下限的最佳变长码为熵编码。

2．离散信源序列变长编码定理

定理：对于平均符号熵为 $H_L(X)$ 的离散平稳无记忆信源，必存在一种无失真编码方法，使平均每个符号的信息率满足下式。

$$H_L(X)\leqslant\overline{K}\leqslant H_L(X)+\varepsilon \tag{5-12}$$

其中，ε 为任意正数。

证明 设信息序列的长度为 L 个信源符号，用 m 进制进行变长编码，由式（5-12）可得平均码长 $\overline{K}_L$ 满足下式。

$$\frac{LH_L(X)}{\log_2 m}\leqslant\overline{K}_L\leqslant\frac{LH_L(X)}{\log_2 m}+1$$

由于平均输出信息率为 $\overline{K}=\frac{\overline{K}_L}{L}\log_2 m$，则有

$$H_L(X)\leqslant\overline{K}\leqslant H_L(X)+\frac{\log_2 m}{L}$$

当 L 足够大时，只要 $L>\frac{\log_2 m}{\varepsilon}$，就可使不等式（5-7）成立。

证毕。

同样，定义变长编码的效率为

$$\eta=\frac{H_L(X)}{\overline{K}} \tag{5-13}$$

η 总是小于 1 的。用 η 来衡量各种编码方法的优劣。

另外，为了衡量各种编码方法与最佳码的差距，定义码的剩余度为

$$\gamma=1-\eta=1-\frac{H_L(X)}{\overline{K}} \tag{5-14}$$

显然，编码的效率越高，则剩余度就越小。

由定理可以得到编码效率的下界为

$$\eta=\frac{H_L(X)}{\bar{K}}>\frac{H_L(X)}{H_L(X)+\log_2 m/L} \tag{5-15}$$

用变长编码来达到相当高的编码效率，一般所要求的符号长度 L 可以比定长编码小得多。以例 5.2 为例，若使用变长二进制码，m=2，$\log_2 m=1$，$H(X)$=2.55bit/符号，要求编码效率 $\eta>90\%$，由 $\frac{2.55}{2.55+1/L}=0.9$，可得 $L=\frac{0.9}{2.55(1-0.9)}=3.53\approx 4$。这就是说，只要序列长度为 L=4 就可以达到 90%的编码效率了。

变长编码不仅适用于离散信源，而且对连续信源也同样适用。尽管对连续信源无法进行无失真的编码，但在限失真条件下，可作类似的推广。

例 5-4 设离散无记忆信源的概率空间为

$$\begin{bmatrix}X\\P\end{bmatrix}=\begin{bmatrix}x_1 & x_2\\ \frac{3}{4} & \frac{1}{4}\end{bmatrix}$$

信源熵为：$H(X)=-\frac{3}{4}\log_2\frac{3}{4}-\frac{1}{4}\log_2\frac{1}{4}=0.811$ bit/符号。

若采用二元定长编码（0,1）来构造一个即时码

$$x_1\to 0,\quad x_2\to 1$$

则平均码长为

$$\bar{K}=1 \text{ 二元码符号/信源符号}$$

编码效率为

$$\eta=\frac{H(X)}{\bar{K}}=0.811$$

输出的信息率为

$$R=0.811\text{bit/二元码符号}$$

若对长度为 2 的信源序列进行变长编码，则结果如表 5-1 所示。

表 5-1　　长度为 2 的信源序列及变长码字

序列	序列概率	即时码
x_1x_1	9/16	0
x_1x_2	3/16	10
x_2x_1	3/16	110
x_2x_2	1/16	111

码字的平均长度：$\bar{K}_2=\frac{9}{16}\times 1+\frac{3}{16}\times 2+\frac{3}{16}\times 3+\frac{1}{16}\times 3=\frac{27}{16}$ 二元码符号/信源序列。

每一单个符号的平均码长：$\bar{K}=\frac{\bar{K}_2}{2}=\frac{27}{32}$ 二元码符号/信源符号。

相应的编码效率：$\eta_2=\frac{H_2(\boldsymbol{X})}{\bar{K}}=\frac{32\times 0.811}{27}=0.961$。

输出的信息率：$R_2 = 0.961$ bit/二元码符号。

比较一下，可以看到，虽然编码复杂了一些，但信息传输率提高了。

用同样的方法可以求得信源序列的长度 L=3 和 L=4 两种情况下的编码效率分别为：$\eta_3 = 0.985$ 和 $\eta_4 = 0.991$。相对应的信息传输率为：$R_3 = 0.985$ bit/二元码符号和 $R_4 = 0.991$ bit/二元码符号。

如果对该信源采用定长二元码编码，要求编码效率达到 96%时，所允许的译码错误概率 $\delta \leqslant 10^{-5}$，那么由公式可计算出自信息的方差为

$$\begin{aligned}\sigma^2(X) &= \sum_{i=1}^{2} p_i(\log_2 p_i)^2 - [H(X)]^2 \\ &= 0.4715\end{aligned}$$

所需的信源序列长度为

$$L \geqslant \frac{0.4715}{0.811^2} \times \frac{0.96^2}{0.04^2 \times 10^{-5}} = 4.13 \times 10^7$$

可见，定长编码需要的信源序列很长，并且存在译码差错。

而变长编码要求编码效率达到 96%时，只需 L=2。并且随着信源长度 L 的增加，编码的效率越来越接近于 1，编码后的信息传输率也越来越接近于无噪、无损的二元对称信道的信道容量 C（1bit/二元符号），达到信源与信道相匹配，信道得以充分利用。

5.2.3 最佳变长编码

最佳码是凡能保证传输一定的信息量，且各码字的平均长度为最短的码字集合。它意味着采用最佳编码的等效信源所发出的平均信息量为最大；或者说，在不改变信源的条件下，采用最佳编码后信道的编码效率也是最高的。

为了达到此目的，通常是利用构成信源序列的符号出现的概率统计特性，把信源符号集合中出现概率大的符号编成较短的码字，而对那些出现概率小的符号编成较长的码字，使平均码长最短。下面介绍三种最佳码的编码方法：香农编码、费诺编码和霍夫曼编码。

1．香农编码

香农第一定理指出了平均码长与信源之间的关系，也指出了可以通过编码使平均码长达到极限值。那么，如何构造这种码呢？香农第一定理指出，选择每个码字的长度 K_i 满足下列不等式，即

$$I(x_i) \leqslant K_i \leqslant I(x_i) + 1, \qquad \forall i$$

则所得到的码字就是平均码长达到极限值的香农码。这种编码方法称为香农编码方法。

利用香农编码方法得到的码字多余度稍大，其实用性不大，但是具有很重要的理论意义。编码的具体步骤如下。

（1）把信源消息符号按其概率递减顺序进行排列，得

$$p(x_1) \geqslant p(x_2) \geqslant \cdots \geqslant p(x_n)$$

（2）确定满足下列不等式的整数码长 K_i

$$-\log_2 p(x_i) \leqslant K_i \leqslant -\log_2 p(x_i) + 1$$

（3）为了编成唯一可译码，计算第 i 个消息符号的累加概率。

$$P_i = \sum_{k=1}^{i-1} p(x_k)$$

（4）将累加概率 P_i 变换成二进制数。

（5）取 P_i 二进制数的小数点后 K_i 即为该消息符号的二进制码字。

例 5-5 某一离散信源共有 7 个消息符号，其概率和累加概率如表 5-2 所示。

表 5-2　　信源符号的香农编码结果

信源符号 x_i	符号概率 p_i	累加概率 P_i	$-\log_2 p_i$	码字	码长
x_1	0.2	0	2.34	000	3
x_2	0.19	0.2	2.41	001	3
x_3	0.18	0.39	2.48	011	3
x_4	0.17	0.57	2.56	100	3
x_5	0.15	0.74	2.73	101	3
x_6	0.10	0.89	3.34	1110	4
x_7	0.01	0.99	6.66	1111110	7

以第 4 个符号为例，x_4 出现的概率为 0.17，确定其码长

$$-\log_2 0.17 \leqslant K_4 < -\log_2 0.17 + 1$$

即 $2.56 \leqslant K_4 < 3.56$，则 $K_4 = 3$；x_4 的累加概率 $P_4 = 0.57$，变换成二进制为：0.1001…，小数点后 3 位的 100 即为 x_4 的编码码字。用同样的方法可得其他几个符号的编码码字，如表 5-2 第 5 列所示。

从编码的结果可以看到，这种编码是唯一可译码，每一个码字都是即时码。

信源熵为：$H(X) = -\sum_{i=1}^{7} p(x_i)\log_2 p(x_i) = 2.61$ bit/符号。

平均码长：$\overline{K} = \sum_{i=1}^{7} p(x_i)K_i = 3.14$ 码元/符号。

编码效率：$\eta = \dfrac{H(X)}{\overline{K}} = \dfrac{2.61}{3.14} = 83.1\%$。

2．费诺编码

费诺编码方法的具体步骤如下。

（1）把信源消息符号按其概率递减顺序进行排列，得

$$p(x_1) \geqslant p(x_2) \geqslant \cdots \geqslant p(x_n)$$

（2）将依次排列的信源符号分为两大组，使各组的概率之和尽可能接近相等，对各组赋予一个二进制码元“0”和“1”。

（3）将每一组的信源符号进一步再分成两组，使划分后的两个组的概率之和接近相等，

再对各组赋予一个二进制码元“0”和“1”。

（4）重复这个步骤，直到每个组只剩下一个信源符号为止。

（5）从左至右，各信源符号所对应的码字即为编好的费诺码。

例 5-6 某一离散信源共有 7 个消息符号，其概率和费诺编码过程如表 5-3 所示。

表 5-3 **信源符号的费诺编码结果**

信源符号 x_i	符号概率 p_i	分组及编码				码字	码长
x_1	0.2	0	0			00	2
x_2	0.19		1	0		010	3
x_3	0.18			1		011	3
x_4	0.17	1	0			10	2
x_5	0.15		1	0		110	3
x_6	0.10			1	0	1110	4
x_7	0.01				1	1111	4

信源各符号的码字及码长如表 5-3 第 7 列和第 8 列所示。

信源熵：$H(X)=-\sum_{i=1}^{7}p(x_i)\log_2 p(x_i)=2.61\,\text{bit/符号}$。

平均码长：$\overline{K}=\sum_{i=1}^{7}p(x_i)K_i=2.74$ 码元/符号。

编码效率：$\eta=\dfrac{H(X)}{\overline{K}}=\dfrac{2.61}{3.14}=95.3\%$。

本例中使用的信源和前面的香农编码方法使用的信源是相同的，从编码的结果可以看到，费诺编码方法比香农编码方法所得到的码字的平均码长短，信息的传输速率高，编码的效率也高。

3．霍夫曼编码

在霍夫曼编码过程中，利用了构成信息序列的符号出现的概率统计特性，对出现概率较大的符号赋予短码，出现概率较小的符号赋予长码，使平均码长最短，是一种编码效率很高的变长编码方法。

霍夫曼编码方法的具体步骤如下。

① 将 n 个信源消息符号按概率大小依次进行排列。

$$p(x_1)\geqslant p(x_2)\geqslant\cdots\geqslant p(x_n)$$

② 对两个概率最小的符号分别赋予 0 和 1 两个码元，然后将这两个概率相加作为一个新的符号的概率，与其他符号重新按概率大小较小排列。

③ 对重排后两个概率最小的符号重复步骤（2）的过程。

④ 不断继续上述过程，直到最后两个符号分别赋予 0 和 1 为止。

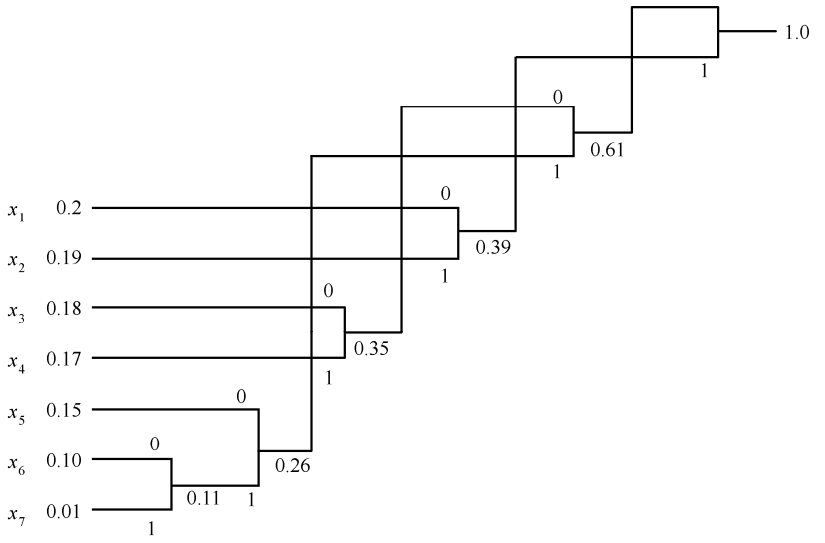
⑤ 从最后一级开始，向前返回，就得到各个信源符号所对应的码元序列。

例 5-7 设某离散无记忆信源为

$$\begin{bmatrix} X \\ P \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ 0.2 & 0.19 & 0.18 & 0.17 & 0.15 & 0.10 & 0.01 \end{bmatrix}$$

试对该信源进行二进制霍夫曼编码。

按照霍夫曼编码步骤，其编码过程如下。



各符号的霍夫曼码字为

x_1：10，x_2：11，x_3：000，x_4：001，x_5：010，x_6：0110，x_7：0111

信源熵：$H(X) = -\sum_{i=1}^{7} p(x_i)\log_2 p(x_i) = 2.61\,\text{bit/符号}$。

平均码长：$\overline{K} = \sum_{i=1}^{7} p(x_i)K_i = 2.72$ 码元/符号。

编码效率：$\eta = \dfrac{H(X)}{\overline{K}} = \dfrac{2.61}{3.14} = 95.96\%$ 。

可以看出，霍夫曼码的平均码长最短，信息的传输速率最大，编码的效率最高。

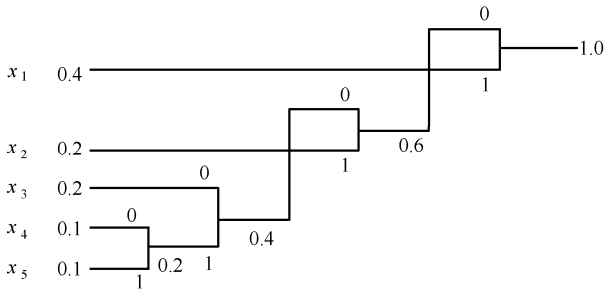
需要说明的是，由霍夫曼编码方法给出的码字不是唯一的。因为每次赋予两个概率最小的符号 0 和 1 时，也可以赋予 1 和 0，这样编码后的各符号的码字将会不同，但码字的长度是一样的，平均脉冲也是一样的。另外，在将两个概率最小的符号合并后的新符号的概率与其他信源符号的概率相等时，重新排列时新符号的位置次序可以放在相等概率符号的上面，也可以放在下面。不同的放法就会得到不同的霍夫曼码字，相应的码字的质量是不一样的。通常将合并后的符号放在上面，以减小符号合并的次数，得到码字的方差较小的霍夫曼码。

例 5-8 设某离散无记忆信源为

$$\begin{bmatrix} X \\ P \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \\ 0.4 & 0.2 & 0.2 & 0.1 & 0.1 \end{bmatrix}$$

对其进行霍夫曼编码。

方法一按霍夫曼编码方法的步骤，有



各信源符号的码字为：x_1：1，x_2：01，x_3：000，x_4：0010，x_5：0011。

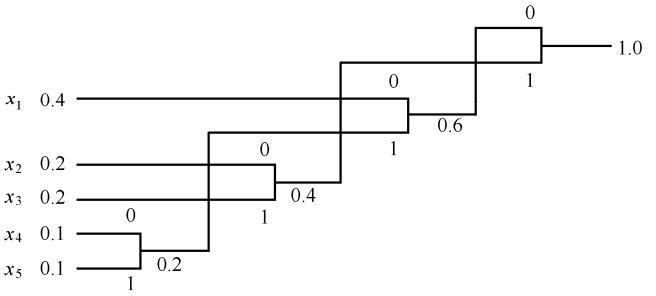
信源熵：$H(X)=-\sum_{i=1}^{5}p_i\log p_i=2.123\,\text{bit/符号}$。

平均码长：$\overline{K}=\sum_{i=1}^{5}p_iK_i=2.2$ 二进制码元/符号。

编码效率：$\eta=\dfrac{H(X)}{\overline{K}}=\dfrac{2.213}{2.2}=0.965$。

码方差：$\sigma_1^2(X)=\sum_{i=1}^{5}p_i(K_i-\overline{K})^2=1.36$。

方法二按霍夫曼编码方法的步骤，有



各信源符号的码字为：x_1：00，x_2：10，x_3：11，x_4：010，x_5：011。

平均码长：$\overline{K}=\sum_{i=1}^{5}p_iK_i=2.2$ 二进制码元/符号。

编码效率：$\eta=\dfrac{H(X)}{\overline{K}}=\dfrac{2.213}{2.2}=0.965$。

码方差：$\sigma_2^2(X)=\sum_{i=1}^{5}p_i(K_i-\overline{K})^2=0.16$。

可见，两种霍夫曼码字的平均码长和编码效率是一样的，而码的方差却不相同，即码的质量不完全相同。

通过例子看到，为了得到码方差最小的码，应使合并后的新信源符号（概率与其他符号

的概率相同时）位于缩减信源序列尽可能高的位置上，以减少再次合并的次数。

霍夫曼编码由于其编码方法保证了概率大的符号对应于短码，概率小的符号对应于长码，以及合并信源的最后两个码字总是最后一位不同，从而保证了编出的码字是即时码。

一般来说，变长编码只适应于有限长度的信息传输，并且可以做到无失真地译码。

5.3 限失真信源编码定理

在很多实际信源中，特别是在模拟的连续信源中，无失真编码的要求是完全没有必要的，也是达不到的。译码输出与信源输出之间有一定的失真在许多情况下是可以允许的。

香农建立的信息率失真函数理论，给出了平均失真小于 D 时所必须具有的最小的信息率 $R(D)$。只要信息率大于 $R(D)$，一定可以编一种码，使译码后的失真小于 D。

定理：设离散无记忆平稳信源的信息率失真函数为 $R(D)$，则当系统的信息率 $R>R(D)$时，只要信源序列的长度 L 足够长，就一定存在一种编码方法，使译码后的失真小于或等于 $D+\varepsilon$，ε 为任意小的正数，且当 $L\to\infty$时，$\varepsilon\to 0$；反之，若 $R<R(D)$，则无论采用什么样的编码方法，其译码失真必大于 D。

该定理可用式子表示为

$$\frac{1}{L}\log K(L,D+\varepsilon)<R(D),\quad L \text{ 足够大。}$$

或

$$\frac{1}{L}\log K(L,D)>R(D)$$

其中，$K(L,D)$为与 L、D 有关的编码长度。

上述定理指出了在允许的失真限度内，使信息率任意接近于 $R(D)$的编码方法是存在的。

若信源为二元信源 $X\in(0,1)$，对任意小的 $\varepsilon>0$，每个信源符号的平均长度满足

$$R(D)\leqslant \overline{K}<R(D)+\varepsilon$$

限失真信源编码定理是有关存在性定理，指明了理论上的极限性能和方向，说明了最佳编码是存在的。但具体的编码方法不能像无失真信源编码定理那样，能直接给出无失真编码的构造性指导，只能从优化的思路去寻求最佳的编码方式。

5.4 其他无失真信源编码方法

5.4.1 算术编码

算术编码是一种高效的编码方法。对于二元、平稳的马尔可夫信源来说，该方法实现起来简单，编码效率可以达到 95%以上。与霍夫曼编码方法不同，算术编码是一种非分组码的编码方法。

算术编码的基本思路是：从序列出发，将各信源序列的概率映射到[0,1]区间上，使每个序列对应该区间内的一个点，即一个二进制的小数。这些点将[0,1]区间分成许多小段，每段的长度等于某一序列的概率，整个编码过程就是单位区间[0,1]内的子分过程。

若某一信源的符号集 $A=\{a_1,a_2,\cdots,a_{n-1}\}$，信源输出的序列为 $X=(x_0x_1\cdots x_l\cdots x_{L-1})$，$x_l\in A$，则总共可以组成 n^L 种不同的序列。要得到某一序列的概率在[0,1]区间内对应的点，只能从已知的信源符号概率 $P=\{p_0,p_1,\cdots,p_{n-1}\}$ 中递推得到。

定义各符号的积累概率为

$$P=\sum_{i=0}^{r-1}p_i \tag{5-16}$$

可以得

$$p_r=P_{r+1}-P_r \tag{5-17}$$

P_r，P_{r+1} 都是大于 0、小于 1 的正数，可用[0,1]区间内的两个点来表示，而 p_r 就是这两个点之间的小区间的长度，如图 5-5 所示。

由图 5-5 可见，不同的符号有不同的小区间，它们互不重叠。因此，可以将这种小区间内的任一点作为该符号的代码。

假定某一离散信源输出序列的符号集为{a，b，c，d}，相应的概率为 $\{p(a)=\frac{1}{2},p(b)=\frac{1}{4},p(c)=p(d)=\frac{1}{8}\}$，其二进制表示为{$p(a)$=0.1，$p(b)$=0.01，$p(c)$=$p(d)$=0.001}，如图 5-6 所示。各符号的积累概率分别为：$P_a=0$，$P_b=p(a)=0.1$，$P_c=p(a)+p(b)=0.11$，$P_d=p(a)+p(b)+p(c)=0.111$。

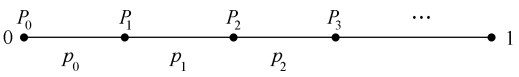


图 5-5 子区间的划分

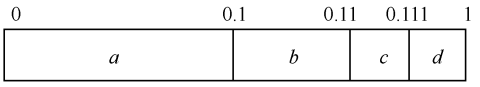


图 5-6 4 个子区间的划分

这些小区间的端点也称为码点，每一个码点值就是它前面所出现的各符号的概率之和，即积累概率。将每一个码点作为右端点，每一个子分过程中所得区间的宽度对应于该符号的概率。

对于给定的待编码的信源序列，通过双重递推运算，将输入序列与子区间的码点值联系起来，这个值就是输入序列的算术编码输出。

设输入符号序列 s 后的子区间宽度为 $A(s)$，码点为 $C(s)$，当输入为 s_i 时，由下列递推关系计算 $A(s_i)$、$C(s_i)$。

$$\begin{cases}C(s_i)=C(s)+A(s_i)\times P_i\\A(s_i)=A(s)\cdot p_i\end{cases} \tag{5-18}$$

递推运算的初始条件为：$C(\varphi)=0$，$A(\varphi)=1$，φ 表示空序列。

下面通过一个例子来说明算术编码的编码过程。

假定信源的符号集为{a，b，c，d}，各符号的概率及积累概率为

符号	概率 p_i	积累概率 P_i
a	$\frac{1}{2}(0.1)$	0

b	$\frac{1}{4}(0.01)$	0.1
c	$\frac{1}{8}(0.001)$	0.11
d	$\frac{1}{8}(0.001)$	0.111

若信源产生的序列为 s=($abda$)，按照编码的递推运算式（5-21），有如下步骤。

第一个符号“a”产生的码点和区间为

$$\begin{cases} C(a) = C(\varphi) + A(\varphi) \times P_a = 0 + 1 \times 0 = 0 \\ A(a) = A(\varphi) \times p_a = 1 \times 0.1 = 0.1 \end{cases}$$

对第二个符号“b”编码后，得到新的码点和区间，即

$$\begin{cases} C(ab) = C(a) + A(a) \times P_b = 0 + 0.1 \times 0.1 = 0.01 \\ A(ab) = A(a) \times p_b = 0.1 \times 0.01 = 0.001 \end{cases}$$

对第三个符号“d”重复该迭代过程，得

$$\begin{cases} C(abd) = C(ab) + A(ab) \times P_d = 0.01 + 0.001 \times 0.111 = 0.010\,111 \\ A(abd) = A(ab) \times p_d = 0.001 \times 0.001 = 0.000\,001 \end{cases}$$

对第四个符号“a”重复该迭代过程，得

$$\begin{cases} C(abda) = C(abd) + A(abd) \times P_a = 0.010\,111 + 0.000\,001 \times 0 = 0.010\ 111 \\ A(abda) = A(abd) \times p_a = 0.000\,001 \times 0.1 = 0.000\,000\,1 \end{cases}$$

经此编码过程后，信源产生的序列 s=($abda$)的算术码字为：C($abda$)=0.010 111，在传输时，只需发送码字 0.010 111 即代表发送的是序列($abda$)。

可以看到，尽管在编码的计算过程中有乘法运算，但由于都是和 2 的负数幂进行相乘，可以很容易地用右移运算来实现。因此，编码运算过程中只有加法和移位的算术运算，这正是将这种编码方法称为算术编码的原因。

对信源产生的序列 s=($abda$)的算术编码过程可用图 5-7 表示。

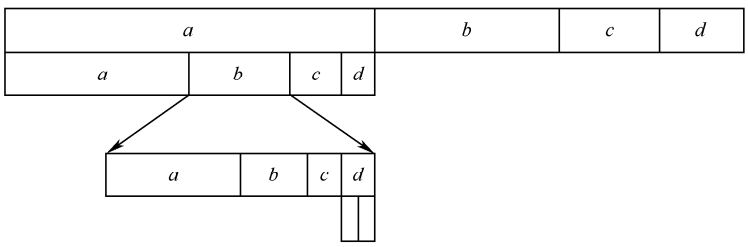


图 5-7　算术编码过程

算术码字的译码过程，是将编码后的码字序列看做是一个幅度值通过逐次比较而实现的，即判定码字 C(s)落在哪一个区间，就可以得出一个相应的信源符号，具体的译码步骤如下。

接收到的码字为 C($abda$)=0.010111，在[0,0.1]范围内，可译出第一个符号为 a。

由于在编码过程中，进行 A 的迭代运算时第二次的子区间的宽度乘过 0.1，解码时应乘

以 0.1 的倒数。

$$C(abcd)\times\frac{1}{0.1}=0.010\ 111\times\frac{1}{0.1}=0.101\ 11$$

该值属于区间[0,0.1]，因此，译出第二个符号为 b。

在编码过程中，第二次的子分区间的新码点是和符号 b 的积累概率 P_b=0.1 相加的，译码时应将其减去：0.101 11−0.1=0.001 11。

再除以符号 b 的概率 0.01，有：$0.00111\times\frac{1}{0.01}=0.111$。

该值属于区间[0.111,1]，因此，译出第三个符号为 d。

用同样的步骤，去掉所加的符号 d 的积累概率 P_d=0.111，有 0.111−0.111=0。

再除以符号 d 的概率 0.001，有 $0\times\frac{1}{0.001}=0$。

该值属于区间[0,0.1]，因此，可译出第四个符号为 a。

综合整个译码过程，可译出码字 $C(abda)$=0.010 111 所对应的信源序列为 s=($abda$)。

5.4.2 游程长度编码

游程长度指的是由字符构成的数据流中，各个字符连续重复出现而形成字符串的长度。如果给出了形成串的字符、串的长度及串的位置，就能准确地恢复出原来的数据流。游程长度编码就是用二进制码字给出上述信息的各种方法。

我们知道，在二元序列中，只有两种符号“0”和“1”，符号“0”和符号“1”可能连续出现，连“0”这段称作 0 游程，连“1”这段称作 1 游程。它们各自的长度称为游程长度，用 $L(0)$和 $L(1)$表示。

将任何二元序列变换成游程长度序列，这种变换是唯一的、可逆的。

例如：某二元序列为 000101110010001…，用游程序列表示为 3113213…。如果已知二元序列是以符号 0（或 1）开始的，那么就可以很容易地从游程序列中恢复出原来的二元序列。

游程序列是多元序列，若能够事先测得各游程长度的概率分布，就可以按照霍夫曼编码的方法或其他的编码方法对其进行进一步的编码处理，以达到最大限度地压缩数据的目的。

游程长度编码仍然是变长码，有其固有的缺点，实际应用时，还需要采取一些措施来改进。

一般情况下，游程长度越大，其出现的概率就越小。在实际应用中，可对长游程不严格按照霍夫曼编码步骤进行，通常采用截断的方法进行处理。

游程长度编码只适用于二元信源序列。对于多元信源，一般不能直接利用长度游程编码。只有当多元信源中存在有较多的冗余字符时，可对其中的冗余字符进行游程长度编码。

在某些信源序列中，存在着很多不携带信息的字符，这些字符就称为冗余字符，它们完全可以不被传送。例如，话音通信中的讲话停顿和间隙、图像通信中的背景等。如果在传输前将这些冗余信息去除，那么就可以得到较大的数据压缩。

对于某多元序列 $x_1x_2\cdots x_lyy\cdots yx_{l+1}x_{l+2}\cdots yy.$，其中的 x 是包含有信息码字，而 y 是不含有信息的冗余字符。即使不传送这些冗余字符，在接收端也能恢复。这样一个多元序列可分解为两个序列

$$11\cdots100\cdots011\cdots100\cdots \quad 和 \quad x_1x_2\cdots x_lx_{l+1}x_{l+2}\cdots x_{l+m}\cdots$$

在第一个序列中，用“1”表示所有的信息码字，“0”表示所有的冗余字符；第二个序列是取消冗余字符后，留下的所有信息码字，变成了长度缩短的多元序列。对这两个序列可采用不同的编码方法分别进行编码，达到压缩数据的目的。

若将编码后的两个序列进行传送，只要不出现差错，在接收端就能够准确地恢复出原来的多元信源序列。

5.5 矢量量化编码

对于连续信源来说，限失真编码的主要方法之一是量化。量化就是将连续的样值转化为有限个离散值，并对这有限个离散值用不同的码字来代替。

量化方式有均匀量化和非均匀量化。

均匀量化：将输入信号的取值域按等间隔分层的量化方式。此时，每个量化区间的量化电平均取在各区间的中点。

非均匀量化：将输入信号的取值域按不等间隔进行分层的量化方式。输入信号取值小的区间，其量化分层的间隔就小；取值大的区间，量化分层的间隔就大。

由于量化是以有限个离散值近似表示无限多个连续值，这样就一定会产生误差，称为量化误差。由此产生的失真称为量化失真或量化噪声。我们希望经过量化编码以后的数据率尽可能的低，同时又希望量化噪声小，还要求在工程实践中容易实现。这些希望与要求是相互矛盾的，不可能同时满足，只能作出一种最佳的折衷，即满足一定条件的最佳设计或接近最佳设计。

5.5.1 最佳标量量化编码

量化过程始于取样，每一次取样得到一个取样值，其理论上的值域为$(-\infty, +\infty)$。通常将每个取样值孤立起来，作为一个一维的连续取值的变量进行处理。将其取值范围按某种规则划分成若干段，所有落在同一段中的样值，都用一个代表值代替。也就是划分量化区间和决定量化输出值两个过程。量化区间的划分及量化输出值的选定，要符合失真度最小的原则。这种将每一次取样得到的单个取样值进行独立处理的编码方法称为标量量化编码。

比如连续的样值为 x，经量化后为 $y_1, y_2,\cdots,y_n$，即由 n 个离散值的相应码字 $y_1, y_2,\cdots,y_n$ 来代替 x，该离散值的个数 n 称为量化级数。若用二进制数来表示这 n 个不同的数，则需要用 $\log_2 n$ 位二进制符号来代表这有限个集合中的每一个数。称这样的量化器输出的信息率为每个样值 $\log_2 n$ (bit)。

假定信源符号的取值区间为 (a_0, a_n)，即 $a_0 < x < a_n$。量化就是将该区间分成 n 个小区间，每个区间内确定一个量化值 y_i。若各区间的端点为 a_{i-1} 和 a_i，则有

$$a_0 \leqslant y_1 \leqslant a_1 \leqslant y_2 \leqslant \cdots \leqslant a_{n-1} \leqslant y_n \leqslant a_n$$

最佳的标量量化就是在一定的 n 值时，选择各 a_i 和 y_i 以使失真最小。

从信息论的观点来看，量化过程实质上是一个舍弃一部分多余的信息，保留为恢复满足一定保真度要求的原信号的必要信息的过程。所谓的保真度要求，实际上就是对量化误差的

一个限制，也是对量化器性能的一个要求。

设任意一个取样值 x 输入到量化器的概率密度函数为 $p(x)$，把量化误差的产生过程视作一个加性随机噪声过程，如图 5-8 所示，其函数关系为

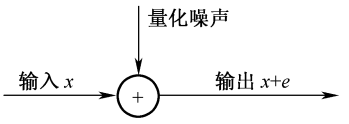


图 5-8 量化器模型

$$e(x)=Q(x)-x$$

$Q(\cdot)$：量化器的量化特性。这里的噪声是由输入、输出值所决定。该量化器被看做一个随机变量。因此，只能以某种统计平均值来测量。

最普遍采用的是均方误差测量，即

$$D=\int_{-\infty}^{\infty}[Q(x)-x]^2\,p(x)\mathrm{d}x \tag{5-19}$$

若量化级数 n 足够大，以致于量化区域的间隔 $\Delta_i=a_i-a_{i-1}$ 很小。这样，对每一个小区域来说，$p(x)$几乎相等，可用 $p(y_i)$ 来代表。

若量化区域 Δ_i 的输出值 y_i 取区域的中央值，即

$$y_i=\frac{a_{i-1}+a_i}{2}$$

则有

$$\begin{aligned}D&=\sum_{i=1}^{n}\int_{a_{i-1}}^{a_i}(y_i-x)^2\,p(x)\mathrm{d}x\\&\approx\sum_{i=1}^{n}p(y_i)\int_{a_{i-1}}^{a_i}(y_i-x)^2\mathrm{d}x\\&=\frac{1}{12}\sum_{i=1}^{n}p(y_i)\,\Delta_i^3\end{aligned} \tag{5-20}$$

若量化为均匀量化，即量化区域是等间隔分布的量化设计，此时 $\Delta_i=\Delta$，于是有

$$D=\frac{\Delta^2}{12}\sum_{i=1}^{n}p(y_i)\,\Delta=\frac{\Delta^2}{12} \tag{5-21}$$

说明：均匀间隔量化器的均方误差与量化区间间隔的平方成正比，间隔越小，则平均失真就越小。

在均方误差测量下得到的平均失真 D 并不完全反映误差的客观效果。比如说对于相同的 D，输入信号大小不同时，所产生的效果也就不同。对大信号来说，其影响可以忽略。而对于小信号来说，却不能忽略。从这个意义上来说，采用一种叫做“信噪比”的测量则更为合适。定义为“信噪比”为

$$\frac{S}{N}=\frac{\sigma^2}{D}$$

σ^2 为输入到量化器的信号的方差，D 为误差信号的能量，即上面提到的均方误差值。

从减少比特率的途径来说，在实际应用中多采用非均匀量化、压扩量化、广适量化等。

下面我们介绍一种当输入信号的概率分布已知时，在均方误差最小的情况下，对量化器进行最优化设计的算法，即 Lloyd-Max 算法。

定义量化误差为量化器的输入、输出信号的差值的某种函数的均值，即

$$D = E[f(s_i - s_0)] = \sum_{j=0}^{n-1} \int_{x_j}^{x_{j+1}} f(x - y_i) p(x) \mathrm{d}x \tag{5-22}$$

其中，$f(d)$表示 d 的一个测量函数，在均方误差测量情况下，$f(d) = d^2$。而 x_i 表示量化层划分的判决电平，且 $x_0 = -\infty$，$x_n = +\infty$；y_i 表示第 i 层的输出电平。

现在的目的就是对某确定的 n，求量化误差 D 的极小值，变量为 x_i 和 y_i。即设法找出最佳的分层与最佳的输出电平。

分别将 D 对 x_i 和 y_i 求导，并分布令其为 0，有

$$\frac{\partial D}{\partial x_i} = f(x_i - y_{i-1}) p(x_i) - f(x_i - y_i) p(x_i) = 0, \quad i = 1, 2, \cdots, n-1 \tag{5-23}$$

以及

$$\frac{\partial D}{\partial y_i} = -\int_{x_i}^{x_{i+1}} f'(x - y_i) p(x) \mathrm{d}x = 0, \quad i = 1, 2, \cdots, n-1 \tag{5-24}$$

由于 $p(x_i) \neq 0$，因此式（5-23）的偏导数变为

$$f(x_i - y_{i-1}) = f(x_i - y_i), \quad i = 1, 2, \cdots, n-1$$

由于并不知道 $f(\cdot)$ 的具体形式，要满足这两个方程，还需进一步的假设。在常用的均方误差测量中，函数 $f(\cdot)$ 是单独递增的偶函数，那么有

$$| x_i - y_{i-1} | = | x_i - y_i |, \quad i = 1, 2, \cdots, n-1$$

又因为

$$y_{i-1} < x_i < y_i$$

则上式变为

$$x_i - y_{i-1} = y_i - x_i, \quad i = 1, 2, \cdots, n-1$$

所以

$$x_i = \frac{y_i + y_{i-1}}{2}, \quad i = 1, 2, \cdots, n-1 \tag{5-25}$$

该式说明量化分层的判决电平应设在两个输出电平的中点为最佳。量化的分层示意图如图 5-9 所示。

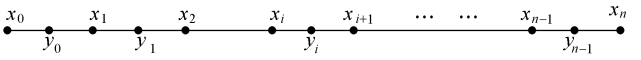


图 5-9　量化分层示意图

若假设 $f(d) = d^2$，则 $f'(d) = 2d$，因此式（5-24）的偏导数为

$$\int_{x_i}^{x_{i+1}} (x - y_i) p(x) \mathrm{d}x = 0, \quad i = 1, 2, \cdots, n-1 \tag{5-26}$$

类似于物理学中质量中心的公式，可以认为 y_i 的最佳位置便是概率密度 $p(x)$ 在 x_i 与 x_{i+1} 段的概率中心。

结论听起来比较简单，但要从该式子中计算出 y_i 并不容易，因为积分限并不确定，并且积分限又依赖于 y_i 的数值。

如何来解决这个问题呢？显然，可以利用反复迭代的办法。首先任意选一个 y_0，由式 $\int_{-\infty}^{x_1}(x-y_0)p(x)\mathrm{d}x=0$ 解出 x_1；再由 y_0 和 x_1 确定出 y_1，依次计算出 x_2、y_2、x_3、y_3、…，直到算出 y_{n-1}。

这样计算的结果当然不会满足最佳的条件，因为 y_0 是任选的。还要验证 y_{n-1} 是不是 $(x_{n-1}\sim+\infty)$ 段的概率中心。如果是，那么计算就完成；若不是，则需要重新估计并重新计算，直到结果正确为止。

在计算技术非常发达的今天，这个递归过程是能够实现的。对于高斯分布、拉普拉斯分布、伽马分布等，都有计算结果，在此不一一详述。

式（5-25）和式（5-26）仅仅是得到最小量化误差的必要条件，并不是充分条件。也就是说，即使满足了这两个方程，其解答不一定就是最佳的。在理论上还需要寻找充分条件。

上述最佳量化算法的意义，还不仅仅在于由它来设计一个标量量化器，更重要的是将这个思想推广到多自由度量化即矢量量化中，构成矢量量化的一种码书构成算法，具有更大的实用意义。

5.5.2 矢量量化编码

将标量量化的概念推广到多维，就是矢量量化，简称 VQ。VQ 的原理和算法相对来说比较简单，其压缩比大，在图像编码、语音编码等领域受到重视，其缺点是运算量较大。

1. VQ 的基本原理

若将每 K 个样值组成一个矢量 $X=(x_1,x_2,\cdots,x_K)$，称为 K 维随机矢量。定义一个 N 点的矢量量化器为一个映射 $Q(X)$，它把输入矢量 X 映射为 N 个输出矢量 $Y_1,Y_2,\cdots,Y_N$ 中的一个。换句话说，如果矢量 X 是属于某个区间 S_i，则其输出为 Y_i，用式子表示为

$$Q(X)=Y_i,\quad i=1,2,\cdots,N$$

这个过程为：首先将所有可能的输入取样组成一个 K 度实空间 R^K，然后按一定的规则将该空间分成 N 个“分割”，并标以 1,2,…,N，分别将每一个“分割”的区域叫做 $S_1,S_2,\cdots,S_N$，在每一个分割中根据 Lloyd 最优量化的思想，找出一个概率中心矢量 Y_i 来作为输出矢量。凡落在 S_i 中的输入矢量，就用 Y_i 作为输出，每一个输出矢量 Y_i 称为码字，所有这些 Y_i 的集合称为码书，如图 5-10 所示。

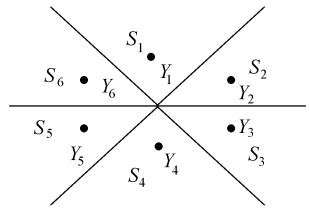


图 5-10 实空间 R^K 的分割

对某一输入矢量 X，确定其落入哪一块分割，这个过程称为“搜索”。因此，建立码书和快速搜索是矢量量化的两项关键技术。

矢量量化器的原理图如图 5-11 所示。

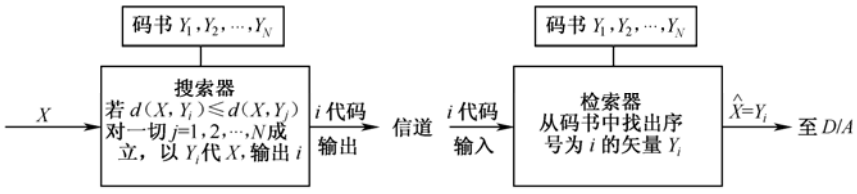


图 5-11 VQ 原理框图

这里的每一个输出矢量 Y_i 都是由输入矢量 X 的特征所决定的，即由 K 个样值联合起来共同确定，而不是由各个样值单独确定输出的。

2．码书的产生

码书的产生，主要解决两个基本问题：一是求出对空间的最佳分割，或者说对某一类矢量进行最佳分群的问题；二是在各个分割中找出最合适的代表矢量的问题。这两个问题需要反复迭代优化来解决。

Linde、Buzo、Gray 等人在 Lloyd 关于最优标量量化的基础上，发展了一种算法——LBG 算法。该算法的主要特点是将每个分割 S_i 的分布中心矢量作为代表矢量 Y_i，然后根据保真度要求，计算平均失真并重新进行分割，直至满足保真度要求为止。

矢量量化的失真度可表示为

$$d = E\left\|X - Q(X)\right\|^z$$

其中，$\|\cdot\|$ 表示范式或叫做模；E 表示均值。当 z=2 时，就是均方误差测量，此时

$$d = E\left\|X - Y\right\|^2 = E[(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_K - y_K)^2]$$

而每个取样的平均失真为

$$D = \frac{d}{K} = \frac{1}{K}\sum_{i=1}^{K} E[(x_i - y_i)^2]$$

下面介绍两种通过 LBG 算法来产生矢量量化中的码书的流程。

一种是在已知分布密度条件下的 LBG 算法。

① 初始化条件：给定量化级数 N、失真门限 ε、初始码书 Y_0 及信源分布 F。

② 对已知码书 $Y_m = \{Y_{mi};\ i = 1,2,\cdots,N\}$，从 m=0 开始，找出最小失真分割。即对所有的样值 X，作如下的比较判决：若 $d(X,Y_{mi}) \leqslant d(X,Y_{mj})$ 对所有的 j=1,2,⋯,N 都成立，则判定该 $X \in S_i$。

③ 计算平均失真

$$D_m = E\min_{Y \in \mathbf{Y}_m} d(X,Y)$$

若

$$\frac{D_{m-1} - D_m}{D_m} \leqslant \varepsilon,\ D_m \leqslant D$$

则输出 Y_m 作为码书；否则用 m+1，并执行下一步。

④ 求各分割的分布中心 C_i，i=1,2,⋯,N，并令 $Y_{m+1} = \{Y_{m+1,i} = C_i,\ i = 1,2,\cdots,N\}$，回到②。

另一种是在未知信源分布特性时的码书建立流程。

① 初始化条件：给定量化级数 N、失真门限 ε、初始码书 Y_0 及训练序列 $\{X_i;\ i = 1,2,\cdots,N\}$。（注：训练序列是对信源取得足够多的样值构成的样值序列。）

② 对码书 $Y_m = \{Y_{mi};\ i = 1,2,\cdots,N\}$，从 m=0 开始，找出最小失真分割，即若

$$d(X,Y_{mi}) = \frac{1}{K}\sum_{l=1}^{K}(x_l - y_{mi})^2 \leqslant d(X,Y_{mj}) = \frac{1}{K}\sum_{l=1}^{K}(x_l - y_{mj})^2$$

对所有的 j=1,2,⋯,K 都成立，则判定 $X \in S_i$。

③ 计算平均失真

$$D_m = \frac{1}{n}\sum_{i=1}^{n}\min_{Y\in Y_m} d(X,Y)$$

若

$$\frac{D_{m-1}-D_m}{D_m} \leqslant \varepsilon,\ D_m \leqslant D$$

则输出 Y_m 作为码书；否则用 m+1，并执行下一步。

④ 计算各分割的算术平均值或几何中心 G_i，i=1,2,…,N，并令 $Y_{m+1}=\{Y_{m+1,i}=G_i,\ i=1,2,\cdots,N\}$，回到②。

这两种建立码书的流程例子是可行的，但不一定是最好的，更不是唯一的，还有许多关键技术与理论问题有待进一步地研究和探讨。

3．编码和译码

建立了码书以后，矢量量化的编码和译码就是一种搜索或称作检索的过程。

由于编码输出为码书中矢量的序号代码，因此，译码过程就是按顺序进行查找，而不是进行比较判决的问题，比较简单。我们只讨论编码问题。

对随机输入的一个矢量 X，应该用码书中的哪一个矢量来代表呢？这里，按最小失真或称作最邻近映射的原则来作出判决，即若

$$d(X,Y_i) \leqslant d(X,Y_j),\ j\neq i,\ j=1,2,\cdots,N$$

对所有的 j 都成立，则 Y_i 便是码书中 X 的最小失真映射，i 的代码就是矢量信号 X 的编码输出。这里的 d 按下式

$$d = E\left\|X-Q(X)\right\|^2$$

或

$$D=\frac{d}{K}=\frac{1}{K}\sum_{i=1}^{K}E[(x_i-y_j)^2]$$

进行计算。

如何来实现“对所有的 j 都成立”这个判决呢？最直接、最简单的方法就是进行“全搜索”的方法。全搜索的意思就是将矢量 X 与码书中 N 个矢量逐个进行失真计算，然后找出失真最小的矢量 Y_i。由于 N 是有限的，因此，这样的比较寻找总是可以得到结果的。

但是，如果 N 很大时，所需的运算量与搜索的时间显然也是相当大的。可设法寻找一些快速搜索的方法。

总之，由于矢量量化在理论上依循了香农信息论编码定理的条件，在实际中得到了优质、低速率的编码效果，引起了人们的极大兴趣与广泛的研究。

5.6 预测编码

由信号的取样值构成的信号序列中，样值间存在着相关性。或者说由这些信号的取样值构成的信源是有记忆的信源。对这些有记忆的信源，通过去相关处理的方式实现信源的

编码。

解除信源相关性的方式主要包含两大类：一类是从时间域进行，称之为预测编码；另一类是应用函数的正交变换从频域上进行，称为变换编码。

预测就是从已收到的信号来提取关于未收到的信号的信息，将信号的预测值与实际值之间的差值进行编码，以达到压缩数据的目的。

预测的方法可以是线性的，也可以是非线性的。我们重点讨论线性预测编码。

5.6.1 线性预测编码的基本原理

线性预测编码，通常称为差值脉冲编码调制，简称 DPCM。

我们知道，信源有记忆时，信源输出的各个分量之间是统计相关的。这种统计相关性可以充分地加以利用。如果不直接对信源输出的信号进行量化编码，而是将信源输出信号先进行预测，然后再对信源输出的实际值与预测值之间的差值进行量化编码输出，此即 DPCM。编码器的原理框图如图 5-12 所示。

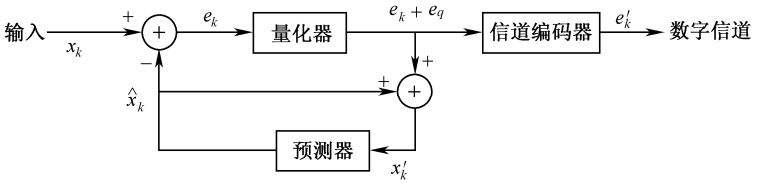


图 5-12 编码器原理

在接收端恢复信号的时候，需要一个相同的预测器作为译码器，其原理框图如图 5-13 所示。

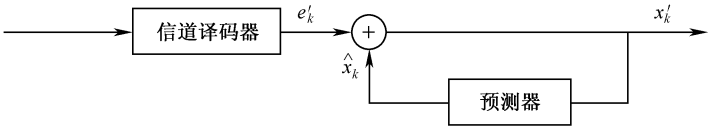


图 5-13 译码器原理

假定信源产生的序列为 $(x_1, x_2, \cdots, x_{k-1}, x_k, \cdots)$，$k$ 时刻的符号值 x_k 用过去的 $k-1$ 个符号值的线性组合来进行预测。

$$\hat{x}_k = a_1 x_1 + a_2 x_2 + \cdots + a_{k-1} x_{k-1} = \sum_{i=1}^{k-1} a_i x_i$$

k 时刻的实际值 x_k 与预测值 $\hat{x}_k$ 之间的预测误差值为

$$e_k = x_k - \hat{x}_k = x_k - \sum_{i=1}^{k-1} a_i x_i$$

若直接对信源的输出 x_k 进行编码，其平均码长要大于对预测误差值 e_k 进行编码的平均码长。因此可以说，经过预测以后可以大大压缩信源数码率。

从图 5-12 中可以看到，整个预测编码器的失真完全由量化器产生，如果信道传输没有误

差，则在接收端的失真就是量化失真。若输入信号不是模拟信号，而是数字信号，则量化器可以去掉。

5.6.2 最佳线性预测编码

我们希望预测尽可能的准确，从数据压缩的观点出发，如果预测误差为 0，则无需传送信息。从信息论的观点来说，能够被完全预测的信号是不带有任何信息量的。因此，就整个系统来说，预测误差还不能全为 0，否则就不可能传送信息了。

那么，如何做到使预测误差为非 0 的最小值呢？我们采用最小均方误差准则来进行讨论，这也称为在均方误差意义下的最佳线性预测器的设计。

用 k 个采样信号 $x_1, x_2, \cdots, x_{k+1}$ 进行讨论。假定 x_{k+1} 是当前时刻的信号值，用前 k 个采样信号对 x_{k+1} 进行预测。

$$\hat{x}_{k+1} = a_1 x_1 + a_2 x_2 + \cdots + a_k x_k = \sum_{i=1}^{k} a_i x_i$$

预测误差值为：$e_k = x_{k+1} - \hat{x}_{k+1} = x_{k+1} - \sum_{i=1}^{k} a_i x_i$。

均方误差为

$$\begin{aligned} D &= E[(x_{k+1} - \hat{x}_{k+1})^2] \\ &= E[(x_{k+1} - a_1 x_1 - a_2 x_2 - \cdots - a_k x_k)^2] \end{aligned}$$

要使 D 的值为极小值，需令

$$\frac{\partial D}{\partial a_i} = 0, \quad i = 1, 2, \cdots, k$$

可得

$$\frac{\partial D}{\partial a_i} = -2E[(x_{k+1} - a_1 x_1 - a_2 x_2 - \cdots - a_k x_k) \cdot x_i]$$

或

$$E[(x_{k+1} - \hat{x}_{k+1}) x_i] = 0, \qquad i = 1, 2, \cdots, k$$

这是由 k 个线性方程组成的方程组，可以解出 k 各预测系数 $a_1, a_2, \cdots, a_k$，使均方误差最小。

最简单的预测是 $\hat{x}_{k+1} = x_k$，即用前一个符号值作为当前的符号值，这种预测称为前值预测。

线性预测编码在语音、图像等编码中都有广泛的应用。

5.7 变换编码

对有记忆的连续信源进行信源编码的另一大类方式就是采用变换编码。变换编码就是将通常在时间域描写的信号（如语音信号）或空间域描写的信号（如图像信号），变换到另外一些正交矢量空间中进行描写，并使变换域中描写的各信号的分量之间相关性很小，或者不相关，从而达到压缩数据的目的。

假定信源输出为一维的矢量 $\boldsymbol{X}=(x_1,x_2,\cdots,x_n)$，经变换以后输出为 $\boldsymbol{Y}=(y_1,y_2,\cdots,y_n)$，而 $\boldsymbol{Y}=\boldsymbol{AX}$，$\boldsymbol{A}$ 为变换中的正交矩阵。

如果在正交变换后，只传送 $m<n$ 个样值，而将剩余的 $n-m$ 个能量较小的样值舍弃。在接收端，根据 m 个分量来恢复原始信息。因此，变换编码的问题归结为如何选择正交矩阵 A，使 m 值较小，使被舍弃的 $n-m$ 个取值足够小，以使变换后既能得到最大的信源压缩率，又能保证舍弃 $n-m$ 个值以后，所产生的误差不超过允许的失真范围。换句话说，正交变换的主要问题就是在一定的误差准则下，寻找最佳的或次最佳的正交变换，以达到最大限度地消除原始信源之间的相关性。

运用正交变换进行数据压缩编码的原理如图 5-14 所示。

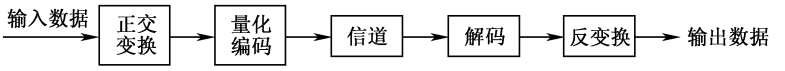


图 5-14　正交变换的原理

常用的正交变换的方法有 KLT、傅里叶变换、离散余弦变换、沃尔什-哈达玛变换、Haar 变换等。

5.7.1　正交变换与正交矩阵

1．正交矩阵

在 n 维线性空间中，有一个矢量信号 $\boldsymbol{X}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}$，一定可以找到一个被称作它的像的另一个矢量信号 $\boldsymbol{Y}=(y_1,y_2,\cdots,y_n)^{\mathrm{T}}$，且有关系

$$\boldsymbol{Y}=\boldsymbol{AX} \tag{5-27}$$

其中，$\boldsymbol{A}$ 为矩阵。

$$\boldsymbol{A}=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

称 $\boldsymbol{Y}$ 为 $\boldsymbol{X}$ 的一个线性变换。$\boldsymbol{A}$ 为变换矩阵，即变换在某组基矢量下的的矩阵。式（5-27）反映了 y 坐标系和 x 坐标系之间的基矢量的关系。

若线性变换保持 n 维空间中的矢量 $\boldsymbol{X}$ 的模不变，就称这种变换为正交变换，对应的变换矩阵 $\boldsymbol{A}$ 为正交矩阵。

由线性代数的知识可知，一个矩阵为正交矩阵的充分必要条件为

$$\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}=\boldsymbol{I} \quad 或 \quad \boldsymbol{A}=\boldsymbol{A}^{\mathrm{T}}$$

对于正交矩阵，由于其是满秩的，这样就保证了变换 $\boldsymbol{Y}=\boldsymbol{AX}$ 的各元素是一一对应的，就能够用反变换得到唯一确定的复原信号。

2．将实对称矩阵化为对角型矩阵

矩阵代数已经证明，对于一个实对称矩阵 $\boldsymbol{\Phi}$，必存在一个正交矩阵 $\boldsymbol{Q}$，使得

$$\boldsymbol{Q}\boldsymbol{\Phi}\boldsymbol{Q}^{-1}=\boldsymbol{Q}\boldsymbol{\Phi}\boldsymbol{Q}^{\mathrm{T}}=\begin{bmatrix}\lambda_1 & & & 0\\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n\end{bmatrix} \tag{5-28}$$

其中，$\lambda_1,\lambda_2,\cdots,\lambda_n$ 是实对称矩阵 $\boldsymbol{\Phi}$ 的 n 个特征根。正交矩阵 $\boldsymbol{Q}$ 的第 i 个行向量是 $\boldsymbol{\Phi}$ 的第 i 个特征根 λ_i 所对应的、满足归一化正交条件的特征向量 q_i 的转置。特征向量 q_i 满足

$$\boldsymbol{\Phi}\boldsymbol{q}_i=\lambda_i\boldsymbol{q}_i \tag{5-29}$$

及

$$\boldsymbol{q}_i^{\mathrm{T}}\boldsymbol{q}_j=\begin{cases}1, & i=j\\ 0, & i\neq j\end{cases} \tag{5-30}$$

5.7.2 K-L 变换

对于一个离散的信号序列，若每个信号由 n 个样点组成，那么就可以将该信号看做是 n 维空间中的一个点，每一个样值就代表 n 维信号空间中该矢量信号的一个分量，记为

$$\boldsymbol{X}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}$$

矢量 X 的协方差矩阵表示为

$$\boldsymbol{\Phi}_X=\begin{bmatrix}\varphi_{11} & \varphi_{12} & \cdots & \varphi_{1n}\\ \varphi_{21} & \varphi_{22} & \cdots & \varphi_{2n}\\ \vdots & \vdots & \ddots & \vdots\\ \varphi_{n1} & \varphi_{n2} & \cdots & \varphi_{nn}\end{bmatrix}=[\varphi_{ij}]$$

该矩阵的每一个元素定义为：$\varphi_{ij}=E[(x_i-Ex_i)(x_j-Ex_j)]$。

因此，$\boldsymbol{\Phi}_X$ 是一个实对称矩阵，它反映了信号 $\boldsymbol{X}$ 的各分量之间的相关性。若各分量之间互不相关，则该矩阵只有对角线上的元素值不为 0，且代表了各分量的方差。

由式（5-29）可知，一定存在一个与协方差矩阵 $\boldsymbol{\Phi}_X$ 相对应的正交矩阵 $\boldsymbol{Q}$，$\boldsymbol{Q}$ 的行向量就是 $\boldsymbol{\Phi}_X$ 的特征矢量的转置。

用正交矩阵 $\boldsymbol{Q}$ 对信号 $\boldsymbol{X}$ 进行正交变换，有 $\boldsymbol{Y}=\boldsymbol{Q}\boldsymbol{X}$，这个变换就称为 K-L 变换，$\boldsymbol{Q}$ 为 K-L 变换矩阵。

