

信息论与编码

曹雪虹 张宗橙 编

北京邮电大学出版社

·北京·

内 容 简 介

本书重点介绍信息论的基本理论以及编码的理论和实现原理。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、信道编码和密码学中的理论知识及其实现原理。本书注重概念,采用通俗的文字,联系目前通信系统,用较多的例题和图示阐述基本概念、基本理论及实现原理,尽量减少繁杂的公式定理证明;在各章的最后还附有大量习题,便于加深理解。本书可作为理工科高等院校信息工程、通信工程及相关专业的本科学生教材,亦可供信息、通信、电子等有关专业的科技人员作为参考书。

图书在版编目(CIP)数据

信息论与编码/曹雪虹,张宗橙编.—北京:北京邮电大学出版社,2001.8

ISBN 7 5635 0524 5

. 信 曹 ... 张 信息论 信息—编码 . TN911.2

中国版本图书馆 CIP 数据核字(2001)第 048272 号

出版发行:北京邮电大学出版社

网 址: www.buptpress.com

社 址:北京市海淀区西土城路 10 号(100876)

电话传真:010 62282185(发行部)/010 62283578(FAX)

E mail: publish@bupt.edu.cn

经 销:各地新华书店

印 刷:北京忠信诚胶印厂印刷

开 本:787mm×1 092mm 1/16

印 张:12

字 数:308 千字

印 数:3 001—6 000 册

版 次:2001 年 8 月第 1 版 2002 年 1 月第 2 次印刷

前 言

当前信息产业发展很快,需要大量从事信息、通信、电子工程类专业的人才,而《信息论与编码》是这些专业的基础课,必须掌握,它可以指导理论研究和工程应用。

由于《信息论与编码》这门课本身理论性很强,介绍的内容是信息论基础和编码理论,现有的一些教材除了介绍理论和公式外,都用了大量篇幅来证明这些理论和公式,这些用作研究生教材是比较适合的。

而作为电子、信息、通信工程的本科生及相关专业的工程技术人员,由于他们理论基础的不足以及实际应用的需要,不可能花很多精力去研读那些在他们看来是非常难懂而枯燥乏味的证明,迫切需要一本介绍有关信息理论的基本知识且与实际应用紧密联系的书籍,本书就是出于这样的目的而写。

本书注重基本概念,用较通俗的文字解释其物理意义,辅以一定的例题和图示说明,不再用繁杂的公式来证明这些早已是非常成熟的公理。联系当前实际通信技术来讲述,使读者研读本书后概念清晰,有目标地应用在实际工作中。

本书共 6 章,由曹雪虹主编。第 5 章由张宗橙编写,其余各章由曹雪虹编写。在编写过程中,得到了徐澄圻教授和胡建彰教授的大力帮助,在此表示衷心感谢。

限于编者的水平,书中不妥或谬误之处难免,殷切希望读者指正。

编 者

2001 年 6 月

目 录

第 1 章 绪 论	1
1.1 信息论的形成和发展	1
1.2 通信系统的模型	3
第 2 章 信源及信源熵.....	6
2.1 信源的描述和分类	6
2.2 离散信源熵和互信息	7
2.2.1 自信息量	7
2.2.2 离散信源熵	8
2.2.3 互信息.....	10
2.2.4 数据处理中信息的变化.....	12
2.2.5 熵的性质.....	13
2.3 连续信源的熵和互信息.....	15
2.3.1 连续信源熵.....	15
2.3.2 最大熵定理.....	17
2.4 离散序列信源的熵.....	18
2.4.1 离散无记忆信源的序列熵.....	18
2.4.2 离散有记忆信源的序列熵.....	18
2.5 冗余度.....	27
习 题.....	29
第 3 章 无失真信源编码	34
3.1 编码的定义.....	34
3.2 定长编码定理.....	37
3.3 变长编码定理.....	40
3.4 最佳编码.....	42
3.4.1 香农编码方法.....	42
3.4.2 费诺编码方法.....	43
3.4.3 哈夫曼编码方法.....	44
习 题.....	48
第 4 章 限失真信源编码	51
4.1 平均失真和信息率失真函数.....	51
4.1.1 失真函数.....	51
4.1.2 平均失真.....	52

4.1.3	信息率失真函数 $R(D)$	53
4.1.4	信息率失真函数的性质.....	55
4.2	离散信源和连续信源的 $R(D)$ 计算.....	58
4.3	限失真信源编码定理.....	60
4.4	常用信源编码方法简介.....	61
4.4.1	游程编码.....	61
4.4.2	算术编码.....	62
4.4.3	矢量量化.....	66
4.4.4	预测编码.....	68
4.4.5	变换编码.....	70
习 题	73
第 5 章	信道编码	75
5.1	信道模型和信道容量.....	75
5.1.1	信道模型.....	75
5.1.2	信道容量.....	77
5.2	有扰离散信道的编码定理.....	83
5.2.1	随机编码.....	83
5.2.2	编码定理.....	85
5.3	差错控制与信道编译码的基本原理.....	88
5.3.1	差错控制的途径.....	88
5.3.2	码距与纠、检错能力	91
5.3.3	最优译码与最大似然译码.....	92
5.4	线性分组码.....	94
5.4.1	线性分组码基本概念.....	94
5.4.2	生成矩阵和校验矩阵.....	97
5.4.3	伴随式与译码	101
5.4.4	循环码	106
5.5	卷积码	114
5.5.1	卷积码的基本概念和描述方法	114
5.5.2	卷积码的最大似然译码——维特比算法	120
5.5.3	卷积码的性能限与距离特点	127
5.6	网格编码调制与级联码简介	130
5.6.1	网格编码调制	130
5.6.2	级联码简介	136
习 题	141
第 6 章	密码学.....	145
6.1	密码学的基础知识	145
6.1.1	密码学的基本概念	145
6.1.2	密码学中的熵概念	148

6.2 数据加密标准 DES	150
6.2.1 换位和替代密码	150
6.2.2 DES 密码算法	151
6.2.3 DES 密码的安全性	155
6.2.4 DES 密码的改进	157
6.3 国际数据加密算法	158
6.3.1 算法原理	158
6.3.2 加密解密过程	159
6.3.3 算法的安全性	161
6.4 公开密钥加密法	161
6.4.1 公开密钥密码体制	161
6.4.2 RSA 密码体制	162
6.4.3 报文摘要	164
6.5 模拟信号加密	168
6.6 通信网络中的加密	168
6.7 信息安全和确认技术	169
6.7.1 信息安全的基本概念	170
6.7.2 数字签名	170
6.7.3 防火墙	173
6.7.4 密码学在电子支付系统中的应用	174
6.7.5 密码学在电子数据交换中的应用	175
习 题	175
附录:符号及含义	176
部分习题参考答案.....	179
参考文献.....	183

第 1 章 绪 论

“信息”这个词相信大家不陌生,几乎每时每刻都会接触到。不仅在通信、电子行业,其他各个行业也都十分重视信息,所谓进入了“信息时代”。信息不是静止的,它会产生也会消亡,人们需要获取它,并完成它的传输、交换、处理、检测、识别、存储、显示等功能。研究这方面的科学就是信息科学,信息论是信息科学的主要理论基础之一。它研究信息的基本理论 (Information Theory),主要研究可能性和存在性问题,为具体实现提供理论依据。与之对应的是信息技术 (Information Technology),主要研究如何实现、怎样实现的问题。

通过本章的学习,可以了解下列问题:信息论的形成和发展;信息论研究的内容及信息的基本概念;并结合通信系统模型介绍模型中各部分的作用及编码的种类和研究内容。

1.1 信息论的形成和发展

信息论理论基础的建立,一般来说开始于香农 (C. E. Shannon) 研究通信系统时所发表的论文。随着研究的深入与发展,信息论具有了较为宽广的内容。

信息在早些时期的定义是由奈奎斯特 (Nyquist, H.) 和哈特莱 (Hartley, L. V. R.) 在 20 世纪 20 年代提出来的。1924 年奈奎斯特解释了信号带宽和信息速率之间的关系;1928 年哈特莱最早研究了通信系统传输信息的能力,给出了信息度量方法;1936 年阿姆斯特朗 (Armstrong) 提出了增大带宽可以使抗干扰能力加强。这些工作都给香农很大的影响,他在 1941 ~ 1944 年对通信和密码进行深入研究,用概率论的方法研究通信系统,揭示了通信系统传递的对象就是信息,并对信息给以科学的定量描述,提出了信息熵的概念。指出通信系统的中心问题是在噪声下如何有效而可靠地传送信息以及实现这一目标的主要方法是编码等。这一成果于 1948 年以《通信的数学理论》 (A mathematical theory of communication) 为题公开发表。这是一篇关于现代信息论的开创性的权威论文,为信息论的创立作出了独特的贡献。香农因此成为信息论的奠基人。

50 年代信息论在学术界引起了巨大的反响。1951 年美国 IRE 成立了信息论组,并于 1955 年正式出版了信息论汇刊。60 年代信道编码技术有较大进展,使它成为信息论的又一重要分支。它把代数方法引入到纠错码的研究,使分组码技术发展到了高峰,找到了大量可纠正多个错误的码,而且提出了可实现的译码方法。其次是卷积码和概率译码有了重大突破;提出了序列译码和 Viterbi 译码方法。

信源编码的研究落后于信道编码。香农 1959 年的文章 (Coding theorems for a discrete source with a fidelity criterion) 系统地提出了信息率失真理论,它是数据压缩的数学基础,为各种信源编码的研究奠定了基础。

到 70 年代,有关信息论的研究,从点与点间的单用户通信推广到多用户系统的研究。1972 年盖弗 (Cover) 发表了有关广播信道的研究,以后陆续有关于多接入信道和广播信道模型的研究,但由于这些问题比较难,到目前为止,多用户信息论研究得不多,还有许多尚待

解决的课题。

信息论主要应用在通信领域,在含噪信道中传输信息的最优方法到今天还不十分清楚。特别是当数据的信息量大于信道容量的情况,更是毫无所知,这是经常遇到的情况。因为从信源提取的信息常常是连续的,也就是信号的信息含量为无限大。在一般信道中传输这样的信号,是不可能不产生误差的。引入信道容量和信息量的概念以后,这类问题就可以得到满意的解释,并可给出一个通信系统的最佳效果,这样就为设计通信系统提供了理论依据。

信息论是在信息可以量度的基础上,研究有效地和可靠地传递信息的科学,它涉及信息量度、信息特性、信息传输速率、信道容量、干扰对信息传输的影响等方面的知识。通常把上述范围的信息论称为狭义信息论,又因为它的创始人是香农,故又称为香农信息论。广义信息论则包含通信的全部统计问题的研究,除了香农信息论之外,还包括信号设计、噪声理论、信号的检测与估值等。当信息在传输、存储和处理的过程中,不可避免地要受到噪声或其它无用信号的干扰,信息理论就是为能可靠地有效地从数据中提取信息,提供必要的根据和方法。这就必须研究噪声和干扰的性质以及它们与信息本质上的差别,噪声与干扰往往具有按某种统计规律的随机特性,信息则具有一定的概率特性,如度量信息量的熵值就是概率性质的。因此,信息论、概率论、随机过程和数理统计学是信息论应用的基础和工具。

本书讲述的信息理论的基本内容是与通信科学密切相关的狭义信息论,涉及到信息理论中很多基本问题。例如:

- (1) 什么是信息?如何度量信息?
- (2) 在信息传输中,基本的极限条件是什么?
- (3) 信息的压缩和恢复的极限条件是什么?
- (4) 从环境中抽取信息的极限条件是什么?
- (5) 设计什么样的设备才能达到这些极限?
- (6) 实际上接近极限的设备是否存在?

在信息论和通信理论中经常会遇到信息、消息和信号这三个既有联系又有区别的名词。下面将它们的定义比较如下:

信息:信息是指各个事物运动的状态及状态变化的方式。人们从来自对周围世界的观察得到的数据中获得信息。信息是抽象的意识或知识,它是看不见、摸不到的。人脑的思维活动产生的一种想法,当它仍储存在脑子中的时候它就是一种信息。

消息:消息是指包含有信息的语言、文字和图像等,例如我们每天从广播节目、报纸和电视节目中获得各种新闻及其他消息。在通信中,消息是指担负着传送信息任务的单个符号或符号序列。这些符号包括字母、文字、数字和语言等。单个符号消息的情况,例如用 x_1 表示晴天, x_2 表示阴天, x_3 表示雨天。符号序列消息的情况,例如“今天是晴天”这一消息由 5 个汉字构成。可见消息是具体的,它载荷信息,但它不是物理性的。

信号:信号是消息的物理体现,为了在信道上传输消息,就必须把消息加载(调制)到具有某种物理特征的信号上去。信号是信息的载荷子或载体,是物理性的。如电信号、光信号等。

按照信息论或控制论的观点,在通信和控制系统中传送的本质内容是信息,系统中实际传输的则是测量的信号,信息包含在信号之中,信号是信息的载体。信号到了接收端(信息论里称为信宿)经过处理变成文字、语声或图像,人们再从中得到有用的信息。在接收端将

含有噪声的信号经过各种处理和变换,从而取得有用信息的过程就是信息提取,提取有用信息的过程或方法主要有检测和估计两类。载有信息的可观测、可传输、可存储及可处理的信号均称为数据。

- 信息的基本概念在于它的不确定性,任何已确定的事物都不含有信息。其特征有:
- 接收者在收到信息之前,对它的内容是不知道的,所以信息是新知识、新内容;
- 信息是能使认识主体对某一事物的未知性或不确定性减少的有用知识;
- 信息可以产生,也可以消失,同时信息可以被携带、贮存及处理;
- 信息是可以量度的,信息量有多少的差别。

1.2 通信系统的模型

图 1 2 1 是目前较常用的、也是较完整的通信系统模型,下面介绍模型中各个部分的作用及需要研究的核心问题。

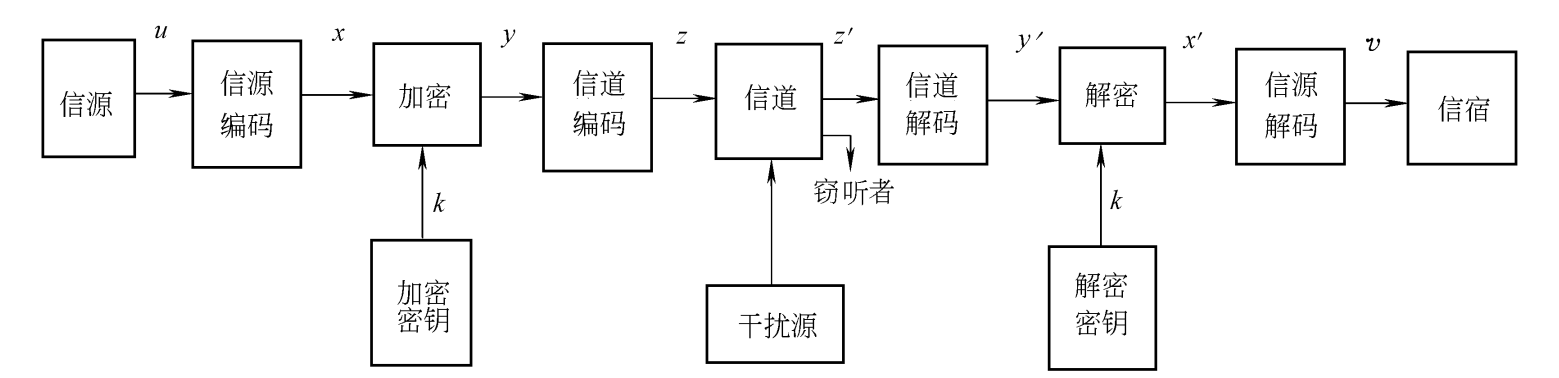


图 1 2 1 通信系统的物理模型

信源是向通信系统提供消息 (u) 的人和机器。信源本身是十分复杂的,在信息论中我们仅研究信源的输出。信源输出的是以符号形式出现的具体消息,它载荷信息。信源输出的消息可以有多种形式,但可归纳成两类: 离散消息,例如由字母、文字、数字等符号组成的符号序列或者单个符号。 连续消息,例如话音、图像、在时间上连续变化的电参数等。因为通信系统的接收者(信宿)在收到消息之前并不知道信源所发出消息的内容,所以一般地说信源发出的是随机性的消息。但因信源发出的消息都携带着信息,可见消息的变化是具有一定规律性的,因此严格地说信源发出消息并不是完全随机性的。信源的核心问题是它包含的信息到底有多少,怎样将信息定量地表示出来,即如何确定信息量。

信宿是消息传递的对象,即接收消息的人或机器。根据实际需要,信宿接收的消息 (v) 其形式可以与信源发出的消息 (u) 相同,也可以不相同,当两者形式不相同时, v 是 u 的一个映射。信宿需要研究的问题是能收到或提取多少信息。

信道是传递消息的通道,又是传送物理信号的设施。信道可以是一对导线、一条同轴电缆、传输电磁波的空间、一条光导纤维等传输信号的媒质。信道的问题主要是它能够传送多少信息的问题,即信道容量的大小。