1．K-L 变换的步骤

根据已知信号的协方差矩阵，确定变换矩阵 $\boldsymbol{Q}$。

① 确定 $\boldsymbol{\Phi}_X$ 的特征根 λ_i，i=1,2,⋯,n。

② 根据 $\boldsymbol{\Phi}\boldsymbol{q}_i=\lambda_i\boldsymbol{q}_i$ 求特征向量。

③ 以 $\boldsymbol{q}_i$ 为列向量所得到的矩阵就是 $\boldsymbol{Q}$。

例 5-9 假定某随机信号 X 的协方差矩阵为

$$\boldsymbol{\Phi}_X=\begin{bmatrix}1 & 1 & 0\\ 1 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$$

求相应的正交矩阵 $\boldsymbol{Q}$。

解 ① 由 $\boldsymbol{\Phi}_X$ 求特征值，令

$$\begin{vmatrix} 1-\lambda & 1 & 0 \\ 1 & 1-\lambda & 0 \\ 0 & 0 & 1-\lambda \end{vmatrix} = 0$$

按 $\lambda_1 > \lambda_2 > \lambda_3$ 的次序排列，解出 $\lambda_1 = 2$，$\lambda_2 = 1$，$\lambda_3 = 0$。

② 求对应于 λ_i 的特征向量 $\boldsymbol{q}_i = (q_{i1}, q_{i2}, q_{31})^{\mathrm{T}}$，$i$=1,2,3。

由式（5-29），得三个方程组为

$$\begin{cases} q_{11} + q_{12} = 2q_{11} \\ q_{11} + q_{12} = 2q_{12} \\ q_{13} = 2q_{13} \end{cases} \Rightarrow q_{11} = q_{12} = a,\ q_{13} = 0,\ \text{即}\ \boldsymbol{q}_1 = (a, a, 0)^{\mathrm{T}}$$

$$\begin{cases} q_{21} + q_{22} = q_{21} \\ q_{21} + q_{22} = q_{12} \\ q_{23} = q_{23} \end{cases} \Rightarrow q_{21} = q_{22} = 0,\ q_{23} = b,\ \text{即}\ \boldsymbol{q}_2 = (0, 0, b)^{\mathrm{T}}$$

$$\begin{cases} q_{31} + q_{32} = 0 \\ q_{31} + q_{32} = 0 \\ q_{33} = 0 \end{cases} \Rightarrow q_{31} = -q_{32} = c,\ q_{33} = 0,\ \text{即}\ \boldsymbol{q}_3 = (c, -c, 0)^{\mathrm{T}}$$

a、b、c 为待定的实数。

由式（5-30），可解出：$a = \dfrac{\sqrt{2}}{2}$，$b = 1$，$c = \dfrac{\sqrt{2}}{2}$。

因此，对应于 λ_1、λ_2、λ_3 的三个归一化特征向量为

$$\boldsymbol{q}_1 = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0\right)^{\mathrm{T}},\quad \boldsymbol{q}_2 = (0, 0, 1)^{\mathrm{T}},\quad \boldsymbol{q}_3 = \left(\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}, 0\right)^{\mathrm{T}}$$

③ 写出正交矩阵 $\boldsymbol{Q}$

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{q}_1^{\mathrm{T}} \\ \boldsymbol{q}_2^{\mathrm{T}} \\ \boldsymbol{q}_3^{\mathrm{T}} \end{bmatrix} = \frac{\sqrt{2}}{2} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & \sqrt{2} \\ 1 & -1 & 0 \end{bmatrix}$$

可以验证一下，用正交矩阵 $\boldsymbol{Q}$ 对 $\boldsymbol{\Phi}_X$ 进行变换。

$$\boldsymbol{Q}\boldsymbol{\Phi}_X\boldsymbol{Q}^{\mathrm{T}} = \frac{\sqrt{2}}{2} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & \sqrt{2} \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & -1 \\ 0 & \sqrt{2} & \sqrt{2} \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

说明经 K-L 变换后，X 的各分量之间的相关性被完全解除。

2．K-L 变换的性质

K-L 变换大多是对离散信号进行变换的，归纳起来，该变换有如下几条重要的性质。

① 去相关性最彻底。对矢量信号 $\boldsymbol{X}$ 进行 K-L 变换后，所得矢量信号 $\boldsymbol{Y}$ 的各分量之间互

不相关。

② 能量集中特征。对 n 维矢量信号 X 进行 K-L 变换后，最大的方差将集中在矢量信号 Y 的前 m 个分量之中（$m<n$）。

③ 最佳性。K-L 变换是在均方误差准则下，失真最小的一种变换，所以又称为最佳变换。其失真量为被略去的各分量的方差之和。

④ 没有快速算法。K-L 变换矩阵随不同的矢量信号而不同，且无快速算法，这就造成了 K-L 变换难以应用于实际的信源。

既考虑变换的效果，同时又兼顾到实用性价值，陆续出现了一些准最佳的变换方法。所谓准最佳正交变换是指变换后的协方差矩阵是近似的对角线矩阵，下面分别讨论。

5.7.3 离散傅里叶变换

一维傅里叶变换已为大家所熟悉，这里，我们只讨论应用于信源编码的二维离散傅里叶变换。

定义信号 $\boldsymbol{X}$ 的二维离散傅里叶变换为

$$\boldsymbol{Y}=\boldsymbol{A}_{\mathrm{DF}}\boldsymbol{X}\boldsymbol{A}_{\mathrm{DF}}^{*\mathrm{T}},\quad \boldsymbol{X}=\boldsymbol{A}_{\mathrm{DF}}^{*\mathrm{T}}Y\boldsymbol{A}_{\mathrm{DF}} \tag{5-31}$$

若已知信号 $\boldsymbol{X}$ 的协方差矩阵，则定义为

$$\boldsymbol{\Phi}_Y=\boldsymbol{A}_{\mathrm{DF}}\boldsymbol{\Phi}_X\boldsymbol{A}_{\mathrm{DF}}^{*\mathrm{T}},\quad \boldsymbol{\Phi}_X=\boldsymbol{A}_{\mathrm{DF}}^{*\mathrm{T}}\boldsymbol{\Phi}_Y\boldsymbol{A}_{\mathrm{DF}} \tag{5-32}$$

变换式中的 $\boldsymbol{A}_{\mathrm{DF}}$ 为二维离散傅里叶变换的变换矩阵

$$\boldsymbol{A}_{\mathrm{DF}}=\frac{1}{\sqrt{N}}\begin{bmatrix} W^0 & W^0 & W^0 & \cdots & W^0 \\ W^0 & W^1 & W^2 & \cdots & W^{N-1} \\ W^0 & W^2 & W^4 & \cdots & W^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ W^0 & W^{N-1} & W^{2(N-1)} & \cdots & W^{(N-1)^2} \end{bmatrix}$$

其中，$W=\mathrm{e}^{-\mathrm{j}\frac{2\pi}{2}}$，$\boldsymbol{A}_{\mathrm{DF}}^{*\mathrm{T}}$ 为 $\boldsymbol{A}_{\mathrm{DF}}$ 的共轭转置。

一般情况下，N 取 2 的幂次（N=2,4,8,…）。N=4 时，变换矩阵为

$$\boldsymbol{A}_{\mathrm{DF}}=\frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \mathrm{e}^{-\mathrm{j}\frac{\pi}{2}} & \mathrm{e}^{-\mathrm{j}\pi} & \mathrm{e}^{-\mathrm{j}\frac{3\pi}{2}} \\ 1 & \mathrm{e}^{-\mathrm{j}\pi} & \mathrm{e}^{-\mathrm{j}2\pi} & \mathrm{e}^{-\mathrm{j}3\pi} \\ 1 & \mathrm{e}^{-\mathrm{j}\frac{3\pi}{2}} & \mathrm{e}^{-\mathrm{j}3\pi} & \mathrm{e}^{-\mathrm{j}\frac{9\pi}{2}} \end{bmatrix}=\frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -\mathrm{j} & -1 & \mathrm{j} \\ 1 & -1 & 1 & -1 \\ 1 & \mathrm{j} & -1 & -\mathrm{j} \end{bmatrix}$$

例 5-10 已知信源 $\boldsymbol{X}$ 的协方差矩阵为

$$\boldsymbol{\Phi}_X=\begin{bmatrix} 1 & 0.9 & 0.9 & 0.9 \\ 0.9 & 1 & 0.9 & 0.9 \\ 0.9 & 0.9 & 1 & 0.9 \\ 0.9 & 0.9 & 0.9 & 1 \end{bmatrix}$$

对其进行 DFT。

解 由给定的协方差矩阵可知，N 取 4，按定义，有

$$\begin{aligned}\boldsymbol{\Phi}_Y &= \boldsymbol{A}_{\mathrm{DF}}\boldsymbol{\Phi}_X\boldsymbol{A}_{\mathrm{DF}}^{*\mathrm{T}}\\ &= \frac{1}{2}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -\mathrm{j} & -1 & \mathrm{j}\\ 1 & -1 & 1 & -1\\ 1 & \mathrm{j} & -1 & -\mathrm{j}\end{bmatrix}\begin{bmatrix}1 & 0.9 & 0.9 & 0.9\\ 0.9 & 1 & 0.9 & 0.9\\ 0.9 & 0.9 & 1 & 0.9\\ 0.9 & 0.9 & 0.9 & 1\end{bmatrix}\frac{1}{2}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & \mathrm{j} & -1 & -\mathrm{j}\\ 1 & -1 & 1 & -1\\ 1 & -\mathrm{j} & -1 & \mathrm{j}\end{bmatrix}\\ &= \begin{bmatrix}3.7 & 0 & 0 & 0\\ 0 & 0.1 & 0 & 0\\ 0 & 0 & 0.1 & 0\\ 0 & 0 & 0 & 0.1\end{bmatrix}\end{aligned}$$

可见，$\boldsymbol{X}$ 的协方差经 DFT 以后，所得 $\boldsymbol{Y}$ 的协方差矩阵只有对角线上的元素，说明信号 $\boldsymbol{X}$ 的各分量之间的相关性已全部得到解除。

在 DFT 中，存在着复数运算，若变换的 N 值很大时，将使运算量很大。好在 DFT 有快速算法 FFT。因此，运算速度、存储容量等要求都可以得到满足，是一种很有实用价值的准最佳变换方法。

另外，经 DFT 以后的协方差矩阵与信源 X 的统计特性有关，而变换矩阵在 N 值确定以后则不会发生改变，也是确定的。这样，不是所有的信源经 DFT 以后都能达到很理想的效果。总体说来，大多数信源经 DFT 以后的相关性都可以得到解除。

5.7.4 离散余弦变换

对时间域、空间域的信号进行 DFT 的时候，需要进行复数域的运算。当变换的点数 N 很大时，由于复数运算量大，以致难以实时处理。为了克服 DFT 的这些缺点，在寻求快速算法的同时，人们还构造了一种实数域的变换——离散余弦变换，简称 DCT。

DCT 的性能很适用于人类语言和图像信号的特点，从这个意义上，常被认为是接近于 K-LT 变换性能的准最佳变换。

定义信号 X 的二维离散余弦变换为

$$\boldsymbol{Y} = \boldsymbol{C}_N\boldsymbol{X}\boldsymbol{C}_N^{\mathrm{T}},\quad \boldsymbol{X} = \boldsymbol{C}_N^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{C}_N \tag{5-33}$$

若已知信号 X 的协方差矩阵，则定义为

$$\boldsymbol{\Phi}_Y = \boldsymbol{C}_N\boldsymbol{\Phi}_X\boldsymbol{C}_N^{\mathrm{T}},\quad \boldsymbol{\Phi}_X = \boldsymbol{C}_N^{\mathrm{T}}\boldsymbol{\Phi}_Y\boldsymbol{C}_N \tag{5-34}$$

变换式中的 $\boldsymbol{C}_N$ 为二维离散余弦变换的变换矩阵

$$\boldsymbol{C}_N = \frac{1}{\sqrt{N}}\begin{bmatrix}1 & 1 & \cdots & 1\\ \sqrt{2}[\cos\dfrac{\pi}{2N} & \cos\dfrac{3\pi}{2N} & \cdots & \cos\dfrac{2N-1}{2N}\pi]\\ \sqrt{2}[\cos\dfrac{2\pi}{2N} & \cos\dfrac{6\pi}{2N} & \cdots & \cos\dfrac{2N-1}{2N}2\pi]\\ \vdots & \vdots & \ddots & \vdots\\ \sqrt{2}[\cos\dfrac{N-1}{2N}\pi & \cos\dfrac{N-1}{2N}3\pi & \cdots & \cos\dfrac{N-1}{2N}(2N-1)\pi]\end{bmatrix}$$

例 5-11 已知信源 X 的协方差矩阵为

$$\boldsymbol{\Phi}_X = \begin{bmatrix} a & b & b & b \\ b & a & b & b \\ b & b & a & b \\ b & b & b & a \end{bmatrix}$$

对其进行 DCT。

解 由给定的协方差矩阵可知，N 取 4，变换矩阵为

$$\boldsymbol{C}_4 = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ c & c & -s & c \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ s & -c & c & -s \end{bmatrix}$$

其中，$c = \frac{1}{\sqrt{2}}\cos\frac{\pi}{8}$，$s = \frac{1}{\sqrt{2}}\sin\frac{\pi}{8}$。

由定义，可得

$$\begin{aligned} \boldsymbol{\Phi}_Y &= \boldsymbol{C}_4 \boldsymbol{\Phi}_X \boldsymbol{C}_4^{\mathrm{T}} \\ &= \begin{bmatrix} a+3b & 0 & 0 & 0 \\ 0 & a-b & 0 & 0 \\ 0 & 0 & a-b & 0 \\ 0 & 0 & 0 & a-b \end{bmatrix} \end{aligned}$$

从本例的变换结果可见，信号各分量之间的相关性已被完全解除。

DCT 的快速算法已出现了多种，使得实用中的成本降低、实时处理速度提高，且变换性能很好，在语音信号和图像信号的处理中，得到了非常广泛的应用。

5.7.5 离散沃尔什–哈达玛变换

沃尔什（Walsh）函数系是一种完备的正交函数系，它只取两种值：在归一化条件下，只取 ±1 而没有中间值。它具有和傅立叶函数系相类似的一系列性质，但其运算却要简单的多。

Walsh 函数有 3 种不同的编码形式：Walsh 编码、Hadamard 编码和 Paley 编码。

1．Walsh 编码

在函数正交区间为[0,1]内定义

$$\begin{aligned} Wal(n,t) &= \prod_{m=0}^{p-1} [r(m+1,t)]^{g_m} \\ &= \prod_{m=0}^{p-1} (-1)^{t_m g_m} \\ &= (-1)^{\sum_{m=0}^{p-1} t_m g_m} \end{aligned} \tag{5-35}$$

式中，n 称为编号，n=0,1,2,…。比如 n=2 就叫作第 2 号 Walsh 函数。

p 是 n 的二进制编码的位数，若 n=6，其二进制码为 110，则 p 就是 3。

g_m 是 n 的 Gray 码的第 m 位的数值。Gray 码又叫反射二进码，由自然二进码演变产生，其法则为：若自然二进码为 $(a_n a_{n-1}\cdots a_1 a_0)$，则对应的 Gray 码便为 $(g_n g_{n-1}\cdots g_1 g_0)$，其中 $g_n = a_n$，$g_{n-1} = a_n \oplus a_{n-1}$，…，$g_0 = a_1 \oplus a_0$。即保留自然码的最高位不变，其余各位都由自然码相邻两位模 2 加产生。

比如：$n = 6 \Rightarrow$ 自然二进码为$(110) \Rightarrow$ Gray 码为 $(101)_g$。

Walsh 函数的定义式看起来很复杂，但实际上就是+1 或−1 的连乘运算。

若将[0,1)区间分为 N 等分（$N = 2^p$），在各区间内对函数进行采样，便可得到离散的 Walsh 函数。

$$Wal(n,T) = (-1)^{\sum_{m=0}^{p-1} T_{(p-1-m)} \cdot g_m} \tag{5-36}$$

T 为取样点的序数，依次为第 0 次、第 1 次、…、第 N−1 次；$T_{(p-1-m)}$ 为序数 T 的二进制表示，即：$(T)_{10} = (\cdots T_j \cdots T_1 T_0)_2$。

给定 n 值、T 值（T=0,1，2,…, N−1），将函数值排列成矩阵，就是 Walsh 变换矩阵。n=8 时，变换矩阵为

$$\boldsymbol{W}_8 = \begin{vmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{vmatrix}$$

2．Hadamard 编码

定义 Hadamard 编码的 Walsh 函数为

$$\begin{aligned} Wal_h(n,t) &= \prod_{m=0}^{p-1} [r(m+1,t)]^{<n_m>} \\ &= (-1)^{\sum_{m=0}^{p-1} t_m n_{p-1-m}} \end{aligned} \tag{5-37}$$

其中，$<n_m>$ 表示 n 的二进制比特倒置后的第 m 位，即

$$<n_m> = n_{p-1-m}$$

离散 Hadamard 变换矩阵可以对连续函数波形取样得到，也可以对离散 Walsh 变换矩阵调整得到。

Hadamard 变换矩阵有一个突出的优点，它的构造特别有规律。任意 N 阶 Hadamard 变换矩阵 H_N 都可由 2 阶 Hadamard 变换矩阵 H_2 通过直积而产生。

若定义 2 阶 Hadamard 变换矩阵为

$$\boldsymbol{H}_2=\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$$

则直积 $\boldsymbol{H}_2\otimes\boldsymbol{H}_A(A=2^n)$ 就是以矩阵 $\boldsymbol{H}_A$ 去代替 $\boldsymbol{H}_2$ 中的 1，以 $-\boldsymbol{H}_A$ 去代替 $\boldsymbol{H}_2$ 中的−1。那么，4 阶 Hadamard 变换矩阵 $\boldsymbol{H}_4$ 可由直积直接得到。

$$\boldsymbol{H}_4=\boldsymbol{H}_2\otimes\boldsymbol{H}_2=\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}$$

8 阶变换矩阵为

$$\boldsymbol{H}_8=\boldsymbol{H}_2\otimes\boldsymbol{H}_4=\begin{bmatrix}\boldsymbol{H}_4 & \boldsymbol{H}_4\\ \boldsymbol{H}_4 & -\boldsymbol{H}_4\end{bmatrix}$$

2^n 阶变换矩阵为

$$\boldsymbol{H}_{2^n}=\boldsymbol{H}_2\otimes\boldsymbol{H}_{2^{n-1}}=\begin{bmatrix}\boldsymbol{H}_{2^{n-1}} & \boldsymbol{H}_{2^{n-1}}\\ \boldsymbol{H}_{2^{n-1}} & -\boldsymbol{H}_{2^{n-1}}\end{bmatrix}$$

3．Walsh-Hadamard 变换

定义信号 $\boldsymbol{X}$ 的二维离散 Walsh 变换为

$$\boldsymbol{Y}=\frac{1}{N^2}\boldsymbol{W}_N\boldsymbol{X}\boldsymbol{W}_N,\quad \boldsymbol{X}=\boldsymbol{W}_N\boldsymbol{Y} \tag{5-38}$$

若已知信号 $\boldsymbol{X}$ 的协方差矩阵，则定义为

$$\boldsymbol{\Phi}_Y=\frac{1}{N^2}\boldsymbol{W}_N\boldsymbol{\Phi}_X\boldsymbol{W}_N,\quad \boldsymbol{\Phi}_X=\boldsymbol{W}_N\boldsymbol{\Phi}_Y\boldsymbol{W}_N \tag{5-39}$$

定义信号 $\boldsymbol{X}$ 的二维离散 Hadamard 变换为

$$\boldsymbol{Y}=\frac{1}{N^2}\boldsymbol{H}_N\boldsymbol{X}\boldsymbol{H}_N,\quad \boldsymbol{X}=\boldsymbol{H}_N\boldsymbol{Y} \tag{5-40}$$

若已知信号 $\boldsymbol{X}$ 的协方差矩阵，则定义为

$$\boldsymbol{\Phi}_Y=\frac{1}{N^2}\boldsymbol{H}_N\boldsymbol{\Phi}_X\boldsymbol{H}_N,\quad \boldsymbol{\Phi}_X=\boldsymbol{H}_N\boldsymbol{\Phi}_Y\boldsymbol{H}_N \tag{5-41}$$

例 5-12 设 $\boldsymbol{X}=[1,2,1,1,3,2,1,2]^{\mathrm{T}}$，分别计算其 Walsh 变换和 Hadamard 变换。

解 可以看到，$\boldsymbol{X}$ 有 8 各分量，变换矩阵应为 8 × 8，有

Walsh 变换：$\boldsymbol{Y}=\frac{1}{8}\boldsymbol{W}_8\boldsymbol{X}=\left[\frac{13}{8},-\frac{3}{8},-\frac{1}{8},\frac{3}{8},\frac{1}{8},-\frac{3}{8},-\frac{1}{8},-\frac{3}{8}\right]^{\mathrm{T}}$。

Hadamard 变换：$\boldsymbol{Y}=\frac{1}{8}\boldsymbol{H}_8\boldsymbol{X}=\left[\frac{13}{8},-\frac{1}{8},\frac{3}{8},\frac{1}{8},-\frac{1}{8},-\frac{1}{8},-\frac{1}{8},-\frac{3}{8}\right]^{\mathrm{T}}$。

例 5-13 已知信源 $\boldsymbol{X}$ 的协方差矩阵为

$$\boldsymbol{\Phi}_X = \begin{bmatrix} 1 & 0 & 0 & 0.9 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0.9 & 0 & 0 & 1 \end{bmatrix}$$

对其进行 Hadamard 变换。

解 由变换的定义式，有

$$\boldsymbol{\Phi}_Y = \frac{1}{4}\boldsymbol{H}_4\boldsymbol{\Phi}_X\boldsymbol{H}_4 == \begin{bmatrix} -1.7 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix}$$

可见，经变换后，$\boldsymbol{X}$ 的相关性已被解除。

5.7.6 离散 Haar 变换

在各种正交变换中，Haar 变换是比较简单又容易实现的一种变换方法，它具有既反映全体又反映局部的特点。

1. Haar 函数

Haar 函数组是完备的、正交的非正弦型周期函数，保证了逆变换是唯一的。

连续的 Haar 函数记为：$har(n,m,t)$，t 为时间变量，n 和 m 为两个参数，定义为

$$har(0,0,t) = 1,\quad t \in [0,1)$$

$$har(n,m,t) = \begin{cases} \sqrt{2^n}, & (m-1)/2^n \leqslant t < (m-\frac{1}{2})/2^n \\ -\sqrt{2^n}, & (m-\frac{1}{2})/2^n \leqslant t < m/2^n \\ 0, & \text{对} t \in [0,1) \text{其他地方} \end{cases} \tag{5-42}$$

参数 n 称为阶数，当函数图形占满区间[0,1)时，称 0 阶 Haar 函数。若把 $har(0,1,t)$ 在区间[0,1)上压缩一半，构成的图形称 1 阶 $har(1,1,t)$，再压缩一半就构成 2 阶 $har(2,1,t)$ …，依此类推，记为 n=0,1,2,…。

参数 m 称次数，经压缩以后的图形向右移位的次数，m=1,2,…。

比如，称为 2 阶 3 次 Haar 函数，它是由 $har(0,1,t)$ 经压缩 2 次、右移 2 次而成，每压缩一次，函数的取值要乘以 $\sqrt{2}$。每次经压缩、右移后，函数占满[0,1)区间为止。

对连续 Haar 函数作 $N = 2^r$ 次等间隔取样，就得到离散的 Haar 函数，记为 $H(r,m,t)$或 $H(r)$，r=n+1 代表总阶数，包括 0 阶。将所有的取样值排列成一个矩阵，就构成了离散 Haar 变换矩阵。

N=4 的 Haar 变换矩阵为

$$\boldsymbol{H}(2) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix}$$

N=8 的 Haar 变换矩阵为

$$\boldsymbol{H}(3)=\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix}$$

2．Haar 变换

定义二维离散 Haar 变换为

$$\boldsymbol{Y}=\frac{1}{N}\boldsymbol{H}(r)\boldsymbol{X}\boldsymbol{H}(r)^{\mathrm{T}},\quad \boldsymbol{X}=\frac{1}{N}\boldsymbol{H}(r)^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{H}(r) \tag{5-43}$$

若已知信号 X 的协方差矩阵，则定义为

$$\boldsymbol{\Phi}_Y=\frac{1}{N}\boldsymbol{H}(r)\boldsymbol{\Phi}_X\boldsymbol{H}(r)^{\mathrm{T}},\quad \boldsymbol{\Phi}_X=\frac{1}{N}\boldsymbol{H}(r)^{\mathrm{T}}\boldsymbol{\Phi}_X\boldsymbol{H}(r) \tag{5-44}$$

例 5-14 已知信源 $\boldsymbol{X}$ 的协方差矩阵为

$$\boldsymbol{\Phi}_X=\begin{bmatrix} a & b & b & b \\ b & a & b & b \\ b & b & a & b \\ b & b & b & a \end{bmatrix}$$

对其进行 Haar 变换。

解 由变换的定义式，有

$$\begin{aligned}\boldsymbol{\Phi}_Y&=\frac{1}{4}\boldsymbol{H}(2)\boldsymbol{\Phi}_X\boldsymbol{H}(2)^{\mathrm{T}}\\&=\begin{bmatrix} a+3b & 0 & 0 & 0 \\ 0 & a-b & 0 & 0 \\ 0 & 0 & a-b & 0 \\ 0 & 0 & 0 & a-b \end{bmatrix}\end{aligned}$$

从本例的变换结果可见，信号各分量之间的相关性已被完全解除。

在实际应用中，要进行变换编码，首先要选择一种合适的正交变换。在性能满足要求的条件下，尽可能选择实现容易、简单的变换方法。

习　　题

5.1 有一信源，它有 6 种可能的输出，其概率分布如下表所示，表中给出了对应的 6 种编码 C_1、C_2、C_3、C_4、C_5 和 C_6。

（1）求这些码中哪些是唯一可译码。

（2）求哪些是非延长码（即时码）。

（3）对所有唯一可译码求出其平均码长。

消息	概率	C_1	C_2	C_3	C_4	C_5	C_6
a_1	1/2	000	0	0	0	1	01
a_2	1/4	001	01	10	10	000	001
a_3	1/16	010	011	110	1101	001	100
a_4	1/16	011	0111	1110	1100	010	101
a_5	1/16	100	01111	11110	1001	110	110
a_6	1/16	101	011111	111110	1111	110	111

5.2　证明若存在一个码长为 $l_1,l_2,\cdots,l_q$ 的唯一可译码，则一定存在具有相同码长的即时码。

5.3　设信源 $\begin{bmatrix} S \\ P(s) \end{bmatrix}=\begin{bmatrix} s_1 & s_2 & \cdots & s_6 \\ p_1 & p_2 & \cdots & p_6 \end{bmatrix}$，$\sum_{i=1}^{6} p_i=1$。将此信源编码成为 r 元唯一可译变长码（即码符号集 $X=\{x_1,x_2,\cdots,x_r\}$），其对应的码长为（$l_1,l_2,\cdots,l_6$）=（1，1，2，3，2，3），求 r 值的最小下限。

5.4　设某城市有 805 门公务电话和 60 000 门居民电话。作为系统工程师，你需要为这些用户分配电话号码。所有号码均是十进制数，且不考虑电话系统中 0、1 不可用在号码首位的限制。（提示：用异前缀码概念）

（1）如果要求所有公务电话号码为 3 位长，所有居民电话号码等长，求居民号码长度 L_1 的最小值。

（2）设城市分为 A、B 两个区，其中 A 区有 9 000 门电话，B 区有 51 000 门电话。现进一步要求 A 区的电话号码比 B 区的短 1 位，试求 A 区号码长度 L_2 的最小值。

5.5　求概率分布为 $\left(\frac{1}{3},\frac{1}{5},\frac{1}{5},\frac{2}{15},\frac{2}{15}\right)$ 的信源的二元霍夫曼码。讨论此码对于概率分布为 $\left(\frac{1}{5},\frac{1}{5},\frac{1}{5},\frac{1}{5},\frac{1}{5}\right)$ 的信源也是最佳二元码。

5.6　设二元霍夫曼码为(00, 01, 10, 11)和(0, 10, 110, 111)，求出可以编这些霍夫曼码的信源的所有概率分布。

5.7　设一信源有 K=6 个符号，其概率分别为：$P(s_1)=1/2$，$P(s_2)=1/4$，$P(s_3)=1/8$，$P(s_4)=P(s_5)=1/20$，$P(s_6)=1/40$，对该信源进行霍夫曼二进制编码，并求编码效率。

5.8　设信源概率空间为：

$$\begin{bmatrix} S \\ P(s) \end{bmatrix}=\begin{bmatrix} s_1, & s_2 \\ 0.1, & 0.9 \end{bmatrix},$$

（1）求 $H(S)$ 和信源冗余度；

（2）设码符号为 X={0,1}，编出 S 的紧致码，并求紧致码的平均码长 $\overline{L}$；

（3）把信源的 N 次无记忆扩展信源 S^N 编成紧致码，试求 N=2，3，4，∞ 时的平均码长 $\left(\overline{\frac{L_N}{N}}\right)$；

（4）计算上述 N=1, 2, 3, 4 这 4 种码的编码效率和码冗余度。

5.9 设信源空间为：$\begin{bmatrix} S \\ P(\mathrm{s}) \end{bmatrix}=\begin{bmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 \\ 0.4 & 0.2 & 0.1 & 0.1 & 0.05 & 0.05 & 0.05 & 0.05 \end{bmatrix}$，码符号为 X={0,1,2}，试构造一种三元紧致码。

5.10 某气象员报告气象状态，有 4 种可能的消息：晴、云、雨和雾。若每个消息是等概的，那么发送每个消息最少所需的二元脉冲数是多少？又若 4 个消息出现的概率分别是 1/4，1/8，1/8 和 1/2，问在此情况下消息所需的二元脉冲数是多少？如何编码？

5.11 若某一信源有 N 个符号，并且每个符号等概率出现，对这信源用最佳霍夫曼码进行二元编码，问当 $N=2^i$ 和 $N=2^i+1$ （i 是正整数）时，每个码字的长度等于多少？平均码长是多少？

5.12 若有一信源

$$\begin{bmatrix} S \\ P(s) \end{bmatrix}=\begin{bmatrix} s_1 \ , \ s_2 \\ 0.8, 0.2 \end{bmatrix}$$

每秒钟发出 2.66 个信源符号。将此信源的输出符号送入某一个二元信道中进行传输（假设信道是无噪无损的），而信道每秒钟只传递 2 个二元符号。试问信源不通过编码能否直接与信道连接？若通过适当编码能否在此信道中进行无失真传输？若能连接，试说明如何编码并说明原因。

5.13 现有一幅已离散量化后的图像，图像的灰度量化分成 8 级，如下表所示。表中数字为相应像素上的灰度级。

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	6	6	6	6
7	7	7	7	7	8	8	8	8	8

另有一无噪无损二元信道，单位时间（秒）内传输 100 个二元符号。

（1）现将图像通过给定的信道传输，不考虑图像的任何统计特性，并采用二元等长码，问需多长时间才能传送完这幅图像？

（2）若考虑图像的统计特性（不考虑图像的像素之间的依赖性），求这图像的信源熵 $H(S)$，并对每个灰度级进行霍夫曼最佳二元编码，问平均每个像素需用多少二元码符号来表示？这时需多少时间才能传送完这幅图像？

（3）从理论上简要说明这幅图像还可以压缩，而且平均每个像素所需的二元码符号数可

以小于 $H(S)$ 比特。

5.14 设某无记忆二元信源，概率 $p_1=P(1)=0.1$， $p_0=P(0)=0.9$，采用下述游程编码方案：第一步，根据 0 的游程长度编成 8 个码字，第二步，将 8 个码字变换成二元变长码，如下表所示。

信源符号序列	中间码	二元码字
1	S_0	1000
01	S_1	1001
001	S_2	1010
0001	S_3	1011
00001	S_4	1100
000001	S_5	1101
0000001	S_6	1110
00000001	S_7	1111
00000000	S_8	0

（1）试问最后的二元变长码是否是唯一可译码。

（2）试求中间码对应的信源序列的平均长度 $\overline{L_1}$。

（3）试求中间码对应的变长码二元码码字的平均长度 $\overline{L_2}$。

（4）计算比值 $\overline{L_2}/\overline{L_1}$，解释它的意义，并计算这种游程编码的编码效率。

（5）若用霍夫曼编码，对信源的四次扩展信源进行直接编码，求它的平均码长 $\overline{L}$（对应于每一个信源符号），并计算编码效率，试将此方法与游程编码方法进行比较。

（6）将上述游程编码方法一般化，可把 2^s+1 个信源序列（上例中 s=3）变换成二元变长码，即 2^s 个连零的信源序列编为码字 0，而其他信源序列都编成 s+1 位的码字。若信源输出零的概率为 p_0，求 $\overline{L_2}/\overline{L_1}$ 的一般表达式，并求 p_0=0.995 时 s 的最佳值。

5.15 有两个信源 X 和 Y 如下：

$$\begin{bmatrix} X \\ P(x) \end{bmatrix}=\begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ 0.2 & 0.19 & 0.18 & 0.17 & 0.15 & 0.1 & 0.01 \end{bmatrix}$$

$$\begin{bmatrix} Y \\ P(y) \end{bmatrix}=\begin{bmatrix} y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 & y_8 & y_9 \\ 0.49 & 0.14 & 0.14 & 0.07 & 0.07 & 0.04 & 0.02 & 0.02 & 0.01 \end{bmatrix}$$

（1）分别用霍夫曼码编成二元变长唯一可译码，并计算编码效率；

（2）分别用香农编码法编成二元变长唯一可译码，并计算编码效率；

（3）分别用费诺编码法编成二元变长唯一可译码，并计算编码效率；

（4）从 X， Y 两种不同信源来比较这 3 种编码方法的优缺点。

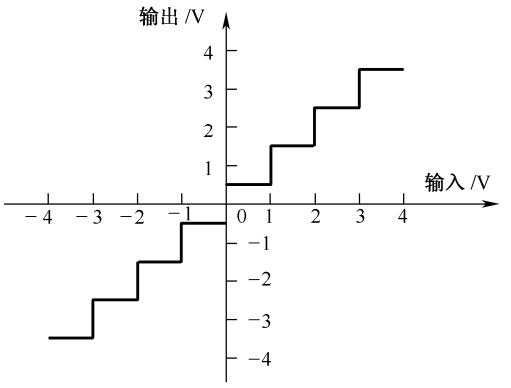
5.16 将幅度为 3.25V、频率为 800Hz 的正弦信号输入采样频率为 8 000Hz 采样保持器后，通过一个如题图 5-16 所示量化数为 8 的中升均匀量化器。试画出均匀量化器的输出波形。

5.17 已知某采样时刻的信号值 x 的概率密度函数 $p(x)$ 如题图 5-17 所示，将 x 通过一个量化数为 4 的中升均匀量化器得到输出 x_q。试求：

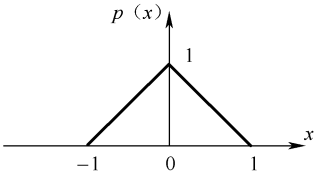
（1）输出 x_q 的平均功率 $S=E[x_q^2]$；

（2）量化噪声 $e=\left|x_q-x\right|$ 的平均功率 $N_q=E[e^2]$；

（3）量化信噪比 S/N_q。



题图 5-16



题图 5-17

5.18 在 CD 播放机中，假设音乐是均匀分布，采样频率为 44.1kHz，采用 16 比特的中升均匀量化器进行量化。试确定 50 分钟音乐所需要的比特数，并求量化信噪比 S/N_q。

5.19 采用 13 折线 A 律非均匀量化编码，设最小量化间隔为 Δ，已知某采样时刻的信号值 x=635Δ。

（1）试求该非均匀量化编码 c，并求其量化噪声 e。

（2）试求对应于该非均匀量化编码的 12 位均匀量化编码 c'。

5.20 将正弦信号 $x(t)=\sin(1\,600\pi t)$ 输入采样频率为 8kHz 采样保持器后通过 13 折线 A 律非均匀量化编码器，设该编码器的输入范围是[−1,1]。试求在一个周期内信号值 $x_i=\sin(0.2i\pi)$，$i=0,1,\cdots,9$ 的非均匀量化编码 c_i，$i=0,1,\cdots,9$。

5.21 将正弦信号 $x(t)=0.25\sin(400\pi t)$ 输入采样频率为 4kHz 采样保持器后通过差分脉冲编码调制器，设该调制器的初始值 $d_{q0}=0$，$\tilde{x}_0=0$。采用码长为 4 的均匀量化编码，量化间隔 Δ=0.031 25。试求在半个周期内信号值 $x_i=0.25\sin(0.1i\pi)$，$i=0,1,\cdots,9$ 的差分脉冲编码 c_i 和量化值 x_i'，$i=0,1,\cdots,9$。

第6章 信道编码定理

内容简介

本章主要介绍信道编码定理以及相关知识。具体包括信道译码的准则、费诺不等式、差错概率上限等基础知识，同时介绍了信道编码定理以及信源、信道联合编码定理。

6.1 基础知识

最大后验概率译码准则是最小平均误码率的最优准则，其次是最大似然概率准则，这些都是常用的译码准则。掌握译码准则和费诺不等式、典型序列的基础概念将有助于对信道编码的理解。

6.1.1 译码准则

本节的内容包括了最大后验概率、最大似然准则、离散信道的译码以及高斯白噪声信道的译码等内容。

1．最大后验概率准则

在数字、数据通信中常用的准则是最小平均误码率准则，平均误码率由（6-1）式决定。

$$p_{\mathrm{e}}=\sum_{\boldsymbol{r}} p(\boldsymbol{r})p(\boldsymbol{e}\,|\,\boldsymbol{r}) \tag{6-1}$$

其中，$p(\boldsymbol{e}\,|\,\boldsymbol{r})=p(\hat{\boldsymbol{c}}\neq\boldsymbol{c}\,|\,\boldsymbol{r})$ 表示接收码字为 $\boldsymbol{r}$ 的时候产生的误码判决；$p(\boldsymbol{r})$ 表示 $\boldsymbol{r}$ 发生的概率；$\hat{\boldsymbol{c}}$ 为译码后的码字；$\boldsymbol{c}$ 为发送的码字。

故可以得到下面的最小平均误码率公式。

$$\begin{aligned}\min p_{\mathrm{e}}&=\min\sum_{\boldsymbol{r}} p(\boldsymbol{r})p(\hat{\boldsymbol{c}}\neq\boldsymbol{c}\,|\,\boldsymbol{r})\\&=\sum_{\boldsymbol{r}} p(\boldsymbol{r})\min p(\hat{\boldsymbol{c}}\neq\boldsymbol{c}\,|\,\boldsymbol{r})\\&=\sum_{\boldsymbol{r}} p(\boldsymbol{r})\min\{1-p(\hat{\boldsymbol{c}}=\boldsymbol{c}\,|\,\boldsymbol{r})\}\\&=\sum_{\boldsymbol{r}} p(\boldsymbol{r})\max p(\hat{\boldsymbol{c}}=\boldsymbol{c}\,|\,\boldsymbol{r})\end{aligned} \tag{6-2}$$

因此从式（6-2）可以看出，最小平均误码等效于最大后验概率。下面就根据一个例子来

设计译码准则。

例 6-1　假设有一个 BSC 信道，如图 6-1 所示，在发送端信息等概发送时，

试求一种译码算法使得平均概率最小。

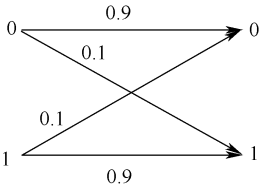


图 6-1　BSC 信道转移概率

解　（1）若收到“0”译作“0”，收到“1”译作“1”，则平均错误概率为

$$P_{\mathrm{E}} = P(0)P_{\mathrm{e}}^{(0)} + P(1)P_{\mathrm{e}}^{(1)} = 0.1$$

（2）若收到“0”译作“1”，收到“1”译作“0”，则平均错误概率为

$$P_{\mathrm{E}} = P(0)P_{\mathrm{e}}^{(0)} + P(1)P_{\mathrm{e}}^{(1)} = 0.9$$

所以方法（1）的平均差错概率明显小于方法（2）。

因此，从例 6-1 看出错误概率与译码准则有关，好的译码准则可以很好地降低平均差错概率。

定义译码准则的相关参数：输入符号集 $A=\{a_i\}$，$1 \leqslant i \leqslant r$；输出符号集 $B=\{b_i\}$，$1 \leqslant j \leqslant s$；译码规则 $F(b_j)=a_i$。

在推导中我们借用下面的信道参数来进行分析，此例中 i=j=3，假设转移矩阵为

$$\boldsymbol{P} = \begin{bmatrix} 0.6 & 0.3 & 0.1 \\ 0.1 & 0.3 & 0.6 \\ 0.3 & 0.4 & 0.3 \end{bmatrix}$$

则可以设计 2 种译码准则。

A 准则：$\begin{cases} F(b_1)=a_1 \\ F(b_2)=a_2 \\ F(b_3)=a_3 \end{cases}$

B 准则：$\begin{cases} F(b_1)=a_1 \\ F(b_2)=a_3 \\ F(b_3)=a_2 \end{cases}$

在确定了译码规则以后，收到 b_j 的情况下，译码的条件正确概率为

$$P(F(b_j)\,|\,b_j) = P(a_i\,|\,b_j) \tag{6-3}$$

而错误译码的概率为收到 b_j 后，推测发出除了 a_i 之外其他符号的概率。

$$P(e\,|\,b_j) = 1 - P(a_i\,|\,b_j) \tag{6-4}$$

可以得到平均错误译码概率为

$$P_{\mathrm{e}} = \sum_{j=1}^{m} p(b_j)P(e\,|\,b_j) = \sum_{j=1}^{s} p(b_j)(1 - P(a_i\,|\,b_j)) \tag{6-5}$$

信道译码的关键问题就是如何选择 $P(a_i\,|\,b_j)$，经过前边的讨论可以看出，为使 $P(e\,|\,b_j)$ 最小，就应选择 $P(F(b_j)\,|\,b_j)$ 为最大，即选择译码函数 $F(b_j)=a^*$ 并使之满足下列条件。

$$P(a^*\,|\,b_j) \geqslant P(a_i\,|\,b_j) \qquad a_i \neq a^* \tag{6-6}$$

即收到一个符号以后译成具有最大后验概率的那个输入符号。这种译码准则称为“最大后验概率准则”或“最小错误概率准则”。

2．最大似然概率准则

由于后验概率一般比较难以计算，因此可以对上节的最大后验概率准则进行进一步的推导，得到最大似然概率准则。

根据 Bayes 公式

$$P(\boldsymbol{c}\,|\,\boldsymbol{r})=\frac{P(\boldsymbol{c})P(\boldsymbol{r}\,|\,\boldsymbol{c})}{P(\boldsymbol{r})} \tag{6-7}$$

我们可以得到

$$\frac{P(b_j\,|\,a^*)P(a^*)}{P(b_j)}\geqslant\frac{P(b_j\,|\,a_i)P(a_i)}{P(b_j)} \tag{6-8}$$

即有 $P(b_j\,|\,a^*)P(a^*)\geqslant P(b_j\,|\,a_i)P(a_i)$。 (6-9)

当信息为等概发送时，即 $P(a_i)=\dfrac{1}{r}$，此时可以得到

$$P(b_j\,|\,a^*)\geqslant P(b_j\,|\,a_i) \tag{6-10}$$

该译码准则称为**最大似然译码准则**，方法是收到一个 b_j 后，在信道矩阵的第 j 列，选择最大的值所对应的输入符号作为译码输出。

根据上面的分析，则当发送码字的先验概率相等的时候，最大似然译码准则等价于最大后验概率，即式（6-10）和式（6-6）是等效的。

3．离散无记忆信道的译码

对于离散无记忆信道（DMC）有

$$\max P(\boldsymbol{r}\,|\,\hat{\boldsymbol{c}}=\boldsymbol{c})=\max\prod_{l=0}^{L-1}P(r_l\,|\,\hat{c}_l=c_l) \tag{6-11}$$

由于对数函数为单调函数，故可将其改写为对数函数形式。

$$\max\log\prod_{l=0}^{L-1}P(r_l\,|\,\hat{c}_l=c_l)=\max\sum_{l=0}^{L-1}\log P(r_l|\hat{c}_l=c_l) \tag{6-12}$$

按上述公式进行译码的算法为最大似然译码算法，同时称公式中的 $\log P(r_l\,|\,\hat{c}_l=c_l)$ 为对数似然函数，有时简称为似然函数。

进一步，对于 BSC 信道（二进制对数信道），有

$$P(\boldsymbol{r}\,|\,\hat{\boldsymbol{c}}=\boldsymbol{c})=p^{d(\boldsymbol{r},\boldsymbol{c})}(1-p)^{n-d(\boldsymbol{r},\boldsymbol{c})} \tag{6-13}$$

似然函数可以进一步改写为

$$\log P(\boldsymbol{r}\,|\,\hat{\boldsymbol{c}}=\boldsymbol{c})=d(\boldsymbol{r},\boldsymbol{c})\log\frac{p}{1-p}+n\log(1-p) \tag{6-14}$$

其中，$d(\boldsymbol{r},\boldsymbol{c})$ 为 $\boldsymbol{r}$ 和 $\boldsymbol{c}$ 之间的汉明距离。

由于对于 BSC 信道而言，一般 $p<\frac{1}{2}$，所以有 $\log\frac{p}{1-p}<0$，$n\log(1-p)$ 为常数，$\boldsymbol{r}$ 和 $\boldsymbol{c}$ 之间的汉明距离由 $d(\boldsymbol{r},\boldsymbol{c})=\sum_{l=0}^{L-1}d(r_l,c_l)$ 式确定，所以

$$\begin{aligned}\max\log P(\boldsymbol{r}\mid\hat{\boldsymbol{c}}=\boldsymbol{c})&=\min d(\boldsymbol{r},\boldsymbol{c})\\&=\min\sum_{l=0}^{L-1}d(r_l,c_l)\end{aligned}$$

因此对于 BSC 信道来说，求最大似然等效于求最小汉明距离。

4．高斯白噪声信道的译码

对于均值为 0，双边功率谱密度 $N_0/2$ 高斯白噪声信道

$$p(y\mid x_{m'})=\prod_{k=1}^{N}\left(\frac{1}{\pi N_0}\right)^{1/2}\exp\left\{-\frac{(y_k-x_{m'k})^2}{N_0}\right\}\tag{6-15}$$

根据式（6-10），所以有

$$\exp\left\{-\frac{1}{N_0}\sum_{k=1}^{N}(y_k-x_{m'k})^2\right\}\geqslant\exp\left\{-\frac{1}{N_0}\sum_{k=1}^{N}(y_k-x_{mk})^2\right\}\quad\text{所有的}m\neq m'\tag{6-16}$$

即为最小欧氏距离译码准则。

$$\sum_{k=1}^{N}(y_k-x_{m'k})^2\leqslant\sum_{k=1}^{N}(y_k-x_{mk})^2\quad\text{所有的}m\neq m'\tag{6-17}$$

对上式进行展开，得到

$$\sum_{k=1}^{N}y_k^2+\sum_{k=1}^{N}x_{mk}^{'2}-2\sum_{k=1}^{N}y_kx_{mk}^{'}\leqslant\sum_{k=1}^{N}y_k^2+\sum_{k=1}^{N}x_{mk}^2-2\sum_{k=1}^{N}y_kx_{mk}\tag{6-18}$$

所以当 $\|\boldsymbol{x_m}\|^2=\sum_{k=1}^{N}x_{mk}^2$ 都相等，即等能量发送时，式（6-18）就变为

$$\sum_{k=1}^{N}y_kx_{mk}^{'}\geqslant\sum_{k=1}^{N}y_kx_{mk}\quad\text{所有的}m\neq m'\tag{6-19}$$

6.1.2　费诺不等式

译码时发生错误是由信道中的噪声引起的，而信道噪声的影响使在接收端收到输出符号 Y 后对发送端发送的符号仍然存在不确定性，因此平均错误概率与信道疑义度存在着一定的关系，这个关系可以用费诺不等式表示。

费诺不等式：平均错误概率 P_{E} 与信道疑义度 $H(X\mid Y)$ 满足以下关系。

$$H(X\mid Y)\leqslant H(P_{\mathrm{E}})+P_{\mathrm{E}}\log(r-1)\tag{6-20}$$

费诺不等式的物理含义是：接收到 Y 后关于 X 的平均不确定性可以分为两部分，第一部分是指接收到 Y 后是否产生错误的不确定性，用 $H(P_{\mathrm{E}})$ 表示；第二部分 $P_{\mathrm{E}}\log(r-1)$ 是当错误 P_{E} 发生后，判断是哪个输入符号造成错误的最大不确定性，是(r−1)个符号不确定性的最大值与 P_{E} 的乘积。

若以 $H(X|Y)$ 为纵坐标，P_E 为横坐标，不等式右半部分的函数曲线如图 6-2 所示。

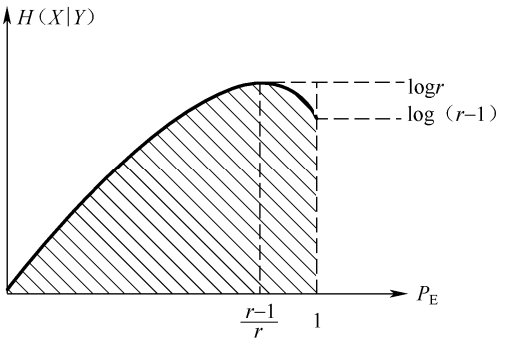


图 6-2　费诺不等式曲线图

6.1.3　ε 典型序列及其性质

1．ε 典型序列

对于离散的无记忆信源有

$$\begin{bmatrix} X \\ p(x) \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & \cdots & x_q \\ p(x_1) & p(x_2) & \cdots & p(x_q) \end{bmatrix}, \sum_{i=1}^{q} p(x_i) = 1 \tag{6-21}$$

则它的 N 次扩展信源为 $X^N = (X_1, X_2, \cdots, X_N)$，且有

$$\begin{bmatrix} X^N \\ p(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & \cdots & x_{q^N} \\ p(\boldsymbol{x}_1) & p(\boldsymbol{x}_2) & \cdots & p(\boldsymbol{x}_{q^N}) \end{bmatrix}, \sum_{j=1}^{q^N} p(\boldsymbol{x}_j) = 1 \tag{6-22}$$

其中，$\boldsymbol{x}_j = (x_{j1}, x_{j2}, \cdots, x_{jN}) \in X^N$，即为（6-21）中 x 变量的 N 次扩展。

渐进等分割性：如果 $X_1, X_2, \cdots, X_N$ 中的 X_i（$1 \leqslant i \leqslant N$）相互统计独立且服从同一分布 $p(x)$，且令 $\boldsymbol{x}_j = (x_{j1}, x_{j2}, \cdots, x_{jN}) \in (X_1, X_2, \cdots, X_N)$，其中$1 \leqslant j \leqslant q^N$、$1 \leqslant j1, j2, \cdots, jN \leqslant q$，则

$-\frac{1}{N}\log p(\boldsymbol{x}_j) = -\frac{1}{N}\log p(x_{j1}, x_{j2}, \cdots, x_{jN})$以概率收敛于$\mathrm{H}(X)$。

渐进等分割性的物理意义是：对于联合概率 $P(X_1 X_2 \cdots X_N)$，如果 $X_1, X_2, \cdots, X_N$ 是独立同分布的随机变量，则当 N 足够大的时候，$-\frac{1}{N}\log p(X_1 X_2 \cdots X_N)$ 接近于信源熵 $H(X)$。

定义：N 长的 $\boldsymbol{x} \in X^N$ 序列，对于任意小的正数 ε，满足

$$\left| -\frac{1}{N}\log p(\boldsymbol{x}) - H(X) \right| < \varepsilon \tag{6-23}$$

或者 $\left|\frac{1}{N} I(\boldsymbol{x}) - H(X)\right| < \varepsilon$，则称为 N 长的 $\boldsymbol{x}$ 序列为 ε **典型序列**，其中 $I(\boldsymbol{x})$ 是 $\boldsymbol{x}$ 的自信息，$H(X)$ 是 X 的信息熵。

反之，如果有 $\left| -\frac{1}{N}\log p(\boldsymbol{x}) - H(X) \right| \geqslant \varepsilon$，则称为 N 长的 $\boldsymbol{x}$ 序列为非$\boldsymbol{\varepsilon}$典型序列。

用 $T_X(N, \varepsilon)$ 表示典型序列的集合，用 $\overline{T}_X(N, \varepsilon)$ 表示非典型序列的集合，如图 6-3 所示，即有

$$T_X(N,\varepsilon)=\left\{\boldsymbol{x}:\left|-\frac{1}{N}\log p(\boldsymbol{x})-H(X)\right|<\varepsilon\right\}$$
$$\overline{T}_X(N,\varepsilon)=\left\{\boldsymbol{x}:\left|-\frac{1}{N}\log p(\boldsymbol{x})-H(X)\right|\geqslant\varepsilon\right\} \tag{6-24}$$

典型序列的意义：N 次扩展信源分为两大类，一类为高概率集合，即经常出现的序列，称作 ε 典型序列，当 $N\to\infty$ 时这类序列出现的概率为 1，且每个序列的概率分布接近于等概分布；另一类为低概率集合，称作非 ε 典型序列，当 $N\to\infty$ 时这类序列出现的概率为 0。

2．联合 ε 典型序列

对于单符号的离散信道输入 x，输出 y，则有：$[X,p(y|x),Y]$，见图 6-4（a），它的 N 次无记忆扩展信道满足 $[X^N,p(\boldsymbol{y}|\boldsymbol{x}),Y^N]$，见图 6-4（b）。其中，

$\boldsymbol{x}=(x_1,x_2,\cdots,x_N)\in X^N,\boldsymbol{y}=(y_1,y_2,\cdots,y_N)\in Y^N$ 。

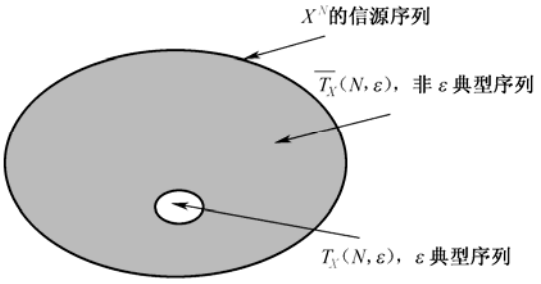


图 6-3　典型序列和非典型序列

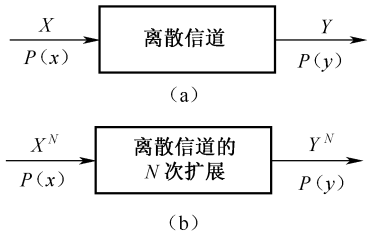


图 6-4　单符号离散信道以及 N 次扩展信道

如果假设 $\boldsymbol{x}$，$\boldsymbol{y}$ 是典型序列，则对任意小的正数 $\boldsymbol{\varepsilon}$ 有

$$\left|-\frac{1}{N}\log p(\boldsymbol{x})-H(X)\right|<\varepsilon$$
$$\left|-\frac{1}{N}\log p(\boldsymbol{y})-H(Y)\right|<\varepsilon \tag{6-25}$$

则 ***xy* 联合典型序列**满足

$$\left|-\frac{1}{N}\log p(\boldsymbol{xy})-H(XY)\right|<\varepsilon \tag{6-26}$$

用 $T_{XY}(N,\varepsilon)$ 表示联合典型序列的集合，则有

$$T_{XY}(N,\varepsilon)=\left\{(\boldsymbol{x},\boldsymbol{y}):\begin{array}{l}\left|-\frac{1}{N}\log p(\boldsymbol{x})-H(X)\right|<\varepsilon\\ \left|-\frac{1}{N}\log p(\boldsymbol{y})-H(Y)\right|<\varepsilon\\ \left|-\frac{1}{N}\log p(\boldsymbol{xy})-H(XY)\right|<\varepsilon\end{array}\right\} \tag{6-27}$$

如果用 $\|T_X(N,\varepsilon)\|$、$\|T_Y(N,\varepsilon)\|$、$\|T_{XY}(N,\varepsilon)\|$ 分别表示集合 $T_X(N,\varepsilon)$、$T_Y(N,\varepsilon)$、$T_{XY}(N,\varepsilon)$ 的大小，对于任意小的正数 η、ε，可以证明它们满足

$$
\begin{aligned}
(1-\eta)2^{n[H(X)-\varepsilon]} &\leqslant \left\|T_X(N,\varepsilon)\right\| \leqslant 2^{n[H(X)+\varepsilon]} \\
(1-\eta)2^{n[H(Y)-\varepsilon]} &\leqslant \left\|T_Y(N,\varepsilon)\right\| \leqslant 2^{n[H(Y)+\varepsilon]} \\
(1-\eta)2^{n[H(XY)-\varepsilon]} &\leqslant \left\|T_{XY}(N,\varepsilon)\right\| \leqslant 2^{n[H(XY)+\varepsilon]}
\end{aligned}
\tag{6-28}
$$

联合典型序列的概念对于信道编码非常重要，可以通过图 6-5 来理解它的物理意义。

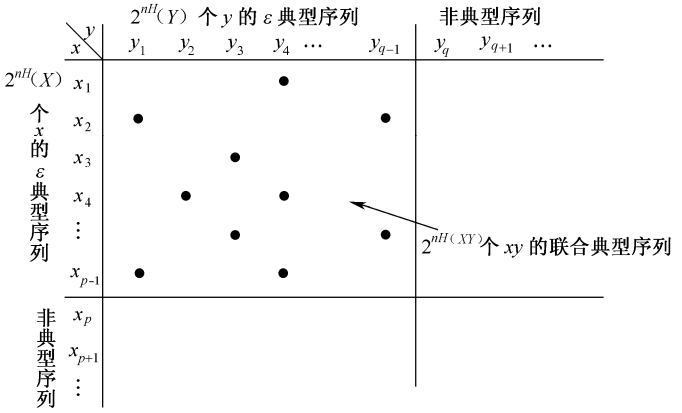


图 6-5　联合典型序列示意图

在对 $\boldsymbol{x}$ 典型序列进行编码的时候希望输出的 $\boldsymbol{y}$ 序列之间没有重合。如果对 $\boldsymbol{y}$ 典型序列的进行不同子集的划分，而且各个子集没有重复，假设不同 $\boldsymbol{x}$ 典型序列传输后能够对应不同的 $\boldsymbol{y}$ 典型序列，根据判决准则就可以完全无误地检测出发送信息。

对于在给定 ε 典型序列 $\boldsymbol{x}$ 的条件下，与 $\boldsymbol{x}$ 构成联合典型序列的 $\boldsymbol{y}$ 典型序列集合的大小满足：$\left\|T_{Y|\boldsymbol{x}}(N,\varepsilon)\right\| \leqslant 2^{n[H(Y|X)+2\varepsilon]}$。因此对于 $\boldsymbol{y}$ 典型序列的进行不同子集的大小应该满足此条件，所以，$\boldsymbol{y}$ 典型序列可以划分的子集合的数目就小于或等于 $2^{nH(Y)}/2^{nH(Y|X)}=2^{n[H(Y)-H(Y|X)]}=2^{nI(X,Y)}$，也就是说当 $\boldsymbol{x}$ 典型序列的数目小于 $2^{nI(X,Y)}$ 的时候，则在输出端的子集合就完全不重合，就可以无误地译出。所以对于 $\boldsymbol{x}$ 典型序列可以完全识别的数目大约为 $2^{nI(X,Y)}$。

6.2　信道编码定理

信道编码定理：如一个离散无记忆信道，信道容量为 C，当信息传输率 $R \leqslant C$ 时，只要码长足够长，总可以在输入符号集中 X^n 找到 $M(M=2^{nR})$ 个码字组成的一组码和相应的译码准则，使信道输出端的平均错误译码概率达到任意小。该定理也称为香农第二编码定理。

信道编码逆定理：如一个离散无记忆信道，信道容量为 C，当信息传输率 $R>C$ 时，则无论码长 n 多长，总找不到一种编码 $(2^{nR},n)$ 使信道输出端的平均错误译码概率达到任意小。

信道编码定理只是一个存在定理，它说明在一定条件下错误概率趋于零的好码是存在的，但是没有说明如何构造这个好码。尽管如此，信道编码（香农第二定理）仍然具有重要的理论意义和实践指导作用，它可以指导各种通信系统的设计，有助于评价各种通信系统及编码效率。

6.3　信源信道联合编码定理

从香农第一编码定理（信源编码定理）和第二定理可以看出，要做到有效、可靠地传输信息，可以考虑将通信系统设计成相对独立的两部分组合，即信源编码和信道编码两部分，如图 6-6 所示。

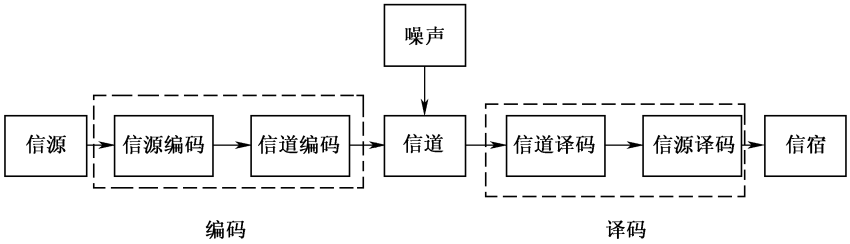


图 6-6　通信系统模型

图 6-6 中首先通过信源编码，用尽可能少的信道符号来表达信源，尽可能减少编码后信源数据的剩余率，然后根据具体的信道，对信源编码后的数据独立进行信道编码，适当增加一些剩余度，使其能纠正和克服信道中引起的错误和干扰。这个思路就是联合编码定理。

信源信道联合编码定理：如果 $X^N=(X_1X_2\cdots X_N)$ 是有限符号集，并且满足渐进等分割行，且信源的极限熵小于信道容量（$H_\infty<C$），则必然存在信源信道编码，使得差错概率任意小。

习　　题

6.1　什么是最大后验概率译码准则和极大似然译码准则？

6.2　说明ε典型序列的物理意义及其性质。

6.3　什么是信道编码定理？

第 7 章 分组码

内容简介

本章主要介绍了分组码的基本概念、主要原理以及应用。内容包括信道编码的基本概念、分类、纠错能力等，具体介绍了汉明码和循环码等常用的线性分组码的编码原理和译码算法，同时对循环码的应用做了分析。

7.1 信道编码的基本概念

通信系统的性能指标涉及有效性、可靠性、适应性、标准性及维护使用等。但若从研究消息的传输来说，通信的有效性与可靠性将是其主要矛盾所在。有效性主要是指消息传输的“速度”，而可靠性主要是指消息传输的“质量”。显然这是一对矛盾，在实际系统中只能根据实际要求取得相对的折中，即在满足一定可靠性指标下，尽量提高消息的传输速度；或者在满足一定传输速度的指标下，尽量提高消息的可靠性。

在通信系统中，一般采用“信源编码”技术来提高数字系统的传输效率。比如语音编码和 MPEG 视频编码技术，通过尽可能地去除信源冗余，可在有限的带宽上提高其传输效率。而通常采用“信道编码”技术，即“差错控制编码”来提高数字系统的可靠传输。

7.1.1 信道编码的作用与分类

1. 信道编码的作用

传输信号在接收端出现误判主要有两方面原因：（1）由于信道传输特性不理想产生的误差，其通常可采用“均衡”的方法加以纠正；（2）由于信号在传输过程叠加的噪声产生的误差，其影响一般需采用“差错控制编码”的方法来加以纠正。

数据信号在信道中传输，所受到的噪声大体分为两类：“随机噪声”和“脉冲噪声”。随机噪声包括热噪声、散弹噪声和传输媒介引起的噪声等，其导致传输中出现“随机差错（又称为独立差错）”，即错码的出现是随机的，且错码之间是统计独立的。存在随机差错的信道称为“无记忆信道”或“随机信道”。脉冲噪声是指突然发生的噪声，包括雷电、开关引起的瞬态电信号变化等，其使传输出现“突发差错”，即错码成串集中出现，也就是说，在一些短促的时间区间内会出现大量的错码，差错之间存在相关性。存在突发差错的信道称为“突发

信道”。此外把既存在随机错码又存在突发错码的信道称为“混合信道”。移动通信中使用的无线信道即为典型的混合信道。对于不同类型的信道，应采用不同类型的差错控制编码技术。比如，一般可采用纠错编码技术对抗“随机差错”，而采用交织编码技术对抗“突发差错”。

2．差错控制方式

常用的差错控制方法有 3 种：检错重发法（简称 ARQ）、前向纠错法（简称 FEC）和混合纠错法（简称 HEC），它们的系统构成如图 7-1 所示。

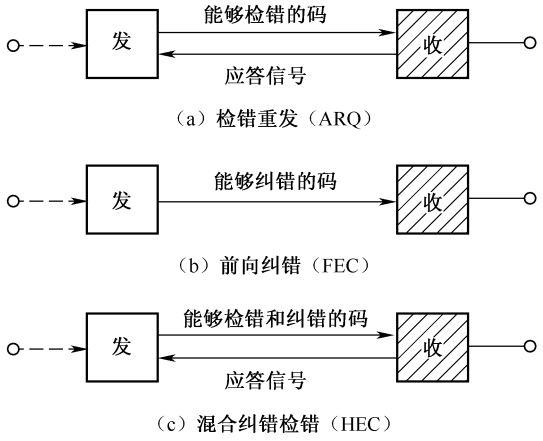


图 7-1　差错控制方式分类

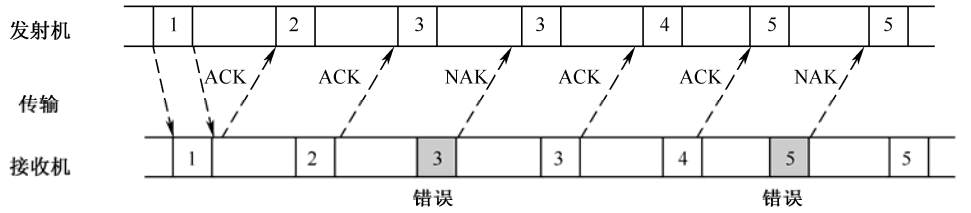
（1）检错重发法。这种差错控制方式也称为“自动请求重传（ARQ）”。在发送端对数据序列进行分组编码，加入一定多余码元使之具有一定的检错能力，成为能够发现错误的码组；接收端收到码组后，按一定规则对其进行有无错误判别，并把判决结果（应答信号）通过反向信道送回发送端。如有错误，发送端把前面发出的信息重新传送一次，直到接收端认为已正确收到信息为止。图 7-2 列举了 3 种比较流行的 ARQ 过程，图中时间从左到右递进。

第一个过程称为“停发-等待（stop-and-wait）”ARQ，如图 7-2（a）所示。发送端送出一个码组，接收端收到后经检验若未发现错误，则发回一个认可信号（Acknowledgment，ACK）给发送端，发送端收到 ACK 信号后再发出下一个码组；如果接收端检测到错误，则会发回一个否认信号（NAK），发送端收到 NAK 信号后重发前一个码组，并再次等候 ACK 或 NAK 信号。这种工作方式只需要半双工连接，在发送两个码组之间有停顿时间，使传输效率受到影响，但由于其工作原理简单，在计算机数据通信中仍得到了应用。

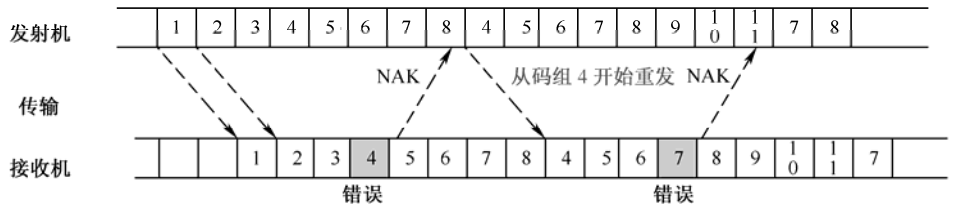
第二个过程称为“具有回拉功能的连续 ARQ”（continuous ARQ with pullback），如图 7-2（b）所示。这里需要一个全双工连接，两个终端同时传输，发射机传送信息数据，接收机传送确认信号。注意，每个分组指定一个连续数字，并且 ACK 和 NAK 需要附带这些数字或预知传输时延，使发送机知道哪个信息与哪个确认信号相联系。在图中，发送信息和确认信号之间有一个固定的 4 个分组距离。例如，当发送消息 8 时，同时接收到错误的消息 4 相对应的 NAK 信号，此时，发射机“拉回”到错误消息处，从错误消息起重发所有的信息数据。这种“具有回拉功能的连续 ARQ”系统比“停发-等待重发”系统有很大改进，其在许多数

据传输系统中得到应用。

第三个过程称为“具有选择性重发功能的连续 ARQ”（continuous ARQ with selective repeat），如图 7-2（c）所示。它与第二个过程一样也需要全双工连接。但在这个过程中，只有错误的消息重发，然后发射机从先前停止的地方继续发送原序列而不重发已经正确接收的消息。显然，“具有选择性重发功能的连续 ARQ”系统的传输效率最高，但它的成本也最昂贵，因为它要求较为复杂的控制，在发送和接收端都要求有数据缓存器。



（a）停止－等待 ARQ（半双工）



（b）具有回拉功能的连续 ARQ（全双工）

（c）具有选择性重发功能的连续 ARQ（全双工）

图 7-2 自动请求重传（ARQ）

根据不同的思路，ARQ 还可以有其他的工作形式，如混合发送形式，它是将等候发送与连续发送结合起来的一种形式。发送端连续发送多个码组以后，再等待收端的应答信号，以决定是重发还是发送新的码组。在国际标准化组织（ISO）建议的高级数据链路控制规程（HDLC）和 CCITT X.25 建议中就推荐采用这一工作形式。

（2）前向纠错法。前向纠错（FEC）系统中，发送端的信道编码器将输入数据序列变换成能够纠正错误的码，接收端的译码器根据编码规律检验出错误的位置并自动纠正。这种纠错方式不需要反向信道（传递重发指令），特别适合于只能提供单向信道的场合。由于能自动纠错，不要求检错重发，因而时延小、实时性好。值得注意的是，所选择的纠错码必须与信道的错误特性密切配合，否则很难达到降低错码率的要求；当需要纠正较多的错码时，就要求附加较多的监督码元，导致译码设备复杂，传输效率低下。

与前向纠错相比，ARQ 的主要优点有：（a）只需要少量的冗余（一般为总码元的 5%～20%）就可获得极低的输出误码率；（b）检错译码器比纠错设备简单得多，降低了成本及复杂度。

但若在以下几种情况下，则需要用 FEC 代替 ARQ 或与检错相结合使用：（a）没有可用的反向信道，比如单向传输系统或广播系统中；（b）当信道干扰增大时，整个系统可能处在重发循环中，导致通信效率降低，甚至不能通信；（c）要求系统中的业务严格实时传输时，ARQ 很难保证系统的实时性。

（3）混合纠错法。混合纠错法（HEC）是前向纠错（FEC）和检错重发（ARQ）方式的结合。在这种系统中，接收端不但有纠正错误的能力，而且对超出纠错能力的错误有检测能力。当遇到后一种情况时，通过反向信道要求发送端重发一遍。混合纠错方式在实时性和译码复杂度方面是前向纠错和检错重发方式的折衷。

3．差错控制编码的分类

从不同的角度出发，差错控制编码可有不同的分类方法。

按照码组的不同功能，可以将其分为**检错码**、**纠错码**和**纠删码**。检错码仅能够检测错误；纠错码仅可纠正误码；纠删码则兼有检错和纠错能力，当在纠错范围内时纠正错码，当发现不可纠错码时，发出错误指示或者简单地删除此错误信息码段。

按照信息码元与附加的监督码元之间的检验关系，可以分为**线性码**和**非线性码**。线性码是指监督码元与信息码元之间的检验关系满足一组线性方程式（呈线性关系）。反之，非线性码是指它们二者呈非线性关系。

按照信息码元在编码后是否保持原来的形式不变，可分为**系统码**和**非系统码**。在系统码中，编码后的信息码元保持原样不变；而非系统码中信息码元则改变了原来的信号形式。由于系统码的编码和译码相对比较简单些，所以得到广泛应用。

按照对信息码元处理方法的不同，可分为**分组码**和**卷积码**。在分组码中，首先将 k 个信息码元划分为一组，然后由这 k 个码元按照一定的规则产生 r 个监督码元，从而组成长度为 $n=k+r$ 的码组。在分组码中，监督码元仅与本码组的信息码元有关，而与其他码组的信息码元无关。卷积码则不同，虽然编码后的序列也划分为码组，但每组的监督码元不但与本码组的信息码元有关，而且还与前面若干组信息码元有关，因此卷积码有时也称为连环码。

按照构造差错控制编码的数学方法，可以分为**代数码**、**几何码**和**算术码**。代数码建立在近代代数学基础上，是目前发展最为完善的编码。其中线性码是代数码的一个最重要的分支。

本小节先简单介绍几种实用的简单检错码，然后在下一节主要讨论纠正随机差错的二进制线性分组码。

4．几种实用的简单检错码

（1）奇偶监督码。奇偶监督码又称为奇偶校验码，是一种最简单的检错码，被广泛用于以随机错误为主的计算机通信系统中。其编码规则是：将所要传输的数据码元分成组，然后在每组数据后附加一位监督位，使得该组码元连同监督位在内的码组中的“1”的个数为偶数（称为偶校验）或者为奇数（称为奇检验）；在接收端按同样的规律检查，如发现不符就说明产生了差错，但它不能确定差错的具体位置，即不能纠错。

奇偶监督码的监督关系可用公式来表示。设码组长度为 n，记为 a_{n-1}，a_{n-2}，…，a_1，a_0，其中前 $n-1$ 位为信息码元，第 n 位为监督位（即 a_0 为监督码元），则有

$$\begin{cases} a_0 \oplus a_1 \oplus \cdots \oplus a_{n-1} = 0 & \text{（偶校验）} \\ a_0 \oplus a_1 \oplus \cdots \oplus a_{n-1} = 1 & \text{（奇校验）} \end{cases} \tag{7-1}$$

监督码元 a_0 可以表示为

$$\begin{cases} a_0 = a_1 \oplus a_2 \cdots \oplus a_{n-1} & \text{（偶校验）} \\ a_0 = a_1 \oplus \cdots \oplus a_{n-1} \oplus 1 & \text{（奇校验）} \end{cases} \tag{7-2}$$

可以看出，这种奇偶校验只能发现单个或奇数个错误，而不能检测出偶数个错误，因此它的检错能力不高。

（2）水平奇偶监督码。针对上述奇偶监督码检错能力不高，尤其是不能检测突发错误的缺点，提出水平奇偶监督码。即将经过奇偶监督编码的码元序列按行排成方阵，每行为一组奇偶监督编码（如表 7-1 所示），但发送时则按列的顺序传输：111011110011…。接收端仍将码元排成发送时的方阵形式，然后按行进行奇偶校验。

表 7-1　　水平奇偶监督码

信息码元	监督码元
1 1 1 0 0 1 1 0 0 0	1
1 1 0 1 0 0 1 1 0 1	0
1 0 0 0 0 1 1 1 0 1	1
0 0 0 1 0 0 0 0 1 0	0
1 1 0 0 1 1 1 0 1 1	1
1 1 1 0 1 1 0 0 0 0	1

可以看出，由于发端是按列发送码元而不是按码组发送码元，因此把本来可能集中在某一个码组内的突发错误分散在方阵的各个码组上，因而可得到整个方阵的行监督。采取这种编码方法可以发现某一行上所有奇数个错误以及所有长度不大于方阵中行数（表 7-1 中为 6）的突发错误。这种编码的代价是信息的延时随着码长的增加而增加。

（3）水平垂直奇偶监督码。水平垂直奇偶监督码是将水平奇偶监督码推广到二维，又称行列监督码和方阵码。它的方法是在水平奇偶监督码的基础上，对表 7-1 所示方阵中每一列再进行奇偶校验，就可得表 7-2 所示的方阵。发送时按列序顺次传输：1110111110011010000010…1010110。

表 7-2　　水平垂直奇偶监督码

	信息码元	监督码元
	1 1 1 0 0 1 1 0 0 0	1
	1 1 0 1 0 0 1 1 0 1	0
	1 0 0 0 0 1 1 1 0 1	1
	0 0 0 1 0 0 0 0 1 0	0
	1 1 0 0 1 1 1 0 1 1	1
	1 1 1 0 1 1 0 0 0 0	1
监督码元	1 0 0 0 0 0 0 0 0 1	0

显然，这种码比水平奇偶监督码有更强的检错能力，它能发现某一行或某一列上的奇数个错误和长度不大于行数（或列数）的突发错误。这种码还有可能检测出偶数个错码，因为如果每行的监督位不能在本行检出偶数个错误时，则在列的方向上有可能检出。当然，当偶

数个错误恰好分布在矩阵的四个顶点上时，这样的偶数错误是检测不出来的。

（4）恒比码。恒比码是从某确定码长的码组中挑选那些“1”和“0”的比例为恒定值的码组作为许用码组。接收时只要计算每个码组中“1”的数目是否对，即可判断是否有错误发生。

我国电传机传输汉字电码时，每个汉字用 4 位阿拉伯数字表示，而每个阿拉伯数字又用 5 位二进制符号构成的码组表示。每个码组的长度为 5，其中恒有 3 个“1”，称为“5 中取 3”恒比码。其可能编成的码组数目等于从 5 中取 3 的组合数，即共有 $C_5^3 = 5!/(3!2!) = 10$，即 10 个许用码组，这 10 个许用码组正好代表 10 个阿拉伯数字，如表 7-3 所示。

表 7-3　　我国五单位保护电码表

数字	电码	数字	电码
0	0 1 1 0 1	5	0 0 1 1 1
1	0 1 0 1 1	6	1 0 1 0 1
2	1 1 0 0 1	7	1 1 1 0 0
3	1 0 1 1 0	8	0 1 1 1 0
4	1 1 0 1 0	9	1 0 0 1 1

恒比码还应用于国际无线电报通信中，广泛采用的是“7 中取 3”恒比码，即它采用三个“1”，四个“0”的恒比码，共有 $C_7^4 = 7!/(3!4!) = 35$ 个许用码组，分别代表 26 个英文字母及其他符号。

恒比码除了能检测出奇数个错误外，还能部分检测出偶数个错误，但不能检测出全部的偶数错误（如成对交换错误）。恒比码简单，适用于来传输电传机或其他键盘设备产生的字母和符号，但对于信源来的二进随机数字序列，恒比码就不宜使用了。

（5）重复码。在发送端把消息重复多遍发送，在接收端接收时使用合适的译码方式降低错误发生概率，以提高通信的可靠性。比如在二元对称信道中，当发送符号 0 时，不是只发一个 0 而是连续发三个 0（即用“000”表示信息 0）；同样，发送符号 1 时也连续发送三个 1（即用“111”表示信息 1）。于是信道输入端有两个码字 000 和 111。

在输出端，由于信道干扰的作用，各个码元都可能发生错误，因此共有 8 个可能的输出序列。这种信道可以看成是三次无记忆扩展信道。其输入是在 8 个可能出现的二元序列中选两个作为码字，其余为禁用码字，而输出端为 8 个可能的输出符号，如表 7-4 所示。

表 7-4　　三位重复码的构成

禁用码字	发送码字		接收到码字
	a1=000	信道	a1=000
a2=001			a2=001
禁用码字	发送码字		接收到码字
a3=010			a3=010
a4=011			a4=011
a5=100		信道	a5=100
a6=101			a6=101
a7=110			a7=110
	a8=111		a8=111

若重复更多次，$n=5$，7，… 可以进一步降低错误概率。对于 n 重复码，当 n 很大时，可以使 P_e 降低到很小的值，同时信息传输率就会降低很多。

n 重复码：码率为 $1/n$，仅有两个码字 C_0 和 C_1，传送 1 个比特（$k=1$）的消息；C_0=（00…0）表示信息“0”，C_1=（11…1）表示信息“1”。n 重复码可以检测出任意小于 $n/2$ 个差错的错误图样。

7.1.2 纠错与检错原理

“差错控制编码”的基本思想是在发送端被传送的信息中附加一些冗余比特（称为监督码元），这些多余的码元与信息码元之间以某种确定的规则相互关联（约束）；接收端通过检验这种既定的规则来获知是否有错码产生，进而纠正这些错码。不同的编码方法，有不同的检纠错能力。一般而言，检（纠）错能力与所付出的代价成正比。

考虑三位二进制码元，它们共可组成 8 种可能的码组：000、001、010、011、100、101、110、111。如果这 8 个码组都用来传送信息，那么在传送过程中一旦发生一个误码，就会由一个码组变成另外一个码组。由于没有多余的信息量可供利用，所以接收端不可能发现错误。因此，这种编码无检错和纠错能力。然而如果只选择 000、011、101 和 110 这 4 种码组（称为许用码组）来传送信息，则相当于实际传送的信息码组为 00、01、10 和 11 4 种信息，而其第三位可视为附加的监督码元。监督码元和前面两位信息比特一起，保证码组中“1”码的个数为偶数。显然，8 个码组中除了以上 4 个许用码组满足这一检验关系外，其他 4 个码组（称为禁用码组）都不满足这种校验关系。在传输过程中，若出现 1 的位数为奇数，则接收端就会收到禁用码组，从而表明传输过程中发生了错误。用这种简单的校验关系可以发现一个和三个错误，但不能纠正错误。如果我们进一步将许用码组限制为两种：000 和 111，用 000 表示 0，111 表示 1，那么根据多数判决法，即收到的 3 位比特如果 0 的个数多于 1 的个数判决为 0；1 的个数多于 0 的个数则判决为 1，因此此码可以纠正一位错码，代价是效率的降低，用 3 比特传输 1 个比特的信息，效率为 1/3。

由此可见，采用差错控制编码来提高数据通信系统的可靠性是以牺牲有效性为代价来换取的。根据 Shannon 有噪编码定理可知，对于有噪声信道，当码长足够长的情况下，存在可以实现可靠通信的信道编码，在传输速率 $R<C$（C 为信道容量）时达到所期望的任意小的差错概率。若 $R>C$，则无论使用什么样的信道编码，都不可能使差错概率趋于零。检错和纠错技术的基本目的是通过在数据中引入冗余，以改善信道的性能。增加的冗余比特虽然会增加信源数据速率所要求的带宽，减小在高信噪比条件下的带宽利用率，但却能够明显地改善低信噪比情况下的误码率。

众所周知，通过增加解调器输入端的信噪比（SNR）也可获得误码率的降低；而采用“差错控制编码”的另一好处是在编码系统中能够以更低的信噪比需求，获得无编码系统中同样的误码率指标。这可以降低系统中的功率预算，带来许多潜在的系统好处。可以采用“编码增益（Coding Gain）”来描述这种优势。所谓“编码增益”定义为：在给定误码率（BER）和同样数据速率的情况下，与未编码系统相比较，编码系统所需的单位比特能量与噪声功率谱密度之比（E_b/N_o）的降低量，见图 7-3。由定义可见，编码增益所给出的量值，实际上表明了一个未编码系统要得到编码系统中同样的解码误码率所需要提供的附加（E_b/N_o）的提高量。

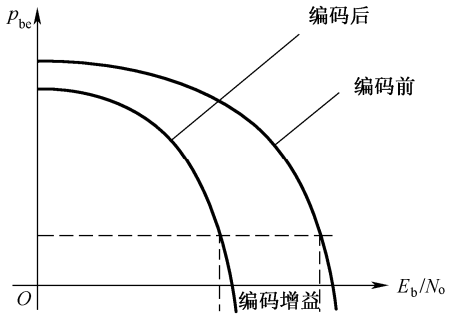


图 7-3 编码增益示意图

7.1.3 纠错与检错能力

假定 C_i 和 C_j 是一 (n,k) 分组码中的任意两个码组，则有如下定义。

① **编码效率 *R***（简称为“码率”）：$R=k/n$。

② **码重**：在信道编码中，码组中非零码元的个数。对于二进制码，码重就是码组中 1 的个数；例如：“010”的码重为 1，“011”的码重为 2。

③ **码距**：两个码组 C_i 和 C_j 中对应码位上具有不同码元的位数，记为 $d(C_i,C_j)$；若采用二进制码，则码距称为“汉明（Hamming）距”。

④ **最小距离 $d_{\min}$**：在一 (n,k) 分组码中，任意两个许用码组距离的最小值，即码组集合中任意两元素间的最小距离。

码距和检纠错能力密切相关。为了说明码距与检错和纠错能力之间的关系，把 3 位二进制码构成的 8 个码组用一个三维立方体来表示，如图 7-4 所示。图中立方体的各顶点分别代表 8 个码组，每一码组的 3 个码元的值 (a_2,a_1,a_0) 就是此立方体各顶点的坐标。由图 7-4 可见，码距 $d(C_i, C_j)$实质上是从 C_i 顶点沿立方体各边到达 C_j 顶点所经过的最少边数。如果 8 种码组都作为许用码组时（无任何检纠错能力），任两码组间最小距离为 1，称这种编码的最小码距为 1，记 $d_{\min}=1$；如果只选 4 种码组为许用码组时（可检出一位错误），则最小码距 $d_{\min}=2$；若只选用两种码组为许用码组时（可纠正一位错误），则最小码距为 $d_{\min}=3$。

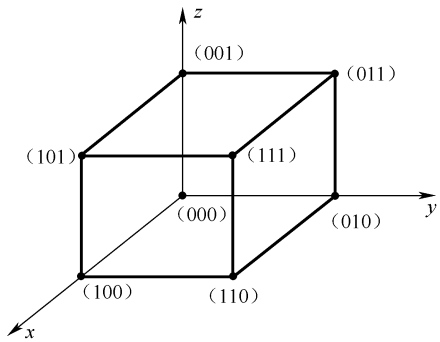


图 7-4 码距的几何解释

由上可见，一种编码的最小码距直接关系到这种码的检错和纠错能力，下面具体说明。

① 在一个码组内为了检测 e 个错码，则要求最小码距

$$d_{\min} \geqslant e+1 \tag{7-3}$$

② 在一个码组内为了纠正 t 个错码，则要求最小码距为

$$d_{\min} \geqslant 2t+1 \tag{7-4}$$

③ 在一个码组内为了纠正 t 个错码，同时检测 e 个 $(e>t)$ 个错码，要求最小码距

$$d_{\min} \geqslant e+t+1 \tag{7-5}$$

注意，这里所述能纠正 t 个错码，同时能检测 e 个错码的含义，是指当错码不超过 t 个时，错码能自动予以纠正；而当错码超过 t 个时，则不可能纠正错误，但仍可检测 e 个错码；这正是前述混合检纠错的控制方式。

上述结论可用图 7-5 所示的几何图形加以简单证明。图 7-5 中 C_1 和 C_2 分别表示 (n,k) 分组码中的任意两个码组。对于结论①，由图（a）可见，若码组 C_1 发生不超过 e 个错码时，该码组的位置移动将不超出以 C_1 为圆心、以 e 为半径的圆。只要其最小码距满足 $d_{\min} \geqslant e+1$，则其他任一许用码组都不会落入此圆内，即 C_1 码组发生 e 个错码时就不可能与其他的许用码组相混淆。换句话说，其他许用码组必须位于以 C_1 为圆心、以 $e+1$ 为半径的圆上或圆外。

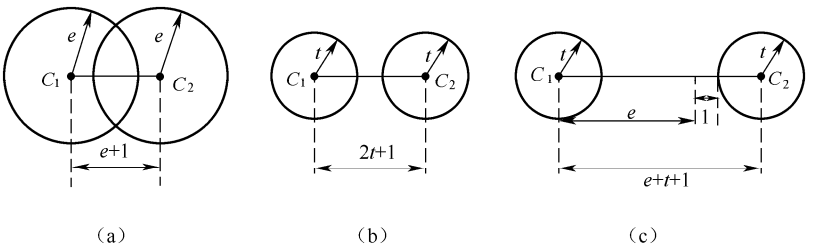


图 7-5 码距与纠检错能力的关系

对于结论②，由图（b）可见，当许用码组 C_1 和 C_2 的各自误码不超过 t 个时，发生错码后两个许用码组的位置移动将分别不会超出以 C_1 和 C_2 为圆心、以 t 为半径的圆。只要这两个圆不相交，则当错码小于 t 个时，可根据它们落在哪个圆内就能正确判决。而以 C_1 和 C_2 为圆心的两个圆不相交的最近圆心距离为 $2t+1$，这就是纠正 t 个错误的最小码距。

对于结论③，可以用图（c）来证明。在最不利情况下，C_1 发生 e 个错码而 C_2 发生 t 个错码。为了保证两码组仍不发生混淆，则要求以 C_1 为圆心、e 为半径的圆必须与以 C_2 为圆心、t 为半径的圆不发生交叠，即要求最小码距 $d_{\min} \geqslant e+t+1$。同时，还可看到若错码超过 t 个，两圆有可能相交，因而不再有纠错的能力，但仍可检测 e 个错码。

现在简要分析采用差错控制编码的效用。假设在随机信道中发送“0”和“1”时的错误概率相等，均为 p，且 $p \ll 1$，则容易证明，在码长为 n 的码组中恰好发生 r 个错误的概率为

$$P_n(r) = C_n^r \cdot p^r \cdot (1-p)^{n-r} \approx \frac{n!}{r!(n-r)!} \cdot p^r \tag{7-6}$$

当码长 $n=7$，$p=10^{-3}$ 时，则有

$$P_7(1) \approx 7p = 7\times 10^{-3}$$

$$P_7(2) \approx 21p^2 = 2.1\times 10^{-5}$$

$$P_7(3) \approx 35p^3 = 3.5\times 10^{-8}$$

可见，采用差错控制编码后，即使只能纠正（或检测）1～2 个错误，也可以使误码率下降几个数量级。这就表明，即使是简单的差错控制编码也具有较大的实用价值。当然，如在

突发信道中传输，由于错码是成串集中出现的，所以上述只能纠正码组中 1～2 个错码的编码，其效用就不像在随机信道中那样明显了，需要采用更为有效的纠错编码方式。

7.2 线性分组码的基本数学理论

本节介绍线性空间、生产矩阵、校验矩阵等分组码的基本数学理论。

7.2.1 线性空间及其性质

线性空间的定义：设 V 是一个非空集合，P 是一个数域，在集合 V 中定义了一种代数运算，叫做加法，即对在 V 中都存在唯一的一个元素 λ，称 λ 为 α 与 β 的和，记为 $\lambda=\alpha+\beta$；在 P 与 V 的元素之间还定义了一种运算，叫做数量乘法，即 $\forall\alpha\in V$，$\forall k\in P$，在 V 中都存在唯一的一个元素 δ 与它们对应，称 δ 为 k 与 α 的数量乘积，记为 $\delta=k\alpha$。如果加法和数量乘法还满足下述规则，则称 V 为数域 P 上的线性空间（$\forall\alpha,\beta,\gamma\in V$）。

对于加法满足
$$\begin{cases}\alpha+\beta=\beta+\alpha\\(\alpha+\beta)+\gamma=\alpha+(\beta+\gamma)\\在V中有一个元素0，对\forall\alpha\in V，有\ \alpha+0=\alpha\\\forall\alpha\in V，都有V中的一个元素\beta，使得\alpha+\beta=0\end{cases}$$

其中 0 称作 0 元素，β 称作 α 的负元素。

对于乘法满足
$$\begin{cases}1\alpha=\alpha\\k(l\alpha)=(kl)\alpha\end{cases}$$

且对于乘和加法满足
$$\begin{cases}(k+l)\alpha=k\alpha+l\alpha\\k(\alpha+\beta)=k\alpha+k\beta\end{cases}$$

线性空间具有下面的性质。

① 零元素是唯一的。

② 负元素是唯一的，$\forall\alpha\in V$，唯一。

③ 关于 0 元素有，$0\alpha=0$，$k0=0$，$(-1)\alpha=-\alpha,k(\alpha-\beta)=k\alpha-k\beta$。

④ 如果 $k\alpha=0$，则有 k=0 或者 α=0。

7.2.2 生成矩阵

分组码是前向纠错（FEC）码，它可以在无需重新发射的情况下检测出有限个错码，并加以纠正，借此来改善通信系统的性能。在分组编码器中，k 个信息比特被编成 n 个比特，从而增加了 $n-k$ 个冗余比特（监督比特），用来检测和纠正错误。这种将信息进行分组，为每组信息码附加若干监督码的编码方式，称为分组码。在分组码中，监督码元仅由本码组中的信息码元确定。分组码一般用符号 (n,k) 表示，其中 k 是每组二进制信息码元的数目，n 是编码码组的总位数，也称为码组长度（码长）。$r=n-k$ 为每码组中的监督码元的数目。

线性分组码定义：(n,k)线性分组码是 GF(q)域上的 n 维线性空间中的一个 k 维子空间。编码的规则仅局限于本码组之内，本码组的监督元仅和本码组的信息元相关。信息码组由 k 个二进制码元组成，共有 2^k 个不同的信息码组；附加 $n-k$ 个码元，每个监督元

取值与该信息码组的 k 个码元有关；编码器输出长度 n；这 2^k 个码字的集合称为(n, k)分组码。

由于(n, k)线性分组码是一个 k 维线性空间。因此必可找到 k 个线性无关的矢量，能张成该线性空间。设 $\boldsymbol{C}_1, \boldsymbol{C}_2, \cdots, \boldsymbol{C}_k$ 是 k 个线性无关的矢量，则对任意 $\boldsymbol{C}$

$$\begin{aligned}\boldsymbol{C} &= m_1\boldsymbol{C}_1 + m_2\boldsymbol{C}_2 + \cdots + m_k\boldsymbol{C}_k \\ &= (m_1, m_2, \cdots, m_k)\begin{pmatrix}\boldsymbol{C}_1 \\ \boldsymbol{C}_2 \\ \vdots \\ \boldsymbol{C}_k\end{pmatrix} \\ &= \boldsymbol{m}\boldsymbol{G}\end{aligned} \tag{7-7}$$

$\boldsymbol{G}$ 称为该分组码的生成矩阵。

当生成矩阵 $\boldsymbol{G}$ 确定之后，(n, k)线性系统码也就完全被确定了，只要找到码的生成矩阵，编码问题也同样被解决了。

例 7-1 一个(7,3)码，$m_2\, m_1\, m_0 \rightarrow c_6\, c_5\, c_4\, c_3\, c_2\, c_1\, c_0$，如果码字的生成规则为

$$\begin{cases} c_6 = m_2 + m_0 \\ c_5 = m_2 + m_1 + m_0 \\ c_4 = m_2 + m_1 \\ c_3 = m_1 + m_0 \\ c_2 = m_2 \\ c_1 = m_1 \\ c_0 = m_0 \end{cases}$$

若用矩阵形式表示这些线性方程组，则

$$\boldsymbol{C} = [m_2 \quad m_1 \quad m_0]\begin{bmatrix}1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1\end{bmatrix}$$

故该码的生成矩阵为

$$\boldsymbol{G} = \begin{bmatrix}1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1\end{bmatrix}$$

生成矩阵具有下面 3 个**性质**。

① 生成矩阵 $\boldsymbol{G}$ 中的每一行都是一个码字。

② 任意 k 个线性独立的码字都可以作为生成矩阵。

③ 给定一个(n, k)线性分组码，其生成矩阵可有多个。

7.2.3 校验矩阵

分组码的结构如图 7-6 所示，其中 $n-k$ 个校验位可用 k 个已知的信息位表示出来，具体如式（7-8）。

$$\begin{cases} c_{n-k-1}=h_{n-k-1,n-1}\,c_{n-1}+h_{n-k-1,n-2}\,c_{n-2}+\cdots+h_{n-k-1,n-k}\,c_{n-k} \\ c_{n-k-2}=h_{n-k-2,n-1}\,c_{n-1}+h_{n-k-2,n-2}\,c_{n-2}+\cdots+h_{n-k-2,n-k}\,c_{n-k} \\ \cdots \\ c_0=h_{0,n-1}\,c_{n-1}+h_{0,n-2}\,c_{n-2}+\cdots+h_{0,n-k}\,c_{n-k} \end{cases} \tag{7-8}$$

上式可以改写为式（7-9）

$$\underbrace{\begin{pmatrix} h_{n-k-1,n-1} & \cdots & h_{n-k-1,n-k} & 1 & 0 & 0\cdots \\ h_{n-k-2,n-1} & \cdots & h_{n-k-2,n-k} & 0 & 1 & 0\cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ h_{0,n-1} & \cdots & h_{01,n-k} & 0 & 0\cdots & 1 \end{pmatrix}}_{\text{校验矩阵}(n-k\text{行，}n\text{列})} \begin{pmatrix} c_{n-1} \\ c_{n-2} \\ \vdots \\ c_0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{7-9}$$

由上式可知**校验矩阵** $\boldsymbol{H}$ 与任意一个码字之积为 0，因此有

$$\boldsymbol{H}\,\boldsymbol{G}^{\mathrm{T}}=\boldsymbol{0} \tag{7-10}$$

7.2.4 对偶码和系统码

系统码和对偶码是分组码中的常用码字，本小节就进行主要分析。

1. 系统码

用标准生成矩阵 $\boldsymbol{G}_{k\times n}$ 编成的码字，前面 k 位为信息数字，后面 $r=n-k$ 位为校验字，这种信息位在前、校验位在后的线性分组码称为线性系统分组码，如图 7-6 所示，前面 k 位 $(a_{n-1}\cdots a_r)$ 为信息位，后面附加 r 个监督位 $(a_{r-1}\cdots a_0)$。

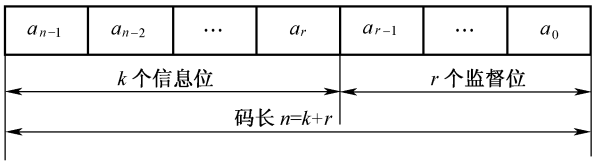


图 7-6 分组码中系统码的结构

系统码由于前面是 k 个信息位，后面是 r 位监督位，因此它的生成矩阵就有一定的特殊性和规律性，可以写为

$$\boldsymbol{G}_{k\times n}=\begin{bmatrix}\boldsymbol{I}_k & \boldsymbol{Q}_{k\times r}\end{bmatrix} \tag{7-11}$$

其中，$\boldsymbol{G}_{k\times n}$ 是 $k\times n$ 矩阵，子矩阵 $\boldsymbol{I}_k$ 是 $k\times k$ 单位阵。

同样，可以写出系统码的校验矩阵

$$\boldsymbol{H}_{r\times n}=\begin{bmatrix}\boldsymbol{Q}^{\mathrm{T}}{}_{r\times k} & \boldsymbol{I}_r\end{bmatrix} \tag{7-12}$$

根据式（7-11）和式（7-12），对于系统码，如果知道生成矩阵就可以非常容易地写出相应的校验矩阵，反之亦然。

2. 对偶码

对一个(n, k)线性码 C，由于 $\boldsymbol{H}_{r\times n}\boldsymbol{G}^{\mathrm{T}}_{n\times k}=\boldsymbol{0}_{r\times k}$，则必然有 $(\boldsymbol{H}_{r\times n}\boldsymbol{G}^{\mathrm{T}}_{n\times k})=\boldsymbol{G}_{k\times n}\boldsymbol{H}^{\mathrm{T}}_{n\times r}=\boldsymbol{0}^{\mathrm{T}}_{r\times k}$。如果

将该公式看做一个新的编码方式，仍然满足分组码的校验矩阵和生成矩阵的关系，所以此时以 $\boldsymbol{G}$ 作监督矩阵，而以 $\boldsymbol{H}$ 作生成矩阵，可构造另一个码 C_{d}，码 C_{d} 是一个$(n, n-k)$线性码，称码 C_{d} 为原码 C 的对偶码。

例 7-2 (7, 4)线性码的对偶码是(7, 3)码。

假设(7, 4)线性系统码的生成矩阵

$$\boldsymbol{G}_{(7,4)}=\begin{bmatrix}1&0&0&0&1&0&1\\0&1&0&0&1&1&1\\0&0&1&0&1&1&0\\0&0&0&1&0&1&1\end{bmatrix}$$

可以写出它的校验矩阵为

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

其对偶码的生成矩阵和检验矩阵分别为

$$\boldsymbol{G}_d=\boldsymbol{H}=\begin{bmatrix}1&1&1&0&1&0&0\\0&1&1&1&0&1&0\\1&1&0&1&0&0&1\end{bmatrix}$$

$$\boldsymbol{H}_d=\boldsymbol{G}=\begin{bmatrix}1&0&0&0&1&0&1\\0&1&0&0&1&1&1\\0&0&1&0&1&1&0\\0&0&0&1&0&1&1\end{bmatrix}$$

即对偶(7, 3)码的生成矩阵和检验矩阵分别是原来(7, 4)码校验矩阵和生成矩阵。

7.2.5 差错图样

发送码组 $\boldsymbol{c}$ 在传输过程中由于噪声、干扰等影响有可能发生误码。设接收到的码组为

$$\boldsymbol{r}=\left[b_{n-1}b_{n-2}\cdots b_0\right]$$

则发送码组与接收码组之差为

$$\boldsymbol{r}-\boldsymbol{c}=\boldsymbol{e}\ (\text{模 }2) \tag{7-13}$$

也可以写为 $\boldsymbol{r}+\boldsymbol{c}=\boldsymbol{e}$ (模 2)，因此在 2 进制中，加法和减法是等效的。

$\boldsymbol{e}$ 称为差错图样或者错误图样，这里

$$\boldsymbol{e}=\left[e_{n-1}e_{n-2}\cdots e_0\right] \tag{7-14}$$

$$e_i=\begin{cases}0,\text{当}b_i=a_i\\1,\text{当}b_i\neq a_i\end{cases}\qquad 0\leqslant i\leqslant n-1$$

因此，若 $e_i=0$，表示第 i 位接收码元无错；若 $e_i=1$，表示第 i 位接收码元有错。差错图样在分组码的译码中具有重要的作用。

7.3 线性分组码的译码

分组码是信道编码的一大类，和编码对应的是译码。分组码有不同的译码算法，常用的包括伴随式译码、标准阵列译码。

7.3.1 线性分组码的伴随式译码

根据线性分组码生成矩阵和校验矩阵间的关系，见式（7-10），有下面的关系式。

$$\boldsymbol{c}\cdot\boldsymbol{H}^{\mathrm{T}}=\boldsymbol{m}\cdot\boldsymbol{G}\cdot\boldsymbol{H}^{\mathrm{T}}=\boldsymbol{m}\cdot\boldsymbol{0}_{k\times r}=\begin{bmatrix}0 & 0 & \cdots & 0\end{bmatrix}_r \tag{7-15}$$

其中，$\boldsymbol{c}$ 为码字，$\boldsymbol{m}$ 是发送的信息，长度为 k 位；$\boldsymbol{G}$ 是生成矩阵，$\boldsymbol{H}$ 是检验矩阵。

假设接收到的码字是 $\boldsymbol{r}$，结合上面关系式，则有

$$\boldsymbol{r}\cdot\boldsymbol{H}^{\mathrm{T}}=(\boldsymbol{c}+\boldsymbol{e})\cdot\boldsymbol{H}^{\mathrm{T}}=\boldsymbol{e}\cdot\boldsymbol{H}^{\mathrm{T}}=\boldsymbol{s} \tag{7-16}$$

接收码字的差错信息就完全包含在 $\boldsymbol{s}$ 中，$\boldsymbol{s}$ 称为伴随式（或者校正子）。

在此基础上，可以设计基于伴随式的译码方法，具体的步骤如下。

① 根据校验矩阵 H 对接收码字 r 计算伴随式 s，如式（7-16）。

$$\boldsymbol{s}=\boldsymbol{r}\cdot\boldsymbol{H}^{\mathrm{T}}$$

② 伴随式 $\boldsymbol{r}$ 和错误图样 $\boldsymbol{e}$ 是一一对应的，因此根据 $\boldsymbol{r}$ 确定 $\boldsymbol{e}$。

③ 根据错误图样计算发送的码字 $\boldsymbol{c}=\boldsymbol{r}+\boldsymbol{e}$ 。

例 7-3 假设某一线性分组码的校验矩阵如下

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

接收的码字矢量为 $\boldsymbol{r}=\begin{bmatrix}1 & 0 & 0 & 0 & 0 & 1\end{bmatrix}$，求发送的码字。

差错图样与伴随式的关系如下。

错误图样 $\boldsymbol{e}$	伴随式 $\boldsymbol{s}$
000001	001
000010	010
000100	100
001000	111
010000	011
100000	110

解 根据上面的译码步骤，有

（1）先计算伴随式

$$\boldsymbol{s}=\boldsymbol{r}\cdot\boldsymbol{H}^{\mathrm{T}}=\begin{bmatrix}1 & 0 & 0 & 0 & 0 & 1\end{bmatrix}\cdot\begin{bmatrix}1 & 1 & 0\\ 0 & 1 & 1\\ 1 & 1 & 1\\ 1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}=\begin{bmatrix}1 & 1 & 1\end{bmatrix}$$

（2）根据差错图样和伴随式的关系式来确定差错图样，根据（1）的计算结果，可知 $\boldsymbol{e}$=[001000]，此时也可以根据校验矩阵来寻找对应的错误位。

（3）计算发送的码字

$$\boldsymbol{c}=\boldsymbol{r}+\boldsymbol{e}=[1\quad 0\quad 0\quad 0\quad 0\quad 1]+[0\quad 0\quad 1\quad 0\quad 0\quad 0]=[1\quad 0\quad 1\quad 0\quad 0\quad 1]$$

因此在接收到的码字为[100001]的情况下，发送的码字为[101001]，即在传送的过程中有一位错误发生，并且被纠正。

7.3.2 标准阵列译码

1．线性分组码的相关参数

发送码字：由于发送的信息共有 k 位，因此发送的信息组合有 2^k 个可能，所以发送的码字是取自于 2^k 个元素的码字集合$\{C\}$。

接收码字：由于接收到的信息共有 n 位，因此发送的信息组合有 2^n 个可能，所以发送的码字是取自于 2^n 个元素的码字集合$\{R\}$。

由于接收码字的集合远远大于发送码字的集合，因此可以设计一种译码方法，利用这些冗余，标准阵列译码就是一种典型的方法。

2．标准阵列译码方法

标准阵列是对给定的(n, k)线性码，将 2^n 个 n 重矢量划分为 2^k 个子集的一种方法。

标准阵列译码的步骤如下。

① 把 2^n 个 n 重矢量划分为 2^k 个互不相交的子集，使得在每个子集中仅含一个码矢。

② 根据码矢和子集间一一对应的关系，若接收矢量 R_l 落在子集 D_l 中，就把 R_l 译为子集 D_l 含有的码字 C_l。

③ 当接收矢量 R 与实际发送码矢在同一子集中时，译码就是正确的，译码时，在标准阵列中查找接收码字，则该元素所在列的首个元素就是发送码字。

④ 如果在标准阵列中查找不到接收码字，则说明码字中发生了不可纠正（或检测）的错误。

3．标准阵列的构造

标准阵列的构造步骤如图 7-7 所示。

标准阵列译码

码字	C_1（陪集首）	C_2	$\cdots C_i$	$\cdots$	C_{2^k}	
陪集首	E_2	C_2+E_2	C_i+E_2	⋮	$C_{2^k}+E_2$	一个陪集
禁用码字	E_3	C_2+E_3	C_i+E_3	⋮	$C_{2^k}+E_3$	
	⋮	⋮	⋮	⋮	⋮	
	$E_{2^{n-k}}$	$C_2+E_{2^{n-k}}$	$C_i+E_{2^{n-k}}$		$C_{2^k}+E_{2^{n-k}}$	

图 7-7 标准阵列的构造

① 先将 2^k 个码矢排成一行，作为标准阵列的第一行，并将全 0 码矢 C_1=（00…0）放在最左面的位置上。

② 然后在剩下的(2^n-2^k)个 n 重中选取一个重量最轻的 n 重 E_2 放在全 0 码矢 C_1 下面，再将 E_2 分别和码矢相加，放在对应码矢下面构成阵列第二行。

③ 在第二次剩下的 n 重中，选取重量最轻的 n 重 E_3，放在 E_2 下面，并将 E_3 分别加到第一行各码矢上，得到第三行。

④ 继续这样做下去，直到全部 n 重用完为止，得到(n, k)线性码的标准阵列。

例 7-4 假设(6, 3)码，生成矩阵为

$$\boldsymbol{G}=\begin{bmatrix}1&0&0&1&1&0\\0&1&0&0&1&1\\0&0&1&1&1&1\end{bmatrix}$$

如果接收到的码字为(110110)，试利用标准阵列发译码。

解：根据生成矩阵则可以写出其发送全部码字。

000	001	010	011	100	101	110	111
000000	001111	010011	011100	100110	101001	110101	111010

此时可以构造出下面的标准阵列。

000000 （陪集首）	100110	010011	001111	110101	101001	011100	111010
100000	000010	110011	101111	010101	001001	111100	011010
010000	110110	000011	011111	100101	111001	001100	101010
001000	101110	011011	000111	111101	100001	010100	110010
000100	100010	0101110	001011	110001	101101	011000	111110
000010	100100	010001	001101	110111	101011	011110	111000
000001	100111	010010	001110	110100	101000	011101	111011
110000	010110	100011	111111	000101	011001	101100	001010

当接收的码字是(110110)时，可以知道发送的码字为(100110)。

7.4 汉明码及译码

汉明码是最早出现的信道编码技术，该编码的特点是只能够纠正一个错误。下面介绍汉明码的编码以及译码。

7.4.1 汉明码编码

由前述可知，线性分组码中信息码元和监督码元是用线性方程联系起来的，建立在代数学群论的基础上。典型的线性分组码有 Hamming（汉明）码、循环码和 BCH 码等。本节将

以汉明码为例来介绍线性分组码构造的一般原理以及相关概念。

1．编码原理

奇偶监督码是一种最简单的线性码，让我们先来回顾一下偶数监督码。由于使用了一位监督位 a_0，它就能和信息位 $a_{n-1}\cdots a_1$ 一起构成一个偶校验关系式，如式（7-1）。在接收端解码时，实际上就是将上式再计算一遍，有

$$S = a_{n-1} \oplus a_{n-2} \oplus \cdots \oplus a_0 \tag{7-17}$$

这里 S 称为校正子，又称为伴随式。若 $S=0$，表示无错误；若 $S=1$，表示有错误；由于奇偶监督码中只有一位监督码元，因此只能表示两种信息：有错或无错，而不能指出错码的位置。

设想监督位增加一位，即变成两位，此时将有两个监督方程式。在接收端按照两个监督关系式可计算出两个校正子 S_1 和 S_2。$S_1\ S_2$ 将有四种组合：00、01、10，11，故能表示 4 种不同的信息。若用其中一种表示无错，则其余 3 种就可以指示 3 种不同的错误图样；对二进制码而言，其余 3 种信息即可指示一位错码的 3 种不同位置。

同理，由 r 个监督方程式可计算得到 r 个校正子，有 2^r 种组合，可以表示 2^r-1 种错误图样，即可指示一位错码的 2^r-1 种可能位置。对于 (n,k) 分组码而言，如果满足 $2^r-1\geqslant n$，则有可能构造出纠正一位或一位以上错误的线性码。

汉明码是 Hamming 在 1950 年提出的第一个设计用来纠正单个错误且编码效率较高的线性分组码，下面以具体例子来说明其构造原理。

设分组码 (n,k) 中 $k=4$，为了能够纠正一位错码，要求满足 $2^r-1\geqslant n$，即需要满足

$$2^r \geqslant n+1 = k+r+1$$

经过计算可知，要求 $r\geqslant 3$。现取 $r=3$，则 $n=k+r=7$。我们用 $a_6a_5a_4a_3a_2a_1a_0$ 表示这 7 个码元，用 $S_1\ S_2\ S_3$ 表示由三个监督方程式所对应的校正子，并假设 $S_1\ S_2\ S_3$ 三位校正子码组与误码位置的对应关系如表 7-5 所示（也可以规定成另一种对应关系，其不影响讨论的一般性）。

由表 7-5 可知，当误码位置在 a_2、a_4、a_5 和 a_6 时，校正子 $S_1=1$；否则 $S_1=0$。这就意味着 a_2、a_4、a_5 和 a_6 四个码元构成偶数监督关系，因此有

$$S_1 = a_6 \oplus a_5 \oplus a_4 \oplus a_2 \tag{7-18}$$

依此类推，a_1、a_3、a_5 和 a_6 构成偶数监督关系，即有

$$S_2 = a_6 \oplus a_5 \oplus a_3 \oplus a_1 \tag{7-19}$$

以及 a_0、a_3、a_4 和 a_6 构成偶数监督关系，即有

$$S_3 = a_6 \oplus a_4 \oplus a_3 \oplus a_0 \tag{7-20}$$

表 7-5　　校正子与误码位置对应关系

$S_1S_2S_3$	误码位置	$S_1S_2S_3$	误码位置
001	a_0	101	a_4
010	a_1	110	a_5
100	a_2	111	a_6
011	a_3	000	无错

在发送端编码时，$a_6a_5a_4a_3$ 为信息码元，取决于被传输的信息。$a_2a_1a_0$ 为三位监督码元，其值应由上述的监督方程式确定，即应使监督方程式满足偶数监督关系（当码组正确传输时应保证 S_1、S_2、S_3 的值为 0），有

$$\begin{cases} a_6 \oplus a_5 \oplus a_4 \oplus a_2 = 0 \\ a_6 \oplus a_5 \oplus a_3 \oplus a_1 = 0 \\ a_6 \oplus a_4 \oplus a_3 \oplus a_0 = 0 \end{cases} \tag{7-21}$$

将上式经移位运算，得到监督位分别为

$$\begin{cases} a_2 = a_6 \oplus a_5 \oplus a_4 \\ a_1 = a_6 \oplus a_5 \oplus a_3 \\ a_0 = a_6 \oplus a_4 \oplus a_3 \end{cases} \tag{7-22}$$

由此得到的 16 个许用码组列于表 7-6 中。

表 7-6　　(7, 4)汉明码许用码组

信息位 $a_6a_5a_4a_3$	监督位 $a_2a_1a_0$	信息位 $a_6a_5a_4a_3$	监督位 $a_2a_1a_0$
0000	000	1000	111
0001	011	1001	100
0010	101	1010	010
0011	110	1011	001
0100	110	1100	001
0101	101	1101	010
0110	011	1110	100
0111	000	1111	111

接收端收到每个码组后，计算出 S_1、S_2 和 S_3，如不全为 0，则可根据表 7-5 确定误码的位置，然后予以纠正。例如接收码组为：0000011，可算得 $S_1S_2S_3$=011，由表 7-5 可知在 a_3 位置上有一错码，纠正接收码组为：0001011。

由上可见，上述(7, 4)汉明码具有以下特点。

码长：$n=2^r-1$；最小码距：$d_{\min}=3$；信息码位：$k=2^r-1-r$，纠错能力=1；监督码位：$r=n-k$；编码效率：$R=k/n$。

由于汉明码的最小码距等于 3，根据“最小码距与纠检错能力”可知，这种码能纠正一个错码或检测两个错码。且其编码效率等于 $R=k/n=(2^r-1-r)/(2^r-1)=1-r/n$。当 n 很大时，编码效率接近于 1。表 7-7 所示为前 8 个汉明码的编码效率。

表 7-7　　汉明码的编码率

r	k	n	$R=k/n$
3	4	7	0.57
4	11	15	0.73
5	26	31	0.84
6	57	63	0.90
7	120	127	0.94
8	247	255	0.97
9	502	511	0.98
10	1013	1023	0.99

2．监督矩阵和生成矩阵

（1）监督矩阵。我们再来讨论线性分组码的一般原理。从上节汉明码的例子出发，将(7,4)汉明码的三个监督方程式（7-21）改写成如下形式

$$\begin{cases}1\cdot a_6+1\cdot a_5+1\cdot a_4+0\cdot a_3+1\cdot a_2+0\cdot a_1+0\cdot a_0=0\\1\cdot a_6+1\cdot a_5+0\cdot a_4+1\cdot a_3+0\cdot a_2+1\cdot a_1+0\cdot a_0=0\\1\cdot a_6+0\cdot a_5+1\cdot a_4+1\cdot a_3+0\cdot a_2+0\cdot a_1+1\cdot a_0=0\end{cases} \tag{7-23}$$

上式中“+”表示“模 2”加，后面除另加说明，“+”均表示是“模 2”加。上述线性方程组可用矩阵表示为

$$\begin{bmatrix}1110100\\1101010\\1011001\end{bmatrix}\cdot[a_6a_5a_4a_3a_2a_1a_0]^{\mathrm{T}}=\begin{bmatrix}0\\0\\0\end{bmatrix} \tag{7-24}$$