干扰源是整个通信系统中各个干扰的集中反映,用以表示消息在信道中传输时遭受干扰的情况。对于任何通信系统而言,干扰的性质、大小是影响系统性能的重要因素。

密钥源是产生密钥 k 的源。信源编码器输出信号 x 经过 k 的加密运算后,就把明文 x

变换为密文 y 。若窃听者未掌握发端采用的密钥 k , 则他就很难从窃听到的信号 z 解出明文。而收端的信宿则因知道事先已约定好的密钥 k , 因此能从收到的信号 z 解出明文。对于二进制的代码而言, 加密相当于 $y = x \oplus p$ 运算(其中序列 p 通常是受密钥控制的伪随机序列), 而解密则相当于 $x = y \oplus p$ 运算。这里 x, y, z 之所以不同于发端的 x, y, z , 是考虑到信号 z 在信道中传输时所受到的干扰影响。但在正常通信条件下, 总会有 $x \rightarrow x, y \rightarrow y, z \rightarrow z$ 的结果。

一般地说, 通信系统的性能指标主要是有效性、可靠性、安全性和经济性。通信系统优化就是使这些指标达到最佳。除了经济性外, 这些指标正是信息论的研究对象。根据信息论的各种编码定理和上述通信系统的指标, 编码问题可分解为三类: 信源编码、信道编码和密码。

信源编码器的作用是把信源发出的消息变换成由二进制码元(或多进制码元)组成的代码组, 这种代码组就是基带信号。同时通过信源编码可以压缩信源的冗余度(即多余度), 以提高通信系统传输消息的效率。信源编码可分为无失真信源编码和限失真信源编码。前者适用于离散信源或数字信号, 后者主要用于连续信源或模拟信号, 如语音、图像等信号的数字处理。从提高通信系统的有效性意义上说, 信源编码器的主要指标是它的编码效率, 即理论上能达到的码率与实际达到的码率之比。一般来说, 效率越高, 编译码器的代价也将越大。信源译码器的作用是把信道译码器输出的代码组变换成信宿所需要的消息形式, 它的作用相当于信源编码器的逆过程。

信道编码器的作用是在信源编码器输出的代码组上有目的地增加一些监督码元, 使之具有检错或纠错的能力。信道译码器具有检错或纠错的功能, 它能将落在其检错或纠错范围内的错传码元检出或纠正, 以提高传输消息的可靠性。信道编码包括调制解调和纠错检错编译码。信道中的干扰常使通信质量下降, 对于模拟信号, 表现在收到的信号的信噪比下降; 对于数字信号, 就是误码率增大。信道编码的主要方法是增大码率或频带, 即增大所需的信道容量。这恰与信源编码相反。

密码学是研究如何隐蔽消息中的信息内容, 使它在传输过程中不被窃听, 提高通信系统的安全性。将明文变换成密文, 通常不需要增大信道容量, 例如在二进码信息流上叠加一密钥流; 但也有些密码要求占用较大的信道容量。

在实际问题中, 上述三类编码应统一考虑来提高通信系统的性能。这些编码的目标往往是相互矛盾的。提高有效性必须去掉信源符号中的冗余部分, 此时信道误码会使接收端不能恢复原来的信息, 也就是必须相应提高传送的可靠性, 不然会使通信质量下降; 反之, 为了可靠而采用信道编码, 往往需扩大码率, 也就降低了有效性。安全性也有类似情况。编成密码, 有时需扩展码位, 这样就降低有效性; 有时也会因失真而使授权用户无法获得信息, 必须重发而降低有效性, 或丢失信息而降低可靠性。从理论方面来说, 若能把三种码合并成一种码来编译, 即同时考虑有效、可靠和安全, 可使编译码器更理想化, 在经济上可能也更优越。这种三码合一的设想是当前众所关心的课题, 但因理论上和技术上的复杂性, 要取得有用的结果, 还是相当困难。值得注意的是信息论分析的问题是存在性问题, 即符合条件的编码是存在的, 但并没有给出如何去寻找。

本书用了大量篇幅讨论编码问题, 着重介绍信源和信道的编码定理, 主要从概念上解释了这些定理的结论, 并没有从严格意义上加以证明。顺便指出, 不是所有的通信系统都采用

如图 1 2 1 所示的那样全面的技术。例如,点对点的有线电话,只要有一对电话机和一条电话线路(铜线)就够了,话音基带信号通过电话机变成相应的电信号(模拟信号),就能在电话线上传送,收端的电话机再把电信号恢复成人耳能听得清的话音。如果是点对点的无线电话,则在发端需要一台发信机,把模拟信号调制到射频上,再用大功率发射机经天线发射出去,然后在无线信道中传输;收端则应使用受信机把收到的调制射频信号解调恢复为发端的原始话音。若在这样的系统中增加加密和解密装置,就构成无线保密通信系统。在干扰大、信道容量有限的通信系统中,就需要采用信源编码和信道编码技术,以提高传输消息的有效性和可靠性。

这里首先举几个例子来说明编码的应用,如电报常用的莫尔斯码就是按信息论的基本编码原则设计出来的;在一些商品上面有一张由粗细条纹组成的标签,从这张标签可以得知该商品的生产厂家、生产日期和价格等信息,这些标签是利用条形码设计出来的,非常方便,非常有用,应用越来越普遍;计算机的运算速度很高,要保证它几乎不出差错,相当于要求有 100 年的时间内不得有一秒钟的误差,这就需要利用纠错码来自动地及时地纠正所发生的错误;每出版一本书,都给定一个国际标准书号(ISBN),大大方便图书的销售、编目和收藏工作。可以说,人们在日常生活和生产实践中,正在越来越多地使用编码技术。

本书的内容安排如下:

第 2 章介绍信息论的一些基本概念,包括自信息量、条件自信息量、互信息量、条件互信息量、平均互信息量、单符号熵、熵的性质以及连续信源熵、最大熵定理和随机序列的熵等,并解释了冗余度的由来及作用。

第 3 章介绍无失真信源编码定理,包括定长编码定理和变长编码定理,并详细阐述了最佳编码中的香农码、费诺码和霍夫曼码的编码方法及其性能比较。

第 4 章主要介绍了失真函数和信息率失真函数的定义及性质,简述了限失真信源编码定理。最后还简单提及了常用的几种信源编码方法。

第 5 章介绍信道及信道编码,其中包括信道、信道容量等基本概念,以及信道编码定理,还介绍了差错控制与信道编译码的基本原理及线性分组码、卷积码、级联码的基本原理。

第 6 章在介绍密码体制的基础知识及其熵的概念后,简述了具有代表性的秘密密钥加密算法 DES, IDEA 和公开密钥加密算法 RSA 和 MD5 等。还引入了信息安全性概念以及数字签名、防火墙等技术。

第 2 章 信源及信源熵

2.1 信源的描述和分类

在信息论中,信源是发出消息的源,信源输出以符号形式出现的具体消息。如果符号是确定的而且预先知道的,那么该消息就无信息可言。只有当符号的出现是随机的,预先无法确定,一旦出现某个符号就给观察者提供了信息。因此可用随机变量或随机矢量来表示信源,运用概率论和随机过程的理论来研究信息,这就是香农信息论的基本点。

实际应用中分析信源所采用的方法往往依信源特性而定。按照信源发出的消息在时间和幅度上的分布情况可将信源分成离散信源和连续信源两大类。离散信源是指发出在时间和幅度上都是离散分布的离散消息的信源,如文字、数字、数据等符号都是离散消息。连续信源是指发出在时间和幅度上都是连续分布的连续消息(模拟消息)的信源,如语言、图像、图形等都是连续消息。

下面来分析离散情况。离散信源可进一步分类:

离散信源	离散无记忆信源	发出单个符号的无记忆信源
		发出符号序列的无记忆信源
	离散有记忆信源	发出符号序列的有记忆信源
		发出符号序列的马尔可夫信源

发出单个符号的信源是指信源每次只发出一个符号代表一个消息;发出符号序列的信源是指信源每次发出一组含二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,各个符号的出现概率是它自身的先验概率。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表示,一种是用信源发出的一个符号序列的整体概率(即联合概率)反映有记忆信源的特征,这就是上图中发出符号序列的有记忆信源。一般情况下,表述有记忆信源要比表述无记忆信源困难得多,尤其当记忆长度很长甚至无限长时。在实际问题中,我们往往试图限制记忆长度,即某一个符号出现的概率只与前面一个或有限个符号有关,而不依赖更前面的那些符号,这样的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记忆特征,这就是发出符号序列的马尔可夫信源。

例如一个离散信源发出的各个符号消息的集合为 $X = \{x_1, x_2, \dots, x_n\}$, 它们的概率分别为 $P = \{p(x_1), p(x_2), \dots, p(x_n)\}$, $p(x_i)$ 称为符号 x_i 的先验概率。通常把它们写到一起,称为概率空间:

$$\begin{array}{ccccccc} X & & x_1 & & x_2 & & \dots & & x_n \\ & = & & & & & & & \\ P & & p(x_1) & & p(x_2) & & \dots & & p(x_n) \end{array}$$

显然有 $p(x_i) \geq 0$, $\sum_{i=1}^n p(x_i) = 1$ 。

最简单的有记忆信源是 $N=2$ 的情况,此时信源 $X = X_1 X_2$,其信源的概率空间为

$$\begin{array}{ccccccc} X & & a_1 a_1 & & a_1 a_2 & & \dots & & a_q a_q \\ P & = & p(a_1 a_1) & & p(a_1 a_2) & & \dots & & p(a_q a_q) \end{array}$$

对于无记忆信源联合概率为 $p(x_1, x_2, \dots, x_n) = p(x_1) p(x_2) \dots p(x_n)$,当进一步满足平稳性时, $p(x_1, x_2, \dots, x_n) = p(x_1) p(x_2) \dots p(x_n) = p^n$ 。

在分析有记忆信源时,有时也可将多个符号合并成一个符号来处理。例如有 L 个符号,每个符号取值于 A 空间,有 n 种可能性。将这 L 个符号组成一个 L 维随机矢量,则该随机矢量取值于 A^L 空间,共 n^L 个可能的取值,这样就把有记忆的 L 个符号的信源转化成单符号问题。

2.2 离散信源熵和互信息

2.2.1 自信息量

一个随机事件的自信息量定义为其出现概率对数的负值。即

$$I(x_i) = -\log p(x_i) \quad (2.2.1)$$

因为概率 $p(x_i)$ 越小, x_i 的出现就越稀罕,一旦出现,所获得的信息量应是较大的。由于 x_i 是随机出现的,它是 X 的一个样值,所以是一个随机量。而 $I(x_i)$ 是 x_i 的函数,它必须也是一个随机量。

自信息量的单位与所用的对数底有关。在信息论中常用的对数底是 2,信息量的单位为比特(bit);若取自然对数,则信息量的单位为奈特(nat);若以 10 为对数底,则信息量的单位为笛特(det)。这三个信息量单位之间的转换关系如下:

$$1 \text{ nat} = \log_2 e \approx 1.433 \text{ bit}, \quad 1 \text{ det} = \log_2 10 \approx 3.322 \text{ bit}$$

如一个以等概率出现的二进制码元(0,1)所包含的自信息量为:

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

若是一个 m 位的二进制数,因为该数的每一位可从 0,1 两个数字中任取一个,因此有 2^m 个等概率的可能组合。所以 $I = -\log_2 \frac{1}{2^m} = m \text{ bit}$,就是需要 m 比特的信息来指明这样的二进制数。

这里要引入随机事件的不确定度概念。根据日常知识,各个出现概率不同的随机事件所包含的不确定度是有差别的。一个出现概率接近于 1 的随机事件,发生的可能性很大,所以它包含的不确定度就很小。反之,一个出现概率很小的随机事件,很难猜测在某个时刻它能否发生,所以它包含的不确定度就很大。若是确定性事件,出现概率为 1,则它包含的不确定度为 0。注意:随机事件的不确定度在数量上等于它的自信息量,两者的单位相同,但含义却不相同。具有某种概率分布的随机事件不管发生与否,都存在不确定度,不确定度表征了该事件的特性,而自信息量是在该事件发生后给予观察者的信息量。

若有两个消息 x_i, y_j 同时出现,可用联合概率 $p(x_i y_j)$ 来表示,这时的联合自信息量定义为

$$I(x_i y_j) = -\log p(x_i y_j) \quad (2.2.2)$$

当 x_i 和 y_j 相互独立时,有 $p(x_i y_j) = p(x_i) p(y_j)$,那么就有 $I(x_i y_j) = I(x_i) + I(y_j)$ 。

所包含的不确定度在数值上也等于它们的自信息量。

若两个消息出现不是独立的,而是有相互联系的,这时可用条件概率 $p(x_i/y_j)$ 来表示,即在事件 y_j 出现的条件下,随机事件 x_i 发生的条件概率,则它的条件自信息量定义为条件概率对数的负值:

$$I(x_i/y_j) = -\log p(x_i/y_j) \quad (2.2.3)$$

在给定 y_j 条件下,随机事件 x_i 所包含的不确定度在数值上与条件自信息量相同,但两者含义不同。

由于一个随机事件的概率和条件概率总是在闭区间 $[0,1]$ 内,所以自信息量和条件自信息量均为非负值。

例 2.2.1 英文字母中“e”的出现概率为 0.105,“c”的出现概率为 0.023,“o”的出现概率为 0.001。分别计算它们的自信息量。

解:“e”的自信息量 $I(e) = -\log_2 0.105 = 3.25 \text{ bit}$

“c”的自信息量 $I(c) = -\log_2 0.023 = 5.44 \text{ bit}$

“o”的自信息量 $I(o) = -\log_2 0.001 = 9.97 \text{ bit}$

2.2.2 离散信源熵

例 2.2.2 一个布袋内放 100 个球,其中 80 个球是红色的,20 个球是白色的,若随机摸取一个球,猜测其颜色,求平均摸取一次所能获得的自信息量。

这一随机事件的概率空间为

$$\begin{array}{c} X \\ P \end{array} = \begin{array}{cc} x_1 & x_2 \\ 0.8 & 0.2 \end{array}$$

其中 x_1 表示摸出的球为红球事件, x_2 表示摸出的球是白球事件。

这是一个随机事件试验。试验结果,当被告知摸出的是红球,则获得的信息量是

$$I(x_1) = -\log_2 p(x_1) = -\log_2 0.8 \text{ bit}$$

当被告知摸出的是白球,那么获得的信息量是

$$I(x_2) = -\log_2 p(x_2) = -\log_2 0.2 \text{ bit}$$

如果每次摸出一个球后又放回袋中,再进行下一次摸取。那么如此摸取 n 次,红球出现的次数为 $np(x_1)$ 次,白球出现的次数为 $np(x_2)$ 次。随机摸取 n 次后总共所获得的信息量为

$$np(x_1)I(x_1) + np(x_2)I(x_2)$$

而平均随机摸取一次所获得的信息量则为

$$\begin{aligned} H(X) &= \frac{1}{n} [np(x_1)I(x_1) + np(x_2)I(x_2)] \\ &= -[p(x_1)\log_2 p(x_1) + p(x_2)\log_2 p(x_2)] \\ &= -\sum_{i=1}^2 p(x_i)\log_2 p(x_i) = 0.72 \text{ 比特/次} \end{aligned}$$

从此例可以看出,自信息量 $I(x_1)$ 和 $I(x_2)$ 只是表征信源中各个符号的不确定度,一个信源总是包含着多个符号消息,各个符号消息又按概率空间的先验概率分布,因而各个符号的自信息量就不同。所以自信息量不能作为信源总体的信息量,而上面求出的平均自信息

量,也即信息熵 $H(X)$ 是从平均意义上来表征信源的总体特征,可以表征信源的平均不确定性。我们定义信源的平均不确定度 $H(X)$ 为信源中各个符号不确定度的数学期望。即

$$H(X) = E[I(X)] = \sum_i p(x_i) I(x_i) = - \sum_i p(x_i) \log p(x_i) \quad (2.2.4)$$

单位为比特/符号或比特/符号序列。

因为 X 中各符号 x_i 的不确定度 $I(x_i)$ 为非负值, $p(x_i)$ 也是非负值,且 $0 < p(x_i) < 1$, 故信源的平均不确定度 $H(X)$ 也是非负量。平均不确定度 $H(X)$ 的定义公式与热力学中熵的表示形式相同,所以又把 $H(X)$ 称为信源 X 的熵。熵是在平均意义上来表征信源的总体特性的。正如不确定度与自信息量的关系那样,信源熵是表征信源的平均不确定度,平均自信息量是消除信源不确定度时所需要的信息的量度,即收到一个信源符号,全部解除了这个符号的不确定度。或者说获得这样大的信息量后,信源不确定度就被消除了。两者在数值上相等,但含义不同。某一信源,不管它是否输出符号,只要这些符号具有某些概率特性,必有信源的熵值;这熵值是在总体平均上才有意义,因而是一个确定值,一般写成 $H(X)$, X 是指随机变量的整体(包括概率分布)。而另一方面,信息量则只有当信源输出符号而被接收者收到后,才有意义,这就是给予接收者的信息度量,这值本身也可以是随机量,也可以与接收者的情况有关,如考虑信息的有用性时就是如此。