上式简记为

$$\boldsymbol{H}\cdot\boldsymbol{A}^{\mathrm{T}}=\boldsymbol{0}^{\mathrm{T}}\text{ 或 }\boldsymbol{A}\cdot\boldsymbol{H}^{\mathrm{T}}=\boldsymbol{0} \tag{7-25}$$

$$\boldsymbol{H}=\begin{bmatrix}1110100\\1101010\\1011001\end{bmatrix}$$

其中

$$\boldsymbol{A}=[a_6a_5a_4a_3a_2a_1a_0] \tag{7-26}$$
$$\boldsymbol{0}=[000]$$

由于式（7-24）来自监督方程式，因此 $\boldsymbol{H}$ 矩阵称为线性码的“监督矩阵”，它决定了信息码元与监督码元之间的校验关系。换句话说，只要监督矩阵 $\boldsymbol{H}$ 给定，编码时监督码元和信息码元的关系就完全确定了。 $\boldsymbol{H}$ 矩阵式（7-26）可以分成两部分。

$$\boldsymbol{H}=\begin{bmatrix}1110\vdots100\\1101\vdots010\\1011\vdots001\end{bmatrix}=[\boldsymbol{P}\vdots\boldsymbol{I}_r] \tag{7-27}$$

式中，$\boldsymbol{P}$ 为 $r\times k$ 阶矩阵，$\boldsymbol{I}_r$ 为 $r\times r$ 阶单位方阵。通常将具有 $[\boldsymbol{P}\vdots\boldsymbol{I}_r]$ 形式的 $\boldsymbol{H}$ 矩阵称为典型形式的监督矩阵（典型阵）。

从上面可看出，$\boldsymbol{H}$ 矩阵的行数就是监督方程的个数，即监督码元的位数 r；$\boldsymbol{H}$ 矩阵的列数就是码长 n，这样 $\boldsymbol{H}$ 为 $r\times n$ 阶矩阵，矩阵中元素“1”表示相应码元之间存在着偶校验关系，由典型阵及信息码元就很容易算出各监督码元。且由线性代数理论得知，$\boldsymbol{H}$ 矩阵的各行应该是线性无关的，任何非典型阵都可以经过行运算化成典型阵。

（2）生成矩阵。若知道典型形式的监督矩阵和信息码元，就能确定各监督码元。式（7-22）可写成矩阵形式，即

$$\begin{bmatrix}a_2\\a_1\\a_0\end{bmatrix}=\begin{bmatrix}1110\\1101\\1011\end{bmatrix}\begin{bmatrix}a_6\\a_5\\a_4\\a_3\end{bmatrix}$$

或者

$$[a_2a_1a_0]=[a_6a_5a_4a_3]\begin{bmatrix}111\\110\\101\\011\end{bmatrix}=[a_6a_5a_4a_3]\boldsymbol{Q} \tag{7-28}$$

式中，$\boldsymbol{Q}$ 为 $k\times r$ 阶矩阵，它为 $\boldsymbol{P}$ 矩阵的转置，即

$$\boldsymbol{Q}=\boldsymbol{P}^{\mathrm{T}} \tag{7-29}$$

根据式（7-28），同样可由信息码元算出监督码元。

在 $\boldsymbol{Q}$ 的左边加上一个 $k\times k$ 阶的单位方阵，构成一个新的矩阵 $\boldsymbol{G}$，即

$$\boldsymbol{G}=[\boldsymbol{I}_k\vdots\boldsymbol{Q}]=\begin{bmatrix}1000\vdots111\\0100\vdots110\\0010\vdots010\\0001\vdots011\end{bmatrix} \tag{7-30}$$

$\boldsymbol{G}$ 称为生成矩阵，因为由它可以产生整个码组 $\boldsymbol{A}$，即有

$$\boldsymbol{A}=[a_6a_5a_4a_3]\boldsymbol{G} \tag{7-31}$$

把具有 $[\boldsymbol{I}_k\vdots\boldsymbol{Q}]$ 形式的生成矩阵称为典型形式生成矩阵。由它产生的码组，信息位不变，监督位附加于其后，把这种码称为系统码。非典型形式的生成矩阵经过行运算也一定可以化成典型形式的生成矩阵。同样，典型生成矩阵的各个行也必定是线性无关的，且每行都是一个许用码组。线性无关的 k 行许用码组经过行运算可产生 2^k 个不同的许用码组（恰好是 k 位信息位的全部码组）。由此可见，如果可以找到 k 个线性无关的码组，则可以用它们作为生成矩阵 $\boldsymbol{G}$，并由它生成其余的码组。

7.4.2 汉明码的伴随式译码

汉明码可以采用上节中提到的伴随式译码进行译码，具体见下面的例子。

例 7-5 假设(7, 4)汉明码的校验矩阵为

$$\boldsymbol{H}=\begin{bmatrix}1110100\\1101010\\1011001\end{bmatrix}$$

接收码组 $\boldsymbol{r}$ 为(0000011)，试确定发送的码字。

解

$$\boldsymbol{S}=\boldsymbol{e}\boldsymbol{H}^{\mathrm{T}}=[0001000]\begin{bmatrix}111\\110\\101\\011\\100\\010\\001\end{bmatrix}=[011]$$

同样可计算出其他错误图样所对应的伴随式

$$0000001 \to 001$$
$$0000010 \to 010$$
$$0000100 \to 100$$
$$0001000 \to 011$$
$$0010000 \to 101$$
$$0100000 \to 110$$
$$1000000 \to 111$$

在接收端译码器将根据接收到的码组 $\boldsymbol{r}$ 计算它的伴随式，结果为

$$\boldsymbol{S} = \boldsymbol{B}\boldsymbol{H}^{\mathrm{T}} = [0000011]\begin{bmatrix} 111 \\ 110 \\ 101 \\ 011 \\ 100 \\ 010 \\ 001 \end{bmatrix} = [011]$$

计算结果与相其对应的错误图样计算出来的校正子相同，译码器通过查找“错误图样与校正子对应关系表”，将 $\boldsymbol{e} = [0001000]$ 作为接收错误图样，则正确的码组即是接收码组和接收错误图样的模 2 和，据此得到

$$\boldsymbol{c} = \boldsymbol{r} - \boldsymbol{e} = \boldsymbol{r} + \boldsymbol{e} = [0001011]$$

所以发送的码字为(0001011)，发送的信息为前 4 位(0001)。

7.4.3 汉明码的主要性质

1．封闭性

汉明码有一个重要的性质，就是它具有封闭性。所谓封闭性，是指一种线性码中的任意两个许用码组之和（逐位模 2 和）仍为这种码中的一个许用码组。这就是说，若 A 和 B 为一种线性码中的两个许用码组，则 $(A+B)$ 仍为其中的一个许用码组。

此性质的证明很简单，若 A 和 B 为一种线性码中的两个许用码组，则它们应满足监督关系式，即

$$\boldsymbol{A} \cdot \boldsymbol{H}^{\mathrm{T}} = 0, \qquad \boldsymbol{B} \cdot \boldsymbol{H}^{\mathrm{T}} = 0 \tag{7-32}$$

将上两式相加，可得

$$\boldsymbol{A} \cdot \boldsymbol{H}^{\mathrm{T}} + \boldsymbol{B} \cdot \boldsymbol{H}^{\mathrm{T}} = (\boldsymbol{A} + \boldsymbol{B}) \cdot \boldsymbol{H}^{\mathrm{T}} = 0 \tag{7-33}$$

由于码组 $(\boldsymbol{A} + \boldsymbol{B})$ 满足监督关系式，因此，码组 $(\boldsymbol{A} + \boldsymbol{B})$ 也是一许用码组。

2．码的最小距离等于非零码的最小重量

因为线性分组码具有封闭性，因而两个码组之间的距离必是另一码组的重量。为此，码的最小距离也就是非零码的最小重量（除全“0”码组）。

该性质也是线性分组码的基本性质。

7.5 循环码（CRC）

循环码是分组码中主要的分支，有着良好的纠错性能，在无线通信、计算机通信等领域中得到广泛的应用。

7.5.1 循环码基础

本小节介绍循环码的定义、生产多项式等基本理论知识。

1．循环码的定义

循环码是线性分组码的一个重要子类，它是在严密的代数学理论基础上建立起来的。它为系统码，即前 k 位为信息码，后 r 位为监督码元。它除了具有线性码的一般性质外，还具有循环性。所谓循环性是指循环码中任一许用码组经过循环移位后得到的码组仍为一许用码组。若 $[a_{n-1}a_{n-2}\cdots a_1a_0]$ 为一许用的循环码组，则它的循环移位 $[a_{n-2}a_{n-3}\cdots a_0a_{n-1}]$、$[a_{n-3}a_{n-4}\cdots a_{n-1}a_{n-2}]$ 等也是许用码组。

2．循环码的码多项式运算

把码组中各码元当作是一个多项式的系数，将长为 n 的码组与 $(n-1)$ 次多项式建立一一对应关系。这样就可以将码组用多项式来表示，称为码多项式。比如码组 $A=[a_{n-1}a_{n-2}\cdots a_1a_0]$，则相应的码多项式可表示为

$$A(x)=a_{n-1}x^{n-1}+a_{n-2}x^{n-2}+\cdots+a_1x^1+a_0 \tag{7-34}$$

在码多项式中，x 为一任意的实变量，只代表码元位置的标记。对于二进制码，多项式系数 $a_i(i=n-1,n-2,\cdots,1,0)$ 取 0 或 1。码多项式中 x^i 的存在只表示该位对应的码信息是“1”码，否则为“0”。

从代数学的角度，每个二进制码组可以看成是只有 0 和 1 两个元素的二元域中的 n 重。所有二元 n 重的集合称为二元域上的一个矢量空间。二元域上只有两种运算：加和乘，所有运算结果也必定在同一个二元集合中。

其运算规则定义如表 7-8 所示。

表 7-8　　二元域的加和乘运算

加	乘
0+0=0	0×0=0
0+1=1	0×1=0
1+0=1	1×0=0
1+1=0	1×1=1

由上可见，加运算其实质上是作模 2 和运算。

一般而言，在整数运算中，有模 n 运算。若一整数 m 可以表示为

$$\frac{m}{n}=P+\frac{q}{n}$$

式中, P为整数，且$q < n$ 。

在模 n 运算下，有

$$m = q\ (\text{模}\ n) \tag{7-35}$$

在码多项式中也有类似的按模运算。若一任意码多项式 $A(x)$ 被一 n 次多项式 $B(x)$ 除，得到商式 $P(x)$ 和一个次数小于 n 的余式 $q(x)$，即

$$\frac{A(x)}{B(x)} = P(x) + \frac{q(x)}{B(x)} \tag{7-36}$$

在模 $B(x)$ 运算下，有

$$A(x) = q(x)\ （\text{模}\ B(x)\ ） \tag{7-37}$$

此时，码多项式的系数仍按模 2 运算，即只取 0 和 1 值。

例 7-6 $x^4 + x\ +1 = x^2 + 1$（模 $x^3 + x + 1$）

$$\begin{array}{r} x \\ x^3 + x + 1 \overline{\big) x^4 + x + 1} \\ \underline{x^4 + x^2 + x} \\ x^2 + 1 \end{array}$$

注意：由于二元域中，码多项式系数的运算实质上是模 2 和运算，因此，减法就相当于加法，所以余式不是 $-x^2 + 1$（二元域中只存在 0 和 1 值），而是 $x^2 + 1$。

3．循环码的循环移位特点

循环码的许用码组 $A(x)$ 可用 $(n-1)$ 次多项式表示为

$$A(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1x^1 + a_0 \tag{7-38}$$

上述许用码组向左循环移一位后得到的码组记为 $A^{(1)} = \left[a_{n-2}a_{n-3}\cdots a_0a_{n-1}\right]$，其码多项式为

$$A^{(1)}(x) = a_{n-2}x^{n-1} + a_{n-3}x^{n-2} + \cdots + a_0x + a_{n-1} \tag{7-39}$$

同理，其左移 i 位的码组 $A^{(i)} = \left[a_{n-1-i}a_{n-2-i}\cdots a_{n-i+1}a_{n-i}\right]$，其码多项式为

$$A^{(i)}(x) = a_{n-1-i}x^{n-1} + a_{n-2-i}x^{n-2} + \cdots + a_0x^i + \cdots + a_{n-i+1}x + a_{n-i} \tag{7-40}$$

其中，左移 i 位的码组 $A^{(i)}(x)$ 可由下式求得

$$\frac{x^i \cdot A(x)}{x^n + 1} = P(x) + \frac{q(x)}{x^n + 1} = P(x) + \frac{A^i(x)}{x^n + 1} \tag{7-41}$$

式（7-41）表明，循环码的一个许用码组 $A^{(i)}(x)$，是 $x^i \cdot A(x)$ 除 $x^n + 1$ 的余式。上式还可另写为

$$A^i(x) = x^i \square A(x)(\text{模}x^n + 1) \tag{7-42}$$

由此可见，在循环码中，若 $A(x)$ 是一个长为 n 的许用码组，则 $x^i \cdot A(x)$ 在按模 $x^n + 1$ 运算下，也是一个许用码组。它的证明较为简单，我们已知

$$x^i \square A(x) = a_{n-1}x^{n-1+i} + a_{n-2}x^{n-2+i} + \cdots + a_{n-1-i}x^{n-1} + \cdots + a_1x^{1+i} + a_0x^i \tag{7-43}$$

对上式 $x^i \cdot A(x)$ 除以 $x^n + 1$ 后得到的余式为

$$q(x) = a_{n-1-i}x^{n-1} + a_{n-2-i}x^{n-2} + \cdots + a_0x^i + \cdots + a_{n-i+1}x + a_{n-i} = A^{(i)}(x) \tag{7-44}$$

可以看到，$q(x)$ 正是许用码组 $A(x)$ 向左循环移位 i 次的结果，即它也是循环码中的一许用码组 $A^{(i)}=\left[a_{n-1-i}a_{n-2-i}\cdots a_{n-i+1}a_{n-i}\right]$。

7.5.2 循环码生成矩阵、生成多项式和监督矩阵

循环码是分组码的主要分支，因此循环码就符合分组码的描述形式，也可以用生成矩阵、校验矩阵来进行描述。

1．循环码的生成矩阵

由前述式（7-31）可知，有了生成矩阵 $\boldsymbol{G}$，就可由 k 个信息位得出整个码组，而且生成矩阵 $\boldsymbol{G}$ 中的每一行都是一个许用码组。我们还知道，如果可以找到 k 个线性无关的许用码组，则可以用它们作为生成矩阵 $\boldsymbol{G}$，并由它生成其余的码组。

在循环码中，一个 (n,k) 循环码有 2^k 个许用码组。若用 $g(x)$ 表示其中前 $(k-1)$ 位皆为“0”的码组，用 $x\cdot g(x)$、$x^2\cdot g(x)$、…、$x^{k-1}\cdot g(x)$ 分别表示其向左移 1、2、…、$k-1$ 位的码组（实际上是 $x^i\cdot g(x)$ 除以 x^n+1 的余式）。根据循环性可知：$g(x)$、$x\cdot g(x)$、$x^2\cdot g(x)$、…、$x^{k-1}\cdot g(x)$ 都是许用码组，而且这 k 个码组将是线性无关的。因此可用它们构成循环码的生成矩阵。其中 $g(x)$ 又被称为循环码的生成多项式。

因此，循环码的生成矩阵 $\boldsymbol{G}_{k\times n}$ 可以写成

$$\boldsymbol{G}_{k\times n}(x)=\begin{bmatrix} x^{k-1}\cdot g(x) \\ x^{k-2}\cdot g(x) \\ \vdots \\ x\cdot g(x) \\ g(x) \end{bmatrix} \tag{7-45}$$

若用 $M(x)$ 表示信息多项式，其定义为

$$M(x)=m_{k-1}x^{k-1}+\cdots+m_1x+m_0 \tag{7-46}$$

式中，$\left[m_{k-1}\cdots m_1m_0\right]$ 表示 k 个信息比特。由此得到的码组为

$$\begin{aligned} A(x)&=M(x)\cdot G(x) \\ &=\left[m_{k-1}\cdots m_1m_0\right]\cdot\begin{bmatrix} x^{k-1}\cdot g(x) \\ \vdots \\ x\cdot g(x) \\ g(x) \end{bmatrix} \\ &=(m_{k-1}x^{k-1}+\cdots+m_1x+m_0)\cdot g(x) \end{aligned} \tag{7-47}$$

式（7-47）表明，所有的许用码组多项式都可被 $g(x)$ 整除，而且任一次数不大于 $(k-1)$ 的多项式乘 $g(x)$ 都是循环码的许用码多项式。且因为 $A(x)$ 是一个阶次小于 n 的多项式，所以由式（7-47）可知，$g(x)$ 应是一常数项不为 0 的 $(n-k)$ 阶多项式。因为如果常数项为 0，则经过右移一位，会得到一个信息位全为 0，而监督位不全为 0 的码组，这在线性码中显然是不可能的。

2．循环码的生成多项式

为了找出循环码的生成矩阵，现在的问题是如何寻找任一 (n,k) 循环码的生成多项式

$g(x)$？由上分析可知，任一循环码许用码组都是 $g(x)$ 的倍数，故有下式成立。

$$A(x)=M(x)\cdot g(x) \tag{7-48}$$

而 $g(x)$ 本身也是一个许用码组，故下式也成立。

$$\frac{x^k\cdot g(x)}{x^n+1}=P(x)+\frac{A(x)}{x^n+1} \tag{7-49}$$

由于 $g(x)$ 为一常数项不为 0 的 $(n-k)$ 次多项式，故 $x^k\cdot g(x)$ 必为一 n 次多项式。上式分子和分母都是 n 次多项式，故商式 $P(x)=1$。由此上式可化为

$$x^k\cdot g(x)=(x^n+1)+A(x) \tag{7-50}$$

将式（7-48）和式（7-49）带入式（7-50）中化简后得

$$x^n+1=\left[x^k+M(x)\right]\cdot g(x) \tag{7-51}$$

由此可见，生成多项式 $g(x)$ 应该是 x^n+1 的一个常数项不为 0、阶次为 $(n-k)$ 次的因子。这就为我们找到循环码的生成多项式找到了途径：当求一个(n, k)循环码编码时，只要分解多项式（x^n+1），从中取出$(n-k)$次因式作生成多项式即可。

例 7-7 求 x^n+1 当 $n=7$ 时的循环码生成多项式。

解 进行因式分解

$$x^7+1=(x+1)(x^3+x^2+1)(x^3+x+1)$$

由上式可形成表 7-9 所示的 $(7,k)$ 循环码。

表 7-9 **x^7+1 因式分解构成的循环码**

(n,k)	$g(x)$
(7,6)	$x+1$
(7,4)	x^3+x^2+1 或 x^3+x+1
(7,3)	$(x+1)\cdot(x^3+x^2+1)$ 或 $(x+1)\cdot(x^3+x+1)$
(7,1)	$(x^3+x^2+1)\cdot(x^3+x+1)$

当 $g(x)=x^3+x^2+1$ 时，由此可写出(7,4)循环的的生成矩阵为

$$G(x)=\begin{bmatrix}x^3\cdot g(x)\\ x^2\cdot g(x)\\ x\cdot g(x)\\ g(x)\end{bmatrix}=\begin{bmatrix}x^6+x^5+x^3\\ x^5+x^4+x^2\\ x^4+x^3+x\\ x^3+x^2+1\end{bmatrix}=\begin{bmatrix}1101000\\ 0110100\\ 0011010\\ 0001101\end{bmatrix}$$

注意：以上两式 x^3+x^2+1 或 x^3+x+1 和 $(x+1)\cdot(x^3+x^2+1)$ 或 $(x+1)\cdot(x^3+x+1)$ 都可作为生成多项式，它们将分别产生不同的循环码组。

3．监督矩阵

式（7-45）给出了循环码的生成矩阵，但此生成矩阵不是典型的，可通过线性变换使之成为典型生成矩阵，记为

$$\boldsymbol{G}_{k\times n}=[\boldsymbol{I}_k\vdots\boldsymbol{Q}] \tag{7-52}$$

由于生成多项式 $g(x)$ 能除尽 x^n+1（因为它是 x^n+1 的一个因子，且可表示为 $g(x)=g_{n-k}x^{n-k}+\cdots+g_1x+g_0$，且 $g_0=1$），因此有

$$x^n+1=g(x)\square h(x) \tag{7-53}$$

由于式（7-53）是循环码许用码组必须要满足的监督关系，因此 $h(x)$ 称为监督多项式，且 $h(x)=h_kx^k+\cdots+h_1x+h_0$。由式（7-45）可知，必定有

$$\begin{aligned}
&g_{n-k}\square h_k=1\\
&g_0\square h_0=1\\
&g_1\square h_0+g_0\square h_1=0\\
&g_2\square h_0+g_1\square h_1+g_0\square h_2=0\\
&\cdots\\
&g_{n-1}\square h_0+g_{n-2}\square h_1+\cdots+g_{n-k}\square h_{k-1}+\cdots+g_0\square h_{n-1}=0
\end{aligned} \tag{7-54}$$

由此可得循环码的监督矩阵为

$$\boldsymbol{H}_{r\times n}=\begin{bmatrix}
h_0 & h_1 & \cdots & & h_{k-1} & h_k & 0 & 0 & & \cdots & & 0\\
0 & h_0 & h_1 & \cdots & & h_{k-1} & h_k & 0 & 0 & \cdots & & 0\\
0 & 0 & h_0 & h_1 & \cdots & & h_{k-1} & h_k & 0 & 0 & \cdots & 0\\
\vdots & \vdots & & & & & & & & & \vdots & \\
0 & 0 & 0 & 0 & 0 & 0 & h_0 & h_1 & \cdots & & h_{k-1} & h_k
\end{bmatrix} \tag{7-55}$$

它完全由监督多项式 $h(x)$ 的系数确定，有 $\boldsymbol{G}\square\boldsymbol{H}^{\mathrm{T}}=0$。

上述监督方程式不是典型形式，可将它经线性变换化成典型形式，即典型形式的监督矩阵为

$$\boldsymbol{H}_{r\times n}=\left[\boldsymbol{Q}^{\mathrm{T}}\vdots\boldsymbol{I}_r\right] \tag{7-56}$$

例 7-8 构造一个(7, 3)循环码。并要求写出典型的生成矩阵、监督矩阵及所有许用码组。

解 因为生成多项式 $g(x)$ 应该是 x^7+1 的一个常数项不为 0、阶次为 $n-k=4$ 次的因子。所以，首先分解 x^7+1

$$x^7+1=(x+1)(x^3+x^2+1)(x^3+x+1)$$

可以求得

$$g(x)=(x+1)(x^3+x^2+1)=x^4+x^2+x+1 \tag{7-57}$$

或

$$g(x)=(x+1)(x^3+x+1)=x^4+x^3+x^2+1 \tag{7-58}$$

式（7-57）和式（7-58）两式都可以生成(7,3)循环码。我们以式（7-57）为例来说明。

由式（7-45）可得其生成矩阵为

$$\boldsymbol{G}_{3\times7}=\begin{bmatrix}x^2g(x)\\ xg(x)\\ g(x)\end{bmatrix}=\begin{bmatrix}x^6+x^4+x^3+x^2\\ x^5+x^3+x^2+x\\ x^4+x^2+x+1\end{bmatrix}=\begin{bmatrix}1011100\\ 0101110\\ 0010111\end{bmatrix}$$

将其化成典型阵得

$$\boldsymbol{G}_{3\times7}=\begin{bmatrix}100\vdots1011\\010\vdots1110\\001\vdots0111\end{bmatrix}=\left[\boldsymbol{I}_k\vdots\boldsymbol{Q}\right]$$

由式（7-53）可知，$x^n+1=g(x)\cdot h(x)$，则该循环码相对应的监督多项式 $h(x)$ 为

$$h(x)=x^3+x+1$$

根据式（7-55）可写出其监督矩阵为

$$\boldsymbol{H}_{4\times7}=\begin{bmatrix}h_0h_1h_2h_3000\\0h_0h_1h_2h_300\\00h_0h_1h_2h_30\\000h_0h_1h_2h_3\end{bmatrix}=\begin{bmatrix}1101000\\0110100\\0011010\\0001101\end{bmatrix}$$

将其化成典型阵得

$$\boldsymbol{H}_{4\times7}=\begin{bmatrix}110\vdots1000\\011\vdots0100\\111\vdots0010\\101\vdots0001\end{bmatrix}=\left[\boldsymbol{Q}^{\mathrm{T}}\vdots\boldsymbol{I}_r\right]$$

容易验证

$$\boldsymbol{G}\cdot\boldsymbol{H}^{\mathrm{T}}=\begin{bmatrix}0000\\0000\\0000\end{bmatrix}=\boldsymbol{0}$$

根据式（7-47），$A(x)=M(x)\cdot G(x)$ 即可得到所有的循环码组，如表 7-10 所示。也可根据式（7-48），$A(x)=M(x)\cdot g(x)$ 得到所有的循环码组，只不过此时不再是系统码而已。

表 7-10　　$g(x)=x^4+x^2+x+1$ 的(7,3)循环码的所有许用码组

码组编号	信息位	监督位	码组编号	信息位	监督位
	$a_6a_5a_4$	$a_3a_2a_1a_0$		$a_6a_5a_4$	$a_3a_2a_1a_0$
1	000	0000	5	100	1011
2	001	0111	6	101	1100
3	010	1110	7	110	0101
4	011	1001	8	111	0010

7.5.3　循环码的编、译码

本小节介绍循环码的编码和译码原理以及实现。

1．循环码的编码器

所谓编码，就是在已知信息位的条件下求得循环码的码组。首先，根据给定的 (n,k) 值选定生成多项式 $g(x)$，它是 x^n+1 的一个常数项不为 0、阶次为 $(n-k)$ 次的因子。再根据式（7-45），由生成多项式得到循环码的生成矩阵 $G(x)$。

若已知输入信息码元为 $(m_{k-1}m_{k-2}\cdots m_1m_0)$ 时，相应的循环码多项式为

$$\begin{aligned} A(x) &= (m_{k-1}m_{k-2}\cdots m_1m_0)\,G(x) \\ &= (m_{k-1}x^{k-1}+m_{k-2}x^{k-2}+\cdots+m_1x+m_0)\,g(x) \\ &= M(x)\,g(x) \end{aligned} \tag{7-59}$$

根据上式就可求得循环码各许用码组，且都为 $g(x)$ 的倍式。

由式(7-59)得到的循环码并非系统码。在系统码中，码组的前 k 位为信息位，随后的 $n-k$ 位是监督位。即此时的码多项式为

$$\begin{aligned} A(x) &= M(x)\,x^{n-k}+r(x) \\ &= m_{k-1}x^{n-1}+\cdots+m_1x^{n-k+1}+m_0x^{n-k}+r_{n-k-1}x^{n-k-1}+\cdots+r_0 \end{aligned} \tag{7-60}$$

这里，$r(x)=r_{n-k-1}x^{n-k-1}+\cdots+r_0$ 为监督码多项式，对应的监督码元为 $(r_{n-k-1}\cdots r_0)$。

由式（7-60）可知

$$\begin{aligned} r(x) &= A(x)+M(x)\,x^{n-k} \\ &\equiv M(x)\,x^{n-k} \bmod g(x) \end{aligned} \tag{7-61}$$

由上式可知，构造循环码系统码时，只需将信息码多项式升 $n-k$ 阶（乘以 x^{n-k}），然后对 $g(x)$ 作模运算（除以 $g(x)$），所得余式 $r(x)$ 即为监督码元。因此循环码的编码步骤可归纳如下。

① 首先利用 x^{n-k} 乘以信息码多项式 $M(x)$。

② 然后再用 $g(x)$ 除 $x^{n-k}\,M(x)$，得到商式 $P(x)$ 和余式 $r(x)$，即

$$\frac{x^{n-k}\,M(x)}{g(x)}=P(x)+\frac{r(x)}{g(x)} \tag{7-62}$$

③ 最后编出码组。

$$A(x)=x^{n-k}\,M(x)+r(x) \tag{7-63}$$

例 7-9 已知(7, 4)循环码的生成多项式为 $g(x)=x^3+x+1$，若信息位为 1101，求编码后的循环码组。

解 信息位为 1101，其对应的信息码多项式为相当于 $M(x)=x^3+x^2+1$，则

$x^{n-k}\,M(x)=x^3\,(x^3+x^2+1)=x^6+x^5+x^3$，它相当于 1101000。这一运算实际上是在信息码后附加上 $n-k$ 个“0”。

代入式（7-53）中可得

$$\frac{x^6+x^5+x^3}{x^3+x+1}=x^3+x^2+x+1+\frac{1}{x^3+x+1}$$

因此，根据式（7-63）可得生成的循环码组为

$$A(x)=x^3\,(x^3+x^2+1)+1=x^6+x^5+x^3+1$$

其对应的码组为：1101001。

根据以上编码规则，其可用除法电路来实现。多项式除法电路可以用带反馈的线性移位寄存器来实现。它主要由移位寄存器、模 2 加法器和反馈线组成。其中，有 $r=n-k$ 个移位寄存器，在移位寄存器之间存在模 2 加法器（是否存在取决于生成多项式 $g(x)$ 的系数，系数

为 1 就存在，系数为 0 不存在）。例如图 7-8 给出了上述(7, 4)循环码编码器的组成示意图。图中有 3 级移位寄存器，分别用 D_1、D_2、D_3 表示。另外有一双刀双掷开关 S。由图 7-8 可见，当信息位输入时，开关 S 倒向下，输入信码一方面送入除法电路进行除法运算，另一方面直接输出。当信息位全部进入除法器后，开关转向上，输出端接到了移位器上，同时切断了反馈线。此时将移位器中存储的除法余项依次取出。

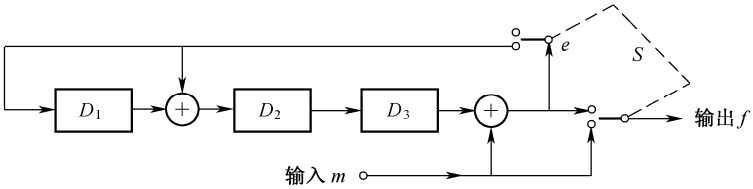


图 7-8 （7，4）循环码编码器

编码器的工作过程示于表 7-11 中。顺便指出，以上编码器是以每次一个比特的方式实现的，对于某些高速应用的场合，编码器可以用每次几个比特的方式来实现，即可以采用带反馈的并联移位寄存器来实现。此外由于微处理器和数字信号处理器的广泛应用，目前宜多采用这些先进的器件和相应的软件来实现上述编码。

表 7-11　　(7, 4)循环码编码器工作过程

输入	移位寄存器	反馈	输出	
m	$D_1D_2D_3$	e	f	
0	000	0	0	
1	110	1	1	$f=m$
1	101	1	1	
0	100	1	0	
1	100	1	1	
0	010	0	0	$f=e$
0	001	0	0	
0	000	1	1	

2．循环码的译码器

接收端译码包括检错和纠错。只需检错的译码原理非常简单。我们知道，循环码任一可用码多项式 $A(x)$ 都能被生成多项式 $g(x)$ 所整除，所以接收端只需将接收到的码多项式 $A'(x)$ 用生成多项式 $g(x)$ 去除。若余式为 0（被生成多项式整除），说明传输过程中未发生错误；若余式不为 0（没有被生成多项式整除），则说明在传输过程中发生了误码。因此可以用余项是否为零来判断码组中有无差错。需要说明的是，当一可用码多项式错成另一可用码多项式时，它也能被 $g(x)$ 所整除，这时的错码就不能被检出了，这种错误称为不可检错误。

在接收端，如果需要纠错，则采用的译码方法要比检错时复杂很多。由前述可知，为了能够纠错，要求每个可纠正地错误图样必须与一个特定的余式有一一对应关系，这样才可能从上述余式中唯一地确定其错误图样，从而纠正错码。如同其他线性分组码，循环码的纠错译码也可以分为三步进行。

① 由接收到的码多项式 $A'(x)$ 计算校正子（伴随式）多项式 $r'(x)$。

对于循环码而言，校正子多项式就是用接收到的码多项式 $A'(x)$ 除以生成多项式 $g(x)$ 所得到的余式，即

$$r'(x) = A'(x) \bmod g(x) \tag{7-64}$$

② 由校正子多项式 $r'(x)$ 确定错误图样 $E(x)$。

③ 将错误图样 $E(x)$ 与接收码多项式 $A'(x)$ 相加，即可纠正错误，恢复原发送码组。

一般而言，纠错译码器的复杂性主要取决于译码过程的第二步。对于纠正突发错误或单个错误的编译码还算简单，如果需要纠正多个错码的编译码却是十分复杂的。下面我们举一个具体例子来说明译码原理。

例 7-10 已知(7, 4)循环码，其生成多项式为 $g(x) = x^3 + x + 1$，说明其能纠正一位错码的译码过程。

解 为了纠正一位错码，错误图样识别电路需要识别一位错码的错误图样就可以了。假设接收码多项式中只有一个错码，则 $E(x)$ 多项式的可能形式为 $x^0, x^1, x^2, x^3, x^4, x^5, x^6$。计算接收到的错码所对应的校正子多项式（除法器中的余式），将它与错误图样一一对应起来列于表 7-12 中。

表 7-12　　错误图样与校正子多项式对应表

校正子多项式（余式）$r(x)$	错误图样 $E(x)$
1	1
x	x
x^2	x^2
$x+1$	x^3
x^2+x	x^4
x^2+x+1	x^5
x^2+1	x^6

由表 7-12 可知，根据余式不同状态可确定发生一个错码的确切位置，从而把它纠正。假定发送码组 $A(x)$ 为 1101001，在传输过程中第 6 位码元发生了错误，则接收码组 $A'(x)$ 变为 1001001。首先计算校正子多项式 $r'(x)$ 为

$$r'(x) = (x^6 + x^3 + 1) \bmod (x^3 + x + 1) = x^2 + x + 1$$

通过表 7-11，可查得此余式所对应的错误图样为：$E(x) = x^5$。

此时恢复接收到的码多项式为

$$A(x) = A'(x) + E(x) = x^6 + x^5 + x^3 + 1 \text{（对应码组 1101001）}$$

这种基于错误图样识别的译码器称为梅吉特译码器，它的原理图如图 7-9 所示。错误图样识别电路是一个具有 $n-k$ 个输入的逻辑电路，原则上采用查表的方法，根据校正子就可找

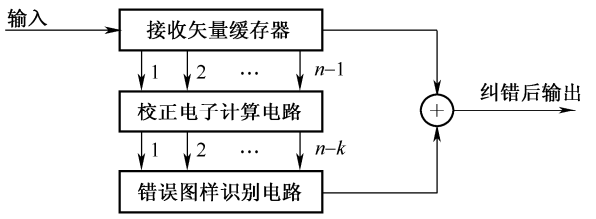


图 7-9　梅吉特译码器原理图

到错误图样，进而纠正错码。此译码方法特别适合于纠正两个及两个以下随机错码的情况。除了上述译码方法之外，循环码的译码还有捕错译码、大数逻辑译码等方法，在此不再一一深究。至于循环码解码原理的详细分析，已超出本书范围，也不再讨论。

7.6 BCH 和 RS 编码以及译码

BCH 和 RS 码是另外两种常用的分组码，也在实际的无线通信中得到了广泛的应用。

7.6.1 BCH 码

1. BCH 码的基本概念

BCH 码是循环码中重要的一类子码，能纠正多个随机错误，它是 1959 年由发现这种码的三个人的名字（Bose-Chaudhuri-Hocquenghem）的缩写来命名的。它的生成多项式 $g(x)$ 与最小码距 $d_{\min}$ 之间有密切的关系，可根据所要求的纠错能力 t，很容易地构造出 BCH 码，且其译码也比较容易实现。由于它具有纠错能力强、构造方便、编码简单、译码也较易实现等一系列优点而被广泛采用。

BCH 码可分为两类：本原 BCH 码和非本原 BCH 码。本原 BCH 码的码长为 $n=2^m-1$（m 为正整数），它的生成多项式 $g(x)$ 中含有最高次数为 m 次的本原多项式；非本原 BCH 码的码长 n 是 2^m-1 的一个因子，它的生成多项式 $g(x)$ 中不含有最高次数为 m 次的本原多项式。

下面我们首先引入本原多项式的概念。若一个 n 次多项式 $f(x)$ 满足下列条件。

① $f(x)$ 是既约的。

② $f(x)$ 可整除 x^n+1，$n=2^m-1$。

③ $f(x)$ 除不尽 x^q+1，$q<n$。

则称 $f(x)$ 为本原多项式。

由前述可知，要确定 (n,k)（$n=2^m-1$）本原循环码是否存在，只需要判断 $r=n-k$ 阶（其常数项不为 0）的生成多项式 $g(x)$ 是否能由 x^n+1 的因式构成。代数理论告诉我们，每个 m 阶既约多项式一定能除尽 x^n+1（$n=2^m-1$ 为其码长）。我们还知道，$x+1$ 必定是 x^n+1 的一个因式。通过计算机分解，表 7-13 列出了阶数为 m（$m<10$）的既约多项式，更高次的既约多项式可从任何一本纠错编码书上查找。

在表 7-13 中，多项式系数均使用 8 进制数；例如，$m=3$ 时，13 意味着 $(13)_8=(1011)_2$，相应的多项式为 $f(x)=x^3+x+1$。多项式系数前的自然数 $1,3,5\cdots$ 是十进制序号，序号为 i 的多项式记作 $m_i(x)$，称为最小多项式；多项式系数后的英文字母的含义如下：A、B、C、D 表示该多项式为非本原多项式；E、F、G、H 则表示其为本原多项式。并且，既约多项式的互反多项式仍是一个既约多项式，因此，表中只列出互反的一对多项式中的一个。以上多项式和其对应的互反多项式，组成了这些阶次的全部既约多项式。换句话说，本原多项式的互反多项式也是一个本原多项式。

例 7-11 对 $x^{31}+1$ 进行因式分解。

解 因为码长 $n=2^m-1=31$，所以 $m=5$，由表 7-13 可得各因式分解如下。

$$m_0(x)=x+1$$
$$m_1(x)=(45)_8=x^5+x^2+1$$
$$m_3(x)=(75)_8=x^5+x^4+x^3+x^2+1$$
$$m_5(x)=(67)_8=x^5+x^4+x^2+x+1$$
$$m_1^*(x)=x^5+x^3+1$$
$$m_3^*(x)=x^5+x^3+x^2+x+1$$
$$m_5^*(x)=x^5+x^4+x^3+x+1$$
$$x^{31}+1=m_0(x)\,m_1(x)\,m_3(x)\,m_5(x)\,m_1^*(x)\,m_3^*(x)\,m_5^*(x)$$

其中，$m_0(x)=x+1$是任意阶x^n+1的必然因子，$m_1^*(x)$、$m_3^*(x)$、$m_5^*(x)$分别是$m_1(x)$、$m_3(x)$、$m_5(x)$的互反多项式。

表 7-13　　部分既约多项式表

2 阶			1	7H				
3 阶			1	13F				
4 阶			1	23F	3	37D	5	07
5 阶			1	45E	3	75G	5	67H
6 阶			1	103F	3	127B	5	147H
	7	111A	9	015	11	155E	21	007
7 阶			1	211E	3	217E	5	235E
	7	367H	9	277E	11	325G	12	203F
	19	313H	21	345G				
8 阶			1	435E	3	567B	5	763D
	7	551E	9	675C	11	747H	13	453F
	15	727D	17	023	19	545E	21	613D
	23	543F	25	433B	27	477B	37	537F
	43	703H	45	471A	51	037	85	007
9 阶			1	1021E	3	1131E	5	1461G
	7	1231A	9	1423G	11	1055E	13	1167F
	15	1541E	17	1333F	19	1605G	21	1027A
	23	1751E	25	1743H	27	1617H	29	1553H
	35	1401C	37	1157F	39	1715E	41	1563H
	43	1713H	45	1175E	51	1725G	53	1225E
	55	1275E	73	0013	75	1773G	77	1511C
	83	1425G	85	1267E				
10 阶			1	2011E	3	2017B	5	2415E
	7	3771G	9	2257B	11	2065A	13	2157F
	15	2653B	17	3515G	19	2773F	21	3753D
	23	2033F	25	2443F	27	3573D	29	2461E
	31	3043D	33	0075C	35	3023H	37	3543F

续表

	39	2107B	41	2745E	43	2431E	45	3061C
	47	3177H	49	3525G	51	2547B	53	2617F
	55	3453D	57	3121C	59	3471G	69	2701A
	71	3323H	73	3507H	75	2437B	77	2413B
	83	3623H	85	2707E	87	2311A	89	2327F
	91	3265G	93	3777D	99	0067	101	2055E
	103	3575G	105	3607C	107	3171G	109	2047F
	147	2355A	149	3025G	155	2251A	165	0051
	171	3315C	173	3337H	179	3211G	341	0007

2．BCH 码的生成

若循环码的生成多项式具有如下形式

$$g(x)=LCM\left[m_1(x),m_3(x),\cdots,m_{2t-1}(x)\right] \tag{7-65}$$

其中，t 为纠错个数，$m_i(x)$ 为最小多项式，LCM 表示取最小公倍数，则由此生成的循环码称为 BCH 码。其最小码距满足 $d_{\min}\geqslant 2t+1$，它可纠正 t 个随机独立差错。由前定义可知，BCH 码的码长为 $n=2^m-1$ 或为 2^m-1 的一个因子，前者称为本原 BCH 码（或狭义 BCH 码），后者称为非本原 BCH 码。

由于 BCH 码是循环码，所以它的编码可用前面讨论过的循环码生成技术来简单地实现。即只要给定生成多项式 $g(x)$，利用下列步骤就可得到具有系统码形式的 BCH 码。

① 首先将消息多项式 $M(x)$ 乘以 x^{n-k}。

② 用生成多项式 $g(x)$ 除 $x^{n-k}\cdot M(x)$，得到余式 $r(x)$。

③ 编出码组 $A(x)=x^{n-k}\cdot M(x)+r(x)$。

下面用例子来说明如何找到 BCH 码的生成多项式。

例 7-12 构造一个能纠正 3 个错码，码长为 15 的 BCH 码。

解 由 $n=2^m-1=15$ 可知，$m=4$，$t=3$。

由式（7-65）可得到该码的生成多项式为

$$g(x)=LCM\left[m_1(x),m_3(x),m_5(x)\right]$$

查既约多项式表（见表 7-13），可得

$$m_1(x)=(23)_8=(010011)_2=x^4+x+1$$

$$m_3(x)=(37)_8=(011111)_2=x^4+x^3+x^2+x+1$$

$$m_5(x)=(07)_8=(000111)_2=x^2+x+1$$

所以

$$\begin{aligned}g(x)&=LCM\left[m_1(x),m_3(x),m_5(x)\right]\\&=(x^4+x+1)\cdot(x^4+x^3+x^2+x+1)\cdot(x^2+x+1)\\&=x^{10}+x^8+x^5+x^4+x^2+x+1\\&=(2\,467)_8\end{aligned}$$

这是一个能纠正 3 位错码的(15, 5)BCH 码。

由上讨论可见，对于能够纠正 t 个错码的本原 BCH 码，其生成多项式为

$$g(x)=m_1(x)\,m_3(x)\cdots m_{2t-1}(x) \tag{7-66}$$

它的最小码距为 $d_{\min}=2t+1$。作为特例，能纠正一位错码的本原 BCH 码，就是循环汉明码。表 7-14 列出了码长不超过 127 的本原 BCH 码。表 7-15 列出了部分非本原 BCH 码。BCH 码的进一步论证说明需要借助于有限域的相关知识，限于篇幅，这里不再进一步讨论。

下面再给出一个特殊的非本原 BCH 码的例子。(23,12)码是一个非本原 BCH 码，又称为戈雷（Golay）码。该码的码距为 7，能纠正 3 个随机独立错误。由表 7-14 可查得其生成多项式为

$$g(x)=(5343)_8=(101011100011)_2$$
$$=x^{11}+x^9+x^7+x^6+x^5+x+1$$

它的互反多项式 $g^*(x)=x^{11}+x^{10}+x^6+x^5+x^4+x^2+1$ 也是生成多项式。它们都是 $(x^{23}+1)$ 的因式，即 $x^{23}+1=(x+1)\,g(x)\,g^*(x)$。

表 7-14　　$n\leqslant 127$ 的本原 BCH 码

n	k	t	$g(x)$（8 进制）
7	4	1	13
15	11	1	23
	7	2	721
	5	3	2467
31	26	1	45
	21	2	3551
	16	3	107657
	11	5	5423325
	6	7	313365047
63	57	1	103
	51	2	12471
	45	3	1701317
	39	4	166623567
	36	5	1033500423
	30	6	157464165547
	24	7	17323260404441
	18	10	1363026512351725
	7	15	5231045543503271737
127	120	1	211
	113	2	41567
	106	3	11554743
	99	4	3447023271
	92	5	624730022327
	85	6	130704476322273
	78	7	26230002166130115

续表

n	k	t	$g(x)$（8 进制）
127	71	9	6255010713253127753
	64	10	1206534025570773100045
	57	11	335265252505705053517721
	50	13	544465125233140124215014 21
	43	14	1772177221365122752122057 4343
	36	15	3146074666522075044764574721735
	29	21	403114461367670603667530141176155
	22	23	123376070404722522435445626637647043
	15	27	220570424456045547705230137622176043 53
	8	31	704726405275103065147622427156773313021 7

表 7-15　　部分非本原 BCH 码

n	k	t	$g(x)$（8 进制）
17	9	2	727
21	12	2	1663
23	12	3	5343
25	5	2	4102041
41	21	4	6647133
47	24	5	43073357
65	53	2	10761

3．BCH 码的译码

BCH 码译码方法大体上分为“频域译码”和“时域译码”两大类。所谓“频域译码”，就是把每个码组看成一个数字信号，将接收到的信号进行离散付氏变换（DFT），然后利用数字信号处理技术在“频域”内译码，最后再进行付氏反变换得到译码后的码组。“时域译码”则是在时域上直接利用码的代数结构进行译码。一旦它的代数基础建立起来，译码器就会很简单。时域译码的方法又有很多，纠正多个错误的 BCH 译码算法很复杂，限于篇幅，本节只简单介绍彼得森译码方法的基本思路。在彼得森译码中，仍然采用计算校正子，然后用校正子寻找错误图样的方法，其译码基本思路如下。

① 用生成多项式 $g(x)$ 的各因式作为除式，对接收到的码多项式求余，得到 t 个余式，称为“部分校正子”或“部分伴随式”。

② 通过下列步骤确定接收多项式中码错误的位置：首先根据“部分校正子”确定错误位置多项式；然后解出多项式的根，由这些根可直接确定接收多项式中错误的位置。

③ 纠正接收多项式中的错误。

至于详细具体的译码过程，限于篇幅，这里不再赘述。

7.6.2 RS 码

RS 码是里德-索洛蒙码的简称，它是一类具有很强纠错能力的多进制 BCH 码。在(n, k)RS 码中，输入信息可分为 $k \cdot m$ 比特一组，每组包括 k 个符号，每个符号由 m 比特组成，而不是二进制 BCH 码中的一个比特。

一个纠正 t 个符号错误的 RS 码具有以下参数。

码长：$n = 2^m - 1$ 符号　　或　m 比特。

信息段：k 符号　　或　k 比特。

监督段：$n - k = 2t$ 符号　　或　$m(n-k) = 2mt$ 比特。

最小码距：$d = 2t + 1$ 符号　　或　$md = m(2t+1)$ 比特。

RS 码具有同时纠正随机错误和突发错误的能力，且特别适合于纠正突发错误。它可纠正以下错误图样：

总长度为 $b_1 = (t-1)m + 1$ 比特的单个突发；

总长度为 $b_2 = (t-3)m + 3$ 比特的两个突发；

…

总长度为 $b_i = (t-2i+1)m + 2i - 1$ 比特的 i 个突发。

对于长度为 $2^m - 1$ 符号的 RS 码，每个符号都可以看成是有限域 $GF(2^m)$ 中的一个元素。且最小距离为 d 符号的 RS 码的生成多项式具有如下形式

$$g(x) = (x+\alpha)(x+\alpha^2)\cdots(x+\alpha^{d-1}) \tag{7-67}$$

这里，α^i 是 $GF(\alpha^m)$ 域中的一个元素。

例 7-13　试构造一个能纠正 3 个错误符号，码长 n=15，m=4 的 RS 码（写出生成多项式）。

解　已知 t=3，n=15，m=4，根据 RS 码参数性质进行构造。

该码码距：$d = 2t + 1 = 2 \times 3 + 1 = 7$ 个符号（$7 \times 4 = 28$bit）。

监督段：$n - k = 2t = 2 \times 3 = 6$ 个符号（$6 \times 4 = 24$bit）。

信息段：$k = n - 6 = 15 - 6 = 9$ 个符号（$9 \times 4 = 36$bit）。

码长：$n = 15$ 个符号（$15 \times 4 = 60$bit）。

因此，该码为$(15, 9)$RS 码（也可看成$(60, 36)$二进制码）。

该码的生成多项式为

$$\begin{aligned} g(x) &= (x+\alpha)(x+\alpha^2)\cdots(x+\alpha^{d-1}) \\ &= (x+\alpha)(x+\alpha^2)\cdots(x+\alpha^6) \\ &= x^6 + \alpha^{10}x^5 + \alpha^{14}x^4 + \alpha^4 x^3 + \alpha^6 x^2 + \alpha^9 x + \alpha^6 \end{aligned}$$

RS 码的编码过程与 BCH 码一样，也是除以 $g(x)$。它同样可以用带反馈的移位寄存器来实现，只不过所有数据通道都是 m 比特宽，即移位寄存器为 m 级并联工作。每个反馈连接必须乘以生成多项式中相应的系数 α^i。编码器示意图如图 7-10 所示。其中与 α^i 的相乘可以用 $2^m \times m$ ROM（只读存储器）查表法实现。

RS 码的译码过程也大体上与纠正 t 个错误的彼得森译码法相似。所不同的是，需要在找到错误位置后，求出错误值。因为 BCH 译码时只有一个错误值“1”，而 RS 译码则有 $2^m - 1$ 种可能值。具体来说，其译码步骤如下。

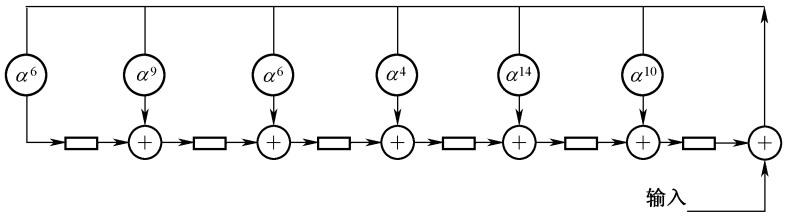


图 7-10 (15,9)RS 码编码器示意图

① 计算校正子。
② 确定错误位置多项式。
③ 寻找错误位置。
④ 求出错误值。
⑤ 纠正错误。

有关 RS 码译码原理的详细讨论，有兴趣的读者可参阅相关专著。

7.7 线性分组码的应用

循环码由于性能优异，而且实现较为简单，因此在移动通信中有广泛的应用。本节主要介绍循环码在 GSM 中的应用。

GSM 的逻辑信道主要分为两大类，如图 7-11 所示。

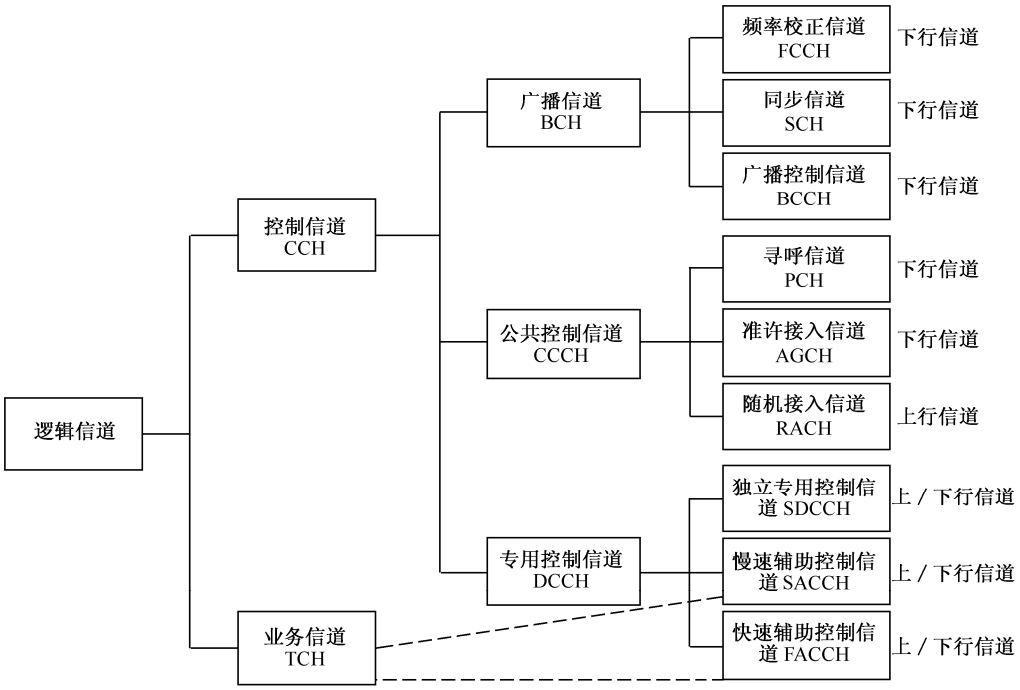


图 7-11 GSM 逻辑信道

① 业务信道。业务信道（TCH）主要传输数字话音或数据，其次还有少量的随路控制信令。

- 话音业务信道，包括全速率（FS）、增强型全速率（EFR）等。
- 数据业务信道，包括全速率、半速率等。

② 控制信道。控制信道（CCH）用于传送信令和同步信号，主要分为 3 种：广播信道（BCH）、公共控制信道（CCCH）和专用控制信道（DCCH）。

循环码在 GSM 中使用的较为普遍，各种信道几乎都使用了循环码，具体如表 7-16 所示。

表 7-16　　GSM 信道中使用的循环码

信息内容	逻辑信道	信息长度	CRC 长度	生成多项式
Class I a 类语音	TCH/FS	50	3	D^3+D+1
信令和控制	控制信道（SACCH，FACCH，BCCH，PCH，AGCH，CBCH）	184	40	（$D^{23}+1$）（$D^{17}+D^3+1$） 法尔码，有可能纠错（Fire code）
接入（RACH）	RACH	8	6	$D^6+D^5+D^3+D^2+D+1$
同步（SCH）	SCH	25	14	（D^5+1）（D^7+1）（$D+1$）^{2}

除了 GSM 信道外，其他的数字通信系统也使用了循环码，具体可以参考相关文献。

在其他的系统中，循环码的使用也很普遍，下面的 3 中结构形式的循环码比较常见。

$$\text{CRC-12}=x^{12}+x^{11}+x^3+x^2+x+1$$
$$\text{CRC-16}=x^{16}+x^{15}+x^2+1$$
$$\text{CRC-CCITT}=x^{16}+x^{12}+x^5+1$$

习　　题

7.1　在通信系统中，采用差错控制的目的是什么？常用的差错控制方法有哪些？

7.2　码的最小码距与其检、纠错能力有何关系？

7.3　什么是线性分组码？其结构特点如何？

7.4　设某二元码为 $C=\{11100，01001，10010，00111\}$，

（1）计算此码的最小距离 $d_{\min}$。

（2）计算此码的码率 R，假设码字等概率分布。

（3）采用最小距离译码准则，试问接收序列 10000，01100 和 00100 应译成什么码字？

（4）此码能纠正几位码元的错误？

7.5　已知(7, 4)码的生成矩阵为

$$\boldsymbol{G}=\begin{bmatrix}1&0&0&0&1&1&1\\0&1&0&0&1&0&1\\0&0&1&0&0&1&1\\0&0&0&1&1&1&0\end{bmatrix}$$

试求解下述问题。

（1）监督矩阵 $\boldsymbol{H}$？

（2）写出所有许用码组？

（3）若接收码组为 1101101，计算伴随式（校正子）？

7.6　已知某线性分组码的监督矩阵为

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

试求解下述问题。

（1）生成矩阵 $\boldsymbol{G}$？

（2）写出所有许用码组？

（3）当输入信息序列为 11011011 时，求输出码序列？

7.7　什么是循环码？如何确定其生成多项式？

7.8　已知 $x^{15}+1=(x+1)(x^4+x+1)(x^4+x^3+1)(x^4+x^3+x^2+x+1)(x^2+x+1)$，试问由它共可构成多少种码长为 15 的循环码？列出它们的生成多项式。

7.9　已知(7, 3)循环码生成多项式为

$$g(x)=1+x^2+x^3+x^4$$

试求解下述问题。

（1）监督多项式。

（2）生成矩阵、监督矩阵。

（3）当输入信息序列为 110 时，求输出码序列。

7.10　什么是 BCH 码？什么是本原 BCH 码？什么是非本原 BCH 码？

7.11　构造一个码长为 63，能纠正 3 个错误的 BCH 码，写出它的生成多项式。

第 8 章 卷积码

内容简介

卷积码是1955年Elias最早提出的，后来Wozencraft和Massey等人分别于1957年和1963年系统地给出了卷积码的矩阵和多项式表示，Forney 于 1970 年从网络的观点系统地分析了卷积码的特点。卷积码的理论研究不像分组码那样有完美、系统的理论支撑，没有系统地像分组码那样的构造方法，一般来说，目前性能好的卷积码都是通过计算机搜索得到的。

另一方面，卷积码的译码理论和方法发展得很快。Wozencraft 在 1957 年提出了序列的译码法，Reffen 于 1960 年对该算法进行了改进，Fano 于 1960 年给出了更有效的算法。1963 年 Massey 提出比较实用的门限译码法（大数逻辑译码），该方法属于代数译码。Viterbi 于 1967 年提出了一种极大似然译码算法，被称作为 Viterbi 算法，该算法属于概率译码。此外，Bussgang1965 年系统地研究了卷积码的反馈译码算法。

卷积码在性能上优于分组码，和分组码相比卷积码的编译码容易实现，特别是其译码极大地降低了实现复杂度，所以在实际中得到广泛的应用。

本章将主要介绍卷积码的基本概念、译码方法以及性能界的分析，最后将涉及级联码的简单介绍。

本章的内容都是基于GF(2)的，所得到的结果可以推广到GF(q)域的一般情况，即 q>2。

8.1 卷积码的基本概念

分组码是把 k 个信息比特的序列编成 n 个比特的码组，每个码组的 $n-k$ 个校验位仅与本码组的 k 个信息位有关，而与其他码组的信息比特无关。为了达到一定的纠错能力和编码效率，分组码的码组长度一般都比较大。编译码时必须把整个信息码组存储起来，由此产生的译码时延随 n 的增加而迅速增加。我们可以将分组码的编码思想加以推广，使校验位与更多的信息比特有关，这样就会起到更好的校验作用，使码的性能更好，这就是卷积码的基本想法。理论上已经证明，卷积码的性能界要优于分组码。

卷积码也是将 k 个信息比特编成 n 个比特，但 k 和 n 通常很小，特别适合以串行形式进行传输，时延小。与分组码不同，卷积码编码后的 n 个码元不仅与当前段的 k 个信息有关，还与前面的 $m-1$ 段信息有关，以获取高性能。

卷积码编码器的一般框图如图 8-1 所示。输入的信息序列被分成长度为 k 的段，并经过

串并转换输入到离散线性系统的 k 个输入端。该系统的输出端为 n 个（一般 $n>k$），且系统最大延时为 m。输出的 n 个编码数字经过并串转换送入信道就完成了编码过程，这就是可表示为(n, k, m)的典型卷积码。一般选 n 和 k 较小，但 m 值较大（m<10 左右）。

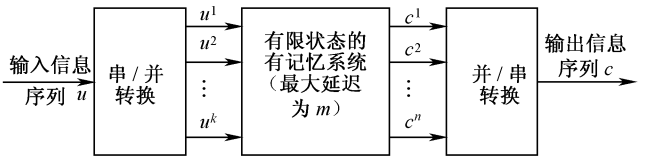


图 8-1 卷积码编码器一般框图

卷积码的纠错性能随 m 的增加而增大，而差错率随 m 的增加而指数下降。在编码器复杂性相当的情况下，卷积码的性能优于分组码。但卷积码没有分组码那样严密的数学分析手段，目前大多是通过计算机进行好码的搜索。

注：卷积码编码器是由一个有 k 个输入端口、n 个输出端且具有 m 节移位寄存器所构成的有限状态有记忆系统,通常也称它为时序网络。

描述卷积码的方法很多，它大致可划分为两大类：解析法和图形法。解析法包括离散卷积法、矩阵生成法以及码多项式法等，主要用于编码部分的描述；图形法包括状态图法、树图法以及网格图法等，主要用于译码的描述和说明。

8.2 卷积码的编码

下面我们用两个具体例子来说明二元域上的卷积码的编码过程以及有关描述。

例 8-1 图 8-2 给出一个(2,1,4)卷积码编码器结构，此时 $n=2$，$k=1$，$m=4$，编码速率 $R=\frac{k}{n}=\frac{1}{2}$，约束长度 $m=4$。求该卷积码编码器的输出。

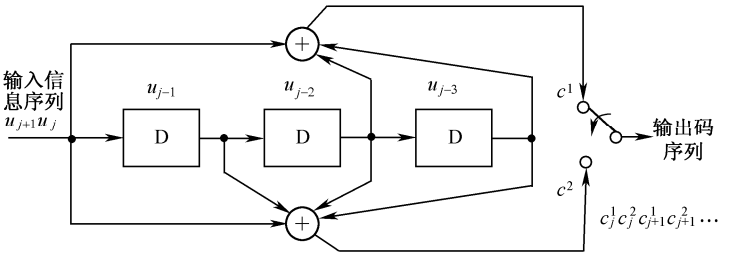


图 8-2 （2,1,4）卷积码编码器结构

解：该编码器由 3 个移位寄存器（D）和两个模 2 加法器组成，每输入一个码元就会产生 2 个输出，输出端第 j 时刻的码元分别由下式确定。

$$\begin{cases} c_j^1 = u_j \oplus u_{j-2} \oplus u_{j-3} \\ c_j^2 = u_j \oplus u_{j-1} \oplus u_{j-2} \oplus u_{j-3} \end{cases} \tag{8-1}$$

假设输出信息序列为

$$u = (u_0 u_1 u_2 \cdots) \tag{8-2}$$

则对应输出两码组为

$$\begin{cases} c^1 = (c_0^1 c_1^1 c_2^1 \cdots) \\ c^2 = (c_0^2 c_1^2 c_2^2 \cdots) \end{cases}, \tag{8-3}$$

最终的编码输出为

$$c = c_0^1 c_0^2 c_1^1 c_1^2 c_2^1 c_2^2 \cdots \tag{8-4}$$

我们假设输入为 $u=(10111)$，编码开始前先对移位寄存器进行复位（即置 0），输入的顺序为 10111，则结合式（8-1）、（8-2）、（8-3）编码的过程如下。

① 输入 $u_j=1$，$c_j^1 = u_j \oplus u_{j-2} \oplus u_{j-3}$=$1\oplus 0\oplus 0=1$，

$c_j^2 = u_j \oplus u_{j-1} \oplus u_{j-2} \oplus u_{j-3}$= $1\oplus 0\oplus 0\oplus 0=1$，编码输出 11

② 输入 $u_{j+1}=0$，$c_{j+1}^1 = u_{j+1} \oplus u_{j-1} \oplus u_{j-2}$=$0\oplus 0\oplus 0=0$，

$c_{j+1}^2 = u_{j+1} \oplus u_j \oplus u_{j-1} \oplus u_{j-2}$= $0\oplus 1\oplus 0\oplus 0=1$，编码输出 01

③ 输入 $u_{j+2}=1$，$c_{j+2}^1 = u_{j+2} \oplus u_j \oplus u_{j-1}$=$1\oplus 1\oplus 0=0$，

$c_{j+2}^2 = u_{j+2} \oplus u_{j+1} \oplus u_j \oplus u_{j-1}$= $1\oplus 0\oplus 1\oplus 0=0$，编码输出 00

④ 输入 $u_{j+3}=1$，$c_{j+3}^1 = u_{j+3} \oplus u_{j+1} \oplus u_j$=$1\oplus 0\oplus 1=0$，

$c_{j+3}^2 = u_{j+3} \oplus u_{j+2} \oplus u_{j+1} \oplus u_j$= $1\oplus 1\oplus 0\oplus 1=1$，编码输出 01

⑤ 输入 $u_{j+4}=1$，$c_{j+4}^1 = u_{j+4} \oplus u_{j+2} \oplus u_{j+1}$=$1\oplus 1\oplus 0=0$

$c_{j+4}^2 = u_{j+4} \oplus u_{j+3} \oplus u_{j+2} \oplus u_{j+1}$= $1\oplus 1\oplus 1\oplus 0=1$，编码输出 01

所以编码器输出为：1101000101，由于是 1/2 码率，所以共有 10 位输出。

如果考虑到信息序列输入完移位寄存器的清零，即增加 3 位 0 输入，所以还有以下 3 步。

⑥ 输入 $u_{j+5}=0$，$c_{j+5}^1 = u_{j+5} \oplus u_{j+3} \oplus u_{j+2}$=$0\oplus 1\oplus 1=0$

$c_{j+5}^2 = u_{j+5} \oplus u_{j+4} \oplus u_{j+3} \oplus u_{j+2}$= $0\oplus 1\oplus 1\oplus 1=1$，编码输出 01

⑦ 输入 $u_{j+6}=0$，$c_{j+6}^1 = u_{j+6} \oplus u_{j+4} \oplus u_{j+3}$=$0\oplus 1\oplus 1=0$

$c_{j+6}^2 = u_{j+6} \oplus u_{j+5} \oplus u_{j+4} \oplus u_{j+3}$= $0\oplus 0\oplus 1\oplus 1=0$，编码输出 00

⑧ 输入 $u_{j+7}=0$，$c_{j+7}^1 = u_{j+7} \oplus u_{j+5} \oplus u_{j+4}$=$0\oplus 0\oplus 1=1$

$c_{j+7}^2 = u_{j+7} \oplus u_{j+6} \oplus u_{j+5} \oplus u_{j+4}$= $0\oplus 0\oplus 0\oplus 1=1$，编码输出 11

故加清零的码元后，编码器最终输出为：1101000101010011。

8.2.1 解析法中的码多项式法描述

类似于第 7 章的表示，我们可以将输入的序列对应写成多项式的形式。

$$\boldsymbol{u} = (u_0 u_1 u_2 u_3 \cdots) \leftrightarrow u(x) = u_0 + u_1 x + u_2 x^2 + u_3 x^3 + \cdots \tag{8-5}$$

所以有

$$\boldsymbol{u} = (10111) \leftrightarrow u(x) = 1 + x^2 + x^3 + x^4 \tag{8-6}$$

在分析中，我们可以用

$$\boldsymbol{g}^{1,n} = (g_0^{1,n} g_1^{1,n} g_2^{1,n} \cdots g_m^{1,n}), \boldsymbol{g}^{2,n} = (g_0^{2,n} g_1^{2,n} g_2^{2,n} \cdots g_m^{2,n}), \cdots, \boldsymbol{g}^{k,n} = (g_0^{k,n} g_1^{k,n} g_2^{k,n} \cdots g_m^{k,n}) \tag{8-7}$$

来表示第 k 个输入端在输出端 $\boldsymbol{C}^1$、$\boldsymbol{C}^2$、…、$\boldsymbol{C}^n$ 的求和式的系数（也是第 k 个输入端在 n 个

输出端的脉冲冲击响应，$1 \leqslant K \leqslant n$），则对于图 8-2 所示结构的卷积码编码器，$k=1$，故为了书写简便起见，可以忽略 k 的角标，所以有

$$\boldsymbol{g}^1=(1011), \boldsymbol{g}^2=(1111)$$

物理意义如图 8-3 所示，分别由反馈系数构成（如果和虚线有交点则取 1，否则取 0）。

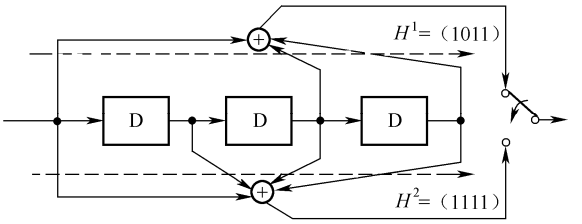


图 8-3 （2，1，4）卷积码的多项式描述系数

类似式（8-5）对应的多项式为

$$g^1(x)=1+x^2+x^3, g^2(x)=1+x+x^2+x^3$$

编码后多项式为（乘积后合并也是模 2，即有 $x^n+x^n=0, x^n+x^n+x^n=x^n$）

$$\begin{cases} c^1(x)=u(x)\times g^1(x) \\ c^2(x)=u(x)\times g^2(x) \end{cases} \tag{8-8}$$

所以有

$$\begin{aligned} c^1(x)=u(x)\times g^1(x)&=(1+x^2+x^3+x^4)(1+x^2+x^3) \\ &=1+x^2+x^3+x^4+x^2+x^4+x^5+x^6+x^3+x^5+x^6+x^7 \\ &=1+x^7 \end{aligned}$$

$$\begin{aligned} c^2(x)=u(x)\times g^2(x)&=(1+x^2+x^3+x^4)(1+x+x^2+x^3) \\ &=1+x^2+x^3+x^4+x+x^3+x^4+x^5+x^2+x^4+x^5+x^6+x^3+x^5+x^6+x^7 \\ &=1+x+x^3+x^4+x^5+x^7 \end{aligned}$$

其中用到模 2 合并，即有

$$x^2+x^2=0, x^3+x^3+x^3=x^3, x^4+x^4+x^4=x^4$$

$$x^5+x^5+x^5=x^5, x^6+x^6=0$$

故对应的码元为

$$\begin{cases} \boldsymbol{c}^1=(10000001) \\ \boldsymbol{c}^2=(11011101) \end{cases}$$

根据式（8-4）的规则，则编码器的输出为：1101000101010011。

8.2.2 矩阵生成法描述

令

$$\boldsymbol{G}^k=\begin{bmatrix} \boldsymbol{g}^{k,1} \\ \boldsymbol{g}^{k,2} \\ \vdots \\ \boldsymbol{g}^{k,n} \end{bmatrix}=[\boldsymbol{G}_0^k, \boldsymbol{G}_1^k, \cdots, \boldsymbol{G}_m^k] \tag{8-9}$$

上式是第 k 个输入端的脉冲相应矩阵，为一个 $n \times m$ 的矩阵。

再令

$$\boldsymbol{G}_\alpha = [\boldsymbol{G}_\alpha^1, \boldsymbol{G}_\alpha^2, \cdots, \boldsymbol{G}_\alpha^k]^{\mathrm{T}} \qquad \text{其中} 0 \leqslant \alpha \leqslant m \tag{8-10}$$

此时，可以得到一个卷积码的半无穷生成矩阵 $\boldsymbol{G}_\infty$。

$$\boldsymbol{G}_\infty = \begin{bmatrix} \boldsymbol{G}_0 & \boldsymbol{G}_1 & \cdots & \boldsymbol{G}_{\mathrm{m}} & & & & \\ & \boldsymbol{G}_0 & \boldsymbol{G}_1 & \cdots & \boldsymbol{G}_{\mathrm{m}} & & & \\ & & \boldsymbol{G}_0 & \boldsymbol{G}_1 & \cdots & \boldsymbol{G}_{\mathrm{m}} & & \\ & & & \boldsymbol{G}_0 & \boldsymbol{G}_1 & \cdots & \boldsymbol{G}_{\mathrm{m}} & \\ & & & & \ddots & \ddots & \ddots & \ddots \end{bmatrix} \tag{8-11}$$

则编码器输出可以由下式来确定。

$$\boldsymbol{c} = \boldsymbol{u}\boldsymbol{G} \tag{8-12}$$

对于本例来说，$k=1$，$n=2$，$m=4$，所以由式（8-9）可知

$$\boldsymbol{G}^1 = \begin{pmatrix} 1011 \\ 1111 \end{pmatrix} = [\boldsymbol{G}_0^1, \boldsymbol{G}_1^1, \boldsymbol{G}_2^1, \boldsymbol{G}_3^1]$$

$$\boldsymbol{G}_0 = [\boldsymbol{G}_0^1]^{\mathrm{T}} = (11), \boldsymbol{G}_1 = [\boldsymbol{G}_1^1]^{\mathrm{T}} = (01), \boldsymbol{G}_2 = [\boldsymbol{G}_2^1]^{\mathrm{T}} = (11), \boldsymbol{G}_3 = [\boldsymbol{G}_3^1]^{\mathrm{T}} = (11)$$

所以

$$\boldsymbol{G} = \begin{bmatrix} 11 & 01 & 11 & 11 & & & & \\ & 11 & 01 & 11 & 11 & & & \\ & & 11 & 01 & 11 & 11 & & \\ & & & 11 & 01 & 11 & 11 & \\ & & & & 11 & 01 & 11 & 11 \end{bmatrix}$$

而输入 $\boldsymbol{U} = (10111)$，所以有

$$\boldsymbol{c} = \boldsymbol{u}\boldsymbol{G} = [10111] \begin{bmatrix} 11 & 01 & 11 & 11 & & & & \\ & 11 & 01 & 11 & 11 & & & \\ & & 11 & 01 & 11 & 11 & & \\ & & & 11 & 01 & 11 & 11 & \\ & & & & 11 & 01 & 11 & 11 \end{bmatrix}$$

$$= (11010001010010011)$$

结果和前面的相同。

也可以直接根据卷积码编码器结构写出式（8-10）中的子矩阵 $\boldsymbol{G}_\alpha$，具体见图 8-4，子矩阵元素分别由每一级移位寄存器的反馈系数构成（如果和虚线有交点则取 1，交点用♥表示；否则取 0）。

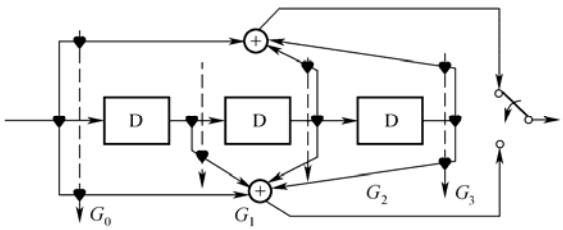


图 8-4 (2,1,4)卷积码的矩阵描述方式

所以有：$\boldsymbol{G}_0=(11)$，$\boldsymbol{G}_1=(01)$，$\boldsymbol{G}_2=(11)$，$\boldsymbol{G}_3=(11)$，可以构成式（8-11）的半无限矩阵，下面的计算相同。

8.2.3 离散卷积法描述

此例对应的编码方程为：$\begin{cases} c^1=u*g^1 \\ c^2=u*g^2 \end{cases}$

其中“*”表示卷积运算，故卷积码因此而得名。而 g^1、g^2 表示编码的两个脉冲冲击响应。

由上面的分析知：$g^1=(1011)$，$g^2=(1111)$，且 $u=(10111)$。

最后可求得

$$c^1=u*g^1=(10111)*(1011)=(10000001)$$

$$c^2=u*g^2=(10111)*(1111)=(11011101)$$

则编码器的输出为：1101000101010011。

离散卷积的有关知识在信号与系统中有详细的介绍，感兴趣的读者可参见有关书中的介绍。

例 8-2 图 8-5 所示是一个(3, 2, 2)卷积码编码器结构，即有 $n=3$，$k=2$，$m=2$。编码器的编码速率 $R=\dfrac{2}{3}$，假设输入为 $\boldsymbol{u}=(110110)$，求编码器的输出。

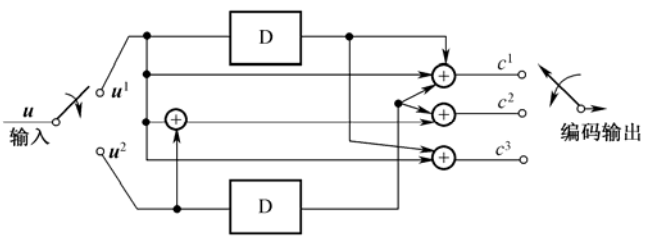


图 8-5 一种（3，2，2）卷积码编码器结构

本例的编码器有 2 个输入端，所以分析会更复杂一些，但是分析过程更具有代表性。

（1）矩阵生成法。根据式（8-9）有

$$\boldsymbol{G}^1=\begin{bmatrix} \boldsymbol{g}^{1,1} \\ \boldsymbol{g}^{1,2} \\ \boldsymbol{g}^{1,3} \end{bmatrix}=\begin{bmatrix} 11 \\ 10 \\ 11 \end{bmatrix}=[\boldsymbol{G}_0^1 \quad \boldsymbol{G}_1^1]$$

$$\boldsymbol{G}^2=\begin{bmatrix} \boldsymbol{g}^{2,1} \\ \boldsymbol{g}^{2,2} \\ \boldsymbol{g}^{2,3} \end{bmatrix}=\begin{bmatrix} 01 \\ 11 \\ 00 \end{bmatrix}=[\boldsymbol{G}_0^2 \quad \boldsymbol{G}_1^2]$$

再根据式（8-10）有

$$\boldsymbol{G}_0=[\boldsymbol{G}_0^1 \ \ \boldsymbol{G}_0^2]^{\mathrm{T}}=\begin{bmatrix} 111 \\ 010 \end{bmatrix}$$

$$\boldsymbol{G}_1=[\boldsymbol{G}_1^1 \ \ \boldsymbol{G}_1^2]^{\mathrm{T}}=\begin{bmatrix} 101 \\ 110 \end{bmatrix}$$

当然也可以直接根据图 8-6 写出子生成矩阵 $\boldsymbol{G}_0$ 和 $\boldsymbol{G}_1$，即 $\boldsymbol{G}_0$ 表示没有延迟的时候 k 个输入端对 n 个输出端的系数矩阵，即为一个 $k \times n$ 矩阵；$\boldsymbol{G}_1$ 表示有一级延迟的时候 k 个输入端对 n 个输出端的系数矩阵，即为一个 $k \times n$ 矩阵。更一般的情况下，则可以类似地写出 $\boldsymbol{G}_2, \boldsymbol{G}_3, \cdots$ 等。

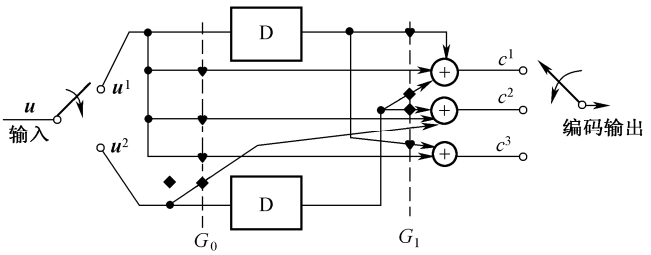


图 8-6 （3，2，2）卷积码子生成矩阵的示意图

对于此例，由于 $k=2$，所以 $\boldsymbol{G}_0$ 和 $\boldsymbol{G}_1$ 子矩阵的第一行为第一个输入 $\boldsymbol{u}^1$ 对 3 个输出的反馈系数，对于 $\boldsymbol{G}_0$ 则为 111，用♥表示；第二行为第二个输入 $\boldsymbol{u}^2$ 对 3 个输出的反馈系数，对于 $\boldsymbol{G}_0$ 则为 010，用♦表示，故可以写出 $\boldsymbol{G}_0 = \begin{bmatrix} 111 \\ 010 \end{bmatrix}$，同理可以写出 $\boldsymbol{G}_1 = \begin{bmatrix} 101 \\ 110 \end{bmatrix}$。所以有

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{G}_0 & \boldsymbol{G}_1 & & \\ & \boldsymbol{G}_0 & \boldsymbol{G}_1 & \\ & & \boldsymbol{G}_0 & \boldsymbol{G}_1 \end{bmatrix} = \begin{bmatrix} 111 & 101 & & \\ 010 & 110 & & \\ & 111 & 101 & \\ & 010 & 110 & \\ & & 111 & 101 \\ & & 010 & 110 \end{bmatrix}$$

故

$$\boldsymbol{C} = \boldsymbol{U}\boldsymbol{G} = [110110] \begin{bmatrix} 111 & 101 & & \\ 010 & 110 & & \\ & 111 & 101 & \\ & 010 & 110 & \\ & & 111 & 101 \\ & & 010 & 110 \end{bmatrix}$$

$$= [101001001101]$$

（2）多项式生成法。因为输入序列 $\boldsymbol{u} = (110110)$，所以经过串并转换可以得到 2 个输入子序列。

$$\boldsymbol{u}^1 = (101), \boldsymbol{u}^2 = (110)$$

所以得到输入的多项式形式

$$\begin{cases} \boldsymbol{u}^1 = (101) \leftrightarrow u^1(x) = 1 + x^2 \\ \boldsymbol{u}^2 = (110) \leftrightarrow u^2(x) = 1 + x \end{cases}$$

由式（8-7）可得

$$\begin{cases} \boldsymbol{g}^{1,1}=(11)\leftrightarrow g^{1,1}(x)=1+x \\ \boldsymbol{g}^{1,2}=(10)\leftrightarrow g^{1,2}(x)=1 \\ \boldsymbol{g}^{1,3}=(11)\leftrightarrow g^{1,3}(x)=1+x \\ \boldsymbol{g}^{2,1}=(01)\leftrightarrow g^{2,1}(x)=x \\ \boldsymbol{g}^{2,2}=(11)\leftrightarrow g^{2,2}(x)=1+x \\ \boldsymbol{g}^{2,3}=(00)\leftrightarrow g^{2,3}(x)=0 \end{cases}$$

$$\boldsymbol{G}(x)=\begin{bmatrix} g^{1,1}(x) & g^{1,2}(x) & g^{1,3}(x) \\ g^{2,1}(x) & g^{2,2}(x) & g^{2,3}(x) \end{bmatrix}=\begin{bmatrix} 1+x & 1 & 1+x \\ x & 1+x & 0 \end{bmatrix} \tag{8-13}$$

$$\begin{aligned} \boldsymbol{c}(x)&=\begin{bmatrix} u^1(x) \\ u^2(x) \end{bmatrix}^{\mathrm{T}}\boldsymbol{G}(x) \\ &=[1+x^2 \quad 1+x]\begin{bmatrix} 1+x & 1 & 1+x \\ x & 1+x & 0 \end{bmatrix} \\ &=[1+x^3 \quad 0 \quad 1+x+x^2+x^3] \end{aligned}$$

所以有

$$\begin{cases} c^1(x)=1+x^3 & \boldsymbol{c}^1=(1001) \\ c^2(x)=0 & \leftrightarrow \boldsymbol{c}^2=(0000) \\ c^3(x)=1+x+x^2+x^3 & \boldsymbol{c}^3=(1111) \end{cases}$$

卷积码编码器的输出 $\boldsymbol{c}=(101001001101)$，其最终的输出结果和矩阵法输出相同。

对于一个（n,k,m）的卷积码来说，**码速率**的定义是 $R=k/n$，卷积码的**约束长度**的定义是 m（注：也有的文献定义约束长度为 $m-1$ 或者 $n_A=mn$ ）。

由式（8-13）给出 $G(x)$ 的卷积码的码生成多项式，对于一个卷积码来说，给定的了 $G(x)$，则其编码结构就确定了。对于(n,k,m)的卷积码，一般通用的码多项式由下面的公式构成。

$$G(x)=\begin{bmatrix} g^{1,1}(x) & g^{1,2}(x) & \cdots & g^{1,n}(x) \\ g^{2,1}(x) & g^{2,2}(x) & \cdots & g^{2,n}(x) \\ \vdots & \vdots & \vdots & \vdots \\ g^{k,1}(x) & g^{k,2}(x) & \cdots & g^{k,n}(x) \end{bmatrix} \tag{8-14}$$

在卷积码的家族中，还有一种特殊的编码，就是系统卷积码。它在后面的 Turbo 编码中有着重要的作用。系统卷积码输入端的信息序列直接输出到输出端，它们之间有如下的关系。

$$c^i(x)=u^i(x), i=1,2,\cdots,k$$

所以它的生成生成多项式矩阵为

$$
\begin{aligned}
G(x)&=\begin{bmatrix}
1 & 0 & \cdots & 0 & g^{1,k+1}(x) & \cdots & g^{1,n}(x)\\
0 & 1 & \cdots & 0 & g^{2,k+1}(x) & \cdots & g^{2,n}(x)\\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\
0 & 0 & \cdots & 1 & g^{k,k+1}(x) & \cdots & g^{k,n}(x)
\end{bmatrix}\\
&=\left[\begin{array}{c|ccc}
 & g^{1,k+1}(x) & \cdots & g^{1,n}(x)\\
 & g^{2,k+1}(x) & \cdots & g^{2,n}(x)\\
\boldsymbol{I}_k & \vdots & \ddots & \vdots\\
 & g^{k,k+1}(x) & \cdots & g^{3,n}(x)
\end{array}\right]\\
&=\left[\boldsymbol{I}_k \mid \boldsymbol{G}_{k,n-k}\right]
\end{aligned}
\tag{8-15}
$$

其中，$\boldsymbol{I}_k$ 是一个 k 的单位阵，这和系统分组码类似，系统卷积码的输入信息序列不变化地置于输出序列的某个位置。

当 $m=1$ 的时候，卷积码就可以看做是一个分组码，此时编码系统就是一个无记忆系统。

类似于分组码，对每个给定的 $k\times n$ 阶多项式生成矩阵 $\boldsymbol{G}(x)$，存在有 $(n-k)\times n$ 阶多项式矩阵 $\boldsymbol{H}(x)$，使得

$$\boldsymbol{G}(x)\boldsymbol{H}^{\mathrm{T}}(x)=[\boldsymbol{0}] \tag{8-16}$$

成立。其中 $[\boldsymbol{0}]$ 是一个 $k\times(n-k)$ 阶 0 矩阵。

由 $\boldsymbol{G}(x)$ 生成的（n，k，m）卷积码中的每个码多项式 $\boldsymbol{c}(x)$ 满足下面的关系式

$$\boldsymbol{c}(x)\boldsymbol{H}^{\mathrm{T}}(x)=[\boldsymbol{0}] \tag{8-17}$$

其中 $[\boldsymbol{0}]$ 是一个 $1\times(n-k)$ 阶 0 矩阵，即 $\boldsymbol{c}(x)$ 是 $\boldsymbol{H}(x)$ 零化空间中的元素。$\boldsymbol{H}(x)$ 被称作由 $\boldsymbol{G}(x)$ 生成的卷积码的一致校验多项式，而且由 $\boldsymbol{H}(x)$ 生成的（n，$n-k$，m）卷积码是 $\boldsymbol{G}(x)$ 生成的(n, k, m)卷积码的对偶码。

类似于分组码，对于系统码，$\boldsymbol{G}(x)$ 和 $\boldsymbol{H}(x)$ 是对应的，给出任何一个都可以求出另外一个。

由式（8-15）可写出系统卷积码的 $\boldsymbol{H}(x)$ 为

$$
\begin{aligned}
\boldsymbol{H}(x)&=\left[\begin{array}{cccc|c}
h^{1,1}(x) & h^{1,2}(x) & \cdots & h^{1,k}(x) & \\
h^{2,1}(x) & h^{2,2}(x) & \cdots & h^{2,k}(x) & \\
\vdots & \vdots & \ddots & \vdots & \boldsymbol{I}_{n-k}\\
h^{n-k,1}(x) & h^{n-k,2}(x) & \cdots & h^{n-k,k}(x) &
\end{array}\right]\\
&=\left[\boldsymbol{H}_{n-k,k} \mid \boldsymbol{I}_{n-k}\right]\\
&=\left[\boldsymbol{G}_{k,n-k}^{T} \mid \boldsymbol{I}_{n-k}\right]
\end{aligned}
\tag{8-18}
$$

除了上面的几种描述方法，卷积码还可以用**状态转移图**、**树图**以及**格图**等方法进行描述。

8.2.4 卷积码的图形描述法

下面讨论图形表示法，首先从**状态图**入手。

例 8-3 设(2,1,3)卷积码的编码器结构如图 8-7 所示，画出它的状态转移图、树图以及格图。

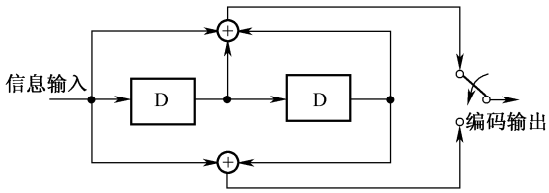


图 8-7　一种（2,1,3）编码器结构

解：

1．状态转移图

- 对于(2,1,3)卷积码：$k=1$，$n=2$，$m=3$，其总状态数为 $2^{k(m-1)}=2^{1\times 2}=4$ 个，即 $a=00$，$b=10$，$c=01$，$d=11$。每个状态的可能输入有 $2^k=2^1=2$ 个，分别用 0 和 1 表示。
- 若输入序列为 $u=$（1011100…），其状态图可以按以下步骤画出。[对照图 8-7 的(2,1,3）编码器结构图。]

具体的有关状态信息为：

① 首先，移位寄存器复 0，其状态为 00；

② 输入 $u_0=1$，状态改为 10，输出 $c_0^{\,1}=1$，$c_0^{\,2}=1$，$c_0=$（11）；

③ 输入 $u_1=0$,状态改为 01,输出 $c_1^{\,1}=1$, $c_1^{\,2}=0$,求得 $c_1=$（10）；

④ 输入 $u_2=1$,状态改为 10，输出 $c_2^{\,1}=0$，$c_2^{\,2}=0$,求得 $c_2=$（00）；

⑤ 输入 $u_3=1$,状态改为 11，输出 $c_3^{\,1}=0$，$c_3^{\,2}=1$，求得 $c_3=$（01）；

⑥ 输入 $u_4=1$,状态仍为 11，输出 $c_4^{\,1}=1$，$c_4^{\,2}=0$，求得 $c_4=$（10）；

⑦ 输入 $u_5=0$,状态改为 01，输出 $c_5^{\,1}=0$，$c_5^{\,2}=1$，求得 $c_5=$（01）；

⑧ 输入 $u_6=0$,状态改为 00，输出 $c_6^{\,1}=1$, $c_6^{\,2}=1$，求得 $c_6=$（11）；

⑨ 输入 $u_7=0$,状态仍为 00，输出 $c_7^{\,1}=0$，$c_7^{\,2}=0$，求得 $c_7=$（00）。

按以上步骤可画出如图 8-8 所示的状态图。

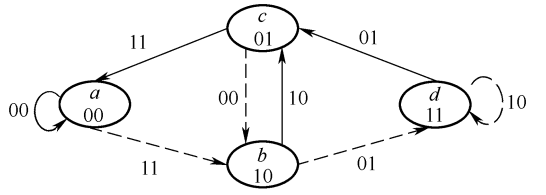


图 8-8　例 8-3 的状态转移图

注：①图中的椭圆内的值为移位寄存器的状态值。

②实线表示输入值为 0 的情况，虚线表示输入值为 1。

③线上的值表示编码器的输出值。

2．树图

树图结构是由状态图按时间展开成的。即按输入信息序列 u 的输入顺序按时间 $l=0,1,2,\cdots$ 展开，并展示所有可能的输入、输出情况。不失一般性，我们以初始状态 $s_0=a$=00 为树根（$l=0$），对每个时刻可能的输入进行分支。在 $l=0$ 的时刻，有 2 个可能的分支，如果输入 $u_0=0$

向上，$u_0=1$ 则向下，都到达下一级节点；在 $l=1$ 的时刻，每个节点根据 u_1 的取值也有 2 个可能的分支，按照上一步的规则向前推进进入二级节点。依此类推，就可以得到一个无线延伸的树状结构，即可求得 $l=0,1,2\cdots$ 不断延伸的树状结构。

和上面的讨论相同，我们用虚线表示输入的值为 0，实线表示输入的值为 1。因此向上的线均为虚线，向下的线均为实线，树图线上的值为输出序列的值。此例由于输入的第一位 $u_0=1$，所以它根据上述状态图规则按照 1 值向下展开的，即加黑的实线和虚线部分，如图 8-9 所示。

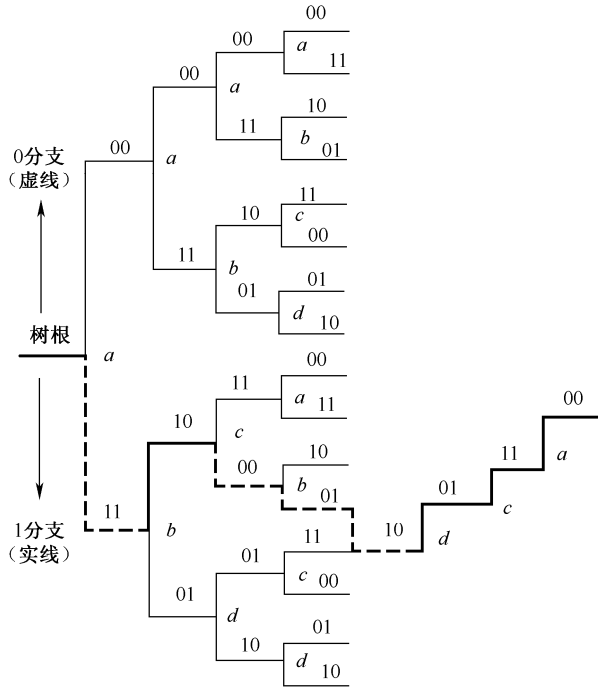


图 8-9 （2，1，3）卷积码的树状图展开

① 对特殊的输入信息序列 $u=(1011100\cdots)$相对应的输出码组 $c=(11\ \ 10\ \ 00\ \ 01\ \ 10\ \ 01\ \ 11\cdots)$，图 8-9 中我们用加粗的黑线表示，且它与前面状态中的结果完全一致。在树图中，编码的过程相当于以输入信息序列为指令沿码树游走，在树图中所经过的路径代码就是相应输出的码序列。

② 树图最大的特点是按时间顺序展开的（即 $l=0,1,2\cdots$）且能将所有时序状态表示为不相重合的路径，但是它也存在很大的缺点，结构太复杂，结构重复性太多。

③ 一般情况下，二元$(n，k，m)$卷积码在每个节点上的可能分支个数为 2^n 个。所有可能的码序列相应于树图上所有路径。码树在研究序列译码的时候比较方便。卷积码是一种线性树码。

3．格图

格图又称篱笆图，它是将码树中处于同一状态的同一级节点合并而成，是一个纵深宽度或者高为 $2^{k(m-1)}$ 的格图。

格图的最大特点是保持了树图的时序展开性，同时又克服了树图中太复杂的缺点，它将树图中产生的重复状态合并起来。例 8-3 的格状图结构如图 8-10 所示。

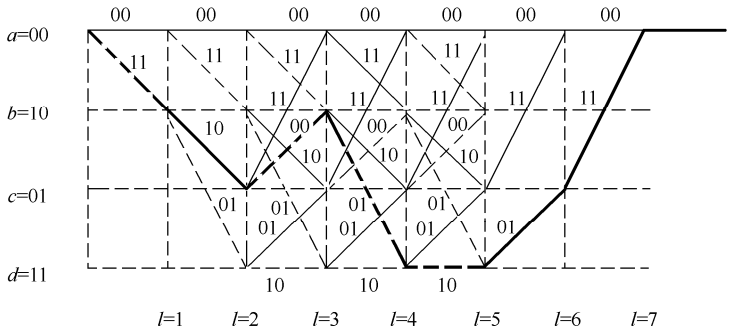


图 8-10 （2，1，3）卷积码的格图

① 格图在 Viterbi 译码中特别有用。

② 将树图转化为格图是很方便的。在树图中，当 $l=m=3$ 时，状态 a、b、c、d 呈现重复，如果将重复状态折合起来加以合并，就可以得到纵深宽度（又称高度）为 $2^{k(m-1)}=2^{1\times 2}=4$ 的格状图。同上面的假设一样，图中实线表示输入为“0”时所走的分支，虚线则表示输入为“1”时所走的分支。

③ 由于格图既能体现时序关系，又能较简洁地表示状态结构，所以它是卷积码的一种简洁的表达形式。

④ 图中粗黑线是表示输出 $\boldsymbol{u}$ =(1011100)时其对应编码为 $\boldsymbol{c}$ =(11100001100111)，这一结果与前面状态图方式、树图方式所得结果完全一致。

⑤ 不同的信息序列在树图上所对应路径完全不重合，但是在格图上则有可能有部分重合，这样两个不同输入序列，只需计算格图中不相重合部分即可，所以在译码时利用格图更加方便。

前面我们讨论的卷积码都可以看做是基于半无限的，但实际的译码中通常仅考察有限段的接收码元，所以就引入了所谓的截断码。当我们观察编码器的某一时间段的码序列的时候，如 $l_1 \leqslant l \leqslant l_2$，就可以把此时的卷积码看做一个 (N,L) 码，其中 $N=n(l_2-l_1+1)$，$L=k(l_2-l_1+1)$，称为卷积码 (n,k,l) 的截断码，我们用 $T_r[l_1,l_2]$ 来表示。

当 $l_2=\infty$ 的时候，就可以得到截去开头 l_1 段的卷积码，或者称作 l_1 **时延码**。当 $l_1=m$ 的时候就是一个 m 时延码，其生成矩阵为式（8-19）。

$$[\boldsymbol{G}]_m=\begin{bmatrix} \boldsymbol{G}_m & & & & & & \\ \boldsymbol{G}_{m-1} & \boldsymbol{G}_m & & & & & \\ \boldsymbol{G}_{m-2} & \boldsymbol{G}_{m-1} & \boldsymbol{G}_m & & & & \\ \vdots & \vdots & \vdots & \ddots & & & \\ \boldsymbol{G}_0 & \boldsymbol{G}_1 & \boldsymbol{G}_2 & \cdots & \boldsymbol{G}_m & & \\ & \boldsymbol{G}_0 & \boldsymbol{G}_1 & \boldsymbol{G}_2 & \cdots & \boldsymbol{G}_m & \\ & & \ddots & \ddots & \ddots & \ddots & \\ & & & \boldsymbol{G}_0 & \boldsymbol{G}_1 & \cdots & \boldsymbol{G}_m \end{bmatrix} \tag{8-19}$$

当 $l_1=0$，$l_2=l$ 的时候就是 (n,k,l) 卷积码的截尾码，记做 $T_r[0,l_2]$；特别是 $l_1=0$，$l_2=m$ 的时候就称作是卷积码 (n,k,l) 的初始码。

衡量卷积码性能的标准仍然是距离，对线性码而言就是最小重量。但是卷积码的距离比

分组码要复杂得多，不同的译码算法采用不同的距离度量。下面我们给出几个卷积码常用距离的定义。

码的重量：给定一个(n, k, m)的卷积码，在长为 L 段的码序列 $\boldsymbol{c}$ 中，非零码元的个数或者 $c(x)$ 多项式中非零系数的个数称为码序列的重量，用 $\omega(\boldsymbol{c}_L)$ 表示。

码间距离：给定一个(n, k, m)的卷积码，在长为 L 段的码序列 $\boldsymbol{c}$ 和 $\boldsymbol{c}'$之间的距离定义为

$$d(\boldsymbol{c}_L, \boldsymbol{c}_L') \overset{\Delta}{=} \omega(\boldsymbol{c}_L - \boldsymbol{c}_L') = \omega(\boldsymbol{c}_L' - \boldsymbol{c}_L) \tag{8-20}$$

最小距离：给定一个(n, k, m)的卷积码，在长为 L 段的码序列的最小距离定义为

$$d_{\min}(L) \overset{\Delta}{=} \min_{\substack{\boldsymbol{c}_L, \boldsymbol{c}_L' \in Tr(0,L) \\ \boldsymbol{c}_L \neq \boldsymbol{c}_L'}} d(\boldsymbol{c}_L - \boldsymbol{c}_L') \tag{8-21}$$

最小自由距离：卷积码的最小自由距离为

$$d_f \overset{\Delta}{=} d_{\min}(\infty) \overset{\Delta}{=} \min_{\substack{\boldsymbol{c}_L, \boldsymbol{c}_L' \in Tr(0,L) \\ \boldsymbol{c}_L \neq \boldsymbol{c}_L'}} d(\boldsymbol{c}_\infty - \boldsymbol{c}_\infty') = \min_{\boldsymbol{u} \neq \boldsymbol{u}'} d(\boldsymbol{uG}, \boldsymbol{u'G}) \tag{8-22}$$

即为任意长编码后序列之间的最小汉明距离。

码的最小重量：卷积码的最小重量为

$$\omega_{\min} \overset{\Delta}{=} \min_{\substack{\boldsymbol{c}_\infty \\ \boldsymbol{c}_\infty \neq 0}} \omega(\boldsymbol{c}_\infty) = \min_{\boldsymbol{u} \neq 0} \omega(\boldsymbol{uG}) \tag{8-23}$$

反馈译码的最小距离：卷积码反馈译码的最小距离的定义为长 $m+1$ 段码序列的最小重量，也就是 m 截尾码的最小距离。

$$d_{FD} \overset{\Delta}{=} \min d(\boldsymbol{c}_m, \boldsymbol{c}_m') = \min_{\boldsymbol{u} \neq \boldsymbol{0}} \omega((\boldsymbol{c})_m) \tag{8-24}$$

8.3 卷积码的译码

卷积码的译码可分为两大类型：代数译码与概率译码。属于代数译码的有 Massey1963 年提出的门限（或称为大数逻辑）译码，属于概率译码的有 1957 年 Wozencraft 提出的序列译码和 1967 年 Viterbi 提出的最大似然译码。后来人们就称它为 Viterbi 译码。本节主要讨论卷积码的译码技术。

8.3.1 卷积码的代数译码

假设卷积码编码器的输出码元序列为

$$\begin{aligned} c(x) &= \sum_{l=0}^{\infty} c l x^l = \left[c^1(x), c^2(x), \cdots c^n(x) \right] \\ c^j(x) &= \sum_{i=0}^{k} u^i(x) g^{i,j}(x) \qquad 1 \leqslant j \leqslant n \end{aligned} \tag{8-25}$$

通过 BSC 信道（二元对称信道）传输，则在接收端收到的序列可以用下式描述

$$\boldsymbol{R}(x) = \boldsymbol{c}(x) + \boldsymbol{E}(x) \tag{8-26}$$

其中， $\boldsymbol{E}(x)=[E^{1}(x),E^{2}(x),\cdots,E^{n}(x)]^{T}$，是干扰序列的多项式。

若信道是离散无记忆的（DMC），即

$$R_l^j=c_l^j+E_l^j,\qquad j=1,2,\cdots,n_0\quad l=1,2,\cdots \tag{8-27}$$

将式（8-26）写成矢量的形式

$$\boldsymbol{R}_l=\boldsymbol{c}_l+\boldsymbol{E}_l,\qquad l=1,2,\cdots \tag{8-28}$$

对应的多项式形式

$$R^j(x)=c^j(x)+E^j(x),\qquad j=1,2,\cdots,n_0 \tag{8-29}$$

译码就是一个寻找发送信息序列的过程，即在有干扰 $\boldsymbol{E}(x)$ 存在的情况下从接收到的序列 $\boldsymbol{R}(x)$ 中按照最大似然准则确定相应的码序列 $\boldsymbol{c}(x)$ 或者信息序列 $\boldsymbol{u}(x)$。我们发现这个过程中只要 $\boldsymbol{E}(x)$ 确定了，就可以通过 $\boldsymbol{R}(x)$ 确定 $\boldsymbol{c}(x)$，所以其等效于确定最大似然错误图样 $\boldsymbol{E}(x)$。卷积码的代数译码类似于我们前面介绍的分组码的译码，其译码过程可以按照下面的步骤来进行：（1）计算 $\boldsymbol{R}(x)$ 的伴随式；（2）计算错误图样 $\boldsymbol{E}(x)$；（3）纠错。

假设卷积码的一致校验矩阵为 $\boldsymbol{H}(x)$，伴随式的计算如下。

$$\begin{aligned}\boldsymbol{s}(x)&=\boldsymbol{R}(x)\boldsymbol{H}^{T}(x)=(\boldsymbol{c}(x)+\boldsymbol{E}(x))\boldsymbol{H}^{T}(x)\\&=\boldsymbol{c}(x)\boldsymbol{H}^{T}(x)+\boldsymbol{E}(x)\boldsymbol{H}^{T}(x)\end{aligned} \tag{8-30}$$

根据式（8-17）可知

$$\boldsymbol{s}(x)=\boldsymbol{0}+\boldsymbol{E}(x)\boldsymbol{H}^{T}(x)=\boldsymbol{E}(x)\boldsymbol{H}^{T}(x) \tag{8-31}$$

从上面的分析我们可以清楚地看到，伴随式只是和信道中的错误图样 $\boldsymbol{E}(x)$ 有关，而与具体发送的码字序列无关，所以就可以通过 $\boldsymbol{s}(x)$ 来确定错误图样 $\boldsymbol{E}(x)$。

在代数译码中，采用的是 m 截尾码，在这种情况下对于系统码和非系统码而言，其最小距离是一样的。因此一般都采用系统结构。对于系统码，根据式（8-18）,我们有

$$\begin{aligned}\boldsymbol{H}^{T}(x)&=\left[\begin{array}{cccc|c}h^{1,1}(x)&h^{1,2}(x)&\cdots&h^{1,k}(x)&\\h^{2,1}(x)&h^{2,2}(x)&\cdots&h^{2,k}(x)&\\\vdots&\vdots&\ddots&\vdots&\boldsymbol{I}_{n-k}\\h^{n-k,1}(x)&h^{n-k,2}(x)&\cdots&h^{n-k,k}(x)&\end{array}\right]^{T}\\&=\left[\begin{array}{cccc}h^{1,1}(x)&h^{2,1}(x)&\cdots&h^{n-k,1}(x)\\h^{1,2}(x)&h^{2,2}(x)&\cdots&h^{n-k,2}(x)\\\vdots&\vdots&\ddots&\vdots\\h^{1,k}(x)&h^{2,k}(x)&\cdots&h^{n-k,k}(x)\\\hline1&0&\cdots&0\\0&1&\cdots&0\\\vdots&\vdots&\ddots&\vdots\\0&0&\cdots&1\end{array}\right]\end{aligned} \tag{8-32}$$

故可以将式（8-31）展开为

$$s^j(x)=\boldsymbol{E}(x)\boldsymbol{H}^j(x)^{\mathrm{T}}=\sum_{i=1}^{k}E^i(x)h^{j,i}(x)+E^{k+j}(x)=\sum_{l=0}^{\infty}s_l^j x^l \qquad (8\text{-}33)$$
$$j=1,2,\cdots,n-k$$

所以有

$$\boldsymbol{s}=[\boldsymbol{s}_1,\boldsymbol{s}_2,\cdots,\boldsymbol{s}_l,\cdots]$$
$$\boldsymbol{s}_l=[s_l^1,s_l^2,\cdots,s_l^{n-k}]$$

将式（8-33）写成标量的形式

$$s_l^j=E_l^{k+j}+\sum_{i=1}^{k}E_l^i h_0^{j,i}+\sum_{i=1}^{k}E_{l-1}^i h_1^{j,i}+\cdots+\sum_{i=1}^{k}E_{l-m}^i h_m^{j,i} \qquad (8\text{-}34)$$

与分组码不同，卷积码的译码是逐段进行的。开始的时候译码器输出值为 0，等接收完第 m 段的码元序列并计算出伴随式后就对第 0 段进行译码；依此类推，在收到第 $m+1$ 段的码元序列并计算出伴随式后就对第 1 段进行译码；…。如果第 0 段有错，并且发生在第 0 段和第 m 段之间的错误没有超出卷积码的纠错能力，就可以由第 $m+1$ 段接收的码元判断第 0 段码元的错误图样，进行纠正后就将第 0 段译出，并进行下一步的译码操作。

在第 0 段序列的译码过程中，如果有反馈回路，由于消除第 0 段错误对后面的影响，就称这类译码方式叫做反馈译码；否则就是定译码。

下面讨论反馈译码是如何修正伴随式的。假设译码器的初始状态都为 0，译码器缓存容量为 mk 个比特，当接收 m 段的码元后，译码器算出前 m 段伴随式；当第 m 段数据到达后，此时共有 $m+1$ 段数据，所以根据式（8-34）可以计算出 $m+1$ 个伴随式，具体为

$$\begin{cases} s_0^j=E_0^{k+j}+\sum_{i=1}^{k}E_0^i h_0^{j,i} \\ s_1^j=E_1^{k+j}+\sum_{i=1}^{k}E_1^i h_0^{j,i}+\sum_{i=1}^{k}E_0^i h_1^{j,i} \\ \quad\vdots \\ s_m^j=E_m^{k+j}+\sum_{i=1}^{k}E_m^i h_0^{j,i}+\cdots+\sum_{i=1}^{k}E_0^i h_m^{j,i} \\ s_{m+1}^j=E_{m+1}^{k+j}+\sum_{i=1}^{k}E_{m+1}^i h_0^{j,i}+\cdots+\sum_{i=1}^{k}E_0^i h_{m+1}^{j,i} \\ \quad\vdots \end{cases} \qquad (8\text{-}35)$$

根据上面的公式，第 0 段的码元只和前 $m+1$ 个方程有关。因此在译第 0 段的时候，只用到前$(m+1)(n-k)$个数字。假设 $\hat{\boldsymbol{E}}_0$ 为前 $m+1$ 段数据对第 0 段码元的错误图样的估计值，则经过纠错的接收序列就变成

$$\tilde{R}^i(x)=R^i(x)+\hat{E}_0^i \qquad i=1,2,\cdots,k \qquad (8\text{-}36)$$

故 $\tilde{R}^i(x)$ 的错误图样为

$$\boldsymbol{E'}(x)=\boldsymbol{E}(x)-\hat{\boldsymbol{E}}_0 \qquad (8\text{-}37)$$

与 $\tilde{\boldsymbol{R}}^i(x)$ 想对应的伴随式为

$$\begin{aligned}\tilde{s}^{j}(x)&=\boldsymbol{E}^{'}(x)\boldsymbol{H}^{j}(x)^{\mathrm{T}}=(\boldsymbol{E}(x)-\hat{\boldsymbol{E}}_0)\boldsymbol{H}^{j}(x)^{\mathrm{T}}\\&=s^{j}(x)-\hat{\boldsymbol{E}}_0\boldsymbol{H}^{j}(x)^{\mathrm{T}}\end{aligned}\tag{8-38}$$

此时 $\hat{\boldsymbol{E}}_0\boldsymbol{H}^{j}(x)^{\mathrm{T}}=\sum_{i=1}^{k}\boldsymbol{E}_0^{i}h^{(j,i)}(x)$ 就称为伴随式修正项。从已经计算的接收序列的伴随式中减去这个修正项就可以得到纠正后接收序列的伴随式，从而消除了第 0 段上信道错误对第 1～m 段数据的影响。

一般采用大数逻辑法和捕猎法对错误图样进行纠错。前者多用于纠正独立的随机性错误，后者主要用于纠正突发错误。在代数译码中，可以将(n,k,m)的卷积码看做$(n(m+1),k(m+1))$的分组码，在逐段译码的情况下，每次译首段，当首段译出后接收下一段，依此类推。下面通过一个具体的例子来说明。

例 8-4 假设一个二元的(2,1,6)卷积码的子生成元为

$$\begin{cases}g^{(1,1)}=(100000)\\g^{(1,2)}=(100111)\end{cases}$$

试分析其译码结构。

此时 $k=1$，$n=2$，$m=6$，则根据式（8-9）、式（8-10）可知

$$\begin{aligned}\boldsymbol{G}^{1}&=(11\quad 00\quad 00\quad 01\quad 01\quad 01)\\&=[\boldsymbol{G}_0^{1}\quad \boldsymbol{G}_1^{1}\quad \boldsymbol{G}_2^{1}\quad \boldsymbol{G}_3^{1}\quad \boldsymbol{G}_4^{1}\quad \boldsymbol{G}_5^{1}]\end{aligned}$$

因为 $k=1$，所以有

$$\begin{cases}\boldsymbol{G}_0=\boldsymbol{G}_0^{1}=11\\\boldsymbol{G}_1=\boldsymbol{G}_1^{1}=00\\\boldsymbol{G}_2=\boldsymbol{G}_2^{1}=00\\\boldsymbol{G}_3=\boldsymbol{G}_3^{1}=01\\\boldsymbol{G}_4=\boldsymbol{G}_4^{1}=01\\\boldsymbol{G}_5=\boldsymbol{G}_5^{1}=01\end{cases}$$

故根据式（8-11），生成矩阵为（半无限）

$$\boldsymbol{G}_{\infty}=\begin{bmatrix}11&00&00&01&01&01&\cdots&\\&11&00&00&01&01&01&\cdots\\&&11&00&00&01&01&01&\cdots\\&&&\ddots&&&&&\end{bmatrix}$$

由于在译码的时候(n,k,m)的卷积码可以看做是一个 $Tr(0,5)$ 的截尾码，所以可以写出相应 $(m+1)\times n(M+1)=6\times 12$ 阶的 m 截尾生成矩阵。

$$\boldsymbol{G}=\begin{bmatrix}11&00&00&01&01&01\\&11&00&00&01&01\\&&11&00&00&01\\&&&11&00&00\\&&&&11&00\\&&&&&11\end{bmatrix}$$

可以看出该卷积码是一个系统码，它的一致校验矩阵根据式（8-18）可以写为

$$\boldsymbol{H}=\begin{bmatrix} 11 & & & & & \\ 00 & 11 & & & & \\ 00 & 00 & 11 & & & \\ 10 & 00 & 00 & 11 & & \\ 10 & 10 & 00 & 00 & 11 & \\ 10 & 10 & 10 & 00 & 00 & 11 \end{bmatrix}$$

进一步可以写出它的编码器结构，如图 8-11 所示。

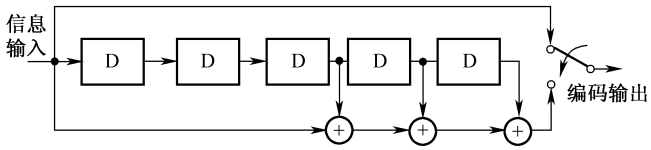


图 8-11 一种结构为(2,1,6)的系统卷积码

该卷积码的一致校验关系为

$$\begin{cases} c_l^1 = u_l \\ c_l^2 = u_l + u_{l-3} + u_{l-4} + u_{l-5} \end{cases}$$

它可采用大数逻辑译码法译码，具体的译码电路如图 8-12 所示。

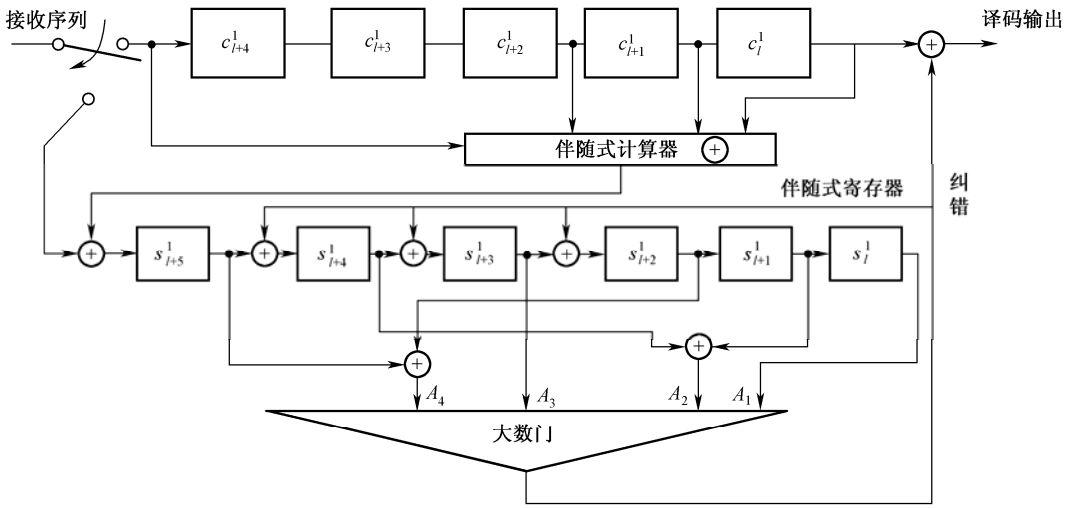


图 8-12 (2,1,6)系统卷积码的译码器结构

下面具体讨论如何从伴随式得到一组正交的校验式，以判断第 0 段是否有错误发生。

在译 u_l 段的时候，需要知道 E_l^1 的估值，我们假定在译第 l 段的时候，前面各段已经正确译码。根据式（8-35）可以得到

$$\begin{cases} s_l = E_l^1 + E_l^2 \\ s_{l+1} = E_{l+1}^1 + E_{l+1}^2 \\ s_{l+2} = E_{l+2}^1 + E_{l+2}^2 \\ s_{l+3} = E_l^1 + E_{l+3}^1 + E_{l+3}^2 \\ s_{l+4} = E_l^1 + E_{l+1}^1 + E_{l+4}^1 + E_{l+4}^2 \\ s_{l+5} = E_l^1 + E_{l+1}^1 + E_{l+2}^1 + E_{l+5}^1 + E_{l+5}^2 \end{cases}$$