在(2.2.4)式中,当某一符号 x_i 的概率 p_i 为零时, $p_i \log p_i$ 在熵公式中无意义,为此规定这时的 $p_i \log p_i$ 也为零。当信源 X 中只含一个符号 x 时,必定有 $p(x) = 1$, 此时信源熵 $H(X)$ 为零。

例 2.2.3 电视屏上约有 $500 \times 600 = 3 \times 10^5$ 个格点,按每点有 10 个不同的灰度等级考虑,则共能组成 $n = 10^{3 \times 10^5}$ 个不同的画面。按等概计算,平均每个画面可提供的信息量为

$$\begin{aligned} H(X) &= - \sum_{i=1}^n p(x_i) \log_2 p(x_i) = - \log_2 10^{-3 \times 10^5} \\ &= 3 \times 10^5 \times 3.32 = 10^6 \text{ 比特/画面} \end{aligned}$$

另外,有一篇千字文章,假定每字可从万字表中任选,则共有不同的千字文

$$N = 10000^{1000} = 10^{4000} \text{ 篇}$$

仍按等概计算,平均每篇千字文可提供的信息量为

$$H(X) = \log_2 N = 4 \times 10^3 \times 3.32 = 1.3 \times 10^4 \text{ 比特/千字文}$$

可见,“一个电视画面”平均提供的信息量要丰富得多,远远超过“一篇千字文”提供的信息量。

例 2.2.4 设信源符号集 $X = \{x_1, x_2, x_3\}$, 每个符号发生的概率分别为 $p(x_1) = 1/2$, $p(x_2) = 1/4$, $p(x_3) = 1/4$, 则信源熵为

$$H(X) = \frac{1}{2} \log_2 2 + \frac{1}{4} \log_2 4 + \frac{1}{4} \log_2 4 = 1.5 \text{ 比特/符号}$$

例 2.2.5 二元信源是离散信源的一个特例。该信源 X 输出符号只有两个,设为 0 和 1。输出符号发生的概率分别为 p 和 q , $p + q = 1$ 。即信源的概率空间为

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

根据(2.2.4)式可得二元信源熵为

$$H(X) = -p \log p - q \log q$$

$$= -p \log p - (1-p) \log (1-p) = H(p)$$

信源信息熵 $H(X)$ 是概率 p 的函数, 通常用 $H(p)$ 表示。 p 取值于 $[0, 1]$ 区间。 $H(p)$ 函数曲线如图 2 2 1 所示。从图中看出, 如果二元信源的输出符号是确定的, 即 $p=1$ 或 $q=1$, 则该信源不提供任何信息。反之, 当二元信源符号 0 和 1 以等概率发生时, 信源熵达到极大值, 等于 1 比特信息量。

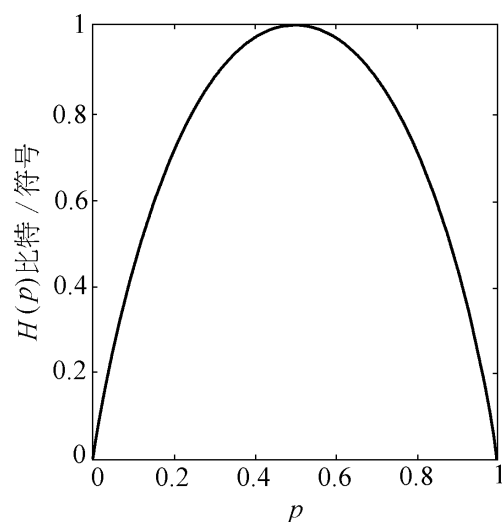


图 2 2 1 熵函数 $H(p)$

在给定 y_j 条件下, x_i 的条件自信息量为 $I(x_i/y_j)$, X 集合的条件熵 $H(X/y_j)$ 为

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

进一步在给定 Y (即各个 y_j) 条件下, X 集合的条件熵 $H(X/Y)$ 定义为

$$\begin{aligned} H(X/Y) &= \sum_j p(y_j) H(X/y_j) = \sum_{i,j} p(y_j) p(x_i/y_j) I(x_i/y_j) \\ &= \sum_{i,j} p(x_i y_j) I(x_i/y_j) \end{aligned} \quad (2 2 5)$$

即条件熵是在联合符号集合 XY 上的条件自信息量的联合概率加权统计平均值。条件熵 $H(X/Y)$ 表示已知 Y 后, X 的不确定度。

相应地, 在给定 X (即各个 x_i) 条件下, Y 集合的条件熵 $H(Y/X)$ 定义为

$$H(Y/X) = \sum_{i,j} p(x_i y_j) I(y_j/x_i) = - \sum_{i,j} p(x_i y_j) \log p(y_j/x_i) \quad (2 2 6)$$

联合熵是联合符号集合 XY 上的每个元素对 $x_i y_j$ 的自信息量的概率加权统计平均值, 定义为

$$H(XY) = \sum_{i,j} p(x_i y_j) I(x_i y_j) = - \sum_{i,j} p(x_i y_j) \log p(x_i y_j) \quad (2 2 7)$$

联合熵 $H(XY)$ 表示 X 和 Y 同时发生的不确定度。联合熵 $H(XY)$ 与熵 $H(X)$ 及条件熵 $H(Y/X)$ 之间存在下列关系:

$$\begin{aligned} H(XY) &= H(X) + H(Y/X) \\ H(XY) &= H(Y) + H(X/Y) \end{aligned} \quad (2 2 8)$$

2.2.3 互信息

最简单的通信系统模型, 如图 2 2 2 所示, X 是信源发出的离散符号集合, Y 是信宿收到的符号集合。由于信宿事先不知道信源在某一时刻发出的是哪一个符号, 所以每个符号消息是一个随机事件。信源发出符号通过有干扰的信道传递给信宿。通常信宿可以预先知道信源 X 发出的各个符号消息的集合, 以及它们的概率分布, 即预知信源 X 的先验概率 $p(x_i)$ 。当信宿收到一个符号消息 y_j 后, 信宿可以计算信源各消息的条件概率 $p(x_i/y_j)$, $i=1, 2, \dots, N$, 这种条件概率称为后验概率。互信息量定义为后验概率与先验概率比值的对数, 即

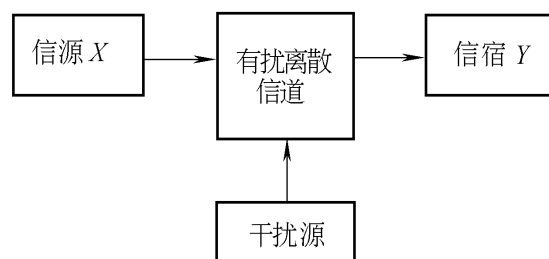


图 2 2 2 简单的通信系统模型

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

由于无法确定 $p(x_i/y_j)$ 和 $p(x_i)$ 的大小关系, 所以 $I(x_i; y_j)$ 不一定大于或等于零。互信息量 $I(x_i; y_j)$ 在 X 集合上的统计平均值

$$I(X; y_j) = \sum_i p(x_i/y_j) I(x_i; y_j) = \sum_i p(x_i/y_j) \log \frac{p(x_i/y_j)}{p(x_i)}$$

平均互信息量 $I(X; Y)$ 为上述 $I(X; y_j)$ 在 Y 集合上的概率加权统计平均值, 即

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

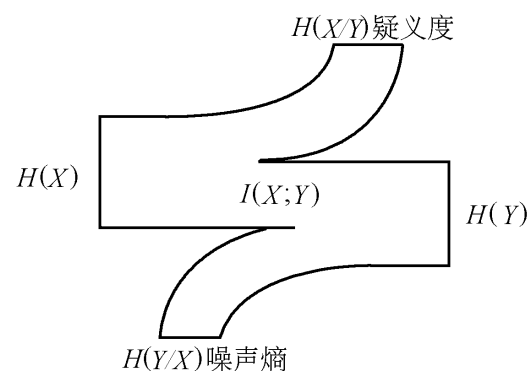
在通信系统中, 若发端的符号是 X , 而收端的符号是 Y , $I(X; Y)$ 就是在接收端收到 Y 后所能获得的关于 X 的信息。若干扰很大, Y 基本上与 X 无关, 或说 X 与 Y 相互独立, 那时就收不到任何关于 X 的信息; 反之, 若没有干扰, Y 是 X 的确知——对应函数, 那就能完全地收到 X 的信息 $H(X)$ 。

由上述定义可推导出下列性质:

$$I(X; Y) = H(X) - H(X/Y) \quad (2.2.9)$$

$$I(Y; X) = H(Y) - H(Y/X) = I(X; Y) \quad (2.2.10)$$

由(2.2.9)式和(2.2.10)式来说明平均互信息量的物理意义: $I(X; Y)$ 是 $H(X)$ 和 $H(X/Y)$ 之差。因为 $H(X)$ 是符号集合 X 的熵或不确定度, 而 $H(X/Y)$ 是当 Y 已知时 X 的不确定度, 那么可见“ Y 已知”这件事使 X 的不确定度减少了 $I(X; Y)$, 这意味着“ Y 已知后”所获得的关于 X 的信息是 $I(X; Y)$ 。这可看成信源符号集合 X , 信宿符号集合 Y , 平均互信息量 $I(X; Y)$ 表示在有扰离散信道上传输的平均信息量。信宿收到的平均信息量等于信宿对信源符号不确定度的平均减少量。具体地说, (2.2.9)式表明在有扰离散信道上, 各个接收符号 y 所提供的有关信源发出的各个符号 x 的平均信息量 $I(X; Y)$ 等于唯一地确定信源符号 x 所需要的平均信息量 $H(X)$, 减去收到符号 y 后要确定 x 所需要的平均信息量 $H(X/Y)$ 。条件熵 $H(X/Y)$ 可看作由于信道上存在干扰和噪声而损失掉的平均信息量。由于损失掉这一部分信息量, 故再要唯一地确定信源发出的符号 x 就显得信息量不足。条件熵 $H(X/Y)$ 又可以看作是信道上的干扰和噪声所造成的对信源符号 x 的平均不确定度, 故又称为疑义度。 $I(X; Y)$ 是有扰离散信道上能传输的平均信息量, 而 $H(X/Y)$ 是在 Y 条件下要唯一地确定信源发出符号所需要的平均信息量。(2.2.10)式表明平均互信息量可看作在有扰离散信道上传递消息时, 唯一地确定接收符号 y 所需要的平均信息量 $H(Y)$, 减去当信源发出符号为已知时需要确定接收符号 y 所需要的平均信息量 $H(Y/X)$ 。因此, 条件熵 $H(Y/X)$ 可看作唯一地确定信道噪声所需要的平均信息量, 故又称噪声熵或散布度。它们之间的关系可以用图 2.2.3 来形象地表达。



如果 X 与 Y 是相互独立的, 那么 Y 已知时 X 的条

图 2.2.3 收、发两端的熵关系

件概率等于 X 的无条件概率,由于熵就是这概率的对数的数学期望, X 的条件熵就等于 X 的无条件熵,此时 $I(X; Y) = 0$ 。这可理解为既然 X 与 Y 相互独立,无法从 Y 中去提取关于 X 的信息。这可看成信道上噪声相当大,以致有 $H(X/Y) = H(X)$ 。在这种情况下,能传输的平均信息量为零。这说明信宿收到符号 y 后不能提供有关信源发出符号 x 的任何信息量。对于这种信道,信源发出的信息量在信道上全部损失掉了,故称为全损离散信道。

如果 Y 是 X 的确定的一一对应函数,那么 Y 已知时 X 的条件概率非“1”即“0”,因为当 X 与 Y 有一一对应关系时,当 X 和 Y 满足该确定函数时,条件概率必为1,而不满足确定函数时条件概率必为零。也就是说, $I(X; Y) = H(X)$ 。可见此时已知 Y 就完全解除了关于 X 的不确定度,所获得的信息就是 X 的不确定度或熵。这可看成无扰离散信道,由于没有噪声,所以信道不损失信息量,疑义度 $H(X/Y)$ 为零,噪声熵也为零。于是有 $I(X; Y) = H(X) = H(Y)$ 。

在一般情况下, X 和 Y 既非相互独立,也不是一一对应,那么从 Y 获得 X 的信息必在零与 $H(X)$ 之间,即常小于 X 的熵。

符号 x_i 与符号对 $y_j z_k$ 之间的互信息量定义为

$$I(x_i; y_j z_k) = \log \frac{p(x_i / y_j z_k)}{p(x_i)} \quad (2.2.11)$$

条件互信息量是在给定 z_k 条件下, x_i 与 y_j 之间的互信息量,定义为

$$I(x_i; y_j / z_k) = \log \frac{p(x_i / y_j z_k)}{p(x_i / z_k)} \quad (2.2.12)$$

引用(2.2.12)式,(2.2.11)式可写成:

$$I(x_i; y_j z_k) = I(x_i; z_k) + I(x_i; y_j / z_k)$$

上述表明: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 $I(x_i; y_j / z_k)$ 。

三维联合集 XYZ 上的平均互信息量有

$$I(X; YZ) = I(X; Y) + I(X; Z/Y) \quad (2.2.13)$$

$$I(YZ; X) = I(Y; X) + I(Z; X/Y) \quad (2.2.14)$$

$$I(X; YZ) = I(X; ZY) = I(X; Z) + I(X; Y/Z) \quad (2.2.15)$$

2.2.4 数据处理中信息的变化

用信息论的观点研究数据处理过程中信息的变化。图 2.2.4 中 X 是输入消息集合, Y 是第一级处理器的输出消息集合, Z 为第二级处理器的输出消息集合,假设在 Y 条件下 X 与 Z 相互独立。

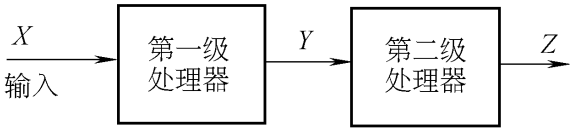


图 2.2.4 级联处理器示意图

由(2.2.13)式和(2.2.15)式得

$$I(X; Z) = I(X; Y) + I(X; Z/Y) - I(X; Y/Z)$$

将(2.2.14)式中的集合符号替代(X 代 Y , Y 代 Z , Z 代 X)得

$$I(XY; Z) = I(X; Z) + I(Y; Z/X) \quad (2.2.16)$$

将(2.2.16)式右边的 X 和 Y 互换得

$$I(XY; Z) = I(Y; Z) + I(X; Z|Y) \quad (2.2.17)$$

由(2.2.16)和(2.2.17)式得

$$I(X; Z) = I(Y; Z) + I(X; Z|Y) - I(Y; Z|X)$$

因为已假设在 Y 条件下 X 与 Z 相互独立, 即 $I(X; Z|Y) = 0$, 而且 $I(X; Y|Z)$ 和 $I(Y; Z|X)$ 均为非负量, 则得出

$$I(X; Z) \leq I(Y; Z), I(X; Z) \leq I(X; Y)$$

说明当消息通过多级处理器时, 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。这就是数据处理定理, 数据处理过程中只会失掉一些信息, 绝不会创造出新的信息, 所谓信息不增性。任何信息处理过程总会失掉信息, 最多保持原来的信息, 一旦失掉了信息, 用任何处理手段, 也不可能再恢复丢失的信息。

2.2.5 熵的性质

1. 非负性

$$H(X) = H(x_1, x_2, \dots, x_n) \geq 0$$

其中等号只有在 $x_i = 1$ 时成立。因为 $0 < p_i \leq 1$, 则 $\log p_i$ 一定是一个负数, 所以熵是非负的。

2. 对称性

熵函数所有变元可以互换, 而不影响函数值。即

$$H(x_1, x_2, \dots, x_n) = H(x_2, x_1, \dots, x_n) \quad (2.2.18)$$

因为熵函数只与随机变量的总体结构有关, 例如下列信源的熵都是相等的:

$$\begin{array}{ccc} X = & x_1 & x_2 & x_3 \\ P = & 1/3 & 1/2 & 1/6 \end{array} \quad \begin{array}{ccc} Y = & y_1 & y_2 & y_3 \\ P = & 1/3 & 1/6 & 1/2 \end{array} \quad \begin{array}{ccc} Z = & z_1 & z_2 & z_3 \\ P = & 1/2 & 1/3 & 1/6 \end{array}$$

3. 确定性

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

只要信源符号表中, 有一个符号的出现概率为 1, 信源熵就等于零。在概率空间中, 如果有两个基本事件, 其中一个是必然事件, 另一个则是不可能事件, 因此没有不肯定性, 熵必为零。当然可以类推到 n 个基本事件构成的概率空间。

4. 香农辅助定理

对于任意 n 维概率矢量 $P = (p_1, p_2, \dots, p_n)$ 和 $Q = (q_1, q_2, \dots, q_n)$, 如下不等式成立

$$H(p_1, p_2, \dots, p_n) \leq - \sum_{i=1}^n p_i \log p_i - \sum_{i=1}^n p_i \log q_i \quad (2.2.19)$$

该式表明, 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。等号仅当 $P = Q$ 时成立。

5. 最大熵定理

离散无记忆信源输出 M 个不同的信息符号, 当且仅当各个符号出现概率相等时 (即 $p_i = 1/M$), 熵最大。因为出现任何符号的可能性相等时, 不肯定性最大, 即