对上式进行组合就可以得到

$$\begin{cases} A_1 = s_l = E_l^1 + E_l^2 \\ A_2 = s_{l+1} + s_{l+4} = E_l^1 + E_{l+1}^2 + E_{l+4}^1 + E_{l+4}^2 \\ A_3 = s_{l+3} = E_l^1 + E_{l+3}^1 + E_{l+3}^2 \\ A_4 = s_{l+2} + s_{l+5} = E_l^1 + E_{l+1}^1 + E_{l+2}^1 \end{cases}$$

其中，每个公式都包含有 E_l^1，而其他的分量则只是在某一个方程中出现一次。所以我们称上式对 E_l^1 构成一个正交一致校验矩阵。如果伴随式矩阵不经过这种组合就可以直接对首段信息码元有正交校验关系，就称这种码为自正交码。

大数逻辑的判决规则可以根据下面的分析来进行，该例的判决分以下的 3 种情况。

① 若 $E_l^1 = 1$，而另外一个错误出现在其他位，则上面组合后 4 个和式中肯定有 3 个和式取值为 1；如果两个错误都在其他位，则有 2 个和式取值为 1。

② 若只有一个错误且 $E_l^1 = 1$，则 4 个和式均取值 1；错误在其他位时，只有 1 个和式取值 1。

③ 无错的时候，4 个和式取值 0。

所以根据上述的分析，当错误的个数小于 2 的时候，如果

$$\sum_{i=1}^{4} A_i > 2$$

则判定 $E_l^1 = 1$，否则 $E_l^1 = 0$，就能保证正确的译码。即该码能够纠正任意 n_A 长字段上的 2 个错误。

对 E_l^1 正确估值后，通过反馈修正原来的伴随式就可以按照相同的方式对下一段的信息进行判决。

该例中，如果长 n_A 中的错误超过 2 个，则会出现译码错误，而且反馈会使错误传播。一般只要连续接收的 m 段无错，就可以中断错误传播。

8.3.2 Viterbi 译码算法

Viterbi 译码算法是一种概率译码算法，也是最大似然译码算法。它是一种有效的卷积码译码算法，在空间通信、无线通信中有广泛的应用。

在数字、数据通信中常用的准则是最小平均误码率准则（最大后验概率准则），平均误码率由下式决定。

$$p_e = \sum_{\boldsymbol{r}} p(\boldsymbol{r}) p(\boldsymbol{e} \mid \boldsymbol{r}) \tag{8-39}$$

其中，$p(\boldsymbol{e} \mid \boldsymbol{r}) = p(\hat{\boldsymbol{c}} \neq \boldsymbol{c} \mid \boldsymbol{r})$ 表示接收码组为 $\boldsymbol{r}$ 的时候产生的误码判决，$\hat{\boldsymbol{c}}$ 为译码后的码组信息，$\boldsymbol{c}$ 为发送码组信息，$p(\boldsymbol{r})$ 表示 $\boldsymbol{r}$ 发生的概率。

故可以得到下面的最小平均误码率公式

$$\begin{aligned}\min p_e &= \min \sum_{\boldsymbol{r}} p(\boldsymbol{r}) p(\hat{\boldsymbol{c}} \neq \boldsymbol{c} \mid \boldsymbol{r}) \\ &= \sum_{\boldsymbol{r}} p(\boldsymbol{r}) \min p(\hat{\boldsymbol{c}} \neq \boldsymbol{c} \mid \boldsymbol{r}) = \sum_{\boldsymbol{r}} p(\boldsymbol{r}) \max p(\hat{\boldsymbol{c}} = \boldsymbol{c} \mid \boldsymbol{r})\end{aligned} \tag{8-40}$$

根据 Bayes 公式，我们可以得到

$$P(\boldsymbol{c} \mid \boldsymbol{r}) = \frac{P(\boldsymbol{c}) P(\boldsymbol{r} \mid \boldsymbol{c})}{P(\boldsymbol{r})} \tag{8-41}$$

假设发送码组（字）是等概率的，这时 $P(\boldsymbol{c})$ 为常数，当已知接收的码组（字）时，$P(\boldsymbol{r})$ 亦为常数，所以

$$\max p(\hat{\boldsymbol{c}} = \boldsymbol{c} \mid \boldsymbol{r}) = \max p(\boldsymbol{r} \mid \hat{\boldsymbol{c}} = \boldsymbol{c}) \tag{8-42}$$

根据式（8-40）,即有

$$\min p_e = \max p(\boldsymbol{r} \mid \hat{\boldsymbol{c}} = \boldsymbol{c})$$

结论：当发送码组等概率时，最小平均误码率准则与最大后验概率准则以及最大似然准则等效。

对于离散无记忆信道（DMC）有

$$\max P(\boldsymbol{r} \mid \hat{\boldsymbol{c}} = \boldsymbol{c}) = \max \prod_{l=0}^{L-1} P(r_e \mid \hat{\boldsymbol{c}}_e = \boldsymbol{c}_e) \tag{8-43}$$

由于对数函数为单调函数，故可将其改写为对数函数形式。

$$\max \log \prod_{l=0}^{L-1} P(r_e \mid \hat{\boldsymbol{c}}_e = \boldsymbol{c}_e) = \max \sum_{l=0}^{L\text{-}1} \log P(r_e \mid \hat{\boldsymbol{c}}_e = \boldsymbol{c}_e) \tag{8-44}$$

我们称按上述公式进行译码的算法为最大似然译码算法，同时称公式中的 $\log P(r_e \mid \hat{\boldsymbol{c}}_e = \boldsymbol{c}_e)$ 为对数似然函数，有时简称为似然函数。

进一步，对于 BSC 信道（二进制对数信道），有

$$P(\boldsymbol{r} \mid \hat{\boldsymbol{c}} = \boldsymbol{c}) = p^{d(\boldsymbol{r},\boldsymbol{c})} (1-p)^{n-d(\boldsymbol{r},\boldsymbol{c})} \tag{8-45}$$

似然函数可以进一步改写为

$$\log P(\boldsymbol{r} \mid \hat{\boldsymbol{c}} = \boldsymbol{c}) = d(\boldsymbol{r},\boldsymbol{c}) \log \frac{p}{1-p} + n \log(1-p) \tag{8-46}$$

其中，$d(\boldsymbol{r},\boldsymbol{c})$ 为 $\boldsymbol{r}$ 与 $\boldsymbol{c}$ 之间的汉明距离。

由于对 BSC 信道而言，一般 $p < \frac{1}{2}$，所以有 $\log \frac{p}{1-p} < 0$，且 $n \log(1-p)$ 为常数，而 $\boldsymbol{r}$ 与 $\boldsymbol{c}$ 之间的汉明距离由下式确定。

$$d(\boldsymbol{r},\boldsymbol{c}) = \sum_{l=0}^{L-1} d(r_e, c_e)$$

则有

$$\begin{aligned}\max \log P(r_e \mid \hat{\boldsymbol{c}}_e = \boldsymbol{c}_e) &= \min d(\boldsymbol{r},\boldsymbol{c}) \\ &= \min \sum_{l=0}^{L-1} d(r_e, c_e)\end{aligned}$$

因此对于 BSC 信道来说，求最大似然**等效于**求最小汉明距离。

1．Viterbi 算法描述

对于用格图表示的卷积码，每个码组序列都是从全 0 状态出发，路经格图上不同的分支，最后回到全 0 状态的一条路径。在二进制对称信道 BSC 下卷积码译码的 Viterbi 算法就是建立在前面介绍的格图基础上的最小汉明距离算法。

在 DMC（或 BSC）中 Viterbi 算法可归结为如下 3 步。

① 从时刻 $l=m$ 开始，（$l<m$ 为起始状态）计算进入每一状态的单个路径的部分度量值，并存储每一状态下的幸存路径及其度量值。

② L 增加 1，即 $l=m+1$，将进入某一状态的分支度量值, 与前一时段的幸存度量值相加，然后计算进入该状态的所有最大度量的路径（或最小汉明距离路径）即幸存路径及其度量，并删去所有其他路径。

③ 若 $l<L+m$，重复步骤 2，否则停止。

上述 3 个步骤中，1 是 2 的初始化，3 是 2 的延续，关键在于第 2 步，它主要包括两部分：一是对每个状态进行度量和比较，并决定幸存路径；另一个是记录幸存路径与度量值。下面仍以具体例子入手来分析。

例 8-5 假设卷积码编码器是例 8-3 中的(2,1,3)卷积码，即 $L=5$，$m=3$，$n=2$，$k=1$，若发送的信息序列为 $\boldsymbol{u}=(10111)$，试用 Viterbi 译码算法进行译码。

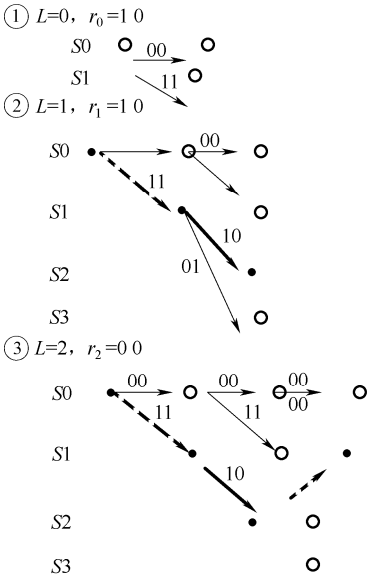
解：根据前面的例题的分析，我们可以得知经过编码后输出的码组（字）为

$$\boldsymbol{c}=(11100001100111)$$

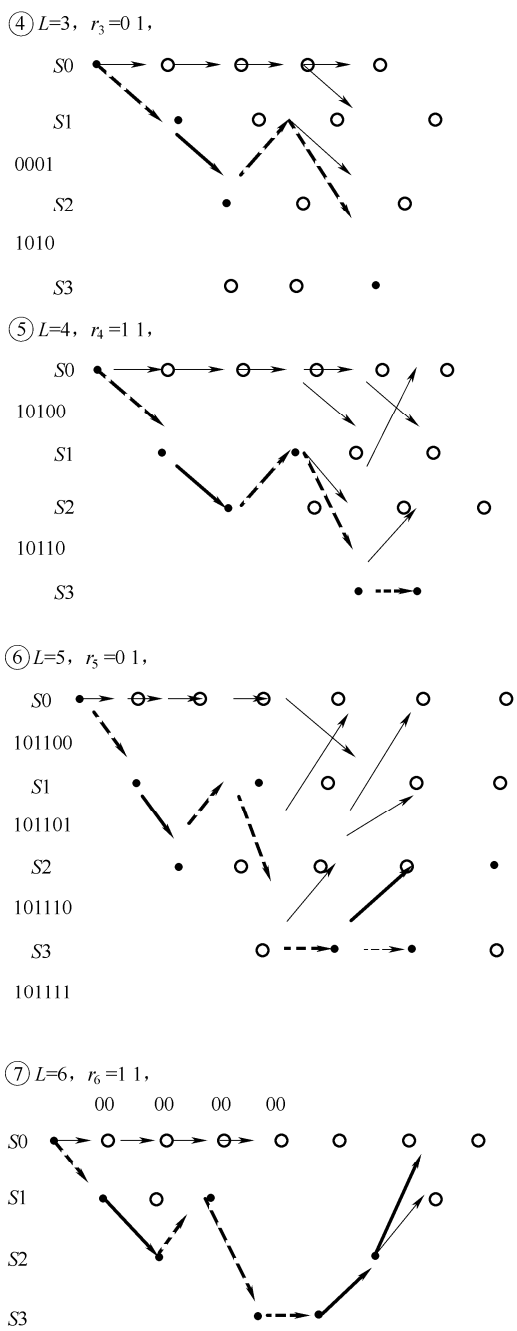
如果接收到的信号序列有 2 位差错，即为：$\boldsymbol{r}=(10100001110111)$，则有汉明距离

$$d(\boldsymbol{r},\boldsymbol{c})=2$$

具体步骤如下:累加的汉明距离是幸存路径的值。（幸存路径相同的随机取一个，本例中取上面的那一条路径。）



汉明距离 d（累加）	估值($\boldsymbol{u}$)
1	0
1	1
2	00
2	01
1	10
3	11
2	000
1	101
3	010
3	011



	汉明距离 d（累加）	估值(u)
④ $L=3$，$r_3=0\ 1$，		
$S0$	3	0000
$S1$		3
0001		
$S2$		3
1010		
$S3$	1	1011
⑤ $L=4$，$r_4=1\ 1$，		
$S0$	3	
10100		
$S1$	3	00001
$S2$		2
10110		
$S3$	2	10111
⑥ $L=5$，$r_5=0\ 1$，		
$S0$		3
101100		
$S1$		3
101101		
$S2$		2
101110		
$S3$		4
101111		
⑦ $L=6$，$r_6=1\ 1$，		
00 00 00 00		
$S0$	2	1011100
$S1$	4	1011101
$S2$		
$S3$		

上面的分析中，如果一个节点有相同的汉明距离，则任选一个继续向下一级节点计算。其中加黑线条表明了译码的路径，估值部分是最后的码元输出。

下面我们用格图来描述具体的距离计算和路径选择。

例 8-6 同例 8-5 条件，假设接收到的信号序列有 2 位差错，为

$$\boldsymbol{r}=(10100101100111)$$

试用 Viterbi 译码算法进行译码。

在 BSC 下，其译码在格图上的每一步的计算结果（距离图）和累计计算结果幸存路径图

如图 8-13 和图 8-14 所示。

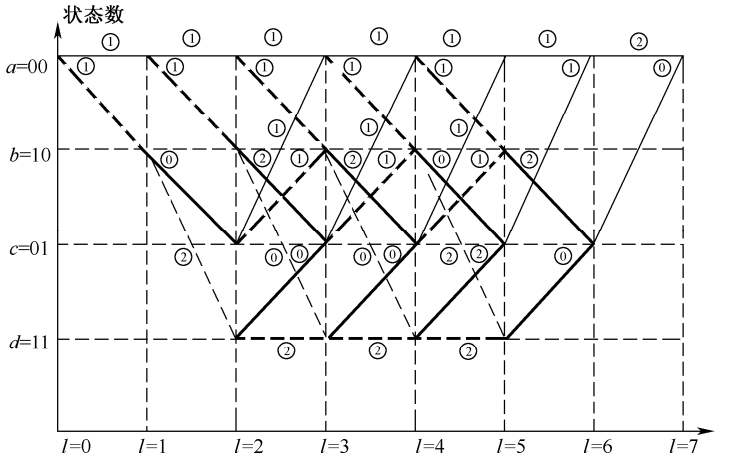


图 8-13 （2，1，3）卷积码 Viterbi 译码汉明距离图

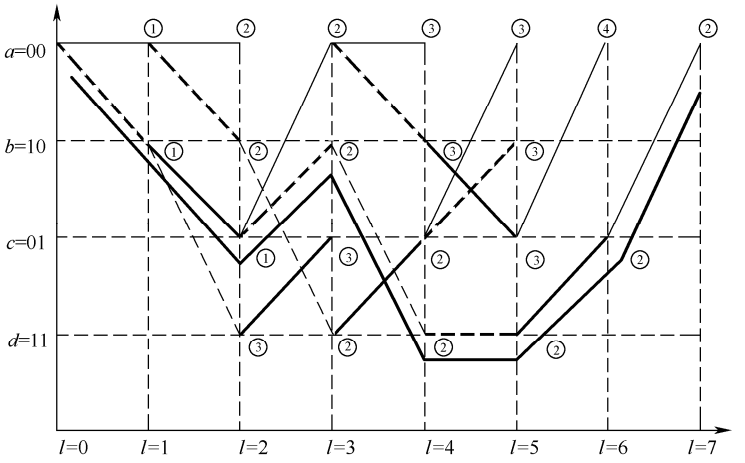


图 8-14 （2，1，3）卷积码 Viterbi 算法的幸存路径

结论：最后求得的最小汉明距离路径如图中粗黑线表示为：$a_0b_1c_2b_3d_4d_5c_6a_7$，最小汉明距离值为 2。

由上面的例子可以总结出如下规律。

维特比算法，对于 DMC 信道（或 BSC 信道），主要体现在上述格形图中寻找具有最大似然值的路径，具体采用迭代法进行处理。即在每一步中，它将进入每一状态的所有路径的度量值进行比较，保存并度量最大度量值（或最小汉明距离值），称为幸存路径,并丢弃其他路径。

每个节点有 2^k 个路径汇入，同时有 2^k 个路径离开。

每个节点在进行路径度量值比较是在 2^k 个路径之间进行，并且选择最大的一个作为幸存路径度量值。每个节点必须至少有 2 个寄存器：一个用于存放幸存路径度量值；另外一个用于存放幸存路径。

DMC（或 BSC）中 Viterbi 算法可归结为如下 3 步。

① 从时刻 $l = m$ 开始，（$l<m$ 为起始状态）计算进入每一状态的单个路径的部分度量值，

并存储每一状态下的幸存路径及其度量值。

② L 增加 1，即 $l=m+1$，将进入某一状态的分支度量值，与前一时段的幸存度量值相加，然后计算进入该状态的所有最大度量的路径（或最小汉明距离路径）即幸存路径及其度量，并删去所有其他路径。

③ 若 $l<L+m=5+2=7$，重复步骤 2，否则停止。

上述 3 个步骤中，1 是 2 的初始化，3 是 2 的延续，关键在于第 2 步，它主要包括两部分：一是对每个状态进行度量和比较，并决定幸存路径，另一个是纪录幸存路径与度量值。

上述 3 步的可以进一步细化。

① 从 $l=m$=2 时刻开始，使网格图充满状态，将路径存储器（PM）和路径度量存储器（MM）从 $l=0$ 到 $l=m$=2，进行初始化。

② $v=l+1=2+1=3$

③ 接收到新的一组数据，它代表 $l=l+1$ 的分支上接收符号组。

④ 对每一状态：

（a）进行分支度量计算；

（b）从 MM 中取出第 l 时刻幸存路径度量值；

（c）进行累加—比较—选择（ACS）基本运算，产生新的幸存路径；

（d）将新的幸存路径及其度量值分别存入 PM 及 MM 中。

⑤ 如果 $l<L+M=5+2=7$，回到步骤 2， 否则继续往下做。

⑥ 求 MM 中最大似然值对应路径，从 PM 中输出判决结果。

Viterbi 译码的实现有 3 种结构:

① 全并行，即采用 2^m 个 ACS 单元同时运算并作 PM、MM 操作；

② 全串行，即采用一个 ACS 单元进行 2^m 次 ACS 与 PM、MM 操作；

③ 时分复用方案，即部分并行，采用少于 2^m 个 ACS 单元，分数次完成全部 2^m 次的 ACS 与相应 PM、MM 操作。

全并行空间复杂度最大，时间复杂度最小，全串行反之，时分复用介于两者之间。

采用 Viterbi 译码时，理论上要求译码器接收完整的数据块后才进行判决，这就要求译码器具有较大的存储空间，同时这也会引入较大的时延。所以，在实际系统中通常采用**截断判决**的方法（截断长度定义为译码深度）。 为了避免解码深度过小而引起系统性能的恶化，通常选择译码深度为约束长度的 5 倍。

2．软判决译码

为了提高译码性能，可将硬判决改为软判决，即多电平判决，这是由于两电平的硬判决在强噪声时容易丢失有用信息，而采用多电平软判决后大约可改进 1.8～2dB。在具体实现时，考虑到实现的复杂度，一般可取 4～8 电平。

对最大似然比作下列线性变换

$$\max[\log_2 p(\boldsymbol{r}\mid\boldsymbol{c})]=\max\sum_{l=0}^{L+M}(a\log_2 p(r_l\mid c_l)+b)$$

其中，a 为任意正整数，b 为任意实数。

例 8-7 以(2,1,3)卷积码为例，试分析通过 DMC 信道采用简单的四电平软判决，如图 8-15 所示。

① DMC 信道转移如图 8-15 所示。

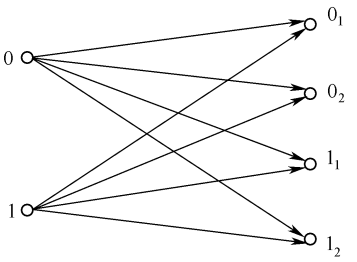


图 8-15 采用四电平软判决

② DMC 转移后的对数比特度量值以及经过上面公式线性变换后的值用表 8-1 和表 8-2 所示。

表 8-1 **经 DMC 转移后的对数比特值**

	0_1	0_2	1_2	1_1
0	−0.4	−0.52	−0.70	−1
1	−1	−0.7	−0.52	−0.40

表 8-2 **取参数 $a=16.8$，$b=1$ 后相应整数度量值**

	0_1	0_2	1_2	1_1
0	10	8	5	0
1	0	5	8	10

③ 若输入：$\boldsymbol{u}$ =（1 0 1 1 1 0 0），后两位为尾比特。

DMC 输入：$\boldsymbol{c}$ =（11，10，00，01，10，01，11）。

DMC 输出：$\boldsymbol{r}$ =（1_20_1，1_10_1，0_10_2，0_11_1，1_11_2，0_21_1，1_20_2）。

软判决译码输出：$\hat{\boldsymbol{u}}$=(1 0 1 1 1 0 0)= $\boldsymbol{u}$ 。

软与硬判决译码过程完全类似，不同之处有:

① 信道模型不一样，硬为 BSC，软为 DMC；

② 度量值及准则不一样，一个为最小汉明，另一个为最大似然；

③ 软比硬复杂度增加不多，但性能确有 1.5dB～2dB 改善。

3．滑动窗维特比译码算法

基本思想：当状态数有限时，给定时刻的各状态残留路径在一定时间（L）之前来自于同一状态的可能性随 L 的增加而迅速趋近于 1。因此当前时刻各残留路径很可能来自于 L 时刻前的同一路径。

①在第 k 时刻，可以将 t–L 时刻前的路径结果直接输出，而在存储空间中不再保存 t–L 时刻前的内容。这里的 L 就被称作译码深度，译码的复杂度不再随码长的增加而增加。因而特别适合信息流的卷积码编译码。在这种情况下甚至不需要对流分段加尾比特。

②滑动窗算法是一种准最优算法，但通常译码深度只要有**编码约束长度的 5～10** 倍，其

性能损失就可以忽略不计。

8.3.3 序列译码

根据上一节的分析我们知道，卷积码的最大似然译码的计算和设备复杂度随着编码约束长度的增加而呈指数增加，每向前推进一个节点就需要将 $2k(m-1)$个路径同时延伸一个分支，每条路径都有 $2k$ 个延伸支路，通过比较，从其中选出一种使路径最短的方式，从而得到 $2k(m\text{-}1)$个新的幸存路径。这样很明显就限制了 Viterbi 译码算法，不适合较大约束长度的卷积码的译码，一般来讲卷积码的性能和约束长度成正比，所以就出现了改进的降低卷积码译码复杂度的算法。其中最主要的一种就是将 Viterbi 算法中每次对所有幸存路径同时延伸改为只对最短路径的延伸，这种改进的算法就被称作是卷积码的序列译码算法。

序列译码的基本思路是在某一级节点，当接收序列的估计路径偏离正确的路径时，随着估计路径向下一级逐步延伸，它的路径度量的增量平均将大于正确的路径度量。

我们令 $\boldsymbol{c}$ 是任意正确的码字，$\boldsymbol{c}'$是输入为 $\boldsymbol{c}$ 的时候的误判码字，$\boldsymbol{R}$ 是接收离散无记忆信道的接收矢量，则序列译码将下式的值作为分支路径度量值。

$$\lambda(\boldsymbol{c})=\sum_{l}\lambda_l \tag{8-47}$$

$$\lambda_l=\ln\frac{p(\boldsymbol{R}_l\mid\boldsymbol{c}_l)}{p(\boldsymbol{R}_l)}-r \tag{8-48}$$

$$p(\boldsymbol{R}_l)=\sum_{\boldsymbol{c}_l}q(\boldsymbol{c}_l)p(\boldsymbol{R}_l\mid\boldsymbol{c}_l) \tag{8-49}$$

其中，$q(\boldsymbol{c}_l)$是$\boldsymbol{c}_l$概率，r 是一个常量（一般取数据速率），该量度称作为费诺量度。

在最大后验概率译码的情况下，判决的准则按$\lambda(\boldsymbol{c})$的最大规则进行，即在判决的时候选择与接收序列互信息量为最大的码字。由于输入等概的时候输出也等概，所以可以将式(8-49)并入常量项 r，它在比较中不起作用，故该值和 Viterbi 算法的路径值是等价的。

下面研究正确路径和错误路径的积累值间的关系。

$$\begin{aligned}\lambda(\boldsymbol{c},\boldsymbol{c}')&=\lambda(\boldsymbol{c})-\lambda(\boldsymbol{c}')\\&=\left(\sum_{l}\ln\frac{p(\boldsymbol{r}_l\mid\boldsymbol{c}_l)}{p(\boldsymbol{r}_l)}-r\right)-\left(\sum_{l}\ln\frac{p(\boldsymbol{r}_l\mid\boldsymbol{c}'_l)}{p(\boldsymbol{r}_l)}-r\right)\\&=\sum_{l}\ln\frac{p(\boldsymbol{r}_l\mid\boldsymbol{c}_l)}{p(\boldsymbol{r}_l\mid\boldsymbol{c}'_l)}\end{aligned} \tag{8-50}$$

上式给出的差值和极大似然译码路径差值刚好异号。序列译码没有必要对每一个路径都作检验后才作出判决，一般只需要检验可能路径中一个小的子集。序列译码要在不同的路径中做一个选择，因此就引入一个偏置量 r，该量对算法的性能有很大的影响。下面我们就进行一些具体的分析。首先我们研究正确路径每增加一个分支时路径值的平均增益。

$$E_{\boldsymbol{r}_l,\boldsymbol{c}_l}(\lambda_l)=\sum_{\boldsymbol{c}_l}q(\boldsymbol{c}_l)\sum_{\boldsymbol{r}_l}p(\boldsymbol{r}_l\mid\boldsymbol{c}_l)\lambda_l \tag{8-51}$$

代入式（8-48），则有

$$\begin{aligned} E_{\boldsymbol{r}_l,\boldsymbol{c}_l}(\lambda_l) &= \sum_{\boldsymbol{c}_l} q(\boldsymbol{c}_l) \sum_{\boldsymbol{r}_l} p(\boldsymbol{r}_l \mid \boldsymbol{c}_l) \left(\ln \frac{p(\boldsymbol{r}_l \mid \boldsymbol{c}_l)}{p(\boldsymbol{r}_l)} - r \right) \\ &= \sum_{\boldsymbol{c}_l} \sum_{\boldsymbol{r}_l} q(\boldsymbol{c}_l) p(\boldsymbol{r}_l \mid \boldsymbol{c}_l) \ln \frac{p(\boldsymbol{r}_l \mid \boldsymbol{c}_l)}{p(\boldsymbol{r}_l)} - \sum_{\boldsymbol{c}_l} \sum_{\boldsymbol{r}_l} q(\boldsymbol{c}_l) p(\boldsymbol{r}_l \mid \boldsymbol{c}_l) r \\ &= I(q) - r \end{aligned} \tag{8-52}$$

其中，$I(q) = \sum_{\boldsymbol{c}_l} \sum_{\boldsymbol{r}_l} q(\boldsymbol{c}_l) p(\boldsymbol{r}_l \mid \boldsymbol{c}_l) \ln \frac{p(\boldsymbol{r}_l \mid \boldsymbol{c}_l)}{p(\boldsymbol{r}_l)}$是输入和输出的互信息。我们知道信道容量 C 是互信息的最大值，即有

$$C = \max_{q} I(q)$$

如果我们取 $r = R$ 是数据速率，则根据式（8-52）知道，当选择合适的 q 使 $I(q)$达到信道容量的时候有

$$E_{\boldsymbol{r}_l,\boldsymbol{c}_l}(\lambda_l) = C - R > 0 \tag{8-53}$$

同理，我们可以算出错误路径度量值的平均增益。

$$\begin{aligned} E_{\boldsymbol{r}_l,\boldsymbol{c}_l,\boldsymbol{c}'_l}(\lambda_l) &= \sum_{\boldsymbol{c}'_l} q(\boldsymbol{c}'_l) \sum_{\boldsymbol{c}_l} q(\boldsymbol{c}_l) \sum_{\boldsymbol{r}_l} p(\boldsymbol{r}_l \mid \boldsymbol{c}_l) \lambda'_l \\ &= \sum_{\boldsymbol{c}'_l} \sum_{\boldsymbol{r}_l} q(\boldsymbol{c}'_l) p(\boldsymbol{r}_l) \lambda'_l \end{aligned} \tag{8-54}$$

代入式（8-48），则有

$$\begin{aligned} E_{\boldsymbol{r}_l,\boldsymbol{c}_l,\boldsymbol{c}'_l}(\lambda_l) &= \sum_{\boldsymbol{c}'_r} \sum_{\boldsymbol{y}_r} q(\boldsymbol{c}'_l) p(\boldsymbol{r}_l) \left(\ln \frac{p(\boldsymbol{r}_l \mid \boldsymbol{c}'_l)}{p(\boldsymbol{r}_l)} - r \right) \\ &= \sum_{\boldsymbol{c}'_r} \sum_{\boldsymbol{y}_r} q(\boldsymbol{c}'_l) p(\boldsymbol{r}_l) \left(\ln \frac{p(\boldsymbol{r}_l \mid \boldsymbol{c}'_l)}{p(\boldsymbol{r}_l)} \right) - r \\ &\leqslant \sum_{\boldsymbol{c}'_r} \sum_{\boldsymbol{y}_r} q(\boldsymbol{c}'_l) p(\boldsymbol{r}_l) \left(\frac{p(\boldsymbol{r}_l \mid \boldsymbol{c}'_l)}{p(\boldsymbol{r}_l)} - 1 \right) - r \qquad \because \ln x \leqslant x - 1 \\ &\leqslant \sum_{\boldsymbol{c}'_r} \sum_{\boldsymbol{r}_l} q(\boldsymbol{c}'_l) p(\boldsymbol{r}_l \mid \boldsymbol{c}'_l) - \sum_{\boldsymbol{c}'_l} \sum_{\boldsymbol{r}_l} q(\boldsymbol{c}'_l) p(\boldsymbol{r}_l) - r \\ &= -r = -R \leqslant 0 \end{aligned} \tag{8-55}$$

根据式（8-53）、式（8-55）可知，正确路径的平均增量总是正值，而由正确路径分叉的错误路径的平均增量总是负值。因此在比较不同长度的路径时，要减去相应的平均增量。虽然选小于 C 的任何值都可以，但是选 $r = R$ 的时候可以使平均计算量为最小。而式（8-55）也说明任何错误的判决将使路径的度量值迟早要低于平均值，只有选择正确的路径的时候才会使路径度量值升到平均值。

1．堆栈序列译码算法

设一个(n,k,m)卷积码，用 $\lambda_u(w)$表示 mk 个信息比特所决定的状态下延伸一个分支路径增量，用 $\hat{\boldsymbol{u}}$ 表示存储器顶部存储的具有最大路径值的路径，其路径值用 $\lambda(\boldsymbol{u})$来表示。图 8-16 所示是堆栈序列译码流程。

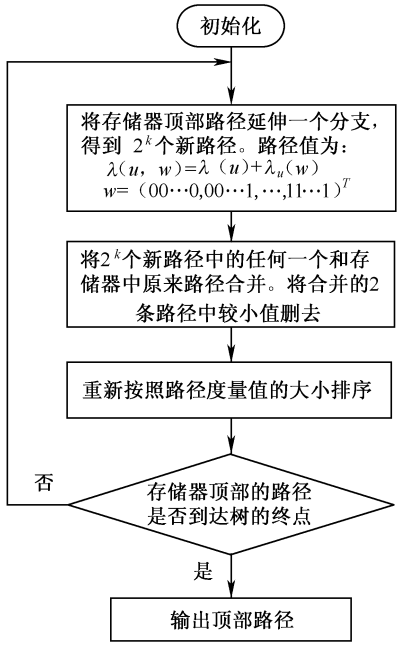
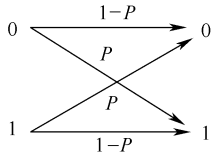


图 8-16 堆栈序列算法流程图

例 8-8 假设卷积码编码器是例 8-3 中的(2,1,3)卷积码，即 $L=5$，$m=3$，$n=2$，$k=1$，码速率 $R=\frac{1}{2}$，若发送的信息序列为 $u=(1011100)$，后两位为尾比特，发送的码字 c 为：$\boldsymbol{c}=11\ \ 10\ \ 00\ \ 01\ \ 10\ \ 01\ \ 11$，假设接收的码矢量为$\boldsymbol{r}=10\ \ 10\ \ 01\ \ 01\ \ 10\ \ 01\ \ 11$，即有 2 个错误产生（在第 2 位和第 6 位）。

如果信息在 BSC 信道中传输，且转移概率 $p=0.02$，信道模型为



试用堆栈序列算法进行译码。

如果信息等概发送，则会有下面的路径度量值。

$$\lambda_l=\begin{cases}2\left[\log_2 2(1-p)-R\right]=2\times0.47=0.94 & \text{分支两位都对}\\ \left[\log_2 2(1-p)-R\right]+\left[\log_2 2p-R\right]=-4.67 & \text{只有一位都对}\\ 2\left[\log_2 2p-R\right]=2\times(-5.12)=-10.29 & \text{分支两位都错}\end{cases}$$

为了计算方便期间，对上式的取值进行整数化，可以得到

$$\lambda_l=\begin{cases}1 & \text{分支两位都对}\\ -5 & \text{只有一位都对}\\ -10 & \text{分支两位都错}\end{cases}$$

图 8-17 所示是根据上面的算法搜索到达的节点以及有关路径值。

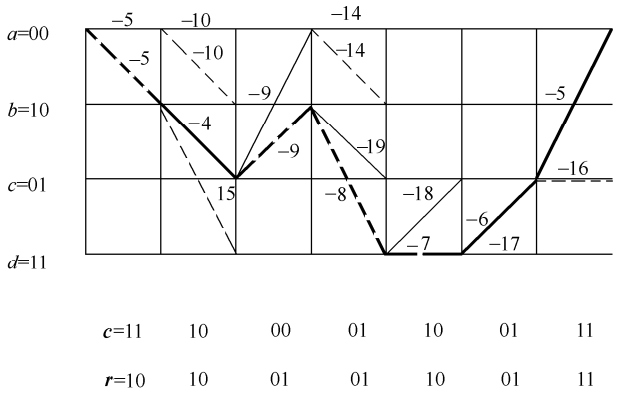


图 8-17 堆栈序列译码算法的路径值以及搜索到达的节点

从图 8-17 中可以看出，只进行了 9 次分支搜索就完成了译码过程，而对于 Viterbi 译码来说，则需要 28 次分支延伸操作，所以序列译码算法大大地减少了译码的计算量。图中路径上的数字分别对应路径值。下面我们用图表的方式进行描述，它有重新的排序操作，看起来更直观一些，如表 8-3 所示。

表 8-3　　堆栈序列译码算法寄存器的值

1	1	2	3	4	5	6	7	8	9
堆栈存储器存储值	0(−5) 1(−5)	1(−5) 00(−10) 01(−10)	10(−4) 00(−10) 01(−10) 11(−15)	100(−9) 101(−9) 00(−10) 01(−10) 11(−15)	101(−9) 00(−10) 01(−10) 1000(−14) 1001(−14) 11(−15)	1011(−8) 00(−10) 01(−10) 1000(−14) 1001(−14) 11(−15) 1010(−19)	10111(−7) 00(−10) 01(−10) 1000(−14) 1001(−14) 11(−15) 10110(−18) 1010(−19)	101110(−6) 00(−10) 01(−10) 1000(−14) 1001(−14) 11(−15) 101111(−17) 10110(−18) 1010(−19)	1011100(−5) 00(−10) 01(−10) 1000(−14) 1001(−14) 11(−15) 1011101(−16) 101111(−17) 10110(−18) 1010(−19)

表 8-3 为译码输出值，括号内是相应的路径值。最后的输出就是在存储器顶部保存的路径，即第 9 步的顶部 1011100，这个时候的路径值为−5，完成了译码。

堆栈序列译码算法是一种在码树中寻求正确路径的有效算法，但是它要求有较大的存储量。因为在每次延伸后要保存已经访问过的所有路径和路径值。如果要减小存储量，就可以直接使用前面描述的序列译码算法，它不仅每次只延伸一个分支路径并对其进行检验，而且只存储一条路径和它对应的路径值。同时，序列译码还保留了接收序列，以便在发现错误时返回到其他路径并能计算相应的路径度量值。

2．费诺序列译码算法

费诺序列译码算法的基本思路是：假设在译码过程中，译码器处于某一个节点，从该节点出发，译码器向前进行观望，如果下一级有它认为合适的节点就向下一级延伸，否则退后一个分支探索那些未访问过的节点。一般来说，如果路径度量值在增加，就继续向下一级节

点推进。反过来，如果某段路径度量值显著减少，则表明译码可能走上错误路径，此时译码器需要往回搜索。译码器向前探索时应加上新的路径度量值，后退探索时同样应该减去相应的分支路径度量值；并且随着译码器前进或后退适时地调整运行判决门限 T（减少或者增加一个分量 Δ），以保证译码器不会在同一门限下访问同一路径。具体的算法流程在图 8-18 中给出。当译码进行到 $m+L$ 级节点的时候就停止搜索。运行门限开始是 0，每次调整为 Δ，所以 T 是 Δ 的倍数。在任何时候 T 的值都小于当前译码路径度量值 λ。

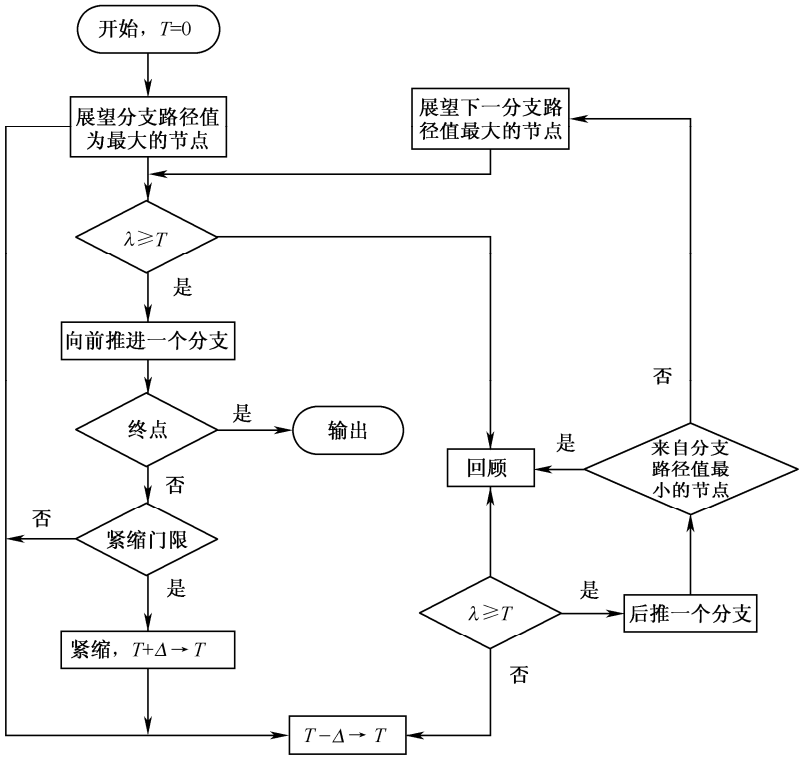


图 8-18 费诺序列译码算法流程图

例 8-9 用费诺序列译码算法重新做例 8-8。

$\lambda 1$ 的计算同例 8-8，我们选择费诺算法中的 $\Delta=4$，表 8-4 是费诺算法的译码过程。

表 8-4 (2,1,3)卷积码的费诺序列译码算法译码过程

译码节拍	展望最大分支 B 或次大分支 NB	前进一个分支路径值（∑）	后退一个分支路径值	到达节点	相应路径值	本步完成后 T 值
0	—	—	—	×	0	0
1	B	−5	−∞	×	0	$-\Delta$
2	B	−5	−∞	×	0	-2Δ
3	B	−5	—	0	−5	-2Δ
4	B	−10	0	×	0	-2Δ
5	NB	−5	—	1	−5	-2Δ
6	B	−4	—	10	−4	-2Δ
7	B	−9	−5	1	−5	-2Δ

续表

译码节拍	展望最大分支 B 或次大分支 NB	前进一个分支路径值（∑）	后退一个分支路径值	到达节点	相应路径值	本步完成后 T 值
8	NB	−15	0	×	0	-2Δ
9	NB	−5	$-\infty$	×	0	-3Δ
10	B	−5	—	0	−5	-3Δ
11	B	−10	—	00	−10	-3Δ
12	B	−15	−10	0	−5	-3Δ
13	NB	−10	—	01	−10	-3Δ
14	B	−9	—	011	−9	-3Δ
15	B	−8	—	0110	−8	-3Δ
16	B	−13	−5	011	−8	-3Δ
17	B	−9	1	01	−9	-3Δ
18	B	−10	1	0	−10	-3Δ
19	B	−5	$-\infty$	×	0	-3Δ
20	NB	−5	—	1	−5	-3Δ
21	B	−4	—	10	−4	-3Δ
22	B	−9	—	101	−9	-3Δ
23	B	−8	—	1011	−8	-2Δ
24	B	−7	—	10111	−7	-2Δ
25	B	−6	—	101110	−6	-2Δ
26	B	−5	—	1011100	−5	-2Δ

Δ 的不同选值对算法的计算量和错误概率都有影响。一般来说 Δ 选择的太小会导致最佳路径总通不过而造成较多的短程往返搜索；而 Δ 选择的大又可能使非最大似然路径通过门限而造成译码错误，而且往往会使错误路径走得太远才返回，也导致计算量的增加。

在译码过程中，如果在往下一级节点的延伸中 2 个分支的路径值是相同的，则任选一个作为最佳路径向下延伸，不失一般性，本例中我们选择 2 个分支中的第 2 个分支作为最佳的延伸路径（即输入为“1”时的路径）。

从表 8-4 所给出的译码过程我们可以看出，费诺算法所需要的计算步骤要多于堆栈序列译码算法。在本例中，节点 1（输出为 1）被访问了 3 次，节点“0”被访问了 4 次，“10”和“01”均被访问了 2 次，根节点也多次被访问了等。这些都是对减少存储器付出的代价。

表 8-4 中的信息说明：

B 表示展望下一级的最佳节点（最大分支路径值），NB 表示展望下一级的次佳节点（次大分支路径值）；

前进一个路径分支值是路径的和值；

后退一个路径分支值是在不满足 $\lambda \geqslant T$ 并且没有改变门限值的时候向后退的路径值；

到达节点表示目前算法搜索到达的最佳节点。

本步完成后门限 T 值是和 Δ 有关，一般取值不变、$T-\Delta \to T$ 或者 $T+\Delta \to T$ 等 3 种情况。

本节主要介绍了卷积码的译码算法。除了这些算法，卷积码的译码还有软译码、迭代译

码等算法，具体的内容请参考有关的参考文献。

8.3.4 卷积码的生成函数

根据前面的分析可知，任何一个非零码序列都和状态转移图中从全零状态出发又回到全零状态的一条路径相对应。因此我们就可以对状态转移图进行修正得到卷积码的生成函数。

我们仍以图 8-8 所示的(2,1,3)卷积码的状态转移图为例进行分析。在修改的时候我们首先按照下面的几个规则：（1）将全零状态节点 a 分为 2 个，a_0 表示输入，a_1 表示输出；（2）将每条路径上的序列汉明重量用 D 的次方形式表示，其中 D 的幂次为该条路径的汉明重量。这样我们就可以得到图 8-19 所示的一个新的状态图。比如路径 $\boldsymbol{P}=a_0bdcbca_1$，则可以计算出该路径的总重量。

$$D^2 \square D^1 \square D^1 \square D^0 \square D^1 \square D^2 = D^7$$

即该路径的重量为 7。

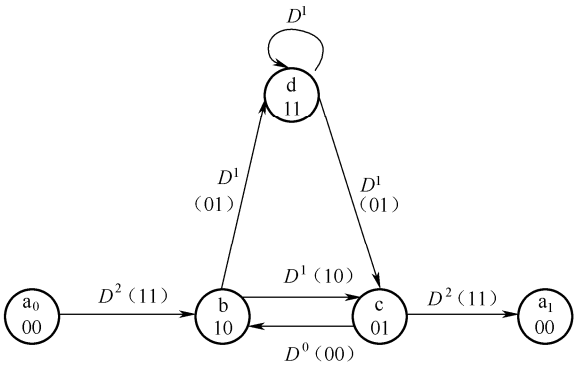


图 8-19 修正后的（2，1，3）卷积码的状态图

假设 A_i 表示重量为 i 的路径的个数，则用式（8-56）表示卷积码的路径重量生成函数。

$$A(D)=\sum_{i=0}^{\infty}A_iD^i \tag{8-56}$$

不难看出使得 $A_i \neq 0$ 的最小的 i 就是码的自由距离 d_{free}。

对于给定的（n,k,m）卷积码，就可以通过标准组合技术由状态转移图获得其生成函数。对于上例的（2，1，3）卷积码，我们可以根据图 8-19 写出状态方程。

$$\begin{cases} Y_b = D^2X_{a_0} + Yc \\ Y_c = DY_b + DY_c \\ Y_d = DY_d + DY_b \\ Y_{a_1} = D^2Y_c \end{cases} \tag{8-57}$$

其生成函数（传递函数）定义为 $A(D)=\dfrac{Y_{a_1}}{X_{a_0}}$，求解（8-57）方程组有

$$
\begin{aligned}
A(D) &= \frac{D^5}{1-2D} \\
&= D^5 + 2D^6 + 4D^7 + \cdots + 2^l D^{l+5} + \cdots \\
&= \sum_{i=5}^{\infty} A_i D^i
\end{aligned}
\tag{8-58}
$$

所以我们可以得到 $A_i \neq 0$ 的最小的 i 取值为 5，所以该卷积码编码方案的自由距离为 5，即可以纠正 2 个错误。重量为 6 的路径有 2 条，重量为 7 的路径有 4 条，…。

除了能提供上面分析的路径的重量特性外，卷积码的转移函数还可以提供更多的有用信息：路径分支的数目和相应信息序列的性质等。我们可以对图 8-19 做进一步的修改，在每一条支路增加一个参数 L，当输入为 1 的时候增加一个参数 N，则我们可以得到图 8-20 所示的转移函数图。

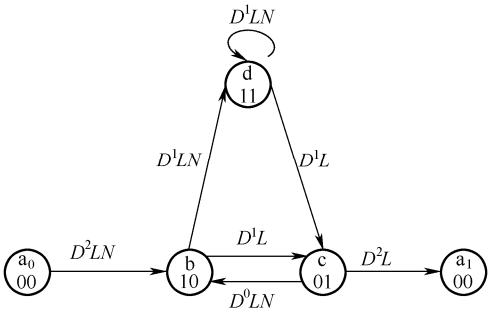


图 8-20 （2，1，3）卷积码的 A（D,L,N）状态图

这样，类似上面的分析，我们可以得到状态方程组

$$
\begin{cases}
Y_b = D^2 LNX_{a_0} + LNYc \\
Y_c = DLY_b + DLY_c \\
Y_d = DLNY_d + DLNY_b \\
Y_{a_1} = D^2 LY_c
\end{cases}
\tag{8-59}
$$

同上，有转移函数

$$
\begin{aligned}
A(D,L,N) &= \frac{D^5 L^3 N}{1 - DLN(1+L)} \\
&= D^5 L^3 N + D^6 L^4 N^2 (1+L) + D^7 L^5 N^3 (1+L)^2 \\
&\quad + \cdots + D^{i+5} L^{i+3} N^{i+1} (1+L)^i + \cdots
\end{aligned}
\tag{8-60}
$$

当 $L = N$=1 时，则退回到式（8-58）。

8.4 卷积码的类型

8.4.1 卷积码中的好码

根据上一节的讨论，可以从卷积码的生成函数中得到有关距离的信息，特别是自由距离。

但是不同于分组码那样有完美的数学理论作支撑，如果只是给定一个(n,k,m)的卷积码，则无法得到自由距离和生成多项式之间的计算公式，必须全部搜索计算所有的编码结构生成函数，才有可能得到具有最大自由距离的好码。随着 k 和 m 的增大，卷积码状态图中的状态数目呈指数增加，使得生成函数的计算变得越来越复杂，甚至达到不可能。因此，目前卷积码的好码一般都是用计算机搜索，列成表格的形式以供查用。表 8-5 给出的就是具有最大自由距离的非恶性码。

表 8-5　　具有最大自由距离的非恶性码（表中子生成元系数为八进制）

码速率	m	$g^{(i,1)}(x)$	$g^{(i,2)}(x)$	$g^{(i,3)}(x)$	$g^{(i,4)}(x)$	d_{free}	d_{free} 上限
1/2	3	5	7			5	5
	4	15	17			6	6
	5	23	35			7	8
	6	53	75			8	8
	7	133	171			10	10
	8	247	371			10	11
	9	561	753			12	12
	10	1167	1545			12	13
	11	2335	3661			14	14
1/3	3	5	7	7		8	8
	4	13	15	17		10	10
	5	25	33	37		12	12
	6	47	53	75		13	13
	7	133	145	175		15	15
	8	225	331	367		16	16
	9	557	663	711		18	18
1/4	3	5	7	7	7	10	10
	4	13	15	15	17	13	15
	5	25	27	33	37	16	16
	6	53	67	71	75	18	18
	7	135	135	147	163	20	20
	8	235	275	313	357	22	22
	9	463	535	733	745	24	24
2/3	2	3 3	1 2	2 3		3	4
	3	2 7	7 5	7 2		5	6
	4	11 16	06 15	15 17		7	7
	5	03 34	16 31	15 17		8	8

例如对于码速率为 1/2、$m=6$ 的卷积码，则有 $(23,35)_8=(010011,011101)_2$，前面的 0 是多余的，所以有

$$G(x)=(1+x^3+x^4,1+x+x^2+x^4)$$

再比如当码速率为 2/3、$m=3$ 的时候，有

$$\begin{bmatrix} 2 & 7 & 7 \\ 7 & 5 & 2 \end{bmatrix}_8 = \begin{bmatrix} 010 & 111 & 111 \\ 111 & 101 & 010 \end{bmatrix}_2$$

所以有

$$\boldsymbol{G}(x) = \begin{bmatrix} x & 1+x+x^2 & 1+x+x^2 \\ 1+x+x^2 & 1+x^2 & x \end{bmatrix}$$

8.4.2 几种类型的卷积码

1. 结尾卷积码

根据前面的分析可以知道，卷积码的码字没有固定的长度，这是和分组码的一个主要区别。但是在实际的应用中，一般需要将卷积码的码字限制在一定的长度范围内，而这个结果可以通过进行所谓的结尾操作来实现。

对于一个(n,k,m)的卷积码，为了使码字的结束状态和初始的全零状态相同，就必须在信息序列中有长度为 km 的全零序列（尾比特），这样得到的码字就是结尾码字。

假设信息序列的长度为 L，则对于(n,k,m)的卷积码，结尾卷积码序列的可以看成码率为 $R'=\dfrac{L-km}{n\left(\dfrac{L}{k}\right)}$ 的分组码。当没有结尾的时候，则其速率仍为 $R=\dfrac{L}{n\left(\dfrac{L}{k}\right)}=\dfrac{k}{n}$。

因此当信息序列的长度 L 远大于 km 的时候，码率的损耗就可以忽略不计了。

2. 删余卷积码

对于一个（n，1，m）卷积码，我们知道其最大的编码效率就是$\dfrac{1}{2}$，即 $n=2$ 时的情况下。而有些应用系统可能要求更高效率（大于$\dfrac{1}{2}$）的编码方案，因此只能选择(n,k,m)（其中 $k>1$）的卷积码。但是对于多个输入的卷积码系统，其译码器电路的复杂度随着输入码元支路数目 k 的增加而呈指数增加。因此较好的方案就是尽可能地减少输入码元支路的数目，尽量使 k 保持为 1。同时，可以在单输入支路的基础上通过删余来提高卷积码的编码效率。

删余就是在编码器输出的码流中系统地输出部分码元，被删码元的个数决定了最终的编码速率，从而提高效率。比如对于前面的(2,1,3)卷积码，其编码效率为$\dfrac{1}{2}$，如果在 4 个比特（此时输入 2 比特）的输出时删掉一个比特，则编码的效率为$\dfrac{2}{3}$（输入 2 比特，输出 3 比特），如果输出为 6 比特时（此时输入为 3 比特），删掉 2 比特，则编码效率为$\dfrac{3}{4}$，删掉 1 比特，则编码效率为$\dfrac{3}{5}$。因此删余的好处就是利用系统的编码器，通过调整删除码元的数目就可以得到一组不同编码效率的编码方案。

在删余的过程中，被删除码元的位置是确定的。通常使用删余矩阵来描述码元删除的情

况。我们用 $\boldsymbol{P}$ 来表示删余矩阵，它是一个 $n \times P_p$ 阶矩阵，其中 P_p 是删余矩阵的周期，删余矩阵中元素值为 1 的位置码元保留，元素值为 0 的位置码元删除。例如假设删余矩阵为

$$\boldsymbol{P} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

则卷积码的效率从 $\frac{1}{2}$ 提高到 $\frac{2}{3}$。

即有：删余前的输出为 $c = c_0^1 c_0^2 c_1^1 c_1^2 c_2^1 c_2^2 c_3^1 c_3^2 \cdots$

删余后的输出为 $c = c_0^1 c_0^2 c_1^1 c_2^1 c_2^2 c_3^1 \cdots$

3．恶性卷积码

有的情况下，当接收序列中有少部分错误，而经过译码却会出现大量的错误，这时该卷积码就叫做恶性卷积码。该过程导致了误差的增加，所以也称作卷积码的误差传播。因此该类的卷积码在实际的应用中应该是尽力避免使用的，一般可以通过其状态转移图来进行区分。图 8-21 就给出两个恶性卷积码的例子。

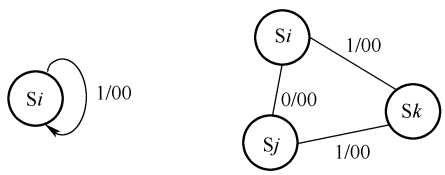


图 8-21 两个恶性卷积码的状态转移图

从图 8-21 中可以看出，当非零序列在卷积码的状态转移图中出现环路，并且其编码输出为全零序列，此卷积码为恶性卷积码。

4．递归卷积码

由于分组码的最小距离由生成矩阵的最大线性无关组的矢量的个数决定，因此在不改变码字最小距离的情况下通过对生成矩阵的初等变换可以得到标准的生成矩阵。我们知道分组码中有系统码的概念，系统码是通过将生成矩阵 $\boldsymbol{G}$ 进行初等变换，得到一个同秩的标准生成矩阵 $\boldsymbol{G'}$，由该标准生成矩阵生成的码序列叫系统码。

对于卷积码，也有类似的概念，也可以在不改变最小距离的情况下改变为系统码的形式。假设卷积码的前馈多项式为 $g_f(D)$，反馈多项式为 $g_b(D)$，信息序列多项式为 $m(D)$，码字多项式为 $c(D)$，则其生成多项式可以写为 $\boldsymbol{G} = \left[g_b(D), g_f(D) \right]$。对于码速率为 $\frac{1}{2}$ 的非系统卷积码而言，系统化的第一步就是计算多项式除法 $\frac{m(D)}{g_b(D)}$ 的余式 $a(D)$；然后利用乘法来计算码字的校验输出 $y(D) = a(D) g_f(D)$，系统的输出就是 $x(D) = m(D)$，即码字输出由信息序列和校验序列构成。这种由前馈多项式和反馈多项式共同决定的系统卷积码叫做递归系统卷积码（Recursive System Convolutional, RSC）。

在递归系统卷积码的编码过程中，需要计算变量和校验输出。

$$\begin{aligned} a_i &= m_i + \sum a_{i-j} g_{bj} \\ yi &= \sum a_{i-j} g_{fj} \end{aligned} \tag{8-61}$$

由于构成传统卷积码编码器的移位寄存器没有反馈部分，因此可以等效为一个有限冲击响应滤波器，而对于递归系统卷积码来说，由于有反馈的存在，其等效为一个无限冲击响应滤波

器，图 8-22 给出了一个(7,5)的递归系统卷积码的例子。图 8-23 所示为该卷积码的状态转移图。

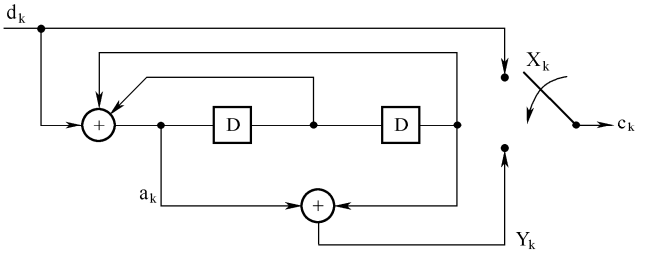


图 8-22 (7,5)递归系统卷积码

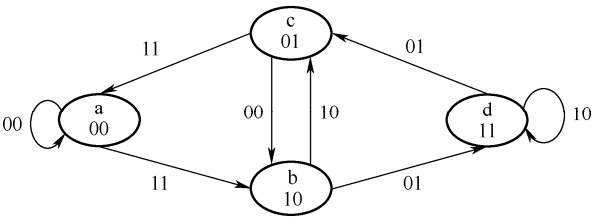


图 8-23 （7，5）递归系统卷积码的状态转移图

注：①图中的椭圆内的值为移位寄存器的状态值；

②实线表示输入值为 0 的情况，虚线表示输入值为 1；

③线上的值表示编码器的输出值。

从图 8-23 可以看出，递归系统卷积码的状态转移图和非系统卷积码的状态转移图类似的，区别主要在于节点 a 和 b 触发状态需要输入的信息比特不同。由于卷积码系统化的时候没有改变生成矩阵的秩，因此系统卷积码的最小距离和相应的非系统卷积码的最小距离是相同的。

根据上一节的讨论我们知道，对于传统的非系统卷积码，在数据帧的末尾添加 km（k 为输入比特流的支路数目，m 是各支路移位寄存器的长度）个零就可以使所有的移位寄存器清零，即回到全零状态。而递归系统卷积码具有无限冲激响应特性，因此仅靠插入 km 个零一般不会回到全零状态。为了能够使递归系统卷积码在状态转移图中能够回到全零状态，必须选择输入信息 m_i，使得余式的对应系数 $a_i = 0\ (n - km \leqslant i \leqslant n - 1)$，这样系统卷积码最后 km 个比特必须满足

$$m_i = \sum_j a_{i-j} g_{bj}$$

递归系统卷积码 RSC 是 Turbo 码的首选方案，具体的内容在第 9 章讨论。

8.5 卷积码的应用

卷积码由于具有良好的纠错性能，在无线通信中使用得非常广泛。在 GSM 系统、cdma 系统以及 OFDM 系统中均有使用，下面对它在移动通信系统中的使用进行分析。

8.5.1 交织编码

交织码主要用于有记忆的信道，特别是无线移动信道。交织码的基本思路与前面介绍的纠错码思路不同，纠错码是为了适应信道，而交织码则是为了改造信道。即将一个有记忆的

突发信道经过交织、去交织变换将信道改造成独立无记忆信道。然后再采用纠正独立随机差错的纠错码，充分发挥其纠错功能。

分组交织（块交织）的基本原理如图 8-24 所示。

① 若待发送的一组信息为

$$\boldsymbol{X}=(x_1, x_2, x_3, \cdots, x_{24}, x_{25})$$

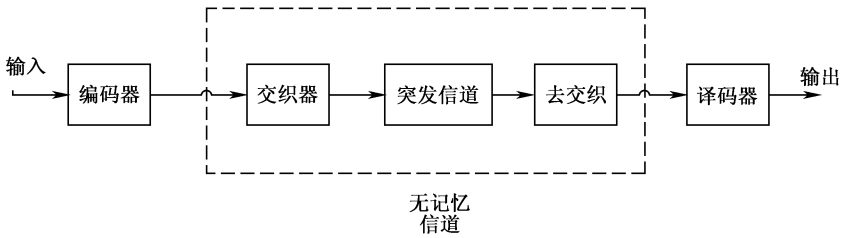


图 8-24 分组交织系统框图

② 交织存储器为一个行列交织矩阵，它按列写入，按行读出。

$$\boldsymbol{A}_1=\begin{pmatrix} x_1 & x_6 & x_{11} & x_{16} & x_{21} \\ x_2 & x_7 & x_{12} & x_{17} & x_{22} \\ x_3 & x_8 & x_{13} & x_{18} & x_{23} \\ x_4 & x_9 & x_{14} & x_{19} & x_{24} \\ x_5 & x_{10} & x_{15} & x_{20} & x_{25} \end{pmatrix}$$

③ 交织器输出并送入突发信道的信息为

$$\boldsymbol{X}'=(x_1\ x_6\ x_{11}\ x_{16}\ x_{21}\ x_2\ x_7\ x_{12}\ x_{17}\ x_{22}\ \cdots x_5\ x_{10}\ x_{15}\ x_{20}\ x_{25})$$

④ 假设突发信道产生了两个突发差错，第一个产生于 x_1 至 x_{21} 连错 5 位，第二个突发产生于 x_{17} 至 x_8 连错 4 位。

⑤ 突发信道输出端信息为 $\boldsymbol{X}''$，它可表示为

$$\boldsymbol{X}''=(\dot{x}_1\ \dot{x}_6\ \dot{x}_{11}\ \dot{x}_{16}\ \dot{x}_{21}\ x_2\ x_7\ x_{12}\ \dot{x}_{17}\ \dot{x}_{22}\dot{x}_3\ \dot{x}_8\ \cdots x_5\ x_{10}\ x_{15}\ x_{20}\ x_{25})$$

⑥ 在接收端，进入去交织器后，送入另一存储器，它也是一个行列交织矩阵，但是它是按行写入按列读出。

$$\boldsymbol{A}_2=\begin{pmatrix} x_1 & x_6 & x_{11} & x_{16} & x_{21} \\ x_2 & x_7 & x_{12} & x_{17} & x_{22} \\ x_3 & x_8 & x_{13} & x_{18} & x_{23} \\ x_4 & x_9 & x_{14} & x_{19} & x_{24} \\ x_5 & x_{10} & x_{15} & x_{20} & x_{25} \end{pmatrix}$$

⑦ 去交织存储器的输出为 $\boldsymbol{X}'''$。

$$\boldsymbol{X}'''=(\dot{x}_1x_2\dot{x}_3x_4x_5\dot{x}_6x_7\dot{x}_8x_9x_{10}\dot{x}_{11}x_{12}\ \cdots\ x_{15}\ x_{20}\ x_{25})$$

⑧ 由上面分析可见，经过交织矩阵与交织矩阵的变换之后，原来信道中的突发差错，即两个突发：连错 5 位与连错 4 位，却变成了 $\boldsymbol{X}'''$ 中随机性的独立差错。

若将上述例子推广至一般情况，设分组长度 $L=M\times N$,即 M 列 N 行的矩阵，利用这一行列交织特性，可实现下列分组交织特性。

① 任何长度 $L\leqslant M$ 的突发差错，经交织后成为至少被 N–1 位隔开后的一些单个独立差错。

② 任何长度 $L>M$ 的突发差错，经交织后可将长突发差错变成长度为 $l_1=\left[\frac{l}{M}\right]$ 的突发差错。

③ 完成上述交织与去交织变换，在不计信道延时条件下，将产生 $2MN$ 个符号的时延，其中收、发端各占一半。

④ 在很特殊情况下，周期为 M 的 k 个单独差错，经交织与去交织后将会产生长度为 L 的突发差错。

为了克服分组交织器的两个主要缺点：时延大与在特殊情况下，可将周期性单个差错交织或突发差错，人们又提出了卷积交织器与随机交织器，相关内容参见有关教科书。

8.5.2 卷积码在移动通信中的应用

1. GSM 中的信道编码

信道编码已广泛使用在移动通信中。在 GSM 中信道（链路）按其功能可分为业务链路（TCH）和控制链路（CCH），其中业务链路用于传送话音与数据，控制链路用于传送信令和同步信息。业务链路中，全速率语音链路表示为 TCH/FS。数据业务链路的表示举例如下：4.8kbit/s 全速率数据业务链路表示为 TCH/F4.8。而 4.8kbit/s 半速率数据业务链路表示为 TCH/H4.8。所谓全速率与半速率，取决于移动终端的类型，由它们来确定移动通信系统的空中接口。具体编、译码方案对不同类型链路采用不同的方案，在本节我们只涉及全速率语音业务的编码系统，其他信道的编码可参考有关书籍或专业文献。

语音编码是逐帧进行的，语音 20ms 帧共计 260 比特，折合成 13bit/s（260 × 50）。其 260 比特的数据序列表示为：$d=\{d(0),d(1),\cdots,d(181),d(182),\cdots,d(259)\}$，其中前 182bit 称为第一类比特，这是在语音中比较重要的比特，它对传输误差最敏感，即这些比特中如果产生差错，将会严重影响语音质量，因此它们必须受到完全保护。第一类比特（182bit）中前 50bit 是重中之重，它不仅受内码纠错保护，还要受到外码检错的双重保护。第一类加上 3 比特的奇偶校验和 4 比特的尾比特共 189 比特，它们采用(2,1,4)卷积码，共输出 378 比特。第 182bit 以后的 78bit，称为第二类比特，它不参加检纠错编码而仅参加交织编码。因此每 20ms 帧的 260 比特数据在加卷积码的信道编码（也叫前向编码）后的总比特数为 456 比特，此时的系统速率就为 $\frac{456\text{bit}}{20\text{ms}}=22.8\text{kbit/s}$。卷积码编码后就是交织，GSM 系统采用的交织深度为 8，对 40ms 的数据进行处理，即对 912 比特的数据进行交织。

（1）GSM 系统中的卷积码。GSM 中全业务信道使用卷积码的生成多项式为

$$\begin{aligned} g^{(1,1)}(x)&=1+x^3+x^4\\ g^{(1,2)}(x)&=1+x+x^3+x^4 \end{aligned} \tag{8-62}$$

故可以写出它相应的编码结构，如图 8-25 所示。

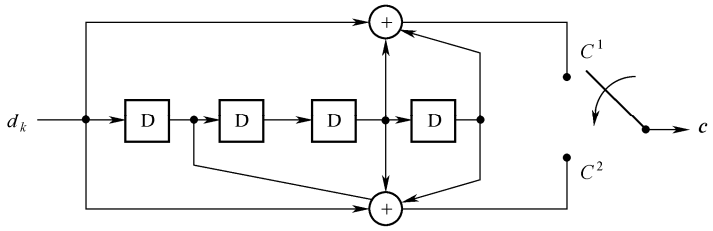


图 8-25 GSM 系统使用的（2，1，5）卷积码

（2）重排和交织。

① GSM 中交织器的位置是在编码器以后的最后基带处理，然后经重复后送入信道。

② 其交织分为两步：首先是按照某种模运算规则对数据进行重排，然后再按照一定规进行帧间交织。

③ 具体处理过程如下。

（a）先将由内编码得到的每个语音帧 456bit，按照安排表格重排后分成 8 个子块，每个子块 57bit，分散至 8 个 TDMA 帧中。

（b）重排公式为 $D(x,y) = 57x + 64y$,mod456，其中 $x = 0$，1，2，…，7，表示每个 456bit 语音帧被分成 8 个子块的序号（TDMA 帧号），$y = 0,1,2,\cdots,56$,表示每个子块中比特序号。

④ 重排交织原理性方框图如图 8-26 所示。

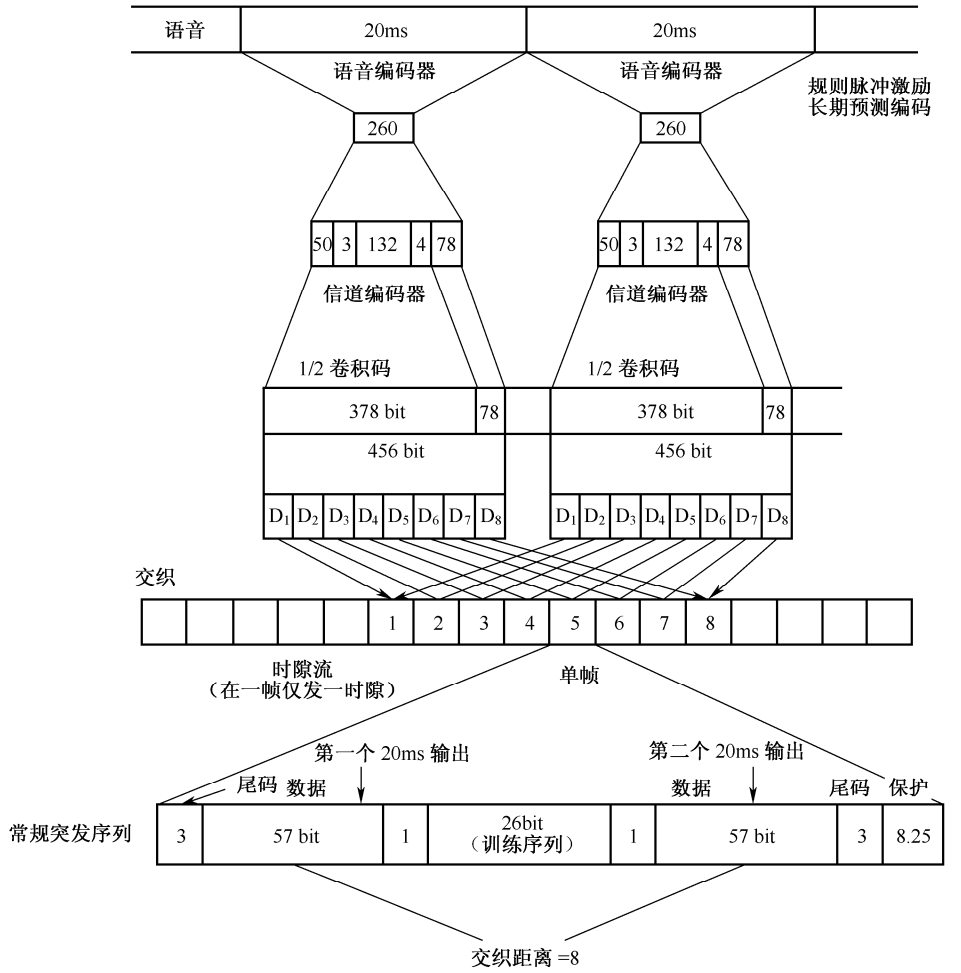


图 8-26 GSM 的编码、重排与交织

2．CDMA 中的卷积码

根据前面的分析我们知道，在复杂度相当的情况下，卷积码的性能要好于分组码，因此 CDMA 系统广泛地使用了卷积码。3G 系统（WCDMA 和 cdma2000）中使用的卷积码的码型和编码结构基本上是对窄带 CDMA 系统的卷积码的继承，因此我们主要介绍 IS-95 系统中纠错码的设计和使用。

IS-95 中的上行业务信道交织过程如图 8-27 所示。

图 8-27 IS-95 上行业务信道交织过程

交织器块分组交织长度为 20ms 一个语音帧，交织矩阵是一个 32 行 18 列的矩阵，即 $32 \times 18 = 576$ 单元符号。

IS-95 上行业务信道交织矩阵操作顺序如图 8-28 所示。

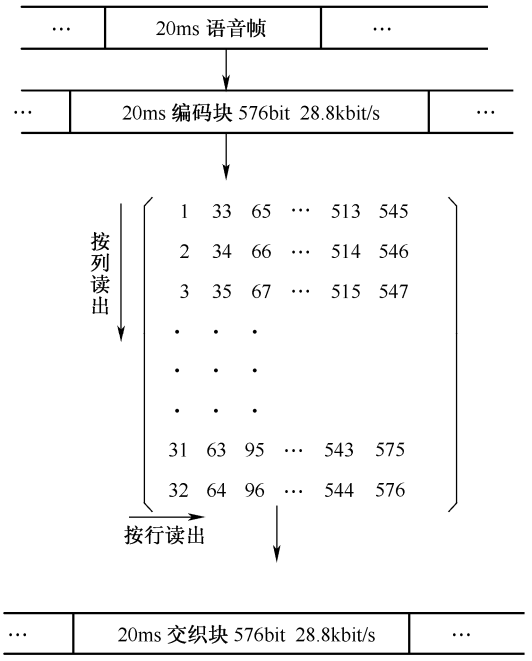


图 8-28 IS-95 上行业务信道交织顺序

IS-95 下行信道使用的是同步码分多址，而上行信道是异步码分多址，因此上行信道要求具有更强纠错能力的卷积码。

下行信道中，同步、寻呼以及前向业务信道均采用(2,1,9)卷积码，该结构是可以产生最大自由距离的好码。该卷积码编码结构如图 8-29 所示。生成多项式系数：$(753,\ 561)_8$=(111101011, 101110001)。

它的生成多项式为

$$\begin{aligned} g^{(1,1)}(x) &= 1 + x + x^2 + x^3 + x^5 + x^7 + x^8 \\ g^{(1,2)}(x) &= 1 + x^2 + x^3 + x^4 + x^8 \\ \boldsymbol{G}(x) &= [g^{(1,1)}(x), g^{(1,2)}(x)] \end{aligned} \tag{8-63}$$

类似的上行业务信道均采用（3，1，9）卷积码，该结构也是可以产生最大自由距离的好码，编码器结构如图 8-30 所示。生成多项式系数

$$(557, 663, 711)_8 = (101101111, 110110001, 111001001)$$

它的生成多项式为

$$\begin{aligned} g^{(1,1)}(x) &= 1 + x^2 + x^3 + x^5 + x^6 + x^7 + x^8 \\ g^{(1,2)}(x) &= 1 + x + x^3 + x^4 + x^7 + x^8 \\ g^{(1,3)}(x) &= 1 + x + x^2 + x^5 + x^8 \end{aligned} \tag{8-64}$$

$$G(x)=[g^{(1,1)}(x),g^{(1,2)}(x),g^{(1,3)}(x)]$$

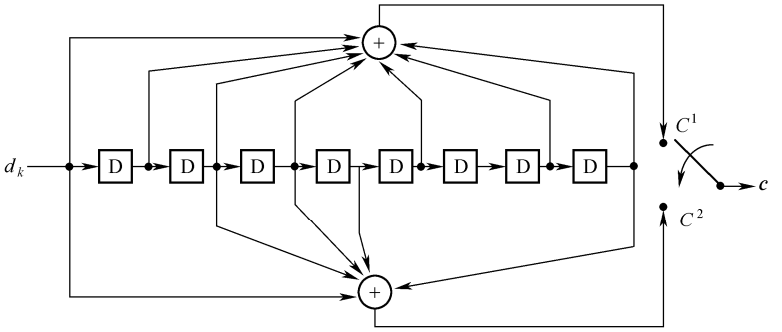


图 8-29 cdma 系统中（2，1，9）卷积码结构

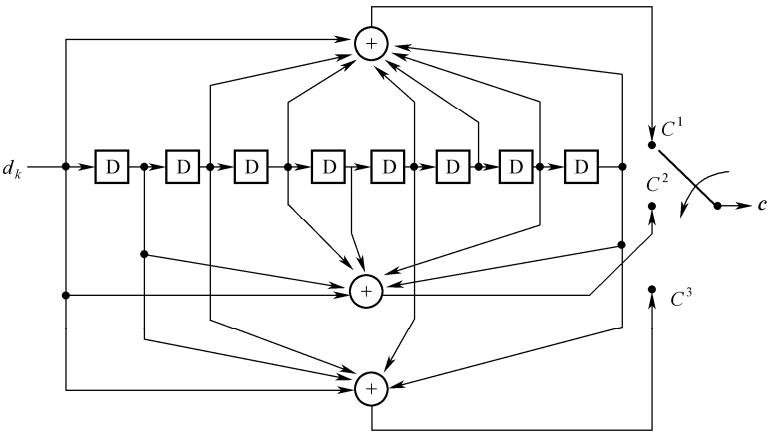


图 8-30 cdma 系统中（3，1，9）卷积码结构

8.6 级联编码

级联码是由短码构造长码的一种有效的方法，它是由 Forney 首先提出。级联码可用于纠正混合差错。既有独立随机差错又有突发差错，且纠错能力很强。级联码是由内外两个短码构成一个长码,内码可设计成一般复杂度以纠正随机独立差错为主的纠错码，外码可设计为较复杂，且纠错能力较强，即可纠正内码不能纠正的随机独立差错和部分突发差错的码。

理论上讲，无论是内码还是外码，均可采用分组码和卷积码，但目前采用最多的是内码为卷积码,外码是 RS 分组码。

1．基本原理

它由两个短码即内码与外码串接而成:即$(n,k)=[n_1\times n_2,k_1\times k_2]=[(n_1,k_1)(n_2,k_2)]$。

称(n_1,k_1)为内码，(n_2,k_2)为外码。因为一组 k 位信息可分解为 k_2 个字节，而每个字节中有 k_1 个信息位，(n_1,k_1)可纠字节内独立随机差错,一般由卷积码构成,称为内码; (n_2,k_2)可

纠正余下的差错以及字节间的突发差错，一般由纠错能力强的 RS 码构成,称为外码。

基本方框图如图 8-31 所示。

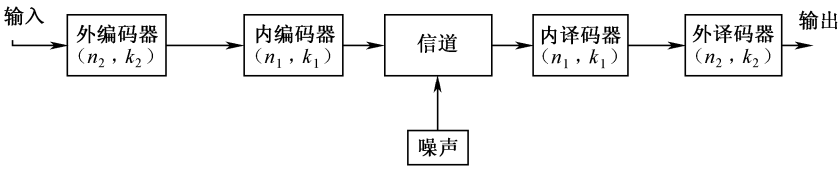


图 8-31 级联编码原理

若(n_1,k_1)最小距离为 d_1，(n_2,k_2)最小距离为 d_2，串接后级联码最小距离 $d \geqslant d_1 \times d_2$。

级联码结构是由内外码串接而成，又称为串行级联码，其设备仅是两者的直接组合，它要比采用一个长码时所需的设备简单得多。

2．级联码标准

① 最早采用级联码的是 20 世纪 80 年代美国宇航局（NASA）将它用于深空遥测数据的纠错。

② 1987 年 NASA 制定了级联码深空遥测标准 CCSDS。

典型标准级联码系统框图如 8-32 所示。

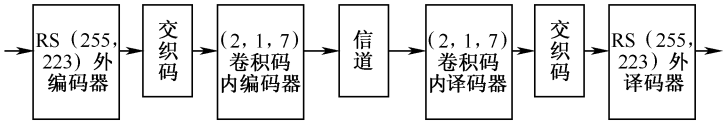


图 8-32 级联码框图

1984 年，NASA 采用上述的典型标准级联系统，内码为(2,1,7)卷积码，外码为(255,223)RS 码。而交织器深度为 2～8 个外码块，其性能很优异。当 $E_b/N_0=2.53$dB，误码率达 $P_b \leqslant 10^{-6}$。

3．级联码的性能

表 8-6 给出了在 $P_b \leqslant 10^{-6}$ 条件下（P_b 为误比特率）的性能。

表 8-6 **一些级联码的性能**

内码	外码	$P_b=10^{-6}$ 条件下 E_b/N_0（dB）	较标准（基准）系统功率增益（dB）
(4,1,5)	(255,223)	0.91	1.62
(5,1,14)	(1023,927)	0.57	1.96
(5,1,15)	(1023,959)	0.50	2.03

(6,1,14)	(1023,959)	0.47	2.06
(6,1,15)	(1023,959)	0.42	2.11
(5,1,13)	(255,229)	0.91	1.62
(5,1,14)	(255,231)	0.79	1.74
(6,1,14)	(255,233)	0.63	1.90

附：

1. 下面的程序是基于 MATLAB 仿真的例 8-3 中（2，1，3）编码结构的卷积码:

主程序:

```
clear all
L = 100;  %编码长度
I = randint（1,L）; 产生 L 个随机二进制数
G_0 = [1 1  0 ;0 1 1];
G_1 = [0 0 1 ;1 1 0];
n = 3;
b = 2;
k = 1;
I_e = [];
ss = [];I_e = [];
C_out = encoding（b,n,k,L,I,G_0,G_1）;% 编码输出


ss = viterbi_dec1（b,n,k,C_err,G_0,G_1）; 译码输出
for r = 1:L
    I_e（r） = ss（r）;
end
p = biterr（I,I_e）/L;  误码比较
```

encoding 函数

```
function[C_out] = encoding（b,n,k,L,I,G_0,G_1）
I_t = zeros（1,b）;
j = L + b*k;
I_add = zeros（1,j）;
for r = 1:L
    I_add（r） = I（r）;
end          %add b*k  0
m = L/b + k;
C_out = [];
for s = 1:m
    C_s_2 = I_t*G_1;
    for r = 1:b
        I_s（r） = I_add（r +（s-1）*b）;
    end
        I_t = I_s;
        C_s_1 = I_s*G_0;
        C_s = C_s_1 + C_s_2;
        C_s = mod（C_s,2）;
        C_out = [C_out C_s]
End
```

Viterbi_dec1 函数

```
function[sur_path_end] = viterbi_dec1
（b,n,k,C_out,g_0,g_1）
sur_add = [];
sur_path_end = [];
sd = [];
sur_path = [];%viterbi_dec
s_in = [];
sur_path_f = [];
L_t = size（C_out）;
L_dec = L_t（2）/n;
C = zeros（1,n）;
state = [0 0];

%------------------first step------------------%
  for s = 1:n
      C（s） = C_out（s）;
   end
sd_1 = compare_con1（state,C,g_0,g_1）;%the first step
for r = 1:8 %output of 4 paths

    s_in（r） = sd_1（r）;
end
```

```
xs = 1:4;       %constructrue 4x2 s_path matrix
sur_path（xs,1）= s_in（xs）;
sur_path（xs,2）= s_in（xs + 4）;
sur_path_1 = sur_path;% copy
for r = 1:4    % 4x1 sur_add matrix
    sur_add_t（r）= sd_1（8 + r）;
end
sur_add = sur_add_t';
%-------------------------------------------------%

%--------------second to end step-----------------%

for r = 2:L_dec   %  L_dec bits divided to the length of n' samller blocks
    for s = 1:n
        C( s ) = C_out( s + ( r-1 )*n );% recieve C of r step
    end
    for x = 1:4
        state = sur_path_1（x,:）; % 4 path extended by evryone node
        sd = compare_con（state,C,g_0,g_1）;  % return 2bits sur_path and 1bit sur_add
        for r = 1:2
            sur_path_t（x,r）= sd（r）;
        end
%       sur_path_f = sur_path（x,:）;
        s_add_t = sd（3）;
%       sur_path（x,:）= [sur_path（x,:）sur_path_t];
        sur_add（x）= sur_add（x）+ s_add_t;    % add refresh
%       sur_path（x,:）=sur_path_f;
    end
    sur_path = [sur_path sur_path_t];
    sur_path_1 = sur_path_t;
end

% ------- find optional path----------------%
dd = min（sur_add）;

for sw = 1:4
    if sur_add（sw）==dd
        a = sw;
    end
end
sur_path_end = sur_path（a,:）;
```

compare_con 函数

```
function[s_in] = compare_con（state,C,g_0,g_1）
s_in = [];
a = 0;
nn = zeros（4,1）;
state_in = [0 0;0 1;1 0;1 1];
state_c = [state;state;state;state];
t_out = state_in*g_0 + state_c*g_1;
T_out = mod（t_out,2）;
for s = 1:4    %cmparence and save
    for t = 1:3
        if C（t）~ = T_out（s,t）
            nn（s）= nn（s）+ 1;
        end
    end
end
d = min（nn）;

for sw = 1:4
    if nn（sw）==d
        a = sw;
    end
end
s_in = [state_in（a,:）  d];
```

compare_con1 函数

```
function[s_in] = compare_con1（state,C,g_0,g_1）
s_in = [];
a = 0;
nn = zeros（4,1）;
state_in = [0 0;0 1;1 0;1 1];
state_c = [state;state;state;state];
t_out = state_in*g_0 + state_c*g_1;
T_out = mod（t_out,2）;
for s = 1:4    %cmparence and save
    for t = 1:3
        if C（t）~ = T_out（s,t）
            nn（s）= nn（s）+ 1;
        end
    end
```

```
end
% d = min（nn）;

% for sw = 1:4
%       if nn（sw）==d
%             a = sw;
%       end
% end
s_in = [state_in nn];
```

2．GSM 中解交织程序（c 语言）

```
//子函数 deinterleave
//功能：解交织
//输入：8 × 114
//输出：1 × 456
//**************************************
*************
void deinterleave（int *in,int *out）
{int out[456],mid[9][115],demidter[457];
int i,j;
for（i = 0;i<8;i++）
{for（j = 0;j<114;j++）
mid[i + 1][j + 1] = in[i][j];
}                                          //将输入
的 8 × 114 数据放入 mid 中，不包括第 0 行第 0 列

demidter[1] = mid[1][1];  //解交织，每个元素
对应于输出中的 1 个元素
demidter[2] = mid[2][99];
demidter[3] = mid[3][83];
demidter[4] = mid[4][67];
demidter[5] = mid[5][52];
demidter[6] = mid[6][36];
demidter[7] = mid[7][20];
demidter[8] = mid[8][4];
demidter[9] = mid[1][101];
demidter[10] = mid[2][85];
demidter[11] = mid[3][69];
demidter[12] = mid[4][53];
demidter[13] = mid[5][38];
demidter[14] = mid[6][22];
demidter[15] = mid[7][6];
demidter[16] = mid[8][104];
demidter[17] = mid[1][87];
demidter[18] = mid[2][71];
demidter[19] = mid[3][55];
demidter[20] = mid[4][39];
demidter[21] = mid[5][24];
demidter[22] = mid[6][8];
demidter[23] = mid[7][106];
demidter[24] = mid[8][90];
demidter[25] = mid[1][73];
demidter[26] = mid[2][57];
demidter[27] = mid[3][41];
demidter[28] = mid[4][25];
demidter[29] = mid[5][10];
demidter[30] = mid[6][108];
demidter[31] = mid[7][92];
demidter[32] = mid[8][76];
demidter[33] = mid[1][59];
demidter[34] = mid[2][43];
demidter[35] = mid[3][27];
demidter[36] = mid[4][11];
demidter[37] = mid[5][110];
demidter[38] = mid[6][94];
demidter[39] = mid[7][78];
demidter[40] = mid[8][62];
demidter[41] = mid[1][45];
demidter[42] = mid[2][29];
demidter[43] = mid[3][13];
demidter[44] = mid[4][111];
demidter[45] = mid[5][96];
demidter[46] = mid[6][80];
demidter[47] = mid[7][64];
demidter[48] = mid[8][48];
demidter[49] = mid[1][31];
demidter[50] = mid[2][15];
demidter[51] = mid[3][113];
demidter[52] = mid[4][97];
demidter[53] = mid[5][82];
demidter[54] = mid[6][66];
demidter[55] = mid[7][50];
demidter[56] = mid[8][34];
demidter[57] = mid[1][17];
demidter[58] = mid[2][1];
demidter[59] = mid[3][99];
demidter[60] = mid[4][83];
demidter[61] = mid[5][68];
```

```
demidter[62] = mid[6][52];
demidter[63] = mid[7][36];
demidter[64] = mid[8][20];
demidter[65] = mid[1][3];
demidter[66] = mid[2][101];
demidter[67] = mid[3][85];
demidter[68] = mid[4][69];
demidter[69] = mid[5][54];
demidter[70] = mid[6][38];
demidter[71] = mid[7][22];
demidter[72] = mid[8][6];
demidter[73] = mid[1][103];
demidter[74] = mid[2][87];
demidter[75] = mid[3][71];
demidter[76] = mid[4][55];
demidter[77] = mid[5][40];
demidter[78] = mid[6][24];
demidter[79] = mid[7][8];
demidter[80] = mid[8][106];
demidter[81] = mid[1][89];
demidter[82] = mid[2][73];
demidter[83] = mid[3][57];
demidter[84] = mid[4][41];
demidter[85] = mid[5][26];
demidter[86] = mid[6][10];
demidter[87] = mid[7][108];
demidter[88] = mid[8][92];
demidter[89] = mid[1][75];
demidter[90] = mid[2][59];
demidter[91] = mid[3][43];
demidter[92] = mid[4][27];
demidter[93] = mid[5][12];
demidter[94] = mid[6][110];
demidter[95] = mid[7][94];
demidter[96] = mid[8][78];
demidter[97] = mid[1][61];
demidter[98] = mid[2][45];
demidter[99] = mid[3][29];
demidter[100] = mid[4][13];
demidter[101] = mid[5][112];
demidter[102] = mid[6][96];
demidter[103] = mid[7][80];
demidter[104] = mid[8][64];
demidter[105] = mid[1][47];
demidter[106] = mid[2][31];
demidter[107] = mid[3][15];
demidter[108] = mid[4][113];
demidter[109] = mid[5][98];
demidter[110] = mid[6][82];
demidter[111] = mid[7][66];
demidter[112] = mid[8][50];
demidter[113] = mid[1][33];
demidter[114] = mid[2][17];
demidter[115] = mid[3][1];
demidter[116] = mid[4][99];
demidter[117] = mid[5][84];
demidter[118] = mid[6][68];
demidter[119] = mid[7][52];
demidter[120] = mid[8][36];
demidter[121] = mid[1][19];
demidter[122] = mid[2][3];
demidter[123] = mid[3][101];
demidter[124] = mid[4][85];
demidter[125] = mid[5][70];
demidter[126] = mid[6][54];
demidter[127] = mid[7][38];
demidter[128] = mid[8][22];
demidter[129] = mid[1][5];
demidter[130] = mid[2][103];
demidter[131] = mid[3][87];
demidter[132] = mid[4][71];
demidter[133] = mid[5][56];
demidter[134] = mid[6][40];
demidter[135] = mid[7][24];
demidter[136] = mid[8][8];
demidter[137] = mid[1][105];
demidter[138] = mid[2][89];
demidter[139] = mid[3][73];
demidter[140] = mid[4][57];
demidter[141] = mid[5][42];
demidter[142] = mid[6][26];
demidter[143] = mid[7][10];
demidter[144] = mid[8][108];
demidter[145] = mid[1][91];
demidter[146] = mid[2][75];
demidter[147] = mid[3][59];
demidter[148] = mid[4][43];
demidter[149] = mid[5][28];
```

```
demidter[150] = mid[6][12];
demidter[151] = mid[7][110];
demidter[152] = mid[8][94];
demidter[153] = mid[1][77];
demidter[154] = mid[2][61];
demidter[155] = mid[3][45];
demidter[156] = mid[4][29];
demidter[157] = mid[5][14];
demidter[158] = mid[6][112];
demidter[159] = mid[7][96];
demidter[160] = mid[8][80];
demidter[161] = mid[1][63];
demidter[162] = mid[2][47];
demidter[163] = mid[3][31];
demidter[164] = mid[4][15];
demidter[165] = mid[5][114];
demidter[166] = mid[6][98];
demidter[167] = mid[7][82];
demidter[168] = mid[8][66];
demidter[169] = mid[1][49];
demidter[170] = mid[2][33];
demidter[171] = mid[3][17];
demidter[172] = mid[4][1];
demidter[173] = mid[5][100];
demidter[174] = mid[6][84];
demidter[175] = mid[7][68];
demidter[176] = mid[8][52];
demidter[177] = mid[1][35];
demidter[178] = mid[2][19];
demidter[179] = mid[3][3];
demidter[180] = mid[4][101];
demidter[181] = mid[5][86];
demidter[182] = mid[6][70];
demidter[183] = mid[7][54];
demidter[184] = mid[8][38];
demidter[185] = mid[1][21];
demidter[186] = mid[2][5];
demidter[187] = mid[3][103];
demidter[188] = mid[4][87];
demidter[189] = mid[5][72];
demidter[190] = mid[6][56];
demidter[191] = mid[7][40];
demidter[192] = mid[8][24];
demidter[193] = mid[1][7];
demidter[194] = mid[2][105];
demidter[195] = mid[3][89];
demidter[196] = mid[4][73];
demidter[197] = mid[5][58];
demidter[198] = mid[6][42];
demidter[199] = mid[7][26];
demidter[200] = mid[8][10];
demidter[201] = mid[1][107];
demidter[202] = mid[2][91];
demidter[203] = mid[3][75];
demidter[204] = mid[4][59];
demidter[205] = mid[5][44];
demidter[206] = mid[6][28];
demidter[207] = mid[7][12];
demidter[208] = mid[8][110];
demidter[209] = mid[1][93];
demidter[210] = mid[2][77];
demidter[211] = mid[3][61];
demidter[212] = mid[4][45];
demidter[213] = mid[5][30];
demidter[214] = mid[6][14];
demidter[215] = mid[7][112];
demidter[216] = mid[8][96];
demidter[217] = mid[1][79];
demidter[218] = mid[2][63];
demidter[219] = mid[3][47];
demidter[220] = mid[4][31];
demidter[221] = mid[5][16];
demidter[222] = mid[6][114];
demidter[223] = mid[7][98];
demidter[224] = mid[8][82];
demidter[225] = mid[1][65];
demidter[226] = mid[2][49];
demidter[227] = mid[3][33];
demidter[228] = mid[4][17];
demidter[229] = mid[5][2];
demidter[230] = mid[6][100];
demidter[231] = mid[7][84];
demidter[232] = mid[8][68];
demidter[233] = mid[1][51];
demidter[234] = mid[2][35];
demidter[235] = mid[3][19];
demidter[236] = mid[4][3];
demidter[237] = mid[5][102];
```

```
demidter[238] = mid[6][86];
demidter[239] = mid[7][70];
demidter[240] = mid[8][54];
demidter[241] = mid[1][37];
demidter[242] = mid[2][21];
demidter[243] = mid[3][5];
demidter[244] = mid[4][103];
demidter[245] = mid[5][88];
demidter[246] = mid[6][72];
demidter[247] = mid[7][56];
demidter[248] = mid[8][40];
demidter[249] = mid[1][23];
demidter[250] = mid[2][7];
demidter[251] = mid[3][105];
demidter[252] = mid[4][89];
demidter[253] = mid[5][74];
demidter[254] = mid[6][58];
demidter[255] = mid[7][42];
demidter[256] = mid[8][26];
demidter[257] = mid[1][9];
demidter[258] = mid[2][107];
demidter[259] = mid[3][91];
demidter[260] = mid[4][75];
demidter[261] = mid[5][60];
demidter[262] = mid[6][44];
demidter[263] = mid[7][28];
demidter[264] = mid[8][12];
demidter[265] = mid[1][109];
demidter[266] = mid[2][93];
demidter[267] = mid[3][77];
demidter[268] = mid[4][61];
demidter[269] = mid[5][46];
demidter[270] = mid[6][30];
demidter[271] = mid[7][14];
demidter[272] = mid[8][112];
demidter[273] = mid[1][95];
demidter[274] = mid[2][79];
demidter[275] = mid[3][63];
demidter[276] = mid[4][47];
demidter[277] = mid[5][32];
demidter[278] = mid[6][16];
demidter[279] = mid[7][114];
demidter[280] = mid[8][98];
demidter[281] = mid[1][81];
demidter[282] = mid[2][65];
demidter[283] = mid[3][49];
demidter[284] = mid[4][33];
demidter[285] = mid[5][18];
demidter[286] = mid[6][2];
demidter[287] = mid[7][100];
demidter[288] = mid[8][84];
demidter[289] = mid[1][67];
demidter[290] = mid[2][51];
demidter[291] = mid[3][35];
demidter[292] = mid[4][19];
demidter[293] = mid[5][4];
demidter[294] = mid[6][102];
demidter[295] = mid[7][86];
demidter[296] = mid[8][70];
demidter[297] = mid[1][53];
demidter[298] = mid[2][37];
demidter[299] = mid[3][21];
demidter[300] = mid[4][5];
demidter[301] = mid[5][104];
demidter[302] = mid[6][88];
demidter[303] = mid[7][72];
demidter[304] = mid[8][56];
demidter[305] = mid[1][39];
demidter[306] = mid[2][23];
demidter[307] = mid[3][7];
demidter[308] = mid[4][105];
demidter[309] = mid[5][90];
demidter[310] = mid[6][74];
demidter[311] = mid[7][58];
demidter[312] = mid[8][42];
demidter[313] = mid[1][25];
demidter[314] = mid[2][9];
demidter[315] = mid[3][107];
demidter[316] = mid[4][91];
demidter[317] = mid[5][76];
demidter[318] = mid[6][60];
demidter[319] = mid[7][44];
demidter[320] = mid[8][28];
demidter[321] = mid[1][11];
demidter[322] = mid[2][109];
demidter[323] = mid[3][93];
demidter[324] = mid[4][77];
demidter[325] = mid[5][62];
```

```
demidter[326] = mid[6][46];
demidter[327] = mid[7][30];
demidter[328] = mid[8][14];
demidter[329] = mid[1][111];
demidter[330] = mid[2][95];
demidter[331] = mid[3][79];
demidter[332] = mid[4][63];
demidter[333] = mid[5][48];
demidter[334] = mid[6][32];
demidter[335] = mid[7][16];
demidter[336] = mid[8][114];
demidter[337] = mid[1][97];
demidter[338] = mid[2][81];
demidter[339] = mid[3][65];
demidter[340] = mid[4][49];
demidter[341] = mid[5][34];
demidter[342] = mid[6][18];
demidter[343] = mid[7][2];
demidter[344] = mid[8][100];
demidter[345] = mid[1][83];
demidter[346] = mid[2][67];
demidter[347] = mid[3][51];
demidter[348] = mid[4][35];
demidter[349] = mid[5][20];
demidter[350] = mid[6][4];
demidter[351] = mid[7][102];
demidter[352] = mid[8][86];
demidter[353] = mid[1][69];
demidter[354] = mid[2][53];
demidter[355] = mid[3][37];
demidter[356] = mid[4][21];
demidter[357] = mid[5][6];
demidter[358] = mid[6][104];
demidter[359] = mid[7][88];
demidter[360] = mid[8][72];
demidter[361] = mid[1][55];
demidter[362] = mid[2][39];
demidter[363] = mid[3][23];
demidter[364] = mid[4][7];
demidter[365] = mid[5][106];
demidter[366] = mid[6][90];
demidter[367] = mid[7][74];
demidter[368] = mid[8][58];
demidter[369] = mid[1][41];
demidter[370] = mid[2][25];
demidter[371] = mid[3][9];
demidter[372] = mid[4][107];
demidter[373] = mid[5][92];
demidter[374] = mid[6][76];
demidter[375] = mid[7][60];
demidter[376] = mid[8][44];
demidter[377] = mid[1][27];
demidter[378] = mid[2][11];
demidter[379] = mid[3][109];
demidter[380] = mid[4][93];
demidter[381] = mid[5][78];
demidter[382] = mid[6][62];
demidter[383] = mid[7][46];
demidter[384] = mid[8][30];
demidter[385] = mid[1][13];
demidter[386] = mid[2][111];
demidter[387] = mid[3][95];
demidter[388] = mid[4][79];
demidter[389] = mid[5][64];
demidter[390] = mid[6][48];
demidter[391] = mid[7][32];
demidter[392] = mid[8][16];
demidter[393] = mid[1][113];
demidter[394] = mid[2][97];
demidter[395] = mid[3][81];
demidter[396] = mid[4][65];
demidter[397] = mid[5][50];
demidter[398] = mid[6][34];
demidter[399] = mid[7][18];
demidter[400] = mid[8][2];
demidter[401] = mid[1][99];
demidter[402] = mid[2][83];
demidter[403] = mid[3][67];
demidter[404] = mid[4][51];
demidter[405] = mid[5][36];
demidter[406] = mid[6][20];
demidter[407] = mid[7][4];
demidter[408] = mid[8][102];
demidter[409] = mid[1][85];
demidter[410] = mid[2][69];
demidter[411] = mid[3][53];
demidter[412] = mid[4][37];
```

```
demidter[413] = mid[5][22];
demidter[414] = mid[6][6];
demidter[415] = mid[7][104];
demidter[416] = mid[8][88];
demidter[417] = mid[1][71];
demidter[418] = mid[2][55];
demidter[419] = mid[3][39];
demidter[420] = mid[4][23];
demidter[421] = mid[5][8];
demidter[422] = mid[6][106];
demidter[423] = mid[7][90];
demidter[424] = mid[8][74];
demidter[425] = mid[1][57];
demidter[426] = mid[2][41];
demidter[427] = mid[3][25];
demidter[428] = mid[4][9];
demidter[429] = mid[5][108];
demidter[430] = mid[6][92];
demidter[431] = mid[7][76];
demidter[432] = mid[8][60];
demidter[433] = mid[1][43];
demidter[434] = mid[2][27];
demidter[435] = mid[3][11];
demidter[436] = mid[4][109];
demidter[437] = mid[5][94];
demidter[438] = mid[6][78];
demidter[439] = mid[7][62];
demidter[440] = mid[8][46];
demidter[441] = mid[1][29];
demidter[442] = mid[2][13];
demidter[443] = mid[3][111];
demidter[444] = mid[4][95];
demidter[445] = mid[5][80];
demidter[446] = mid[6][64];
demidter[447] = mid[7][48];
demidter[448] = mid[8][32];
demidter[449] = mid[1][15];
demidter[450] = mid[2][113];
demidter[451] = mid[3][97];
demidter[452] = mid[4][81];
demidter[453] = mid[5][66];
demidter[454] = mid[6][50];
demidter[455] = mid[7][34];
demidter[456] = mid[8][18];

for（i = 1;i<457;i + + ）
{out[i-1] = demidter[i];}//同理将 1 × 457（第一个元素舍弃）的解交织输出放到应该输出的 1 × 456 中
return;
}
```

3. GSM 中(2,1,5)卷积码的 C 语言译码程序

```
#include"stdio.h"
#include"math.h"
int order（int a[],int n）//n 是数组长度
{int i,j,t,tmin,k = 0;
char b[n];
for（i = 0;i<n;i + + ）
{
   b[i] = a[i];
}
for （i = 0;i<n-1;i + + ）
{tmin = i;
for（j = i + 1;j<n;j + + ）
if （b[j]<b[tmin]） tmin = j;
if （tmin! = i）
{t = b[i];b[i] = b[tmin];b[tmin] = t;}
}
printf（"%c\n",b[0]）;
for（i = 0;i<n;i + + ）
{  if（a[i] = =b[0]）
      {k = i;break;}
}
return k;
}
int compare（long int a,long int b）//数值大小比较
{  int flag;
    if（a<b）     flag = 0;
      else      flag = 1;
      return   flag;
}
int weight（int x）//计算码重
{
   int countx = 0;
   while（x）
   {
```

```
                countx++;
                x = x&(x-1);
                    }

        return countx;
    }
    int catch2(int a[])//获取 2 比特数据
    {  int t;
     if(a[0]%2==0&&a[1]%2==0)  t=0x00;
       else if(a[0]%2==0&&a[1]%2==1)  t=0x01;
          else if(a[0]%2==1&&a[1]%2==0) t=0x02;
                else   t=0x03;
         return t;
        }

     vitrebi()
    { int i,j,k,m,n,s,z,t,b;
      int a;
      int Survivor_path[16][2],path_temp[16][189];
      long int Survivor_metrix[16][2];
      long int Survivor_temp1,Survivor_temp2,Survivor_
temp[16][2];
      int up[]={0x00,0x03,0x01,0x02,0x00,0x03,0x01,
0x02,0x03,0x00,0x02,0x01,0x03,0x00,0x02,0x01};
      int down[]={0x03,0x00,0x02,0x01,0x03,0x00,
0x02,0x01,0x00,0x03,0x01,0x02,0x00,0x03,0x01,0x02};
      int v[2],u[16],out[189];
      int   path1[189];
      int *p;
      int code[189];
      int *decode;
      int path[16][189];
    for(i=0;i<16;i++)
    {u[i]=0;}

     for(i=0;i<16;i++)//初始化
       for(j=0;j<2;j++)
        {Survivor_path[i][j]=0;
         Survivor_metrix[i][j]=0;
         Survivor_temp[i][j]=0;

        }
    for(i=0;i<16;i++)//初始化
        for(j=0;j<189;j++)
    path_temp[i][j]=0;

        for(b=1;b<=9;b++)
        {for(k=1;k<=21;k++)
        {
          v[0]=*in;in++;
          v[1]=*in;in++;//数据读入
          a=catch2(v);//获取所要比较的比特

          if(k<=4&&b==1)   //译码开始,前 4
个节点的累加汉明距离,译码值
           {
            {for(i=0;i<pow(2,k-1);i++)
              for(j=0;j<2;j++)
                { Survivor_metrix[2*i+j][0]=2*i+j;
                  Survivor_temp[2*i+j][0]=2*i+j;

Survivor_metrix[2*i+j][1]=Survivor_temp[i][1]+
weight(up[2*i+j]^a );//printf("\n%d",Survivor_
metrix[2*i+j][1]);//距离的累加
                  path[2*i+j][k]=j;//printf
("\n%d",path[2*i+j][k]);//译码值的更新
                  if(k>1){ for(z=1;z<=k-1;z++)
                    {path[2*i+j][z]=path_temp[i][z];}
                    }
                }
            }
           {for(i=0;i<pow(2,k-1);i++)
             for(m=0;m<2;m++)
              {Survivor_temp[2*i+m][1]=
Survivor_metrix[2*i+m][1];
               for(t=1;t<=k;t++)
               path_temp[2*i+m][t]=
path[2*i+m][t];//printf("\ntemp=%d",path_temp
[2*i+j][k]);
               }
            }
           }
           if(k>4||b>1)//从第 5 个节点开始汉明
距离累加以及比较,译码值
          {
           {for(i=0;i<8;i++)
             for(j=0;j<2;j++)
```

```
                {

//Survivor_metrix[2*i + j][0] = 2*i + j;
                    Survivor_temp1 =Survivor_temp[i][1]+
weight（up[2*i + j] ^ a ）;//距离的累加

Survivor_temp2 = Survivor_temp[i + 8] [1]+ weight
( down [2*i + j] ^ a ) ; //printf ( "%d\n",
Survivor_temp2）;
                //同一个节点延伸出的上下支路汉
明距离的累加
               n= compare（Survivor_temp1,Survivor_
temp2） ;
                //上下支路汉明距离的比较
                                if（n = =0）    {
                                Survivor_metrix
[2*i + j][1] = Survivor_temp1;

      //Survivor_path[2*i + j] [0] =  2*i + j;
                                    path[2*i + j][k] = j;
                        if（k>1）{for（z=1;z<=k-1;z++）
                                        {path[2*i + j]
[z] = path_temp[i][z];}
                                    }
                                }
                        else      {

Survivor_metrix[2*i + j] [1] = Survivor_temp2;//printf
( "\n%d",Survivor_metrix [2*i + j][1]）;
                                //
Survivor_path[2*i + j][0] = 2*i + j;
                                    path[2*i + j][k] = j;
                                if ( k>1 )      { for
( z = 1;z< = k-1;z + + )

{path[2*i + j][z] = path_temp[i + 8][z];}
                                    }
                                    }
                                //printf
( "\n%d",Survivor_metrix[2*i + j][1]）;
                                    // printf( "%d",path[2*i +
j][k]）;
    // 根据上下支路比较返回值,选取累加值小的
路径上累加汉明距离,译码值的更新
                }
            }
                {for（i = 0;i<8;i + + )
                    for（m = 0;m<2;m + + )

{Survivor_temp[2*i + m][1] = Survivor_metrix[2*i +
m][1];
                        for（t = 1;t< = k;t + + )

path_temp[2*i + m][t] = path[2*i + m][t];
                    }
                }
        }
    }
        for（i = 0;i<16;i + + )
          {u[i] = Survivor_metrix[i][1];printf（ "%d
",u[i]）;}
            s = order（u,16）;//比较输出 16 个路径
最优路径的序号
            printf（"s = %d\n",s）;
            for（k = 0;k<21;k + + )
            out[（b-1）*21 + k]= path[s][k + 1];
    }
      }printf（"\n"）;
     getchar（）;
        return
    }
```

习　　题

8.1　数字通信中的错误可以分为哪几类？各有什么特点？

8.2　简述最大后验概率准则和极大似然概率准则，并说明在什么条件下两者是等价的。

8.3　给定生成元 $g^{(1,1)}=(1011)$，$g^{(1,2)}=(1111)$，试画出该卷积码编码器的结构图。

8.4 画出下列卷积码的编码器结构图，写出约束长度。

（1）生成多项式为 $g^{(1,1)}=(23)_8$，$g^{(1,2)}=(35)_8$，码率为 $\frac{1}{2}$。

（2）生成多项式为 $g^{(1,1)}=(25)_8$，$g^{(1,2)}=(33)_8$，$g^{(1,2)}=(37)_8$，码率为 $\frac{1}{3}$。

8.5 设(3,1,2)卷积码的生成多项式为

$$g^{(1,1)}(x)=1+x+x^2,\ g^{(1,2)}(x)=1+x+x^2,\ g^{(1,3)}(x)=1+x^2$$

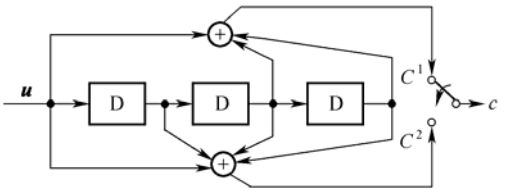
（1）试画出该编码的树图、格图和状态转移图。

（2）画出编码电路。

（3）如果输入序列为(1011)，求出编码后的序列。

（4）假设接收的序列为 110011001001，则试根据 Viterbi 算法进行译码（汉明距离作为量度）。

8.6 一个卷积码的码率为 $\frac{1}{2}$，结构图如下图所示。



（1）画出该卷积码的格图和状态转移图。

（2）求出该卷积码结构的传递函数，并写出它的自由距离。

8.7 已知(3,2,1)卷积码的 $G(D)=\begin{bmatrix}1+D & D & 1+D\\ D & 1 & 1\end{bmatrix}$。

（1）画出该卷积码的编码器结构。

（2）假设输入信息 $M(D)=[1+D+D^3 \quad 1+D+D^2+D^3]$，则求并行的码流 $1C^1(D)C^2(D)C^3(D)$。

8.8 在 GSM 移动通信系统中，信令信号的编码方案为：184 位信令比特用缩短的二进制循环码（FIRE 码）编码为 224 比特，然后再加上 4 比特的尾比特，共构成 228 比特，使用 $\frac{1}{2}$ 的卷积码产生 456 比特，卷积码的生成多项式矩阵为

$$G(D)=\begin{bmatrix}1+D^3+D^4 & 1+D+D^3+D^4\end{bmatrix}$$

（1）画出该卷积码的结构图。

（2）计算该卷积码的自由距离。

8.9 试说明软判决和硬判决的区别。

8.10 试说明交织编码的原理。

8.11 假设级联码的外码为(n,k)的 RS 码，内码为(N,K)的汉明码，则计算：级联码的码长、信息位长度以及码速率。

第 9 章 加密编码

内容简介

研究通信系统的目的就是要提高信息传输的有效性、可靠性、保密性和认证性。由前面几章讨论的结果可以看到，无论是离散信源还是连续波形信源、离散信道还是波形信道、单用户信道还是多用户信道，只要满足一定的条件，总可以通过信源编码和信道编码，使信息传输既有效又可靠。

随着人类进入信息时代，信息的传递、存储和交换日益频繁。现代化的通信网、计算机信息网，以及各种类型数据库、电子文电作业（MHS）系统和电子数据交换（EDI）系统等迅速发展，特别是因特网的迅速发展，使得信息的安全和保密问题与越来越多的人密切相关。个人、公司、集团、政府部门、军事部门等一些有价值的敏感信息在大规模分布式计算机网的环境下，一方面为合法用户提供了极大的方便，另一方面也为非法用户提供了更多介入机密信息的机会。这样密码学就揭开了神秘的面纱，它不仅与政府、军事机构，而且与广大的民间事业和人民大众密切相关，成为为更多人服务的学科。随着 1976 年公开密钥体制的出现和 1977 年美国公布数据加密标准（DES），正式宣告现代密码学的诞生，使这门古老的学科焕发了青春，并引起了更多学者和使用者的广泛注意。尤其当今，由于信息的安全和保密问题更为突出，因此人们不但对密码学在理论上进行了广泛研究，而且还出现了许多可靠实用的密码技术。当前，密码学已经活跃成为一个很重要的科技研究领域。

本章在介绍加密编码基本概念的基础上，简单介绍了几种古典密码，然后着重叙述了现代密码体制中的数据加密标准（DES）、国际数据加密算法（IDEA）以及公开密钥加密法，最后对模拟信号的加密作了简要介绍。

9.1 加密编码的基础知识

密码学有着悠久而神秘的历史，人们很难对密码学的起始时间给出准确的定义，一般认为人类对密码学的研究与应用可以追溯到古巴比伦时代。密码学最早应用在军事和外交领域，随着科技的发展而逐渐进入人们的生活中。本节介绍密码学的发展概况，并给出密码学的基本概念。

9.1.1 密码学的发展概况

研究信息的保密和复原保密信息以获取其真实内容的学科称为密码学，它包括密码编码

学和密码分析学。密码编码学是研究对信息进行编码，实现隐蔽信息的一门学科，而密码分析学则是研究复原保密信息或求解加密算法与密钥的学科。

密码技术的历史源远流长。很久以前，人们为了保存或传递经济、军事、外交上的重要信息，就已经开始使用密码或类似的保密技术，发明了多种多样的加密和解密方法，其手段由手工加密发展到机械加密，直到近代的电子加密。密码技术的发展历史上，记载着许多伟大的成就，书写了丰富的经验，密码技术的成果也曾在二战中发挥了巨大作用。然而直到 20 世纪 40 年代以前，密码技术却一直是纯经验性的学问。1949 年，香农发表了《保密系统的通信理论》，用信息论的观点对密源、密钥、密文等概念进行了数学描述，对保密系统进行了定量化的分析，才使密码学建立在科学的理论基础之上。

我们把 20 世纪 70 年代以前的密码研究与应用称为传统密码学。一般认为现代密码学诞生于 20 世纪 70 年代，标志性的大事有两件：一是 1977 年美国国家标准局正式公布实施了美国数据加密标准（DES），并批准用于非机要部门和商业用途，传统密码学的神秘面纱被揭开；二是 1976 年 11 月，美国斯坦福大学电气工程系研究生 W.Diffie 和副教授 Helman 在 IEEE 上发表了题为《密码学中的新方向》的学术论文，公钥密码研究的序幕就此拉开。从此，密码学的研究进入了一个崭新的时代。

9.1.2 密码学的基本概念

图 9-1 给出的是一个密码系统的模型。

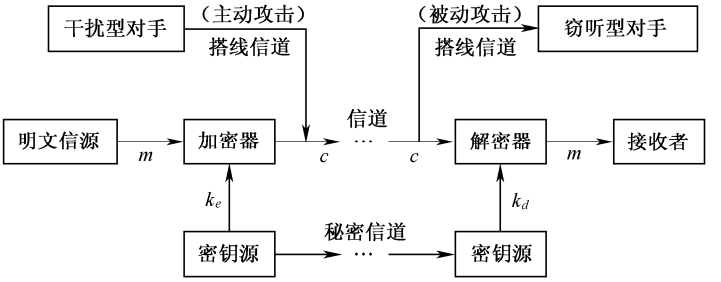


图 9-1 密码系统模型

密码系统的使用者通常称为用户。密码系统的破坏者有时称为对手，对手分为“窃听型”和“干扰型”两种。“窃听型”对手只是截取信道上传送的信息，而“干扰型”对手则会篡改信道上传送的信息。

下面介绍一些密码学的基本术语。

明文——发送方未经过加密处理的信息，其内容是容易理解的。明文常用 m 表示。

密文——发送方经过加密处理后的信息，文字被改变，其内容是难以理解的。密文常用 c 表示。

密钥——发送方加密处理时所用的秘密信息。在传统密码学中，解密也使用同样的密钥。密钥常用 k 表示。

加密——发送方把明文加工成密文的变换，即

$$c = E_k(m) \tag{9-1}$$

解密——接收方把密文解译成明文的变换，即

$$m = D_k(c) \tag{9-2}$$

通常一个密码体制可以有如下几个部分。

① 消息空间 M（又称明文空间）：所有可能明文 m 的集合。

② 密文空间 C：所有可能密文 c 的集合。

③ 密钥空间 K：所有可能密钥 k 的集合，其中每一密钥 k 由加密密钥 k_e 和解密密钥 k_d 组成，即 $k=(k_e,k_d)$。

④ 加密算法 E：一簇由加密密钥控制的、从 M 到 C 的加密变换。

⑤ 解密算法 D：一簇由解密密钥控制的、从 C 到 M 的解密变换。

五元组 $\{M, C, K, E, D,\}$ 就称为一个密码系统。在密码系统中，对于每一个确定的密钥 k，加密算法将确定一个具体的加密变换，解密算法将确定一个具体的解密变换，而且解密变换就是加密变换的逆变换。对于明文空间 M 中的每一个明文 m，加密算法 E 在加密密钥 k_e 的控制下将明文 m 加密成密文 c；而解密算法 D 则在密钥 k_d 的控制下将密文 c 解密成同一明文 m，即对 $\forall m \in M$，$(k_e,k_d) \in K$，有 $D_{k_d}(E_{k_e}(m)) = m$。