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

6. 条件熵小于无条件熵

条件熵小于信源熵： $H(Y/X) < H(Y)$ 。当且仅当 y 和 x 相互独立时， $p(y/x) = p(y)$ ，取等号。

两个条件下的条件熵小于一个条件下的条件熵： $H(Z/XY) < H(Z/Y)$ 。当且仅当 $p(z/xy) = p(z/y)$ 时取等号。

联合熵小于信源熵之和： $H(XY) < H(X) + H(Y)$ 。当且仅当两个集合相互独立时取等号，此时可得联合熵的最大值，即 $H(XY)_{\max} = H(X) + H(Y)$ 。

为了便于记忆，可看图 2 2 5 所示的 $I(X; Y)$, $H(XY)$, $H(Y/X)$, $H(X)$, $H(Y)$ 之间的关系。图中两圆外轮廓表示联合熵 $H(XY)$ ，圆 (1) 表示 $H(X)$ ，圆 (2) 表示 $H(Y)$ ，则

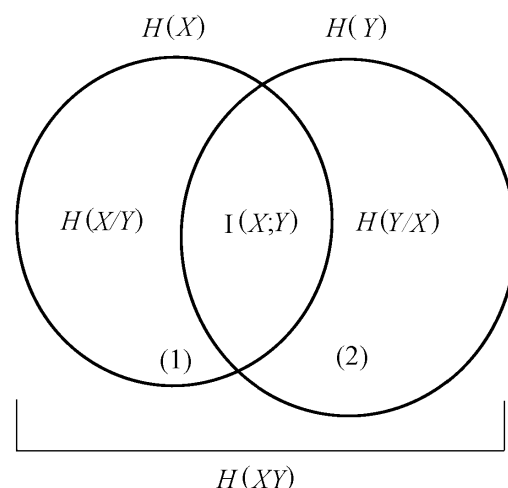


图 2 2 5 互信息量与熵之间的关系

$$H(XY) = H(X) + H(Y/X) = H(Y) + H(X/Y) \quad (2.2.20)$$

$$H(X) = H(X/Y), H(Y) = H(Y/X) \quad (2.2.21)$$

$$\begin{aligned} I(X; Y) &= H(X) - H(X/Y) = H(Y) - H(Y/X) \\ &= H(X) + H(Y) - H(XY) \end{aligned} \quad (2.2.22)$$

$$H(XY) = H(X) + H(Y) \quad (2.2.23)$$

如果 X 与 Y 互相独立，则

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

$$H(XY) = H(X) + H(Y) \quad (2.2.24)$$

$$H(X) = H(X/Y), H(Y) = H(Y/X) \quad (2.2.25)$$

例 2 2 6 二进制通信系统用符号“0”和“1”，由于存在失真，传输时会产生误码，用符号表示下列事件： u_0 ：一个“0”发出； u_1 ：一个“1”发出； v_0 ：一个“0”收到； v_1 ：一个“1”收到。

给定下列概率： $p(u_0) = 1/2$, $p(v_0/u_0) = 3/4$, $p(v_0/u_1) = 1/2$ ，求

- (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) 可求出

$$p(v_1/u_0) = 1 - p(v_0/u_0) = 1/4$$

$$\begin{aligned} H(V/u_0) &= -p(v_0/u_0)\log_2 p(v_0/u_0) - p(v_1/u_0)\log_2 p(v_1/u_0) \\ &= H\left(\frac{1}{4}, \frac{3}{4}\right) = 0.82 \text{ 比特/符号} \end{aligned}$$

(2) 联合概率 $p(u_0 v_0) = p(v_0/u_0)p(u_0) = 3/8$ ，同理可得

$$p(u_0 v_1) = 1/8, p(u_1 v_0) = 1/4, p(u_1 v_1) = 1/4$$

$$H(V/U) = - \sum_{i=0}^1 \sum_{j=0}^1 p(u_i v_j) \log_2 p(v_j/u_i)$$

$$= -\frac{3}{8} \log_2 \frac{3}{4} - \frac{1}{8} \log_2 \frac{1}{4} - 2 \times \frac{1}{4} \log_2 \frac{1}{2}$$

$$= 0.91 \text{ 比特/符号}$$

$$(3) \text{ 解法 1: } H(U, V) = - \sum_{i=1}^1 \sum_{j=0}^1 p(u_i, v_j) \log_2 p(u_i, v_j) = 1.91 \text{ 比特/符号}$$

解法 2: 因为 $p(u_0) = p(u_1) = 1/2$, 所以 $H(U) = 1$ 比特/符号,

$$H(U, V) = H(U) + H(V|U) = 1 + 0.91 = 1.91 \text{ 比特/符号}$$

$$(4) \text{ 可求出 } p(v_0) = \sum_{i=0}^1 p(u_i, v_0) = 5/8, p(v_1) = \sum_{i=0}^1 p(u_i, v_1) = 3/8,$$

$$\text{解法 1: } H(V) = H\left(\frac{3}{8}, \frac{5}{8}\right) = 0.96 \text{ 比特/符号}$$

$$H(U|V) = H(U, V) - H(V) = 1.91 - 0.96 = 0.95 \text{ 比特/符号}$$

解法 2: 利用贝叶斯公式得

$$p(u_0|v_0) = \frac{p(u_0)p(v_0|u_0)}{p(v_0)} = \frac{\frac{1}{2} \times \frac{3}{4}}{\frac{5}{8}} = \frac{3}{5}$$

$$\text{同理可得 } p(u_1|v_0) = 2/5, p(u_0|v_1) = 1/3, p(u_1|v_1) = 2/3$$

$$H(U|V) = - \sum_{i=0}^1 \sum_{j=0}^1 p(u_i, v_j) \log_2 p(u_i|v_j) = 0.95 \text{ 比特/符号}$$

2.3 连续信源的熵和互信息

2.3.1 连续信源熵

连续变量可以用离散变量来逼近, 即连续变量可以认为是离散变量的极限情况。从这个角度来看连续信源的信息量。

假设 $x \in [a, b]$, 令 $\Delta x = (b - a)/n$, $x_i \in [a + (i - 1)\Delta x, a + i\Delta x]$, $p_X(x)$ 为连续变量 X 的概率密度函数, 则利用中值定理 X 取 x_i 的概率是

$$p(x_i) = \int_{a+(i-1)\Delta x}^{a+i\Delta x} p_X(x) dx = p_X(x_i) \Delta x \quad (2.3.1)$$

根据离散信源熵的定义, 则

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

当 $n \rightarrow \infty$ 时, 即 $\Delta x \rightarrow 0$ 时, 由积分定义得

$$H(X) = \lim_{n \rightarrow \infty} H_n(X)$$

$$= - \int_a^b p_X(x) \log p_X(x) dx - \lim_{\Delta x \rightarrow 0} \log \Delta x \int_a^b p_X(x) dx \quad (2.3.2)$$

$$= - \int_a^b p_X(x) \log p_X(x) dx - \lim_{\Delta x \rightarrow 0} \log \Delta x$$

上式的第一项具有离散信源熵的形式, 是定值, 第二项为无穷大。将第二项丢掉后定义连续

信源熵为

$$H_c(X) = - \int_{-\infty}^{\infty} p_X(x) \log p_X(x) dx \quad (2.3.3)$$

连续信源熵与离散信源熵具有相同的形式,但其意义不同。连续信源的不确定度应为无穷大,这是因为连续信源可以假设是一个不可数的无限多个幅度值的信源,需要无限多个二进制位数(比特)来表示,因而它的熵为无穷大。但采用(2.3.3)式来定义连续信源的熵是因为在实际问题中,常遇到的是熵之间的差,如互信息量,只要两者逼近时所取的 x 一致,(2.3.2)式中第二项无穷大量是抵消的。

由此可见,连续信源的熵具有相对性,有时称为相对熵,在取两熵之间的差时才具有信息的所有特性,例如非负性等。

例 2.3.1 有一信源概率密度如图 2.3.1 所示,求连续熵。

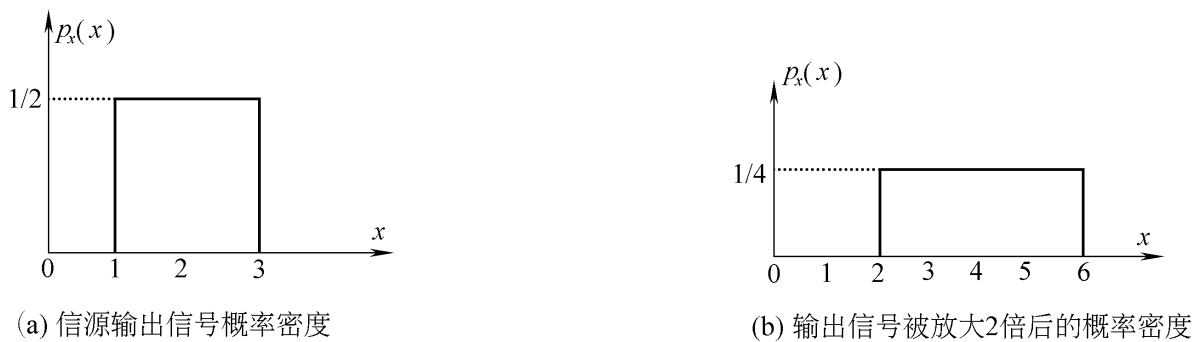


图 2.3.1 信源概率密度

由图(a)得

$$H_c(X) = - \int_1^3 p_X(x) \log_2 p_X(x) dx = - \int_1^3 \frac{1}{2} \log_2 \frac{1}{2} dx = 1 \text{ bit}$$

由图(b)得

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

图(b)是图(a)的放大,计算结果表明信息量增加了,这是荒谬的。因为这两种情况的绝对熵是不会变的,这是由无穷大项所造成的,两者在逼近时所取 x 不一致,使得图(b)比图(a)大了 1 比特。因此 $H_c(X)$ 给出的熵有相对意义,而不是绝对值。

用上述方法同样可定义两个变量 X, Y 的情况,

$$\text{联合熵: } H_c(XY) = - \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p_{XY}(x, y) \log p_{XY}(x, y) dx dy$$

$$\text{条件熵: } H_c(Y/X) = - \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p_X(x) p_{Y|X}(y/x) \log p_{Y|X}(y/x) dx dy$$

它们之间也有与离散信源一样的相互关系,并且可以得到有信息特征的互信息。

$$H_c(XY) = H_c(X) + H_c(Y/X) = H_c(Y) + H_c(X/Y)$$

$$\begin{aligned} I(X; Y) &= I(Y; X) = H_c(X) - H_c(X/Y) \\ &= H_c(X) + H_c(Y) - H_c(XY) \\ &= H_c(Y) - H_c(Y/X) \end{aligned}$$

2.3.2 最大熵定理

在离散信源情况下,已经得到等概率信源的熵为最大值。在连续信源中,当概率密度函数满足什么条件时才能使连续信源熵最大?

在具体应用中,我们只对连续信源的两种情况感兴趣,一是信源输出幅度受限,即限峰功率情况;二是信源输出平均功率受限。下面给出两个定理(证明从略),在此只说明它们的意义。

限峰功率最大熵定理:对于定义域为有限的随机矢量 X ,当它是均匀分布时,具有最大熵。

变量 X 的幅度取值限制在 $[a, b]$, 则有 $\int_a^b p_X(x) dx = 1$, 当任意 $p_X(x)$ 符合平均分布条件

$$p_X(x) = \begin{cases} \frac{1}{b-a}, & a \leq x \leq b \\ 0, & \text{其他} \end{cases}$$

时,信源达到最大熵。该结论与离散信源在以等概率出现时达到最大熵相类似。

限平均功率最大熵定理: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 的概率密度分布为 $p_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-m)^2}{2\sigma^2}}$, 其中 m 为数学期望, σ^2 为方差。则连续熵为

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

可以看到,当信源的概率密度符合正态分布时,其相对熵仅与随机变量的方差 σ^2 有关,而方差在物理含义上往往表示信号的交流功率,即 $P = \sigma^2$ 。在限制信号平均功率的条件下,正态分布的信源可输出最大相对熵 $H_c(X) = \frac{1}{2} \ln(2\pi\sigma^2 e)$, 其值随平均功率的增加而增加。

上述定理说明连续信源在不同限制条件下最大熵是不同的,在无限制条件时,最大熵不存在。

根据最大熵定理可知,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噪声熵。也就是说,高斯白噪声是最有害的干扰,在一定平均功率条件下造成最大数量的有害信息。在通信系统中,往往各种设计都将高斯白噪声作为标准,这不完全是为了简化分析,而是根据最坏的条件进行设计获得可靠系统。

2.4 离散序列信源的熵

前面讨论了单个消息(符号)的离散信源的熵,并较详细地讨论了它的性质。然而实际信源的输出往往是空间或时间的离散随机序列,有无记忆的离散信源序列,当然更多的是有记忆的,即序列中的符号之间有相关性。此时需要用联合概率分布函数或条件概率分布函数来描述信源发出的符号间的关系。下面讨论离散无记忆序列信源和两类较简单的离散有记忆序列信源(平稳序列和齐次遍历马氏链信源)。

2.4.1 离散无记忆信源的序列熵

设信源输出的随机序列为 X , $X = (X_1 X_2 \dots X_l \dots X_L)$, 序列中的变量 $X_l \in \{x_1, x_2, \dots, x_n\}$, $l = 1, 2, \dots, L$, 即序列长为 L 。随机序列的概率为

$$\begin{aligned} p(X = x) &= p(X_1 = x_{i1}, X_2 = x_{i2}, \dots, X_L = x_{iL}) \\ &= p(x_{i1}) p(x_{i2}/x_{i1}) p(x_{i3}/x_{i1} x_{i2}) \dots p(x_{iL}/x_{i1} x_{i2} \dots x_{iL-1}) \\ &= p(x_{i1}) p(x_{i2}/x_{i1}) p(x_{i3}/x_{i1}^2) \dots p(x_{iL}/x_{i1}^{L-1}) \end{aligned}$$

式中 $x_{i1}^{L-1} = x_{i1} x_{i2} \dots x_{iL-1}$ 。当信源无记忆时, $p(x) = p(x_{i1} x_{i2} \dots x_{iL}) = \prod_{l=1}^L p(x_{il})$ 。这时信源的序列熵可表示为

$$H(X) = - \sum_i p(x) \log p(x) = - \sum_{i=1}^L p(x_{il}) \log p(x_{il}) = \sum_{l=1}^L H(X_l)$$

若又满足平稳特性,即与序号 l 无关时,有 $p(x_{i1}) = p(x_{i2}) = \dots = p(x_{iL}) = p$, $p(x) = p^L$, 则信源的序列熵又可表示为 $H(X) = LH(X)$, 平均每个符号(消息)熵为

$$H_L(X) = \frac{1}{L} H(X) = H(X) \quad (2.4.1)$$

可见,离散无记忆信源平均每个符号的符号熵 $H_L(X)$ 就等于单个符号信源的符号熵 $H(X)$ 。例如有一个无记忆信源,随机变量 $X \in (0, 1)$, 等概率分布,以单个符号出现为一事件,则此时的信源熵 $H(X) = 1$ 比特/符号,即用 1 比特就可表示该事件。如果以两个符号出现($L=2$ 的序列)为一事件,则随机序列 $X \in (00, 01, 10, 11)$, 信源的序列熵 $H(X) = \log_2 4 = 2$ 比特/序列,即用 2 比特才能表示该事件。信源的符号熵 $H_2(X) = \frac{1}{2} H(X) = 1$ 比特/符号。

2.4.2 离散有记忆信源的序列熵

对于有记忆信源,就不像无记忆信源那样简单,它必须引入条件熵的概念,而且只能在某些特殊情况下才能得到一些有价值的结论。

对于由两个符号组成的联合信源,有下列结论:

$$(1) H(X_1 X_2) = H(X_1) + H(X_2/X_1) = H(X_2) + H(X_1/X_2)$$

$$(2) H(X_1) = H(X_1/X_2), H(X_2) = H(X_2/X_1)$$

第一式表明信源的联合熵(即前后两个符号 $X_1 X_2$ 同时发生的不确定度)等于信源发出前一个符号 X_1 的信息熵加上前一个符号 X_1 已知时信源发出下一个符号 X_2 的条件熵。

当前后符号无依存关系时,有下列推论:

$$H(X_1 X_2) = H(X_1) + H(X_2), \quad H(X_1 / X_2) = H(X_1), \quad H(X_2 / X_1) = H(X_2)$$

对于一般的有记忆信源如文字、数据等,它们输出的不是单个或两个符号,而是由有限个符号组成的序列,这些输出符号之间存在着相互依存的关系。可依照上述结论来分析序列的熵值。

若信源输出一个 L 长序列,则信源的序列熵为

$$\begin{aligned} H(X) &= H(X_1 X_2 \dots X_L) \\ &= H(X_1) + H(X_2 / X_1) + \dots + H(X_L / X_1 X_2 \dots X_{L-1}) \end{aligned} \tag{2.4.2}$$