从数学的角度来讲，一个密码系统就是一组映射，它在密钥的控制下将明文空间中的每一个元素映射到密文空间上的某个元素。这组映射由密码方案确定，具体使用哪一个映射由密钥决定。

对密码体制的分类方法有多种，常用的分类方法有以下 3 种。

（1）根据密码算法所用的密钥数量。根据加密算法与解密算法所使用的密钥是否相同或是否很容易相互推导出，可以将密码体制分为对称（单密钥）密码体制和非对称（双密钥）密码体制。

如果一个提供保密服务的密码系统，它的加密密钥和解密密钥相同，或者虽然不相同，但由其中的任意一个可以很容易地导出另外一个，那么该系统所采用的就是对称密码体制。如 DES、AES、IDEA 等都是典型的对称密码体制。显然，使用对称密码体制时，如果某个实体拥有加密（或解密）能力，也就必然拥有解密（或加密）能力。

如果一个提供保密服务的密码系统，其加密算法和解密算法分别用两个不同的密钥实现，并且由加密密钥不能推导出解密密钥，则该系统所采用的就是非对称密码体制。采用非对称密钥密码体制的每个用户都有一对选定的密钥。其中一个是可以公开的，称为公开密钥，简称公钥；另一个由用户自己秘密保存，称为私有密钥，简称私钥。

在安全性方面，对称密码体制是基于复杂的非线性变换与迭代运算实现算法安全性的，而非对称密码体制则一般是基于某个公认的数学难题而实现安全性的。由于后者的安全程度与现实的计算能力具有密切的关系，因此，通常认为非对称密码体制的安全强度似乎没有对称密码体制高，但也具有对称密码体制所不具备的一些特性，如它适用于开放性的使用环境，密钥管理问题相对简单，可以方便、安全地实现数字签名和验证等。

由于非对称密码体制的安全性一般是依赖于某个公认的数学难题，普遍认为其安全性没有对称密码体制高。

（2）根据对明文信息的处理方式。根据密码算法对明文信息的处理方式，又可将对称密码体制再分为分组密码和序列密码（也称为流密码）。

分组密码是将消息进行分组，一次处理一个数据块（分组）元素的输入，对每个输入块

产生一个输出块。在用分组密码加密时，一个明文分组被当做一个整体来产生一个等长的密文分组输出。分组密码通常使用的分组大小是 64 比特或 128 比特。如 DES、AES、IDEA 等即为分组密码算法。

序列密码则是连续地处理输入元素，并随着处理过程的进行，一次产生一个元素的输出，在用序列密码加密时，一次加密一个比特或一个字节。典型的序列密码有 RC4、A5、SEAL 等。

（3）根据是否能进行可逆的加密变换。根据密码算法是否能进行可逆的加密变换，又可分为单向函数密码体制和双向变换密码体制。

单向函数密码体制是一类特殊的密码体制，其性质是可以很容易地把明文转换成密文，但再把密文转换成正确的明文却是不可行的，有时甚至是不可能的。单向函数只适用于某种特殊的、不需要解密的应用场合，如用户口令的存储和信息的完整性保护与鉴别等。

双向变换密码体制是指能够进行可逆的加密、解密变换，绝大多数加密算法都属于这一类，它要求所使用的密码算法能够进行可逆的双向加解密变换，否则接收者就无法把密文还原成明文。典型的单向函数包括 MD4、MD5、SHA-1 等。

9.2 几种古典密码

在密码编码体制中，有两种最基本也是最古老的编码体制一直沿用至今，它们是替代密码和换位密码，由于其历史悠久并且是现代密码体制的基本组成部分，因而在密码学的研究与应用中占有重要地位。古典密码和近代密码是密码学发展的重要阶段，也是现代密码学产生和发展的渊源。尽管古典密码体制大多比较简单，一般可用手工或机电方式实现加密和解密过程，破译也较为容易，目前已很少采用，但了解它们的设计原理对于进一步理解、设计和分析现代密码体制都是十分重要的。

替代是古典密码中用到的最基本的处理技巧之一，它在现代密码学中也得到广泛使用。所谓替代就是将明文中的一个字母由其他字母、数字或符号替换的一种方法。替代密码是指先建立一个替换表，加密时将需要加密的明文依次通过查表，替换为相应的字符，明文字符被逐个替换后，生成无任何意义的字符串，即密文；解密时则利用对应的逆替换表，将需要解密的密文依次通过查表，替换为相应的字符即可恢复出明文。替代密码的密钥就是其替换表。

根据密码算法加解密时使用替换表多少的不同，替代密码又可分为单表替代密码和多表替代密码。

① 单表替代密码。密码算法加解密时使用一个固定的替换表。这时对明文消息中出现的同一个字母，在加密时都用同一个固定的字母来替代，而不管它出现在什么地方。

② 多表替代密码。密码算法加解密时使用多个替换表。这样，明文消息中出现的同一个字母，在加密时不是完全被同一个固定的字母替代，而是根据其出现的位置次序，用不同的字母替代。

置换密码又称为换位密码，这种密码通过改变明文消息各元素的相对位置，但明文消息元素本身的取值或内容形式不变；而在替代密码中，则可以认为是保持明文的符号顺序，但

是将它们用其他符号来替代。

这种密码是把明文中各字符的位置次序重新排列来得到密文的一种密码体制。实现的方法多种多样。直接把明文顺序倒过来，然后排成固定长度的字母组作为密文就是一种最简单的置换密码。例如，明文为 this cryptosystem is not secure，密文则为 eruc esto nsim etsy sotp yrcs iht。又如，设明文为 m =Thousands of years ago, cryptography had been used to keep secrets of military operations or treasure，加密算法为将原文忽略空格并不计字母大小写，每 5 字符一组，各组取逆序，连接后得到密文

c =suohtosdnaraeyfcogasotpyrhpargbdahysuneekotdeespeesterclimfoyratiareposnoitertrtroerusa

这里，k =5 是密钥。

下面列举一些古典密码的例子。

9.2.1 凯撒密码

将 26 个英文字母按字母表序号 0～25 编号，如表 9-1 所示。加密编码的方法是把明文的每个字母用向后循环移动 k 位后的字符代替，即

$$c = E_k(m) = m + k \bmod 26 \tag{9-3}$$

移位距离 k 是密钥。

表 9-1　　字母数字映射表

字母	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
数字	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

例如明文是 m =china，密钥 k =3，明文消息对应的数字依次为

2　7　8　13　0

用凯撒（Caesar）密码对其依次进行加密如下：

$E_3(c) = E_3(2) = 2 + 3(\bmod 26) = 5$，对应字母为 f;

$E_3(h) = E_3(7) = 7 + 3(\bmod 26) = 10$，对应字母为 k;

$E_3(i) = E_3(8) = 8 + 3(\bmod 26) = 11$，对应字母为 l;

$E_3(n) = E_3(13) = 13 + 3(\bmod 26) = 16$，对应字母为 q;

$E_3(a) = E_3(0) = 0 + 3(\bmod 26) = 3$，对应字母为 d。

则密文消息为：c =fklqd。

更一般的方法是不仅平移而且作“仿射变换”，即

$$c = k_1 m + k_2 \bmod 26 \tag{9-4}$$

式中，k_1 和 k_2 是两个密钥，k_1 要求与 26 互素。

例如，m =information 对应的字母序号是

8　13　5　14　17　12　0　19　8　14　13

若选 $k_1 = 7,\ k_2 = 10$，则变换后的数字是

14　23　19　4　25　16　10　13　14　4　23

对应的密文为 c =oxtezqknoex。

9.2.2 密钥短语密码

密钥短语密码选用一个英文短语或单词串作为密钥，去掉其中重复的字母得到一个无重复字母的字符串，然后再将字母表中的其他字母依次写于此字母串后，就可构造出一个字母替代表。如密钥短语为 happy new year，按顺序去掉重复字母及空格得到 hapynewr，替代表如表 9-2 所示。

表 9-2　　字母替代表

明文字母	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
密文字母	h	a	p	y	n	e	w	r	b	c	d	f	g	i	j	k	l	m	o	p	q	s	t	u	x	x

当选择上面的密钥进行加密时，若明文为“china”，则密文为“prbih”。

显然，不同的密钥可以得到不同的替换表，对于明文为英文单词或短语的情况时，密钥短语密码最多可能有 $26!=4\times10^{26}$ 个不同的替换表。

9.2.3 维吉尼亚密码

单表替代虽然改变了明文的模样，但密文中同一个字符总来自明文的同一字母，这样就很容易从概率分析上破译（如找到概率最大的字母，就可能是 e 变来的）。为此引入多表替代，原文中的同一个字母出现在不同位置时，会变换成密文的不同字符，改变了单表替代密码中密文的唯一性，使密码分析更加困难。

多表替代密码由莱昂•巴蒂斯塔于 1568 年发明，著名的维吉尼亚（Vigenere）密码和希尔（Hill）密码等均是多表替代密码。

维吉尼亚密码是最古老而且最著名的多表替代密码体制之一，它以法国密码学家 Blaise de Vigenere（1523—1596）的名字命名。与位移密码体制相似，但维吉尼亚密码的密钥是动态周期变化的。

设明文 $m=m_1m_2\cdots m_n$，密钥 $k=k_1k_2\cdots k_n$，则密文

$$c=E_k(m)=c_1c_2\cdots c_n$$

其中，$c_i=m_i+k_i \pmod{26},\ i=1,2,\cdots,n$。

当密钥的长度比明文短时，密钥可以周期性地重复使用，直至完成明文中每个字母的加密。

例如，密钥 k=shift，其字母序号为 18,7,8,5,19，密钥长度 n=5，明文消息为 m=encode algorithm，忽略空格，其字母序号是 4,13,2,14,3,4,0,11,6,14,17,8,19,7,12，用维吉尼亚密码对其进行加密过程如表 9-3 所示。

表 9-3　　维吉尼亚密码加密过程

明文	e	n	c	o	d	e	a	l	g	o	r	i	t	h	m
	4	13	2	14	3	4	0	11	6	14	17	8	19	7	12
密钥	s	h	i	f	t	s	h	i	f	t	s	h	i	f	t
	18	7	8	5	19	18	7	8	5	19	18	7	8	5	19
密文	22	20	10	19	22	22	7	19	11	7	9	15	1	12	5
	w	u	k	t	w	w	h	t	l	h	j	p	b	m	f

最终密文是：c=wuktwwhtlhjpbmf。

9.3　数据加密标准

美国国家标准局 1977 年公布了由 IBM 公司研制的 Data Encryption Standard（DES）作为非机要部门的数据加密标准。它是迄今为止流行最广、时间最长的加密算法，也是现代分组加密技术的典型。尽管 DES 目前已被高级加密标准（AES）所取代，但其设计思想仍然值得借鉴。

9.3.1　DES 加密算法

DES 的加密过程如图 9-2 所示，明文分组长 64 位，明文可表示成：$m = m_1 m_2 \cdots m_{64}$，其中 $m_i \in \{0,1\}$，$1 \leqslant i \leqslant 64$。密钥长 56 位，加上每 7 位一个奇偶校验位，共 64 位。DES 首先利用初始置换 IP 对 m 进行换位处理，然后对如图 9-3 所示的与密钥有关的圈变换进行 16 次迭代加密运算，最后，经过逆初始置换 IP^{-1} 的处理得到密文：$c = c_1 c_2 \cdots c_{64}$，其中，$c_i \in \{0,1\}$，$1 \leqslant i \leqslant 64$。

加密过程可表示为

$$\mathrm{DES}(m) = \mathrm{IP}^{-1} \circ T_{16} \circ T_{15} \cdots T_2 \circ T_1 \circ \mathrm{IP}(m) \tag{9-5}$$

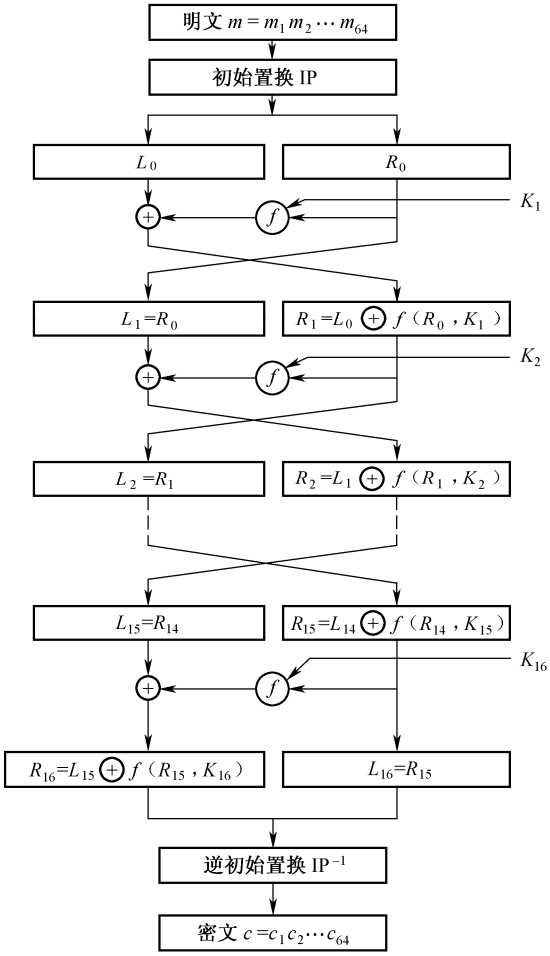


图 9-2　DES 加密算法

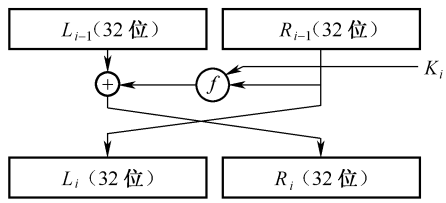


图 9-3　DES 的圈变换

1．置换与逆置换

IP 是初始置换， IP^{-1} 是逆置换，初始置换 IP 如表 9-4 所示，逆初始置换 IP^{-1} 如表 9-5 所示。不难看出 IP^{-1} 是 IP 的逆。初始置换 IP 用于对明文 m 中的各位进行换位，目的在于打乱明文 m 中各位的次序。经过初始置换后， m 变为

$$\begin{aligned} m' &= m_1' m_2' \cdots m_{64}' \\ &= m_{58} m_{50} \cdots m_7 \end{aligned}$$

即 m 中的第 58 位变为 m' 中的第 1 位， m 中的第 50 位变为 m' 中的第 2 位，依此类推，最后将 m 中的第 7 位变为 m' 中的第 64 位。

表 9-4　　初始置换 IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

表 9-5　　逆初始置换 IP^{-1}

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

2．迭代加密运算

将 IP 置换后的 64 位明文分成两半，各 32 位，即 L_0 和 R_0，分别进入加密器的左、右两个入口，先后经 T_1，T_2…，T_{16} 进行 16 轮迭代加密运算。

每轮加密 T_i 流程如图 9-3 所示，其中，$\oplus$ 表示按位模 2 加；f 表示扩展与收缩函数 $f(R_{i-1}, K_i)$。这样，16 次迭代可以形式化地表示为

$$\begin{cases} L_i = R_{i-1} \\ R_i = L_{i-1} \oplus f(R_{i-1}, K_i) \\ i = 1, 2, \cdots, 15, 16 \end{cases} \tag{9-6}$$

其中，L_i 和 R_i 的长度都是 32 位，$L_0 = m_1' m_2' \cdots m_{32}'$，$R_0 = m_{33}' m_{34}' \cdots m_{64}'$，$K_i$ 是由密钥 k 产生的一个 48 位的子密钥。

将 $R_{16}L_{16}$ 进行逆初始置换 IP^{-1} 的处理后就得到密文。

$$c = c_1c_2 \cdots c_{64}$$

不将 R_{16} 与 L_{16} 左右交换而直接对 $R_{16}L_{16}$ 进行逆初始置换处理的目的是为了使加密和解密可以使用同一个算法。这里，$R_{16}L_{16}$ 表示将 L_{16} 排在 R_{16} 的右面。

3．扩展与收缩函数 $f(R_{i-1}, K_i)$

扩展与收缩函数即加密函数 f 是 DES 的核心，如图 9-4 所示。E 为扩展变换，用于将 32 位的输入扩展为 48 位，其方法是将信息的某些比特位重复，即

32 1 2 3 4 5 4 5 6 7 8 9 8 9 10 11 12 13 12 13 14 15 16 17 16 17 18 19 20 21 20 21 22 23 24 25 24 25 26 27 28 29 28 29 30 31 32 1

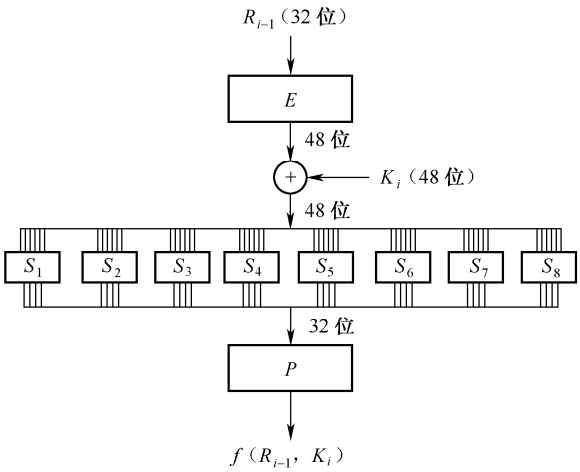


图 9-4　加密函数 $f(R_{i-1}, K_i)$ 的计算

设

$$R_{i-1} = r_1r_2 \cdots r_{32}$$

则

$$E(R_{i-1}) = r_{32}r_1r_2 \cdots r_{32}r_1$$

将 $E(R_{i-1})$ 与子密钥 K_i 进行按位模 2 加运算，对运算结果从左到右分为 8 组，每组 6 位，设

$$E(R_{i-1}) \oplus K_i = B_1B_2B_3B_4B_5B_6B_7B_8$$

其中，B_j 的长度为 6 位，$1 \leqslant j \leqslant 8$。$B_j$ 经过 S_j 后缩减为 4 位，通常称 $S_1, S_2, \cdots, S_8$ 为 S 盒。8 个 S 盒的输出经置换函数 P 进行换位处理后就得到 $f(R_{i-1}, K_i)$。置换函数 P 如表 9-6 所示，S 盒如表 9-7 所示。每个 S 盒有 4 行 16 列。设

$$B_j = b_1b_2b_3b_4b_5b_6$$

将 b_1b_6 和 $b_2b_3b_4b_5$ 作为二进制数，设 b_1b_6 和 $b_2b_3b_4b_5$ 对应的十进制整数分别为 r 和 c，则 S_j 中的第 r 行第 c 列的整数的二进制表示就是 S_j 的输出。例如，$B_1 = 100101$，从表 9-7 可知 S_1 的第 3 行第 2 列的整数为 8，S_1 的输出为 1 000。

表 9-6　　置换函数 P

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

表 9-7　　S 盒数据对照表

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7	
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8	S_1
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0	
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13	
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10	
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5	S_2
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15	
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9	
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8	
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1	S_3
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7	
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12	
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15	
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9	S_4
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4	
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14	
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9	
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6	S_5
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14	
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3	
0	12	1	10	16	9	2	6	8	0	13	3	4	14	7	5	11	
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8	S_6
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6	
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13	
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1	
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6	S_7
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2	
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12	
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7	
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2	S_8
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8	
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11	

4．子密钥的产生

从密钥 k 生成子密钥 K_i 的算法如图 9-5 所示。密钥 k 中有 8 位是奇偶校验位，分别位于第 8,16,24,32,40,48,56,64 位。奇偶校验位用于检查密钥 k 在产生和分配以及存储过程中可能发生的错误。选择置换 PC-1 用于去掉密钥 k 中的 8 个奇偶校验位，并对其余的 56 位打乱重新排列。选择置换 PC-1 如表 9-8 所示。将 PC-1 输出中的前 28 位作为 C_0，后 28 位作为 D_0。C_0 中的各位从左到右依次为密钥 k 中的第 57,49，…，44,36 位，D_0 中的各位从左到右依次为密钥 k 中的第 63,55,…，12,4 位。对于 $1 \leqslant i \leqslant 16$

$$C_i = LS_i(C_{i-1})$$

$$D_i = LS_i(D_{i-1})$$

其中 LS_i 表示对 C_{i-1} 和 D_{i-1} 进行循环左移变换，$LS_1, LS_2, LS_9, LS_{16}$ 是循环左移 1 位变换，其余的 LS_i 是循环左移 2 位变换。选择置换 PC-2 用于从 C_iD_i 中选取 48 位作为子密钥 K_i。选择置换 PC-2 如表 9-9 所示。C_iD_i 表示从左到右将 D_i 排在 C_i 的后面，C_iD_i 的长度为 56 位。子密钥 K_i 中的各位从左到右依次为 C_iD_i 中的第 14,17,…,29,32 位。

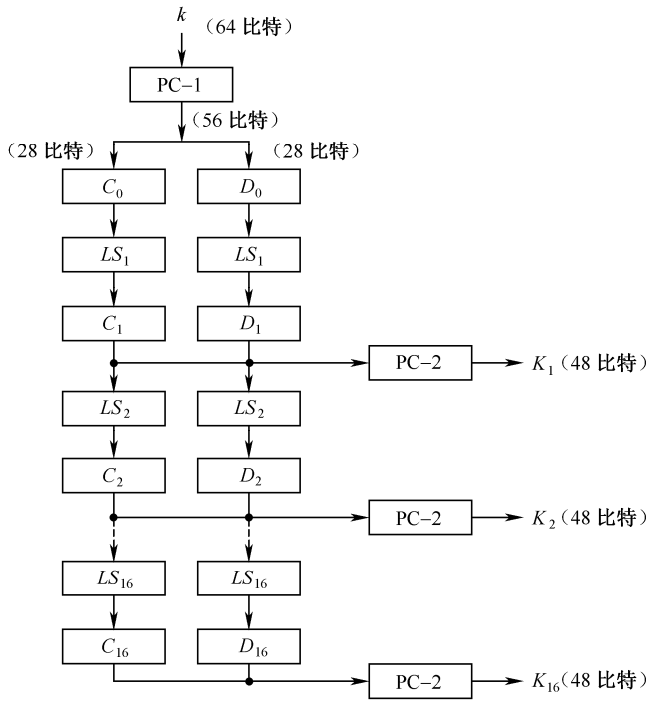


图 9–5　产生子密钥的流程图

表 9-8　　选择置换 PC-1

57	49	41	33	25	17	9	1	58	50	42	34	26	18
10	2	59	51	43	35	27	19	11	3	60	52	44	36
63	55	47	39	31	23	15	7	62	54	46	38	30	22
14	6	61	53	45	37	29	21	13	5	28	20	12	4

表 9-9　　选择置换 PC-2

14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32

9.3.2　DES 的解密过程

DES 的解密过程和加密过程使用同一算法，只不过在 16 次迭代中使用子密钥的次序正好相反。解密时，第 1 次迭代使用子密钥 K_{16}，第 2 次迭代使用子密钥 K_{15}，依此类推，第 16 次迭代使用子密钥 K_1。解密时的 16 次迭代可以形式化地表示为

$$\begin{cases} R_{i-1} = L_i \\ L_{i-1} = R_i \oplus f(L_i, K_i) \\ i = 16,15,\cdots,2,1 \end{cases} \tag{9-7}$$

我们用 $\mathrm{DES}_k(m)$ 表示当密钥为 k 时利用 DES 对明文 m 进行加密得到的密文，用 $\mathrm{DES}_k^{-1}(c)$ 表示当密钥为 k 时利用 DES 对密文 c 进行解密得到的明文，不难验证，对任意明文 m，有

$$\mathrm{DES}_k^{-1}(\mathrm{DES}_k(m)) = m$$

$$\mathrm{DES}_k\ (\mathrm{DES}_k^{-1}(m)) = m$$

9.3.3　DES 的安全性

DES 的出现在密码学史上是一个创举。以前的任何设计者对于密码体制及其设计细节都是严加保密的。而 DES 算法则公开发表，任人测试、研究和分析，无须通过许可就可制作 DES 的芯片和以 DES 为基础的保密设备。DES 的安全性完全依赖于所用的密钥。

在 DES 中，除了 S 盒是非线性变换外，其余变换均为线性变换。因此，S 盒是 DES 的关键。可以看出，任意改变 S 盒输入中的几位，其输出至少有两位发生变化。由于在 DES 中使用了 16 次迭代，所以即使改变明文或密钥中的 1 位，密文中都会有大约 32 位发生变化。S 盒的设计原则一直没有完全公开。人们怀疑 S 盒的设计中可能隐藏着某种陷门，它可以使得了解陷门的人能够成功地进行密码分析。经过多年来的研究，人们的确发现了 S 盒的许多规律，但至今还没有发现 S 盒的致命缺陷。

在 DES 中，子密钥的产生也很有特色，它确保密钥中各位的使用次数基本相等。实验表明，56 位密钥中的每位的使用次数为 12～15 次。

随着密码分析技术和计算能力的提高，DES 的安全性受到质疑和威胁，密钥较短是 DES 的一个主要缺陷。实际上，在 IBM 最初向 NSA 提交的方案当中，密钥的长度为 112 位，但在 DES 成为一个标准时，被削减至 56 位密钥。56 位的短密钥确实太短，下面两个例子很有说服力。

分布式穷举攻击：1997 年 1 月 28 日，美国 RSA 数据安全公司在 Internet 上开展了一项“秘密密钥挑战”的竞赛，悬赏 1 万美元来破解一段 DES 密文。计划公布后，得到了许多网络用户的积极响应。科罗拉多州的程序员 R.Verser 设计了一个可以通过互联网分段运行的密钥搜索程序，组织了一个称为 DESCHALL 的搜索行动，成千上万的志愿者加入到计划中。

第 96 天，即竞赛公布后的第 140 天，1997 年 6 月 17 日晚上，美国盐湖城的 M.Sanders 成功地找到了密钥并解密出明文。

专用搜索机：1998 年 5 月，美国电子边境基金会宣布，他们用一台价值 20 万美元的计算机改装成专用设备，56 小时就破译了 DES。这样，更加直接地说明了 56 位密钥太短了，彻底宣布了 DES 的终结。

9.4　国际数据加密算法

国际数据加密算法（International Data Encryption Algorithm，IDEA）是 1990 年由瑞士联邦技术学院来学嘉（X.J.Lai）和 J.L.Massey 合作提出的建议，标准算法称作 PES（Proposed Encryption Standard）。Lai 和 Massey 在 1991 年进行了改进，1992 年基本定型，并改称为 IDEA。它也是对 64 比特大小的数据块加密的分组加密算法，密钥长度为 128 位，用硬件和软件实现都很容易，且比 DES 在实现上快得多。IDEA 自问世以来，已经经历了大量的详细审查，对密码分析具有很强的抵抗能力，在多种商业产品中被使用。

IDEA 算法将明文中每 64 比特数据块分成 X_1、X_2、X_3 和 X_4 4 个子块，每一子块 16 比特，令这 4 个子块作为第 1 轮迭代的输入，全部共 8 轮迭代。每轮迭代都是 4 个子块彼此间及与 16 比特的子密钥间进行异或运算、模 2^{16}（65 536）加法运算或模 $(2^{16}+1)$ 乘法运算，任何一轮迭代，第 3 和第 4 个子块互换。最后与 4 个 16 比特的子密钥进行输出变换，输出 4 个 16 比特密文数据。每次加密的过程如图 9-6 所示。

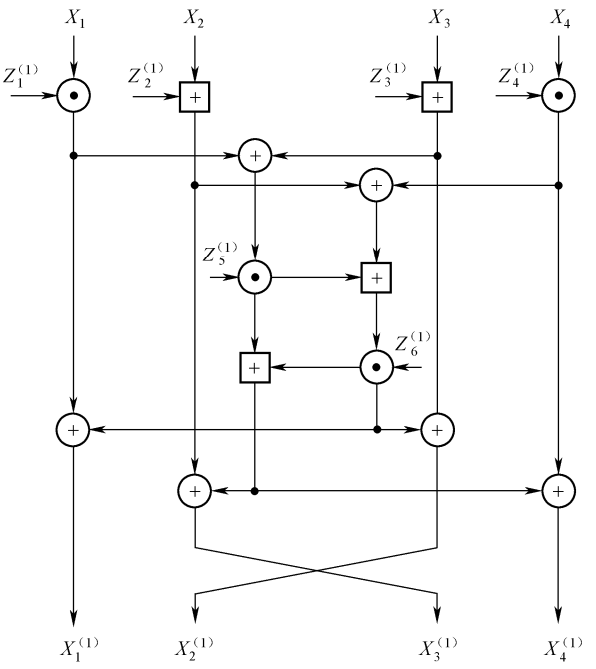


图 9-6　IDEA 迭代加密运算原理

在图 9-6 中，⊕ 表示模 2 加（异或）；⊙ 表示模 $2^{16}+1$（65 537）的乘法；⊞ 表示模 2^{16}（65 536）

的加法。$Z_1^{(1)} \sim Z_6^{(1)}$ 是第一轮加密使用的 6 个子密钥，它们来自 128 比特密钥的顺序分组（每组 16 比特，共 8 组）的前 6 组。第二轮处理的算法相同，只是所用的子密钥不同，$Z_1^{(2)}$ 和 $Z_2^{(2)}$ 是第一轮子密钥取剩下的两个分组，$Z_3^{(2)} \sim Z_6^{(2)}$ 则来自 128 比特密钥循环左移 25 位后再分成 8 组的前 4 个分组，而后 4 个分组留给第三轮子密钥的 $Z_1^{(3)} \sim Z_4^{(3)}$，$Z_5^{(3)}$ 和 $Z_6^{(3)}$ 则来自 128 比特密钥再次循环左移 25 位之后的 8 个分组。如此下去，直到第八轮迭代。第九轮不同于前八轮，不再需要 $Z_5^{(9)}$ 和 $Z_6^{(9)}$ 有关的迭代加密过程，实际上只有如图 9-7 所示的一步。最后将 $X_1^{(9)} \sim X_4^{(9)}$ 连接起来即是密文。

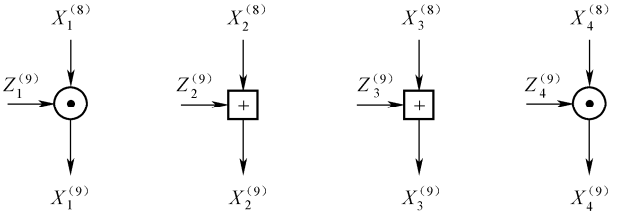


图 9-7　IDEA 迭代加密的第九轮运算

IDEA 解密和加密算法相同，只是子密钥不同，加、解密的密钥如下。

轮次	加密子密钥	解密子密钥
1	$Z_1^{(1)}, Z_2^{(1)}, Z_3^{(1)}, Z_4^{(1)}, Z_5^{(1)}, Z_6^{(1)}$	$(Z_1^{(9)})^{-1}, -Z_2^{(9)}, -Z_3^{(9)}, (Z_4^{(9)})^{-1}, Z_5^{(8)}, Z_6^{(8)}$
2	$Z_1^{(2)}, Z_2^{(2)}, Z_3^{(2)}, Z_4^{(2)}, Z_5^{(2)}, Z_6^{(2)}$	$(Z_1^{(8)})^{-1}, -Z_2^{(8)}, -Z_3^{(8)}, (Z_4^{(8)})^{-1}, Z_5^{(7)}, Z_6^{(7)}$
3	$Z_1^{(3)}, Z_2^{(3)}, Z_3^{(3)}, Z_4^{(3)}, Z_5^{(3)}, Z_6^{(3)}$	$(Z_1^{(7)})^{-1}, -Z_2^{(7)}, -Z_3^{(7)}, (Z_4^{(7)})^{-1}, Z_5^{(6)}, Z_6^{(6)}$
4	$Z_1^{(4)}, Z_2^{(4)}, Z_3^{(4)}, Z_4^{(4)}, Z_5^{(4)}, Z_6^{(4)}$	$(Z_1^{(6)})^{-1}, -Z_2^{(6)}, -Z_3^{(6)}, (Z_4^{(6)})^{-1}, Z_5^{(5)}, Z_6^{(5)}$
5	$Z_1^{(5)}, Z_2^{(5)}, Z_3^{(5)}, Z_4^{(5)}, Z_5^{(5)}, Z_6^{(5)}$	$(Z_1^{(5)})^{-1}, -Z_2^{(5)}, -Z_3^{(5)}, (Z_4^{(5)})^{-1}, Z_5^{(4)}, Z_6^{(4)}$
6	$Z_1^{(6)}, Z_2^{(6)}, Z_3^{(6)}, Z_4^{(6)}, Z_5^{(6)}, Z_6^{(6)}$	$(Z_1^{(4)})^{-1}, -Z_2^{(4)}, -Z_3^{(4)}, (Z_4^{(4)})^{-1}, Z_5^{(3)}, Z_6^{(3)}$
7	$Z_1^{(7)}, Z_2^{(7)}, Z_3^{(7)}, Z_4^{(7)}, Z_5^{(7)}, Z_6^{(7)}$	$(Z_1^{(3)})^{-1}, -Z_2^{(3)}, -Z_3^{(3)}, (Z_4^{(3)})^{-1}, Z_5^{(2)}, Z_6^{(2)}$
8	$Z_1^{(8)}, Z_2^{(8)}, Z_3^{(8)}, Z_4^{(8)}, Z_5^{(8)}, Z_6^{(8)}$	$(Z_1^{(2)})^{-1}, -Z_2^{(2)}, -Z_3^{(2)}, (Z_4^{(2)})^{-1}, Z_5^{(1)}, Z_6^{(1)}$
9	$Z_1^{(9)}, Z_2^{(9)}, Z_3^{(9)}, Z_4^{(9)}$	$(Z_1^{(1)})^{-1}, -Z_2^{(1)}, -Z_3^{(1)}, (Z_4^{(1)})^{-1}$

其中，Z^{-1} 是 Z 的模$(2^{16}+1)$的乘法逆元，$-Z$ 是 Z 的模 2^{16} 的加法逆元。

由于 IDEA 的密钥长度是 DES 的 2 倍，所以用同样的方式攻击 IDEA，所需工作量是 DES 的 $2^{72} = 4.7 \times 10^{21}$ 倍。许多科研部门和军事部门对 IDEA 进行攻击测试，未见成功报道。

9.5　RSA 公钥密码

在传统的密码体制中，由于加密密钥和解密密钥相同或很容易相互推导，因此密钥必须首先经由安全通道分发给通信双方，随后才能利用公开通道建立起安全通信，因而密钥分配和管理十分复杂，特别是在一个有很多用户的通信网中尤为复杂。如果通信网中有 N 个用户，为了保证网内任意两用户间的通信，若采用传统密码体制，需要 $N(N-1)/2$ 个密钥。比如在一个机密通信网中有 1 万个保密用户，这时 $N(N-1)/2 \approx 5 \times 10^7$，即网中需要分配和管理约 5000 万个密钥，这就过于复杂而且危险。此外，随着现代计算机网和现代电子服务系统，特

别是 Internet 的迅速发展，信息系统除了防止接收端非法用户窃听外，还需要在信息发送端防止非法用户主动攻击。消息认证、身份认证、数字签名就属于这类问题。若采用双钥的公开密钥体制，将会对这类信息认证问题的发展起到更大的促进。

1976 年 11 月，美国斯坦福大学电气工程系研究生 W.Diffie 和副教授 M.E.Hellman 在 IEEE 上发表了题为“密码学的新方向”的学术论文，首次提出双密钥思想，用公开的算法解决了单密钥的密钥交换问题。

9.5.1 公钥密码的基本概念

在公钥密码中，加密密钥和解密密钥是不一样的。加密密钥简称公钥（Public Key），解密密钥简称私钥（Private Key）。加密密钥可以公开，解密密钥当然必须保密。

根据加密密钥计算解密密钥的问题是难解的。所谓一个问题是难解的，直观上讲，就是不存在一个计算该问题的有效算法。换句话说，按照我们目前的计算能力，计算一个难解的问题所需要的时间是非常长的，如 100 年甚至更长。一般而言，计算一个难解的问题所需要的时间通常是输入数据长度的一个指数函数。因此，对于一个难解的问题，随着输入数据长度的增加，进行计算所需要的时间将会急剧地增加。所谓有效算法是指可以很快地求得问题的解的算法。对于一个问题，如果存在一个求其解的有效算法，则我们就称这个问题是容易求解的，否则就称其是难解的。到目前为止，还没有能够严格地证明哪一个问题是难解的。但对于某些问题，经过多年来的努力，仍然没有找到进行计算的快速有效的算法的事实，使我们认为这些问题是难解的。对于一个安全的密码体制，进行破译应该是一个难解的问题。

公钥密码的理论基础是单向陷门函数 $y=F(x)$。它满足这些特性：①对自变量 x 的任意给定值，容易计算 $y=F(x)$ 的值；②对于值域中的任意 y 值，即使已知 F，若不知道 F 的某种特殊性质，则求解其对应的 x 值仍然是计算上不可能的；若知道这种特殊性质，就容易计算出 x 值。因此这种特殊性质是很重要的，称为 F 的“陷门”（trapdoor）。在密码体制中，使用者构造出有关单向函数 F 和它的逆函数 F^{-1}。单向函数就是实际上的加密密钥，可公开。它的逆函数就是解密密钥，不公开。只知道公开的加密函数的人要破译密码体制求解逆函数 F^{-1}，这在计算上不可能的。而预定的接收者则可以用陷门信息（解密密钥 K_d）简单地求解 $x=F_{K_d}^{-1}(y)$。

在公开密钥密码体制中，用户 A 要公布他的加密密钥（e 和 n 两个数），图 9-8 所示的即为这种加密体制。B 要将一报文传给 A，首先用用户 A 的公开加密密钥对明文 M 加密，将密文 C 传给用户 A。用户 A 用自己的秘密解密密钥对密文 C 解密即可得到明文 M。其他人由于不知道用户 A 的解密密钥，即使得到密文也无法解读。图中 e 和 n 为加密密钥，d 和 n 为解密密钥，均为正整数。

公开密钥密码体制的另一个用途是在电子邮政和电子资金传送领域内的报文签名。这种签名能使发送者确认接收者的合法性。如图 9-9 所示，用户 B 用自己的秘密解密密钥将明文 M 进行加密得到密文 S，这代表发送者 B 的签名，因为别人是无法制造出这样的密文 S 的。B 再用接收者 A 的公开加密密钥进行加密，得到双重加密的密文 C，发送给用户 A。A 收到双重密文 C 后，先用自己的秘密解密密钥进行解密得到一次密文 S，再用用户 B 的公开加密密钥解出原始明文 M。若不是 B 签发的密文，用用户 B 的公开加密密钥是解不开密文 S 的。

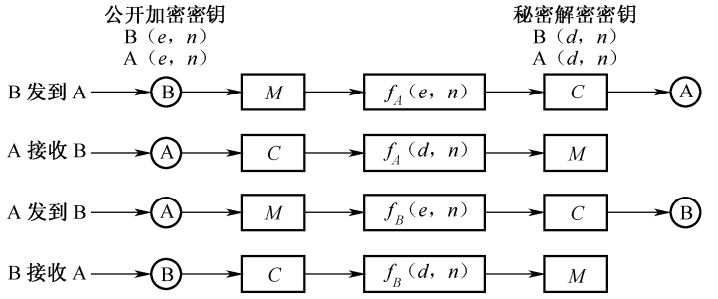


图 9-8　公开密钥密码体制

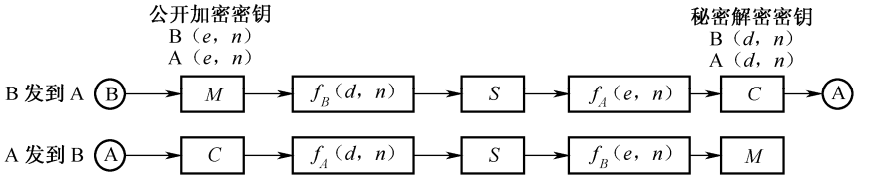


图 9-9　公开密钥密码体制

9.5.2　RSA 公钥密码体制

Ron Rivest 和 Adi Shamir 以及 Leonard Adleman 于 1978 年提出的 RSA 公钥密码体制至今仍被公认为是一个安全性能良好的密码体制。该体制的理论基础是数论中的下述论断：要求得两个大素数（如大到 100 位）的乘积在计算机上很容易实现，但要分解两个大素数的乘积（即从乘积求它的两个素因子）在计算上几乎不可能实现，即为单向函数。

RSA 公钥密码体制描述如下。

① 选取两个很大的素数 p 和 q（保密），计算 $n=p\times q$（公开）。

② 求 n 的欧拉函数 $\Phi(n)=(p-1)\times(q-1)$（保密），随机选取正整数 e，使得 $1<e<\Phi(n)$，且 e 与 $\Phi(n)$ 互素，e 是公开的加密密钥。

③ 解同余方程 $(e\times d)\bmod\Phi(n)=1$，求得保密的解密密钥 d。

④ (e,n) 即为公开密钥，(d,n) 即为秘密密钥。加密时，对明文 M，密文为

$$C=M^e \bmod n \tag{9-8}$$

解密时，对密文 C，明文为

$$M=C^d \bmod n \tag{9-9}$$

用户可将加密密钥 (e,n) 公开，而解密密钥 (d,n) 和构成 n 的两个因子 p、q 是保密的。任何其他人都可用公开密钥 (e,n) 对该用户通信，只有掌握解密密钥的人才能解密，其他人在不知道 p 和 q 的情况下不可能根据已知的 e 推算出 d。

例 9-1　在 RSA 方法中，令 $p=11$，$q=23$，则

$$n=p\times q=11\times 23=253$$

$$\Phi(n)=(p-1)(q-1)=10\times 22=220$$

选取加密密钥 $e=3$，解同余方程 $(e\times d)\bmod\Phi(n)=1$，可解得 $d=147$。容易验证

$$3\times147=441=1(\text{mod } 220)$$

对于明文为 $M=165$，有密文

$$C=M^e=165^3=4492125=110(\text{mod } 253)$$

而对于密文 $C=110$，我们有明文

$$M=110^{147} \bmod 253=165$$

若需发送的报文内容是用英文或其他文字表示的，则可先将文字转换成等效的数字，再进行加密运算。RSA 体制在用于数字签名时，发送者为 A，接收者为 B，具体做法如图 9-10 所示。

$$M_i \xrightarrow{(d_A,\,n_A)} S_i \xrightarrow{(e_B,\,n_B)} S \xrightarrow{(d_B,\,n_B)} S_i \xrightarrow{(e_A,\,n_A)} M_i$$

图 9-10 RSA 公开密钥体制签名略图

发送者 A 用自己的秘密解密密钥 (d_A,n_A) 计算签名：$S_i=M_i^{d_A} \bmod n_A$。

用接收者的公开加密密钥 (e_B,n_B) 再次加密：$S=S_i^{e_B} \bmod n_B$。

接收者用自己的秘密解密密钥 (d_B,n_B) 计算：$S_i=S^{d_B} \bmod n_B$。

查发送者的公开密钥 (e_A,n_A)，计算：$M_i=S_i^{e_A} \bmod n_A$，恢复出发送者的签名，认证密文的来源。

9.5.3 RSA 的安全性

RSA 的安全性是基于分解大整数的困难性假定，之所以为假定是因为其困难性至今还未能证明。如果 RSA 的模数 n 被成功地分解为 pq，则立即获得 $\Phi(n)=(p-1)(q-1)$，从而能够确定 e 模 $\Phi(n)$ 的乘法逆元 d，即 $d=e^{-1} \bmod \Phi(n)$，因此攻击成功。

随着人类计算能力的不断提高，原来被认为是不可能分解的大数已被成功分解。例如，RSA-129（即 n 为 129 位十进制数，大约 428 比特）已在网络上通过分布式计算历时 8 个月于 1994 年 4 月被成功分解，RSA-130 已于 1996 年 4 月被成功分解，RSA-155 已于 1999 年 8 月被成功分解。为保证 RSA 的安全性，在实际应用中选取的素数 p 和 q 越来越大。现在来看，n 的长度为 1 024 位（二进制）至 2 048 位（二进制）是比较合理的。

除了指定 n 的大小之外，为避免选取容易分解的整数 n，研究人员建议对 p 和 q 采取如下限制。

① p 和 q 的长度应该相差不多。

② $p-1$ 和 $q-1$ 都应该包含大的素因子。

③ $\gcd(p-1,q-1)$ 应该很小（gcd 表示最大公约数）。

④ $d<n^{1/4}$，可以证明如果 $d>n^{1/4}$，则 d 可以较容易的被确定。

9.6 模拟信号加密

以上讨论的密码体制都属于数字加密的范畴，下面简要介绍模拟信号的加密问题。对模拟消息比如语音、传真、图像的保密主要有两种方式，一种是直接对模拟信号进行处理、变

换、置乱以进行加密，另一种则采用模/数/模的方式，加密是对数字信号进行的。前者称为模拟置乱加密，后者称为数字化加密。

9.6.1 模拟置乱加密

一个模拟信号的主要物理参量是幅度、频率和时间，只要对这三个参量之一或其组合进行变换和置乱就能改变模拟信号的结构，达到保密的目的。其中单独处理幅度、频率和时间的分别称为幅度置乱、频率（域）置乱和时间（域）置乱，并统一称为一维置乱。同时置乱其中的两种，就称为二维置乱。这种模拟置乱方式最大的优点是不改变信号的带宽，从而加密后的信号仍可以在现有的模拟信号的窄带信道上进行传输。

幅度置乱就是采用一个模拟信号来修正和掩蔽信号，以改变幅度使其发生变化。例如用伪随机噪声、单音、多音组合、音乐等信号压制语音，改变信源幅度随时间的变换。日常的干扰电台常采用这一技术对敌台进行干扰。这类方式安全性差，而且也很难恢复原来语音的质量，实际上很少使用。

频率置乱是语音保密中较早采用的一种体制，至今仍在使用。它包括简单的倒频、移带倒置、裂带倒频组合等方式。倒频是一种将语音信号谱进行简单倒置的变换，即将原来信号谱的低端变为高端，而将原来的高端变为低端。在频率置乱中真正有实用价值的是裂带、倒频组合方式，简称为频域置乱方式。这种方式的基本思想是将原始信号的频谱划分成相等的 m 个子带，m 一般取 5～12，然后各子带位置可以任意排列，若 $m=5$，则共有 5！种排列方法，而每一位置的子带又可进行倒频和不倒频两种选择，因此总共有 $5!\times 2^5=3\,840$ 种可能的选择。一般情况下则有 $(m!)\times 2^m$ 种可能的选择。显然这种组合式置乱以后可供选择的组合大大增加，但是并非任意组合都可以使用，在所有的可能组合中大约有 4/5 由于剩余可懂度大而不能采用。所谓剩余可懂度，是指经组合置乱以后仍然保留对原来语音消息的可懂程度。在能采用的 1/5 可能的选择之中只有 1/20 是好的组合，可供实际使用。为了增大可供使用的组合，可以加大取值 m，但由于实际实现的滤波器制作上的困难和语音质量下降等因素的影响又不能取得太大，故一般 $m=5\sim12$ 即可。在实际实现这种频域置乱体制时，密钥的选取即置乱组合模式的选取可以采用下列 3 种方式：①选择密钥的一种特定排列，并将其存入只读存储器，然后依次读出；②用密钥选择数种特定排列，存入只读存储器，然后依次读出；③将选定的子带置换图样存入只读存储器，并通过一个伪随机序列产生器输出作为只读存储器的地址控制码，伪随机地从只读存储器中选出一种组合。

时域置乱类似于频域置乱，只是被处理的参数由频带改为时隙，它实现起来比频域方便得多。它分为时隙段倒置和时隙置乱，但实际中常用的是后者。时隙置乱的基本原理是首先将模拟信号划分为帧，然后再将每帧划分为 m 个相等的时隙，将它送入一个时隙置乱器（易位器）中置乱，最后输出的就是经时域置乱变换后的加密模拟消息。其示意方框图如图 9-11 所示。

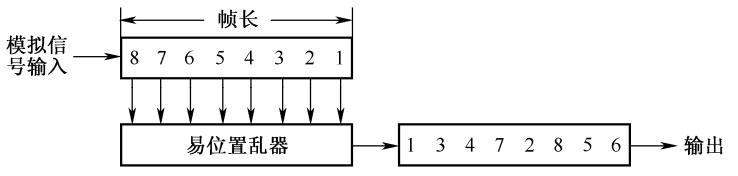


图 9-11 时域置乱原理方框图

由图 9-11 可见，将原始输入一帧模拟信号划分为 8 个时隙送入一个易位置乱器，输出为另一个置乱序号发生变化的 8 时隙模拟信号。若用矩阵形式可写成：输入 $\boldsymbol{m}=(m_1m_2m_3m_4m_5m_6m_7m_8)$，易位置乱为线性置乱，可用一置乱矩阵 $\boldsymbol{T}$ 表示，对本方案

$$\boldsymbol{T}=\begin{bmatrix}1&0&0&0&0&0&0&0\\0&0&0&0&1&0&0&0\\0&1&0&0&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&0&0&0&1&0\\0&0&0&0&0&0&0&1\\0&0&0&1&0&0&0&0\\0&0&0&0&0&1&0&0\end{bmatrix}$$

置乱后输出的密文

$$\boldsymbol{c}=\boldsymbol{mT}=(m_1m_3m_4m_7m_2m_8m_5m_6)$$

密文送出后，接收端收到密文 c 需进行相反的逆变换 T^{-1}，则可恢复出原来模拟语音时隙的正常位置排列，即

$$\boldsymbol{c}\boldsymbol{T}^{-1}=\boldsymbol{m}\boldsymbol{T}\boldsymbol{T}^{-1}=\boldsymbol{m}$$

这里，帧长与时隙长度对体制性能影响很大，一般希望时隙长度小到不能包含一个完整的单词，一般取 16～60ms 为宜。至于帧长度，从保密观点看，帧长越长越好，但从实现的方便和语音质量而言，帧长决定了语音的时延，所以帧长又不能太长，一般采用在一帧取 8～16 时隙为宜。

在确定帧长与时隙长度后，置乱矩阵选择决定了安全性能，若所有帧采用同一个固定的置乱矩阵 T，显然安全、保密性能不会好，当给定时隙个数 m 以后，矩阵 T 可能有 $m!$种置换方式，如果 $m=8$，$8!=40\,320$ 种，若每个置换使用 40ms，则在 3.6 小时以后才会出现重复，由于语音信号的特点并非所有类置换都能采用，一般仍根据剩余可懂度来确定是否可用，由于测试它的工作量大而复杂，且标准也不大一样，一般认为仅有不足 1/10 可用，但也有人认为高标准下仅有 1/1 000 是令人很满意的。

自动改变置乱的方法有两种：一种是采用伪随机数产生器产生任意置换，再检验是否能真正可用；第二种方法是将预先选好的置换存入只读存储器，而后通过伪随机数产生器进行控制，从而从只读存储器中读出一个置换供使用。

上面介绍的是一种定窗时隙置乱技术，它是以 8 个时隙为一帧进行帧内置乱的。另外还有人提出滑动窗时隙置乱与跳动窗时隙置乱技术，它们打破了定窗的界限，可以在一定范围内的相邻时隙之间进行置乱，即置乱窗口位置不再是固定的，而是可以沿着语音的时隙序列滑动或跳动，前者置换窗口宽度仍是固定的，如 m 位。当置乱器选定一位输出后，输入端的 $m+1$ 位将随之滑入置乱寄存器，这个过程逐位地进行下去。然而跳动窗则不同于滑动窗，开始时置乱寄存器中存在 m 位，当置乱器选定某一位输出后，输入端不动，置乱寄存器中少一位，变成 $m-1$ 位，第二次置乱器仅在这 $m-1$ 位中选取下一位输出，第三次置乱则仅在 $m-2$ 位选取一位输出。如此下去，直到置乱寄存器中全部选出，再送入 m 位，这时置乱寄存器的位数变化为：$m\to m-1\to m-2\to\cdots\to 2\to 1\to 0$。选取一个少一个，在不停地跳动。

9.6.2 数字化加密

数字化加密先将模拟信号数字化，而后进行数字信号置乱，最后再通过数/模转换成带宽基本不变的模拟信号送出，其原理图如图 9-12 所示。

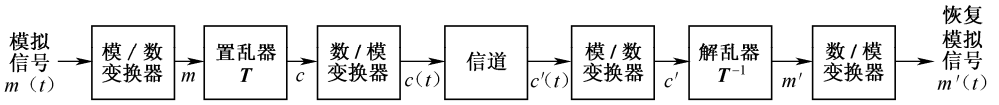


图 9-12 数字化加密系统框图

将原始的模拟明文信号 $m(t)$ 输入模/数变换器，变成一个 n 维的取样变量 $\boldsymbol{m}=[m_1 m_2 \cdots m_n]$，再送入一个 $n\times n$ 置乱矩阵 $\boldsymbol{T}$ 构成的置乱器，输出一个 n 维的密文变量 $\boldsymbol{c}=[c_1 c_2 \cdots c_n]$，即

$$\boldsymbol{c}=\boldsymbol{m}\,\square\,\boldsymbol{T} \tag{9-10}$$

经过数/模变换后输出模拟信号 $c(t)$ 到信道上，在接收端将接收的模拟信号 $c'(t)$ 经过模/数变换后形成 n 维取样变量 $\boldsymbol{c}'$ 送到解乱器，经过 $n\times n$ 维解乱矩阵 $\boldsymbol{T}^{-1}$ 变换后，得到恢复明文取样变量，即

$$\boldsymbol{m}'=\boldsymbol{c}'\boldsymbol{T}^{-1} \tag{9-11}$$

再经数/模变换后最后得到恢复的模拟信号 $m'(t)$。这一体制由于应用了现代的数字信号处理技术，其剩余可懂度低，密钥最大，保密性能好，它将模拟置乱技术提高到一个新的水平。数字化加密是模拟置乱中最有希望的置乱方式，其具体实现方式既可在时域上进行，也可在频域上进行。

习　　题

9.1 已知一种简单加密方式如下

$$c=5m+12 \bmod 26$$

若给出明文“Information”，试求加密后的密文。

9.2 设已知下列分组加密方程为

$$\begin{bmatrix} c_1 \\ c_2 \end{bmatrix}=\begin{bmatrix} 2 & 9 \\ 5 & 8 \end{bmatrix}\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \bmod 26$$

试求当明文为“data security”时，加密后的密文（加密时，可不计单字间空隙）。

9.3 对下面的每种情况求 d，并给出$(e\times d)(\text{mod})=1$：

（1）$p=5$，$q=11$，$e=3$；

（2）$p=3$，$q=41$，$e=23$；

（3）$p=5$，$q=23$，$e=59$；

（4）$p=47$，$q=59$，$e=17$。

9.4 用公开密钥$(e,n)=(5,51)$将报文 ABE,DEAD 用 A = 01，B = 02,…，进行加密。

9.5 用秘密密钥$(d,n)=(13,51)$将报文 4，1，5，1 进行解密。

9.6 若已知 DES 体制中 8 个 S 盒之一的 S 盒选择压缩函数如下：

行号＼列号	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	5	12	8	2	4	9	1	7	5	11	2	14	10	0	6	13

假设输入 S 盒的输入矢量 $\boldsymbol{X}=(010011)$，求通过选择压缩函数 S 变换后的输出矢量。

9.7 用户 A 发送给 B 一份密文，用户 A 用首字母 B = 02 来签署密文，用户 A 知道 3 个密钥：自己的公开加密密钥，秘密解密密钥和接收者的公开加密密钥。

	A	B
公开密钥(e，n)	(7，123)	(13，51)
秘密密钥(d，n)	(23，123)	

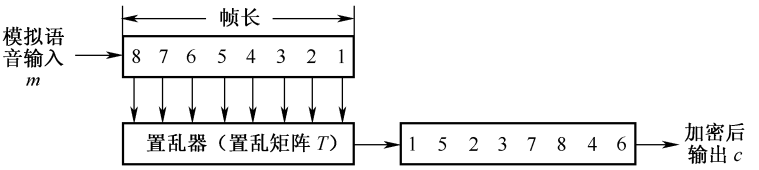
试描述整个数字签名的过程。

9.8 接收者 B 必须恢复出 02 = B 来认证他接收的密文。他也知道 3 个密钥：两个公开加密密钥和自己的秘密解密密钥。

	A	B
公开密钥(e,n)	(7,123)	(13,51)
秘密密钥(d,n)		(5,51)

试描述整个验证签名的过程。

9.9 若有一时域置乱器结构如题图 9-9 所示。



题图 9-9

其中，$\boldsymbol{m}=[1,2,3,4,5,6,7]$，$\boldsymbol{c}=\boldsymbol{mT}=[1,5,2,3,7,8,4,6]$。试求：置乱矩阵 $\boldsymbol{T}$；为了能恢复出原来语音，求 $\boldsymbol{T}^{-1}$。

参 考 文 献

[1] 吴伟陵．信息处理与编码．北京：人民邮电出版社，1999．

[2] 周炯槃．信息理论基础．北京：人民邮电出版社，1983．

[3] 王育民，等．信息与编码理论．西安：西安电子科技大学出版社，1986．

[4] R.J.Mceliece. The Theory of Information and Coding．

[5] 西安交通大学，南京工学院，清华大学合编，信息与系统（上册）．北京：国防工业出版社，1980．

[6] 曹雪红，张宗橙．信息论与编码．北京：北京邮电大学出版社，2001．

[7] 戴善荣．数据压缩．西安：西安电子科技大学出版社，2005．

[8] 陈运，等．信息论与编码．北京：电子工业出版社，2002．

[9] 李亦农，李梅编著．信息论基础教程．北京：北京电电大学出版社，2002.

[10] 陈嘉生，沈世镒．现代密码学．北京：科学出版社，2002．

[11] 刘嘉勇，等．应用密码学．北京：清华大学出版社，2008．

[12] 于工，等．现代密码学原理与实践．西安：西安电子科技大学出版社，2009．

[13] 田丽华．信息论、编码与密码学．西安：西安电子科技大学出版社，2008．