记作
$$H(X) = H(X^L) = \sum_{l=1}^L H(X / X^{l-1})。$$

平均每个符号的熵为

$$H_L(X) = \frac{1}{L} H(X) \tag{2.4.3}$$

若当信源退化为无记忆时,有
$$H(X) = \sum_{l=1}^L H(X_l)。$$

若进一步又满足平稳性时,则有 $H(X) = LH(X)$ 。

这一结论与离散无记忆信源结论是完全一致的。可见,无记忆信源是上述有记忆信源的一个特例。

例 2.4.1 已知离散有记忆信源中各符号的概率空间为
$$\begin{matrix} X & a_0 & a_1 & a_2 \\ P & \frac{11}{36} & \frac{4}{9} & \frac{1}{4} \end{matrix}。$$
 现信源

发出二重符号序列消息 (a_i, a_j) , 这两个符号的概率关联性用条件概率 $p(a_j / a_i)$ 表示,并由下表给出。求信源的序列熵和平均符号熵。

a_i	a_j		
	a_0	a_1	a_2
a_0	9/11	2/11	0
a_1	1/8	3/4	1/8
a_2	0	2/9	7/9

解: 条件熵
$$H(X_2 / X_1) = - \sum_{i=0}^2 \sum_{j=0}^2 p(a_i a_j) \log_2 p(a_j / a_i) = 0.872 \text{ 比特/符号}$$

单符号信源熵
$$H_1(X) = H(X_1) = - \sum_{i=0}^2 p(a_i) \log_2 p(a_i) = 1.543 \text{ 比特/符号}$$

发二重符号序列的熵
$$\begin{aligned} H(X_1 X_2) &= H(X_1) + H(X_2 / X_1) \\ &= 1.543 + 0.872 = 2.415 \text{ 比特/序列} \end{aligned}$$

平均符号熵
$$H_2(X) = \frac{1}{2} H(X^2) = 1.21 \text{ 比特/符号}$$

比较上述结果可得: $H_2(X) < H_1(X)$, 即二重序列的符号熵值较单符号熵变小了, 也就是不确定度减小了, 这是由于符号之间存在关联性(相关性)造成的。

考虑离散平稳信源, 其联合概率具有时间推移不变性:

$$p\{X_{i_1} = x_1, X_{i_2} = x_2, \dots, X_{i_L} = x_L\} = p\{X_{i_1+h} = x_1, X_{i_2+h} = x_2, \dots, X_{i_L+h} = x_L\}$$

即平稳信源发出的符号序列的概率分布与时间起点无关。此时有下列结论：

结论 1 $H(X_L / X^{L-1})$ 是 L 的单调非增函数。

由于条件熵小于或等于无条件熵,条件较多的熵小于或等于减少一些条件的熵,考虑到平稳性,所以

$$\begin{aligned}
 H(X_L / X_1 X_2 \dots X_{L-1}) &= H(X_L / X_2 \dots X_{L-1}) \\
 &= H(X_{L-1} / X_1 \dots X_{L-2}) \quad (\text{平稳性}) \\
 &= H(X_{L-1} / X_2 \dots X_{L-2}) \\
 &= H(X_{L-2} / X_1 \dots X_{L-3}) \\
 &\dots \\
 &= H(X_2 / X_1)
 \end{aligned} \tag{2.4.4}$$

结论 2 $H_L(X) = H(X_L / X^{L-1})$, 因为

$$\begin{aligned}
 H_L(X) &= \frac{1}{L} H(X_1 X_2 \dots X_L) = \frac{1}{L} \sum_{l=1}^L H(X_l / X^{l-1}) \\
 &= \frac{1}{L} [H(X_1) + H(X_2 / X_1) + \dots + H(X_L / X_1 X_2 \dots X_{L-1})]
 \end{aligned}$$

由结论 1 得上式中的 $H(X_L / X_1 X_2 \dots X_{L-1})$ 是和式 L 项中最小的, 所以

$$H_L(X) \geq \frac{1}{L} \cdot L H(X_L / X_1 X_2 \dots X_{L-1}) = H(X_L / X^{L-1})$$

结论 3 $H_L(X)$ 是 L 的单调非增函数, 因为

$$\begin{aligned}
 L H_L(X) &= H(X_1 X_2 \dots X_L) \\
 &= H(X_1 X_2 \dots X_{L-1}) + H(X_L / X_1 X_2 \dots X_{L-1}) \\
 &= (L-1) H_{L-1}(X) + H(X_L / X^{L-1})
 \end{aligned}$$

运用结论 2 得：

$$H_L(X) \geq H_{L-1}(X) \tag{2.4.5}$$

该式说明随着 L 的增大, 增加的熵值 $H(X_L / X^{L-1})$ 越来越小(由结论 1 得), 导致平均符号熵随着 L 的增大而减小。即 $\dots H_{L-1}(X) \geq H_L(X) \geq H_{L+1}(X) \dots$

结论 4 当 $L \rightarrow \infty$ 时,

$$H(X) \geq \lim_{L \rightarrow \infty} H_L(X) = \lim_{L \rightarrow \infty} H(X_L / X_1 X_2 \dots X_{L-1}) \tag{2.4.6}$$

$H(X)$ 称为极限熵, 又称极限信息量。

现在证明(2.4.6)式, 根据上述结论 1 有

$$\begin{aligned}
 H_{L+k}(X) &= \frac{1}{L+k} [H(X_1 \dots X_{L-1}) + H(X_L / X_1 \dots X_{L-1}) \\
 &\quad + \dots + H(X_{L+k} / X_1 \dots X_{L+k-1})] \\
 &= \frac{1}{L+k} [H(X_1 \dots X_{L-1}) + H(X_L / X_1 \dots X_{L-1}) \\
 &\quad + H(X_L / X_1 \dots X_{L-1}) + \dots + H(X_L / X_1 \dots X_{L-1})] \\
 &= \frac{1}{L+k} H(X_1 \dots X_{L-1}) + \frac{k+1}{L+k} H(X_L / X_1 \dots X_{L-1})
 \end{aligned}$$

取足够大的 k ($k \rightarrow \infty$), 固定 L , 前一项可忽略, 后一项系数接近于 1, 得

$$\lim_{k \rightarrow \infty} H_{L+k}(X) = H(X_L / X_1 \dots X_{L-1}) \tag{2.4.7}$$

结论 2 和(2 4 7)式表明,条件熵 $H(X_L/X_1 \dots X_{L-1})$ 的值是在 $H_L(X)$ 和 $H_{L+k}(X)$ 之间,令 $L \rightarrow \infty$, $H_L(X)$ 应等于 $H_{L+k}(X)$ (假设极限存在),故得

$$\lim_{L \rightarrow \infty} H_L(X) = \lim_{L \rightarrow \infty} H(X_L/X_1 X_2 \dots X_{L-1})$$

推广结论 3 可得

$$H_0(X) = H_1(X) = H_2(X) = \dots = H(X) \quad (2 4 8)$$

式中, $H_0(X)$ 为等概率无记忆信源单个符号的熵, $H_1(X)$ 为一般无记忆(不等概率)信源单个符号的熵, $H_2(X)$ 为两个符号组成的序列平均符号熵,依此类推。

结论 4 从理论上定义了平稳离散有记忆信源的极限熵,实际上如按此公式计算极限熵是十分困难的。然而对于一般离散平稳信源,由于取 L 不很大时就能得出非常接近 $H(X)$ 值的 $H_L(X)$,因此在实际应用中常取有限 L 下的条件熵 $H(X_L/X^{L-1})$ 作为 $H(X)$ 的近似值。因为当平稳离散信源输出序列的相关性随着 L 的增加迅速减小时,其序列熵的增加量 $H(X_L/X^{L-1})$ 与相关性有关,相关性很弱,则 $H(X_L/X_1 X_2 \dots X_{L-1}) - H(X_L/X_2 \dots X_{L-1}) = H(X_L/X_1 \dots X_{L-2})$,增加量不再变小,所以平均符号熵也几乎不再减小。

当上述平稳信源满足 m 阶马尔可夫性质时,即信源发出的符号只与前面的 m 个符号有关,而与更前面出现的符号无关。用概率意义表达为

$$p(x_t/x_{t-1}, x_{t-2}, x_{t-3}, \dots, x_{t-m}, \dots) = p(x_t/x_{t-1}, x_{t-2}, \dots, x_{t-m})$$

则根据(2 4 6)式可得

$$\begin{aligned} H(X) &= \lim_{L \rightarrow \infty} H(X_L/X_1 X_2 \dots X_{L-1}) \\ &= H(X_{m+1}/X_1 X_2 \dots X_m) = H_{m+1}(X) \end{aligned} \quad (2 4 9)$$

上述公式在工程上很实用。

由于高阶马氏链过程需要引入矢量进行分析运算,处理较复杂。可将矢量转化为状态变量,通过分析系统状态在输入符号作用下的转移情况,使问题得到简化。对于 m 阶马尔可夫信源

$$\begin{aligned} X &= x_1 \ x_2 \ \dots \ x_q \\ P &= p(x_{m+1}/x_1 x_2 \dots x_m) \end{aligned}$$

在某一时刻($m+1$),信源符号出现的概率,仅与前面已出现的 m 个符号有关,而与更前面出现的符号无关。可通过引入符号条件概率和状态转移概率,从而转化为马尔可夫链,即令

$$s_i = (x_{i_1} x_{i_2} \dots x_{i_m}) \quad i_1, i_2, \dots, i_m = (1, 2, \dots, q) \quad (2 4 10)$$

如果信源符号表中的数目为 q ,则由前面出现的 m 个符号所组成的序列 s_i 共有 $Q = q^m$ 种,将这些序列看作是状态集 $S = \{s_1, s_2, \dots, s_Q\}$,则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,用条件概率表示为 $p(x_j/s_i)$, $i = 1, 2, \dots, Q$; $j = 1, 2, \dots, q$ 。当信源符号 x_j 出现后,信源所处的状态将发生变化,并转入一个新的状态。这种状态的转移可用状态转移概率表示

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

状态转移概率 $p(s_j/s_i)$ 与信源符号条件概率 $p(x_j/s_i)$ 有关。

状态转移概率 $p_{ij}(m, n)$ 表示已知在时刻 m 系统处于状态 s_i , 或 S_m 取值 s_i 的条件下,经 $(n-m)$ 步后转移到状态 s_j 的概率。也可以把 $p_{ij}(m, n)$ 理解为已知在时刻 m 系统处于状态 i 的条件下,在时刻 n 系统处于状态 j 的条件概率,故状态转移概率实际上是一个条件概率。转移概率具有下列性质:

$$p_{ij}(m, n) = 0 \quad i, j \in S$$

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

我们特别关心 $n - m = 1$ 的情况, 即 $p_{ij}(m, m + 1)$ 。把 $p_{ij}(m, m + 1)$ 记为 $p_{ij}(m)$, $m = 0$, 并称为基本转移概率, 也可称为一步转移概率。

$$p_{ij}(m) = p\{S_{m+1} = j | S_m = i\} \quad i, j \in S$$

对于齐次马尔可夫链, 其转移概率具有推移不变性, 即只与状态有关, 与时刻 m 无关, 故转移概率可表示为:

$$p_{ij}(m) = p\{S_{m+1} = j | S_m = i\} = p_{ij} \quad i, j \in S$$

显然 p_{ij} 具有下列性质:

$$p_{ij} \geq 0 \quad i, j \in S$$

$$\sum_j p_{ij} = 1 \quad i \in S$$

类似地, 可以定义 k 步转移概率为

$$p_{ij}^{(k)}(m) = p\{S_{m+k} = j | S_m = i\} = p_{ij}^{(k)} \quad i, j \in S$$

需要指出的是, 平稳信源的概率分布特性具有时间推移不变性, 而齐次马氏链只要求转移概率具有推移不变性, 因此一般情况下平稳包含齐次, 但齐次不包含平稳。

由于系统在任一时刻可处于状态空间 $S = \{s_1, s_2, \dots, s_Q\}$ 中的任意一个状态, 因此状态转移时, 转移概率是一个矩阵

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

由一步转移概率 p_{ij} 可以写出其转移矩阵为

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

或

$$P = (p(s_j | s_i)) = \begin{pmatrix} p_{11} & \dots & p_{1Q} \\ \dots & \dots & \dots \\ p_{Q1} & \dots & p_{QQ} \end{pmatrix}$$

矩阵 P 中第 i 行元素对应于从某一个状态 s_i 转移到所有状态 s_j 的转移概率, 显然矩阵中的每一个元素都是非负的, 并且每行之和均为 1; 第 j 列元素对应于从所有状态 s_i 转移到同一个状态 s_j 的转移概率, 列元素之和不一定为 1。

$$p_{11} \quad \dots \quad p_{1Q}$$

同样也可将符号条件概率写成矩阵: $(p(a_j | s_i)) = \begin{pmatrix} \dots & \dots \\ \dots & \dots \end{pmatrix}$, 这两个矩阵, 一般

$$p_{Q1} \quad \dots \quad p_{Qq}$$

情况下是不同的, 应注意区分。

k 步转移概率 $p_{ij}^{(k)}$ 与 l ($l < k$) 步和 $(k - l)$ 步转移概率之间有所谓的切普曼-柯尔莫戈洛夫方程。即

$$p_{ij}^{(k)} = \sum_r p_{ir}^{(l)} p_{rj}^{(k-l)}$$

上式右侧是对第 l 步的所有可能取值求和, 因而也就是 k 步转移概率。特别地, 当 $l = 1$ 时,

$$p_{ij}^{(k)} = \sum_r p_{ir} p_{rj}^{(k-1)} = \sum_r p_{ir}^{k-1} p_{rj}$$

若用矩阵表示, 则

$$(P^{(k)}) = (P)(P^{(k-1)}) = (P)(P)(P^{(k-2)}) = \dots = (P)^k$$

从这一递推关系式可知,对于齐次马氏链来说,一步转移概率完全决定了 k 步转移概率。为了确定无条件概率 $p(S_k = s_j)$ 还需引入初始概率,令

$$p_{0i} = p(S_0 = s_i)$$

这样

$$\begin{aligned} p(S_k = s_j) &= \sum_i p(S_k = s_j, S_0 = s_i) \\ &= \sum_i p(S_0 = s_i) p(S_k = s_j | S_0 = s_i) \\ &= \sum_i p_{0i} p_{ij}^{(k)} \end{aligned}$$

需要研究 $\lim_k p_{ij}^{(k)}$ 的问题,倘若这极限存在,且等于一个与起始状态 i 无关的被称为平稳分布的 $W_j = p(S_k = s_j)$,则不论起始状态是什么,此马氏链可以最后达到稳定,即所有变量 X_k 的概率分布均不变。在这种情况下,就可以用 (P) 这一矩阵来充分描述稳定的马氏链,起始状态只使前面有限个变量的分布改变,如同电路中的暂态一样。

在求这种信源的熵时,需要先求出稳定分布的概率

$$\lim_k p_{ij}^{(k)} = W_j \tag{2 4 11}$$

但有时是很困难的,事实上只要知道它有极限,平稳分布 W_j 可用下列方程组来求得:

$$\sum_i W_i p_{ij} = W_j \tag{2 4 12}$$

上式中 W_i 和 W_j 均为稳态分布概率。由于 $\sum_i p_{ij} = 1$,所以行列式 $|p_{ij} - \delta_{ij}| = 0$,可见 (2 4 12) 式必有非零解。再用 $\sum_j W_j = 1$ 就可解得各稳态分布概率 W_j 。若 $[p_{ij} - \delta_{ij}]$ 的秩是 $(n - 1)$,则解是唯一的。(2 4 12) 式有唯一解是 $\lim_k p_{ij}^{(k)}$ 存在的必要条件,并不是充分条件。为了使马氏链最后达到稳定,成为遍历的马氏链,还必须要有不可约性和非周期性。

所谓不可约性,就是对任意一对 i 和 j ,都存在至少一个 k ,使 $p_{ij}^{(k)} > 0$,这就是说从 s_i 开始,总有可能到达 s_j ;反之若对所有 k , $p_{ij}^{(k)} = 0$,就意味着一旦出现 s_i 以后不可能到达 s_j ,也就是不能各态遍历,或者状态中应把 s_j 取消,这样就成为可约的了。例如图 2 4 1 中所表示的马氏链,其中 s_1, s_2, s_3 是三种状态,箭头是指从一个状态转移到另一个状态,旁边的数字代表转移概率。这就是香农提出的马尔可夫状态图,也叫香农线图。容易看出由状态 s_3 转移到 s_1 的转移概率 $p_{31}^{(k)} = 0$,因为一进入状态 s_3 就一直继续下去,而不会再转移到其他状态。 $p_{41}^{(k)} = 0$ 也是明显的,因 s_4 和 s_1 之间没有连接箭头,因此这种链就不是不可约的。

所谓非周期性,就是所有 $p_{ii}^{(n)} > 0$ 的 n 中没有比 1 大的公因子。图 2 4 2 中的转移矩阵就是有周期为 2,因为从 s_1 出发再回到 s_1 所需的步数必为 2, 4, 6, ..., 这里的 $p_{ij}^{(n)}$ 矩阵

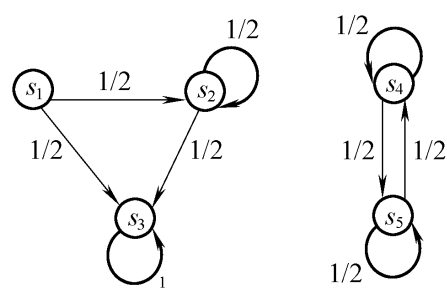


图 2 4 1 非不可约马氏链

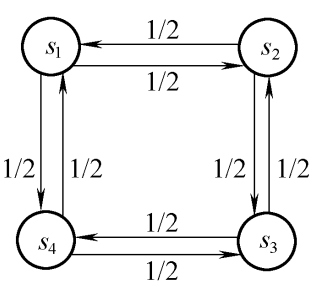


图 2 4 2 周期性马氏链

$$(P^{(k)}) = (P)^k = \begin{pmatrix} 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{pmatrix}^k$$

可以验证,当 k 为奇数时

$$(P^{(k)}) = (P)^k = \begin{pmatrix} 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{pmatrix} = (P)$$

当 k 为偶数时

$$(P^{(k)}) = (P)^k = \begin{pmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix} = (P)$$

若起始状态为 s_1 , 则经奇数步后, $S_k = s_j$ 的概率为

$$p_j = \begin{cases} 0 & j=1 \\ \frac{1}{2} & j=2 \\ 0 & j=3 \\ \frac{1}{2} & j=4 \end{cases}$$

而经偶数步后

$$p_j = \begin{cases} \frac{1}{2} & j=1 \\ 0 & j=2 \\ \frac{1}{2} & j=3 \\ 0 & j=4 \end{cases}$$

这样就达不到稳定状态,虽然方程组(2 4 12)式是有解的,其解为 $p_j = 1/4 (j=1, 2, 3, 4)$ 。

例 2 4 2 如图 2 4 3(a)所示是一个相对码编码器。输入的码 $X_r (r=1, 2, \dots)$ 是相互独立的,取值 0 或 1,且已知 $p(X=0) = p, p(X=1) = 1 - p = q$,输出的码是 Y_r ,显然有

$$Y_1 = X_1, \quad Y_2 = X_2 \quad Y_1, \dots$$

其中 表示模 2 加,那么 Y_r 就是一个一阶马氏链,因 Y_r 确定后, Y_{r+1} 的概率分布只与 Y_r 有关,与 Y_{r-1} 、 Y_{r-2} ... 等无关,且知 Y_r 序列的状态转移概率为

$$p_{00} = p(Y_2 = 0 | Y_1 = 0) = p(X = 0) = p$$

$$p_{01} = p(Y_2 = 1 | Y_1 = 0) = p(X = 1) = q$$

$$p_{10} = p(Y_2 = 0 | Y_1 = 1) = p(X = 1) = q$$

$$p_{11} = p(Y_2 = 1 | Y_1 = 1) = p(X = 0) = p$$

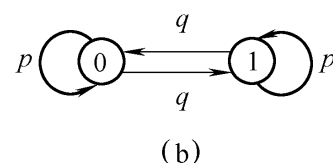
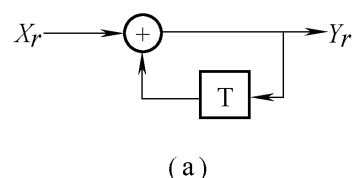


图 2 4 3 相对码编码器及其状态

即转移矩阵为 $\begin{pmatrix} p & q \\ q & p \end{pmatrix}$, 它与 r 无关,因而是齐次的。它的状态图如图 2 4 3(b)所示。

由图容易验证该马氏链具有不可约性和非周期性,由方程组(2 4 12)式可求得平稳概率分布 $p_0 = \frac{1}{2}$, $p_1 = \frac{1}{2}$, 所以这一马氏链是遍历的。

遍历性的直观意义是,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 $p_{ij}^{(k)}$ 都近似等于某个常数 W_j 。反过来说,如果转移步数 k 充分大,就可以用常数 W_j 作为 k 步转移概率 $p_{ij}^{(k)}$ 的近似值。这意味着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。经过足够长时间之后,信源处于什么状态已与初始状态无关。这时每种状态出现的概率已达到一种稳定分布。就像电路中经过暂态后进入稳态一样。

例 2 4 3 有一个二阶马氏链 X (0,1), 其符号条件概率为表 1, 状态变量 S (00, 01, 10, 11), 则可得状态转移概率状态转移概率如比如在状态 01 时, 出现

表 1 符号条件概率 $p(a/s_i)$

起 始 状 态	符 号	
	0	1
00	1/2	1/2
01	1/3	2/3
10	1/4	3/4
11	1/5	4/5

表 2 状态转移概率 $p(s_j/s_i)$

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

符号 0, 则将 0 加到状态 01 的后面, 再将第一位符号 0 挤出, 转移到状态 10, 概率为 1/3。其他状态的变化过程类似。状态转移概率如表 2, 相应的状态图如图 2 4 4 所示。令各状态的平稳分布概率为 W_1, W_2, W_3, W_4 , 利用(2 4 12)式可得方程

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

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

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

解得平稳分布的概率为

$$W_1 = \frac{3}{35}, \quad W_2 = \frac{6}{35}, \quad W_3 = \frac{6}{35}, \quad W_4 = \frac{4}{7}$$

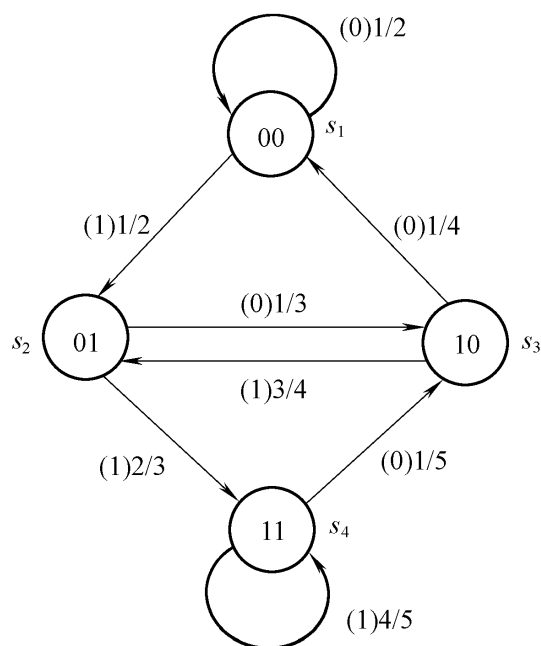


图 2 4 4 二阶马尔可夫信源状态转移图

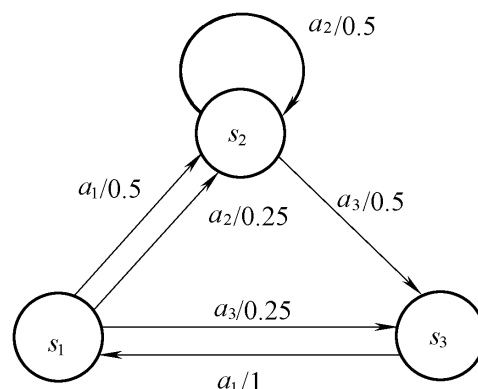


图 2 4 5 三状态马尔可夫信源状态转移图

从上面的例子可见,状态转移概率矩阵与符号条件概率矩阵是不同的,再看如图 2 4 5 所示的情况,该信源的状态转移概率矩阵为

$$P = [p(s_j / s_i)] = \begin{bmatrix} 0 & 3/4 & 1/4 \\ 0 & 1/2 & 1/2 \\ 1 & 0 & 0 \end{bmatrix}$$

符号条件概率矩阵为

$$[p(a_j / s_i)] = \begin{bmatrix} 1/2 & 1/4 & 1/4 \\ 0 & 1/2 & 1/2 \\ 1 & 0 & 0 \end{bmatrix}$$

显然两者也是不一样的。稳定后的状态概率分布为

$$p(s_1) = \frac{2}{7}, \quad p(s_2) = \frac{3}{7}, \quad p(s_3) = \frac{2}{7}$$

符号概率分布为

$$p(a_1) = \frac{3}{7}, \quad p(a_2) = \frac{2}{7}, \quad p(a_3) = \frac{2}{7}$$

下面计算遍历的 m 阶马尔可夫信源所能提供的平均信息量,即信源的极限熵 H 。根据(2 4 9)式 $H = H_{m+1}$,即只需求出 H_{m+1} 。

对于齐次、遍历的马尔可夫链,其状态 s_i 由 $(x_{i_1}, \dots, x_{i_m})$ 唯一确定,因此有

$$p(x_{i_{m+1}} / x_{i_m}, \dots, x_{i_1}) = p(x_{i_{m+1}} / s_i) \quad (2.4.13)$$

上式两边同取对数,并对 $x_{i_1}, \dots, x_{i_m}, x_{i_{m+1}}$ 和 s_i 取统计平均,然后取负,可以得到

$$\begin{aligned} \text{左边} &= - \sum_{i_{m+1}, \dots, i_1; i} p(x_{i_{m+1}}, \dots, x_{i_1}; s_i) \log p(x_{i_{m+1}} / x_{i_m}, \dots, x_{i_1}) \\ &= - \sum_{i_{m+1}, \dots, i_1; i} p(x_{i_{m+1}}, \dots, x_{i_1}) \log p(x_{i_{m+1}} / x_{i_m}, \dots, x_{i_1}) \end{aligned}$$

$$= H(x_{i_{m+1}}/x_{i_m}, \dots, x_{i_1})$$

$$= H_{m+1}$$

$$\begin{aligned} \text{右边} &= - \sum_{i_{m+1}, \dots, i_1; i} p(x_{i_{m+1}}, \dots, x_{i_1}; s_i) \log p(x_{i_{m+1}}/s_i) \\ &= - \sum_{i_{m+1}, \dots, i_1; i} p(x_{i_m}, \dots, x_{i_1}; s_i) p(x_{i_{m+1}}/s_i) \log p(x_{i_{m+1}}/s_i) \\ &= - \sum_{i_{m+1}} p(s_i) p(x_{i_{m+1}}/s_i) \log p(x_{i_{m+1}}/s_i) \\ &= \sum_i p(s_i) H(X/s_i) \end{aligned}$$

即

$$H_{m+1} = \sum_i p(s_i) H(X/s_i)$$

式中, $p(s_i)$ 是马尔可夫链的平稳分布, 它可以由 (2 4 12) 式计算得到; 熵函数 $H(X/s_i)$ 表示信源处于某一状态 s_i 时发出一个消息符号的平均不确定性

$$H(X/s_i) = - \sum_j p(x_j/s_i) \log p(x_j/s_i) \quad (2 4 15)$$

对状态 s_i 的全部可能性作统计平均, 就可得到马尔可夫信源的平均符号熵 H_{m+1} 。

例 2 4 4 如图 2 4 6 所示的三状态马尔可夫信源, 其转移概率矩阵为

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

设平稳分布的概率矢量为 $W = (W_1, W_2, W_3)$, 则

$$WP = W$$

$$\sum_{i=1}^3 W_i = 1 \quad W_i \geq 0$$

解得 $W_1 = 5/59, W_2 = 9/59, W_3 = 45/59$ 。

在 s_i 状态下每输出一个符号的平均信息量为

$$\begin{aligned} H(X/s_1) &= 0.1 \times \log_2 \frac{1}{0.1} + 0.9 \times \log_2 \frac{1}{0.9} \\ &= H(0.1) = 0.468996 \text{ 比特/符号} \\ H(X/s_2) &= H(0.5) = 1 \text{ 比特/符号} \\ H(X/s_3) &= H(0.2) = 0.721928 \text{ 比特/符号} \end{aligned}$$

对三个状态取统计平均后得到信源每输出一个符号的信息量, 即马尔可夫信源的熵:

$$H = \sum_{i=1}^3 W_i H(X/s_i) = 0.742910 \text{ 比特/符号}$$

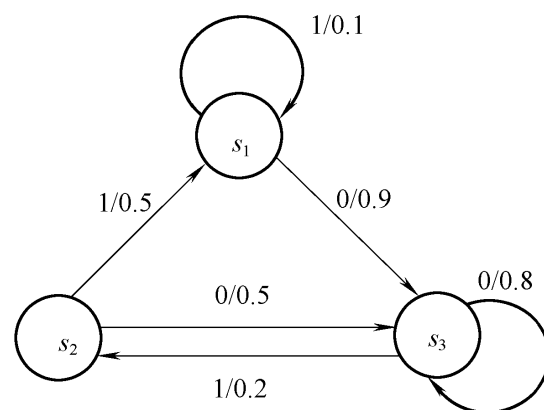


图 2 4 6 三状态马尔可夫信源状态转移图

2 5 冗 余 度

冗余度也称多余度或剩余度。顾名思义, 它表示给定信源在实际发出消息时所包含的多余信息。如果一个消息所包含的符号比表达这个消息所需要的符号多的话, 那么这样的

消息就存在多余度。

冗余度来自两个方面,一是信源符号间的相关性,从(2 4 8)式看出由于信源输出符号间的依赖关系使得信源熵减小,这就是信源的相关性。相关程度越大,信源的实际熵越小,趋于极限熵 $H(X)$;反之相关程度减小,信源实际熵就增大。

另一个方面是信源符号分布的不均匀性,当等概率分布时信源熵最大。而实际应用中大多是不均匀分布,使得实际熵减小。当信源输出符号间彼此不存在依赖关系且为等概率分布时,信源实际熵趋于最大熵 $H_0(X)$ 。

对于一般平稳信源来说,极限熵为 $H(X)$,这就是说我们需要传送这一信源的信息,理论上只需要有传送 $H(X)$ 的手段即可。但实际上我们对它的概率分布未能完全掌握,只能算出 $H_m(X)$,若用能传送 $H_m(X)$ 的手段去传送具有 $H(X)$ 的信源,当然很不经济。定义 为信息效率。

$$\eta = \frac{H(X)}{H_m(X)} \tag{2 5 1}$$

表示不肯定性的程度,由定义可知 $0 \leq \eta \leq 1$ 。(1 - η)表示肯定性的程度,因为肯定性不含有信息量,所以是冗余的。定义冗余度

$$\rho = 1 - \eta = 1 - \frac{H(X)}{H_m(X)} \tag{2 5 2}$$

事实上,当只知道信源符号有 q 种可能取值,而对其概率特性一无所知时,合理的假设是: q 种取值是等可能的,因为此时熵取最大值 $\log q$ 。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 ,显然 $H_0 - H_1 \geq 0$ 是测定一维分布后获得的信息。测 m 维分布后获得的信息就是 $H_0 - H_m$ 。若所有维分布都能测定,就可得到 $H_0 - H$ 。所以压缩传送信息的手段,有赖于已预先从测量中获得的信息,这一部分就无需传送了。

以英文字母的符号为例来计算这些值。英文字母共有 26 个,加上空格共 27 个符号,则最大熵为

$$H_0(X) = \log_2 27 = 4.76 \text{ 比特/符号}$$

对在英文书中各符号出现的概率加以统计,得到表 2 5 1 数值。

表 2 5 1 英文字母出现的概率

符 号	概率 p_i	符 号	概率 p_i	符 号	概率 p_i	符 号	概率 p_i
空 格	0.2	I	0.055	C	0.023	B	0.0105
E	0.105	R	0.054	F, U	0.0225	V	0.008
T	0.072	S	0.052	M	0.021	K	0.003
O	0.0654	H	0.047	P	0.0175	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

如果认为英语字母间是离散无记忆的,则根据表中的概率可求得

$$H_1(X) = - \sum_i p_i \log_2 p_i = 4.03 \text{ 比特/符号}$$

若考虑前后二个、三个、.....若干个字母之间存在相关性,则可根据字母出现的条件概率(统计特性在此未列出)求得

$$H_2(X) = 3.32 \text{ 比特/符号}$$

$$H_3(X) = 3.1 \text{ 比特/符号}$$

.....

$$H(X) = 1.4 \text{ 比特/符号}$$

若用一般传送方式,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为

$$\begin{aligned} &= \frac{1.4}{4.76} = 0.29 \\ &= 1 - 0.29 = 0.71 \end{aligned}$$

由上述例子可看出, $H_1 < H_0$, 是由于各个符号出现的概率不均匀; $H < \dots < H_3 < H_2$, 随着序列增长, 字母间的相关性越来越强。所以正是因为信源符号中存在的这些统计不均匀性和相关性, 才使得信源存在冗余度。当英文字母的结构信息已预先充分获得时, 可用合理的符号来表达英语, 例如传送或存储这些符号, 可大量压缩, 100 页的英语, 大约只要 29 页就可以了。在实际通信系统中, 为了提高传输效率, 往往需要把信源的大量冗余进行压缩, 即所谓信源编码。但是考虑通信中的抗干扰问题, 则需要信源具有一定的冗余度。因此在传输之前通常加入某些特殊的冗余度, 即所谓信道编码, 以达到通信系统理想的传输有效性和可靠性。

习 题

2.1 同时掷两个正常的骰子, 也就是各面呈现的概率都是 $1/6$, 求:

- (1) “3 和 5 同时出现”这事件的自信息量。
- (2) “两个 1 同时出现”这事件的自信息量。
- (3) 两个点数的各种组合(无序对)的熵或平均信息量。
- (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。

2.2 设在一只布袋中装有 100 只对人手的感觉得完全相同的木球, 每只球上涂有一种颜色。100 只球的颜色有下列三种情况:

- (1) 红色球和白色球各 50 只;
- (2) 红色球 99 只, 白色球 1 只;
- (3) 红、黄、蓝、白色各 25 只。

求从布袋中随意取出一只球时, 猜测其颜色所需要的信息量。

2.3 居住某地区的女孩中有 25% 是大学生, 在女大学生中有 75% 是身高 1 米 6 以上的, 而女孩中身高 1 米 6 以上的占总数一半。假如我们得知“身高 1 米 6 以上的某女孩是大学生”的消息, 问获得多少信息量?

2.4 一个消息由符号 0, 1, 2, 3 组成, 已知 $p(0) = 3/8$, $p(1) = 1/4$, $p(2) = 1/4$, $p(3) = 1/8$ 。试求由 60 个符号构成的消息所含的信息量和平均信息量。

2.5 掷两粒骰子, 当其向上的面的小圆点数之和是 3 时, 该消息所包含的信息量是多少? 当小圆点数之和是 7 时, 该消息所包含的信息量又是多少?

2 6 设有一离散无记忆信源,其概率空间为

$$\begin{matrix} X & x_1 = 0 & x_2 = 1 & x_3 = 2 & x_4 = 3 \\ P & 3/8 & 1/4 & 1/4 & 1/8 \end{matrix}$$

该信源发出的消息符号序列为(202 120 130 213 001 203 210 110 321 010 021 032 011 223 210), 求:

- (1) 此消息的自信息量是多少 ?
- (2) 在此消息中平均每个符号携带的信息量是多少 ?

2 7 试问四进制、八进制脉冲所含的信息量是二进制脉冲的多少倍 ?

2 8 国际摩尔斯电码用点和划的序列发送英文字母,划用持续 3 个单位的电流脉冲表示,点用持续 1 个单位的电流脉冲表示。其划出现的概率是点出现概率的 1/ 3。

- (1) 计算点和划的信息量;
- (2) 计算点和划的平均信息量。

2 9 在一个袋中放有 5 个黑球、10 个白球,以摸一个球为一次实验,摸出的球不再放进去。求:

- (1) 一次实验包含的不确定度。
- (2) 第一次实验 X 摸出的是黑球,第二次实验 Y 给出的不确定度;
- (3) 第一次实验 X 摸出的是白球,第二次实验 Y 给出的不确定度;
- (4) 第二次实验 Y 包含的不确定度。

2 10 有一个可旋转的圆盘,盘面上被均匀地分成 38 份,用 1,2,...,38 数字标示,其中有 2 份涂绿色,18 份涂红色,18 份涂黑色,圆盘停转后,盘面上指针指向某一数字和颜色。

- (1) 若仅对颜色感兴趣,计算平均不确定度;
- (2) 若仅对颜色和数字都感兴趣,计算平均不确定度;
- (3) 如果颜色已知时,计算条件熵。

2 11 两个试验 X 和 Y , $X = \{ x_1, x_2, x_3 \}$, $Y = \{ y_1, y_2, y_3 \}$,联合概率 $p(x_i y_j) = p_{ij}$ 已给出,

- (1) 如果有人告诉你 X 和 Y 的试验结果,你得到的平均信息量是多少 ?
- (2) 如果有人告诉你 Y 的试验结果,你得到的平均信息量是多少 ?
- (3) 在已知 Y 试验结果的情况下,告诉你 X 的试验结果,你得到的平均信息量是多少 ?

$$\begin{matrix} p_{11} & p_{12} & p_{13} & 7/24 & 1/24 & 0 \\ p_{21} & p_{22} & p_{23} & 1/24 & 1/4 & 1/24 \\ p_{31} & p_{32} & p_{33} & 0 & 1/24 & 7/24 \end{matrix}$$

2 12 有两个二元随机变量 X 和 Y ,它们的联合概率如右表所示,并定义另一随机变量 $Z = XY$ (一般乘积)。试计算:

$\begin{matrix} X \\ Y \end{matrix}$	X	
	0	1
0	1/8	3/8
1	3/8	1/8

- (1) $H (X)$, $H (Y)$, $H (Z)$, $H (XZ)$, $H (YZ)$ 和 $H (XYZ)$ 。
- (2) $H (X/ Y)$, $H (Y/ X)$, $H (X/ Z)$, $H (Z/ X)$, $H (Y/ Z)$, $H (Z/ Y)$, $H (X/ YZ)$, $H (Y/ XZ)$ 和 $H (Z/ XY)$ 。

(3) $I(X; Y)$, $I(X; Z)$, $I(Y; Z)$, $I(X; Y/Z)$, $I(Y; Z/X)$ 和 $I(X; Z/Y)$ 。

- 2 13 一个信源发出二重符号序列消息 (i, j) , 其中第一个符号 i 可以是 A, B, C 中的任一个, 第二个符号 j 可以是 D, E, F, G 中的任一个。已知各个 $P(i)$ 和 $P(j|i)$ 值列成如下表。求这个信源的熵(联合熵 $H(IJ)$)。

$P(i)$		A	B	C
		1/2	1/3	1/6
$P(j i)$	D	1/4	3/10	1/6
	E	1/4	1/5	1/2
	F	1/4	1/5	1/6
	G	1/4	3/10	1/6

- 2 14 在一个二进制信道中, 信源消息集 $X = \{0, 1\}$, 且 $p(1) = p(0)$, 信宿的消息集 $Y = \{0, 1\}$, 信道传输概率 $p(1/0) = 1/4$, $p(0/1) = 1/8$ 。求:

- (1) 在接收端收到 $y=0$ 后, 所提供的关于传输消息 x 的平均条件互信息量 $I(X; y=0)$ 。
- (2) 该情况所能提供的平均互信息量 $I(X; Y)$ 。

- 2 15 已知信源发出 a_1 和 a_2 两种消息, 且 $p(a_1) = p(a_2) = 1/2$ 。此消息在二进制对称信道上传输, 信道传输特性为 $p(b_1/a_1) = p(b_2/a_2) = 1 - \epsilon$, $p(b_1/a_2) = p(b_2/a_1) = \epsilon$ 。求互信息量 $I(a_1; b_1)$ 和 $I(a_1; b_2)$ 。

- 2 16 黑白传真机的消息元只有黑色和白色两种, 即 $X = \{\text{黑}, \text{白}\}$, 一般气象图上, 黑色的出现概率 $p(\text{黑}) = 0.3$, 白色的出现概率 $p(\text{白}) = 0.7$ 。

- (1) 假设黑白消息视为前后无关, 求信源熵 $H(X)$, 并画出该信源的香农线图。
- (2) 实际上各个元素之间有关联, 其转移概率为: $p(\text{白}/\text{白}) = 0.9143$, $p(\text{黑}/\text{白}) = 0.0857$, $p(\text{白}/\text{黑}) = 0.2$, $p(\text{黑}/\text{黑}) = 0.8$, 求这个一阶马尔可夫信源的信源熵, 并画出该信源的香农线图。
- (3) 比较两种信源熵的大小, 并说明原因。

- 2 17 每帧电视图像可以认为是由 3×10^5 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? 若现有一广播员在约 10000 个汉字的字汇中选 1000 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用多少汉字?

- 2 18 一个随机变量 x 的概率密度函数 $p_X(x) = kx$, $0 \leq x \leq 2V$, 试求该信源的相对熵。

- 2 19 给定语音信号样值 X 的概率密度为 $p_X(x) = \frac{1}{2} e^{-|x|}$, $-\infty < x < \infty$, 求 $H_c(X)$, 并证明它小于同样方差的正态变量的连续熵。

- 2 20 (1) 随机变量 X 表示信号 $x(t)$ 的幅度, $-3V \leq x(t) \leq 3V$, 均匀分布, 求信源熵 $H_0(X)$ 。

- (2) 若 X 在 $-5V$ 和 $5V$ 之间均匀分布, 求信源熵 $H(X)$ 。

- (3) 试解释(1)和(2)的计算结果。

- 2 21 随机信号的样值 X 在 $1V$ 和 $7V$ 之间均匀分布,

(1) 计算信源熵 $H_0(X)$ 。将此结果与上题中的(1)相比较,可得到什么结论?

(2) 计算期望值 $E(X)$ 和方差 $\text{var}(X)$ 。

2 22 连续随机变量 X 和 Y 的联合概率密度为

$$p_{XY}(x, y) = \frac{1}{2} \exp \left[-\frac{1}{2N} \left(x^2 + 1 + \frac{N}{S} - 2xy + y^2 \right) \right]$$

求 $H(X)$, $H(Y)$, $H(Y|X)$ 和 $I(X; Y)$ 。

2 23 连续随机变量 X 和 Y 的联合概率密度为

$$p_{XY}(x, y) = \begin{cases} \frac{1}{r^2}, & x^2 + y^2 \leq r^2 \\ 0 & \text{其他} \end{cases}$$

求 $H(X)$, $H(Y)$, $H(XY)$ 和 $I(X; Y)$ 。

2 24 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) 求符号的平均熵。

(2) 由 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)中的序列的熵。

2 25 设有一个二进制一阶马尔可夫信源,其信源符号为 $X \in (0, 1)$,条件概率为

$$p(0/0) = 0.25, p(0/1) = 0.50, p(1/0) = 0.75, p(1/1) = 0.50$$

画出状态图并求出各符号稳态概率。

2 26 一阶马氏链信源有三个符号 $\{u_1, u_2, u_3\}$,转移概率为:

$$\begin{aligned} p(u_1/u_1) &= 1/2, & p(u_2/u_1) &= 1/2, & p(u_3/u_1) &= 0, & p(u_1/u_2) &= 1/3, \\ p(u_2/u_2) &= 0, & p(u_3/u_2) &= 2/3, & p(u_1/u_3) &= 1/3, & p(u_2/u_3) &= 2/3, \\ p(u_3/u_3) &= 0. \end{aligned}$$

画出状态图并求出各符号稳态概率。

2 27 由符号集 $\{0, 1\}$ 组成的二阶马氏链,转移概率为: $p(0/00) = 0.8$, $p(0/11) = 0.2$, $p(1/00) = 0.2$, $p(1/11) = 0.8$, $p(0/01) = 0.5$, $p(0/10) = 0.5$, $p(1/01) = 0.5$, $p(1/10) = 0.5$ 。画出状态图,并计算各状态的稳态概率。

2 28 有一个马尔可夫信源,已知转移概率为 $p(s_1/s_1) = 2/3$, $p(s_2/s_1) = 1/3$, $p(s_1/s_2) = 1$, $p(s_2/s_2) = 0$ 。试画出状态转移图,并求出信源熵。

2 29 设有一信源,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 时则 X_2 为 a, b, c 的概率为 $1/3$;如果 X_1 为 c 时则 X_2 为 a, b 的概率为 $1/2$,而为 c 的概率是 0。而且后面发出 X_i 的概率只与 X_{i-1} 有关。又 $p(X_i/X_{i-1}) = p(X_2/X_1)$, $i \geq 3$ 。试利用马尔可夫信源的图示法画出状态转移图,并求出转移概率矩阵和信源熵 H 。

2 30 一个马尔可夫过程的基本符号 0, 1, 2, 这三个符号以等概率出现,具有相同的转移概率,并且没有固定约束。

(1) 画出一阶马尔可夫过程的状态图,并求稳定状态下的马尔可夫信源熵 H_1 。

(2) 画出二阶马尔可夫过程的状态图,并求稳定状态下二阶马尔可夫信源熵 H_2 。

2 31 有一个一阶平稳马尔可夫链 $X_1, X_2, \dots, X_r, \dots$, 各 X_r 取值于集 $A = \{a_1, a_2, a_3\}$ 。已

知起始概率 $p(a_i)$ 为: $p_1 = 1/2, p_2 = p_3 = 1/4$, 转移概率如右下表所示, 求

- (1) $X_1 X_2 X_3$ 的联合熵和平均符号熵。
- (2) 这个链的极限平均符号熵。
- (3) H_0, H_1, H_2 和它们所对应的冗余度。

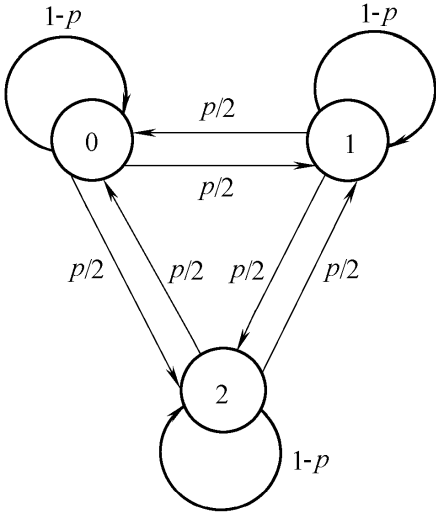
$j \backslash i$	1	2	3
1	1/2	1/4	1/4
2	2/3	0	1/3
3	2/3	1/3	0

2 32 一阶马尔可夫信源的状态图如图题 2 32 所示, 信源 X 的符号集为 $\{0, 1, 2\}$ 。

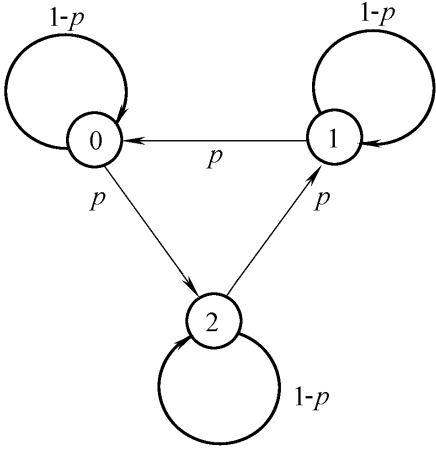
- (1) 求信源平稳后的概率分布 $p(0), p(1)$ 和 $p(2)$ 。
- (2) 求此信源的熵。
- (3) 近似认为此信源为无记忆时, 符号的概率分布等于平稳分布。求近似信源的熵 $H(X)$ 并与 H 进行比较。
- (4) 对一阶马尔可夫信源 p 取何值时 H 取最大值, 又当 $p = 0$ 或 $p = 1$ 时结果如何 ?

2 33 一阶马尔可夫信源的状态图如图题 2 33 所示, 信源 X 符号集为 $\{0, 1, 2\}$, 求

- (1) 平稳后的信源的概率分布;
- (2) 信源熵 H ;
- (3) 当 $p = 0$ 或 $p = 1$ 时信源的熵, 并说明其理由。



图题 2 32



图题 2 33

第 3 章 无失真信源编码

前一章介绍了信源熵的概念,弄清了传送信源信息只需要具有信源极限熵大小的信息率,但是在实际通信系统中,用来传送信源信息的信息率远大于信源极限熵。像信源熵这样的最小信息率是否能够达到或接近,这就是编码定理要回答的问题之一。编码分为信源编码和信道编码,其中信源编码又分为无失真和限失真。由于这些定理都要求符号数很大才能使它的值接近所规定的值,因而这些定理被称为极限定理。一般称无失真信源编码定理为第一极限定理;信道编码定理(包括离散和连续信道)称为第二极限定理;限失真信源编码定理称为第三极限定理。这些定理的完善化,是香农信息论的主要内容。下面将用三章来分别讨论这三大定理。

由于信源符号之间存在分布不均匀和相关性,使得信源存在冗余度,信源编码的主要任务就是减少冗余,提高编码效率。具体说,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。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互相独立,即解除相关性;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:无失真编码定理和限失真编码定理。前者是可逆编码的基础。可逆是指当信源符号转换成代码后,可从代码无失真地恢复原信源符号。当已知信源符号的概率特性时,可计算它的符号熵 H ,这表示每个信源符号所载有的信息量。编码定理不但证明了必存在一种编码方法,使代码的平均长度可任意接近但不能低于符号熵,而且还阐明了达到这目标的途径,就是使概率与码长匹配。无失真编码或可逆编码只适用于离散信源。对于连续信源,编成代码后就无法无失真地恢复原来的连续值,因为后者的取值可有无限多个。此时只能根据限失真编码定理进行限失真编码。信源编码定理出现后,编码方法就趋于合理化。本章讨论离散信源无失真编码,从无失真编码定理出发,重点讨论以香农码、费诺码和哈夫曼码为代表的最佳码。

3.1 编码的定义

将信源消息分成若干组,即符号序列 X_i , $X_i = (X_1 X_2 \dots X_L \dots X_L)$, 序列中的每个符号取自于符号集 A , $X_L \in \{a_1, a_2, \dots, a_i, \dots, a_n\}$ 。而每个符号序列 X_i 依照固定的码表映射成一个码字 Y_i , 这样的码称为分组码,有时也叫块码。只有分组码才有对应的码表,而非分组码中则不存在码表。

如图 3 1 1 所示的信源编码器,如果信源输出的符号序列长度为 L ,即信源符号集

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

信源概率空间为

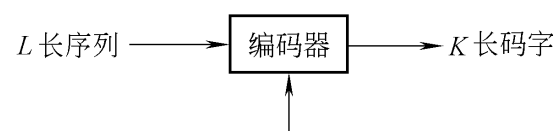


图 3 1 1 信源编码器

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

二元信道是常用的一种信道,它的信道基本符号集为{0,1}。若将信源 X 通过一个二元信道传输,就必须把信源符号 x_i 变换成由 0,1 符号组成的码符号序列,即编码。可用不同的码符号序列,如表3 1 1所示。

表 3 1 1 不同的码符号序列

信源符号 x_i	信源符号出 现概率 $p(x_i)$	码 表		信源符号 x_i	信源符号出 现概率 $p(x_i)$	码 表	
		码 1	码 2			码 1	码 2
x_1	$p(x_1)$	00	0	x_3	$p(x_3)$	10	001
x_2	$p(x_2)$	01	01	x_4	$p(x_4)$	11	111

一般情况下,码可分为两类:一类是固定长度的码,码中所有码字的长度都相同,如表 3 1 1中的码 1 就是定长码。另一类是可变长度码,码中的码字长短不一,如表中码 2 就是变长码。

采用分组编码方法,需要分组码具有某些属性,以保证在接收端能够迅速准确地将码译出。下面首先讨论分组码的一些直观属性。

奇异码和非奇异码:若信源符号和码字是一一对应的,则该码为非奇异码。反之为奇异码。如表 3 1 2 中的码 1 是奇异码,码 2 是非奇异码。

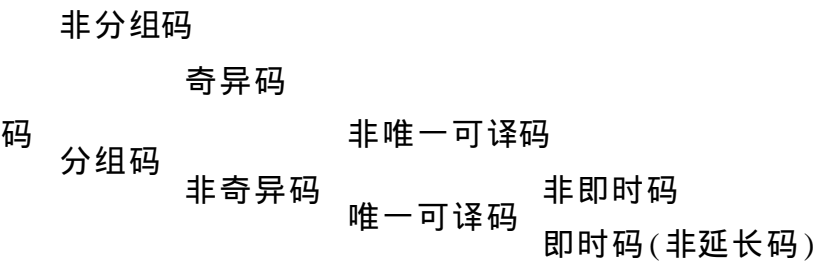
表 3 1 2 不同的码字

信源符号 x_i	符号出现概率 $p(x_i)$	码 1	码 2	码 3	码 4
x_1	1/ 2	0	0	1	1
x_2	1/ 4	11	10	10	01
x_3	1/ 8	00	00	100	001
x_4	1/ 8	11	01	1000	0001

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,便称为唯一可译码。例如{0,10,11}是一种唯一可译码。因为任意一串有限长码序列,如 100111000,只能被分割成 10,0,11,10,0,0。任何其他分割法都会产生一些非定义的码字。显然,奇异码不是唯一可译码,而非奇异码中有非唯一可译码和唯一可译码。表 3 1 2 中码 3 是唯一可译码,但码 2 不是唯一可译码。

唯一可译码中又分为非即时码和即时码:如果接收端收到一个完整的码字后,不能立即译码,还需等下一个码字开始接收后才能判断是否可以译码,这样的码叫做非即时码。表 3 1 2中码 3 是非即时码,而码 4 是即时码。码 4 中只要收到符号 1 就表示该码字已完整,可以立即译码。即时码又称为非延长码,任意一个码字都不是其它码字的前缀部分,有时叫做异前缀码。在延长码中,有的码是唯一可译的,主要取决于码的总体结构,如表 3 1 2 中码 3 的延长码就是唯一可译的。

综上所述,可将码作如下分类:



通常可用码树来表示各码字的构成。对于 r 进制的码树,如图 3 1 2 所示。图(a)是二进制码树,图(b)是三进制码树。其中 A 点是树根,分成 r 个树枝,成为 r 进码树。树枝的尽头是节点,中间节点生出树枝,终端节点安排码字。码树中自根部经过一个分枝到达 r 个节点称为一级节点。二级节点的可能个数为 r^2 个,一般 n 级节点有 r^n 个。图(a)的码树是 4 节,有 $2^4 = 16$ 个可能的终端节点。若将从每个节点发出的 r 个分枝分别标以 $0, 1, \dots, r - 1$, 则每个 n 级节点需要用 n 个 r 元数字表示。如果指定某个 n 级节点为终端节点表示一个信源符号,则该节点就不再延伸,相应的码字即为从树根到此端点的分枝标号序列,其长度为 n 。这样构造的码满足即时码的条件。因为从树根到每一个终端节点所走的路径均不相同,故一定满足对前缀的限制。如果有 q 个信源符号,那么在码树上就要选择 q 个终端节点,用相应的 r 元基本符号表示这些码字。由这样的方法构造出来的码称为树码,若树码的各个分支都延伸到最后一级端点,此时将共有 r^n 个码字,这样的码树称为满树,如图(a)所示。否则就称为非满树,如图(b)所示,这时的码字就不是定长的了。

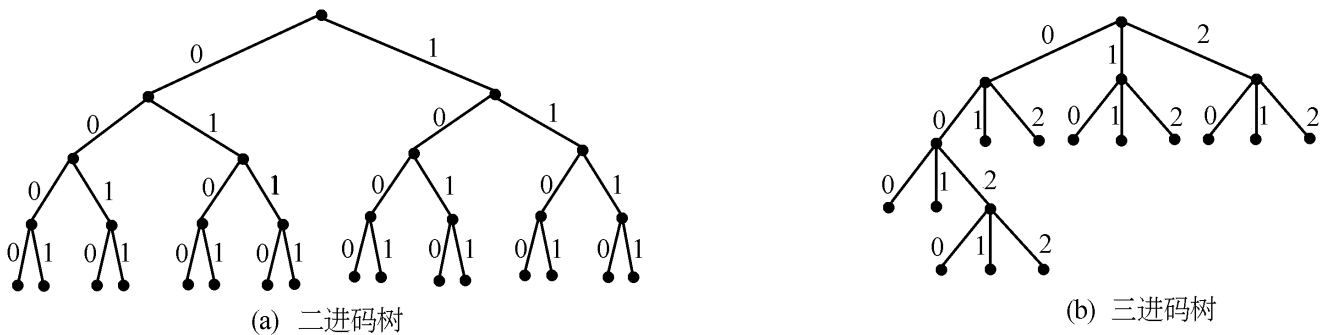


图 3 1 2 码树图

用树的概念可导出唯一可译码存在的充分和必要条件,即各码字的长度 K_i 应符合克劳夫特不等式:

$$\sum_{i=1}^n m^{-K_i} \leq 1 \tag{3 1 1}$$

式中, m 是进制数, n 是信源符号数。

例 3 1 1 设二进制码树中 $X = \{x_1, x_2, x_3, x_4\}$, $K_1 = 1, K_2 = 2, K_3 = 2, K_4 = 3$, 应用上述判断定理,可得

$$\sum_{i=1}^4 2^{-K_i} = 2^{-1} + 2^{-2} + 2^{-2} + 2^{-3} = \frac{9}{8} > 1$$

因此,不存在满足这种 K_i 的唯一可译码。可以用树码进行检查,由图 3 1 3 所示,要形成上述码字,必然在中间节点放置码字,如 $a_1 a_1$ 处,这样就产生了延长码。如码字为 $\{0, 10, 11, 110\}$ 。如果将各码字长度改成 $K_1 = 1, K_2 = 2, K_3 = 3, K_4 = 3$, 则此时

$$\sum_{i=1}^4 2^{-K_i} = 2^{-1} + 2^{-2} + 2^{-3} + 2^{-3} = 1$$

这样的码就存在唯一可译码,如 $\{0, 10, 110, 111\}$ 。但是必须注意,克劳夫特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。如码字 $\{0, 10, 010, 111\}$, 虽然满足克劳夫特不等式,但它不是唯一可译码。

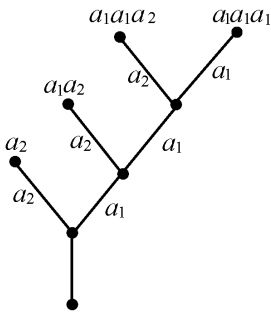


图 3 1 3 树码

若信源输出符号序列的长度 $L \rightarrow \infty$, 即

$$X = (X_1 X_2 \dots X_L \dots X_L), \quad (3.1.2)$$

$$X_L \in \{a_1, a_2, \dots, a_i, \dots, a_n\}$$

变换成由 K_L 个符号组成的码序列(有时也叫做码字, 以下均用码字来叙述。)

$$Y = (Y_1 Y_2 \dots Y_k \dots Y_{K_L}), \quad (3.1.3)$$

$$Y_k \in \{b_1, b_2, \dots, b_j, \dots, b_m\}.$$

变换的要求是能够无失真或无差错地从 Y 恢复 X , 也就是能正确地进行反变换或译码, 同时希望传送 Y 时所需要的信息率最小。由于 Y_k 可取 m 种可能值, 即平均每个符号输出的最大信息量为 $\log m$, K_L 长码字的最大信息量为 $K_L \log m$ 。用该码字表示 L 长的信源序列, 则送出一个信源符号所需要的信息率平均为 $\bar{K} = \frac{K_L}{L} \log m = \frac{1}{L} \log M$, 其中 $M = m^{K_L}$ 是 Y 所能编成的码字的个数。所谓信息率最小, 就是找到一种编码方式使 $\frac{K_L}{L} \log m$ 最小。然而上述最小信息率为多少时, 才能得到无失真的译码? 若小于这个信息率是否还能无失真地译码? 这就是无失真信源编码定理要研究的内容。在无失真的信源编码定理中相应的有定长编码定理和变长编码定理, 下面我们分别加以讨论。

3.2 定长编码定理

在定长编码中, K 是定值, 且 $K = K_L$ 。我们的目的是寻找最小 K 值。要实现无失真的信源编码, 不但要求信源符号 X_i ($i = 1, 2, \dots, q$) 与码字 Y_i ($i = 1, 2, \dots, q$) 是一一对应的, 而且还要求由码字组成的码符号序列的逆变换也是唯一的。也就是说, 由一个码表编出的任意一串有限长的码符号序列只能被唯一地译成所对应的信源符号序列。

定长编码定理: 由 L 个符号组成的、每个符号的熵为 $H_L(X)$ 的无记忆平稳信源符号序列 $X_1 X_2 \dots X_L \dots X_L$, 可用 K_L 个符号 $Y_1, Y_2, \dots, Y_k, \dots, Y_{K_L}$ (每个符号有 m 种可能值) 进行定长编码。对任意 $\epsilon > 0$, $\delta > 0$, 只要

$$\frac{K_L}{L} \log m \geq H_L(X) + \epsilon \quad (3.2.1)$$

则当 L 足够大时, 必可使译码差错小于 δ ; 反之, 当

$$\frac{K_L}{L} \log m \leq H_L(X) - \epsilon \quad (3.2.2)$$

时, 译码差错一定是有限值, 而当 L 足够大时, 译码几乎必定出错。

这个定理的前一部分是正定理, 后一部分为逆定理。定理证明略。

通过上述编码定理, 使我们对信源平均符号熵 $H_L(X)$ 有较好的理解。当编码器容许的输出信息率, 也就是当每个信源符号所必须输出的码长是

$$\bar{K} = \frac{K_L}{L} \log m \quad (3.2.3)$$

时, 只要 $\bar{K} > H_L(X)$, 这种编码器一定可以做到几乎无失真, 也就是收端的译码差错概率接近于零, 条件是所取的符号数 L 足够大。

将上述定理的条件(3 2 1)式改写成

$$K_L \log m > L H_L(X) = H(X) \quad (3 2 4)$$

上式大于号左边为 K_L 长码字所能携带的最大信息量,右边为 L 长信源序列携带的信息量。于是上述定理表明,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,当然条件是 L 足够大。反之,当 $\overline{K} < H_L(X)$ 时,不可能构成无失真的编码,也就是不可能做一种编码器,能使收端译码时差错概率趋于零。 $\overline{K} = H_L(X)$ 时,则为临界状态,可能无失真,也可能有失真。

例如,某信源有 8 种等概率符号, $L = 1$, 信源序列熵达到最大值:

$$H_1(X) = \log_2 8 = 3 \text{ 比特}$$

即该信源符号肯定可以用 3 比特的信息率进行无失真的编码。这就是说,如果采用二进制符号作为码字输出符号, $Y_k \in \{0, 1\}$, 则用 3 个比特就可以表示一个符号,即 $\overline{K} = 3$ 比特/符号 = $H_0(X)$ 。当信源符号输出概率不相等时,如 $p(x_i) = \{0.4, 0.18, 0.1, 0.1, 0.07, 0.06, 0.05, 0.04\}$, 则此时 $H_1(X) = 2.55$ 比特/符号,小于 3 比特。按常理,8 种符号一定要用 3 比特($2^3 = 8$)组成的码字表示才能区别开来,而用 $\overline{K} = H_L(X) = 2.55$ 比特/符号来表示,只有 $2^{2.55} = 5.856$ 种可能码字,还有部分符号没有对应的码字,这些符号一旦出现,被传输至接收端,就没有对应的码字译码,因而引起译码差错。所以定长编码一般都存在译码差错,只是差错大小不同。当 L 足够大时,译码差错可以达到足够小。

设 $X_i = (X_1 X_2 \dots X_L \dots X_L)$ 是信源序列的样本矢量, $X_i \in \{a_1, a_2, \dots, a_i, \dots, a_n\}$, 则共有 n^L 种样本,我们把它分为两个互补的集 A 和 A^C , 集 A 中的元素(样本矢量)有与之对应的不同码字,而集 A^C 中的元素没有对应的输出码字,因而会在译码时发生差错。如果允许一定的差错,则编码时只需对属于 A 中的 M 个样本矢量赋以相应的不同码字,即输出码字的总个数 m^K 只要大于 M 就可以了。在这种编码方式下,差错概率 P_e 即为集 A^C 中元素发生的概率 $P(A^C)$, 此时要求 $P(A^C) \rightarrow 0$, 因而 A^C 集中的样本都应是小概率事件。当 L 增大时,虽然样本数也随着增多,但小概率事件的概率将更小,有望使 $P(A^C)$ 更小。根据切比雪夫不等式可推得(推导从略,见参考文献 1)

$$P_e \leq \frac{\sigma^2(X)}{L^2} \quad (3 2 5)$$

式中, $\sigma^2(X) = E\{[I(X) - H(X)]^2\}$ 为信源序列的自信息方差; ϵ 为一正数。当 $\sigma^2(X)$ 和 ϵ 均为定值时,只要 L 足够大, P_e 可以小于任一正数,即 $\frac{\sigma^2(X)}{L^2} < \epsilon$, 也就是当信源序列长度 L 满足

$$L > \frac{\sigma^2(X)}{\epsilon} \quad (3 2 6)$$

时,就能达到差错率要求。

说得具体一些,就是给定 ϵ 和 $\sigma^2(X)$ 后,用(3 2 6)式规定了 L 的大小,计算所有可能的信源序列样本矢量的概率 $P(X_i)$, 按概率大小排列,选用概率较大的 X_i 作为 A 中的元素,直到 $P(A) \geq 1 - \epsilon$, 使 $P(A^C) \leq \epsilon$ 。这些在 A 中的元素分别用不同码字来代表,就完成了编码过程。如果取足够小的 ϵ , 就可几乎无差错地译码,而所需的信息率就不会超过 $H_L(X) + \epsilon$ 。

在连续信源的情况下,由于信源的信息量趋于无限,显然是不能用离散符号序列 Y 来完成无失真编码的,而只能进行限失真编码。

定义

$$= \frac{H_L(X)}{K}$$

为编码效率。即信源的平均符号熵为 $H(X)$, 采用平均符号码长为 \overline{K} 来编码, 所得的效率。编码效率总是小于 1, 且最佳编码效率为

$$= \frac{H_L(X)}{H_L(X) + \dots}, \quad > 0 \tag{3 2 7}$$

编码定理从理论上阐明了编码效率接近 1 的理想编码器的存在性, 它使输出符号的信息率与信源熵之比接近于 1, 即

$$\frac{H_L(X)}{\frac{K_L}{L} \log m} \rightarrow 1 \tag{3 2 8}$$

但要在实际中实现, 必须取无限长 ($L \rightarrow \infty$) 的信源符号进行统一编码。这样做实际上是不可能的, 因 L 非常大, 无法实现。下面用例子来说明。

例 3 2 1 设离散无记忆信源概率空间为

X	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
P	0.4	0.18	0.1	0.1	0.07	0.06	0.05	0.04

信源熵为

$$H(X) = - \sum_{i=1}^8 p_i \log_2 p_i = 2.55 \text{ 比特/符号}$$

自信息的方差

$$\sigma^2(X) = D[I(x_i)] = \sum_{i=1}^8 p_i (\log_2 p_i)^2 - [H(X)]^2 = 7.82$$

对信源符号采用定长二元编码, 要求编码效率为 $\eta = 90\%$, 无记忆信源有 $H_L(X) = H(X)$, 则据(3 2 7)式

$$\eta = \frac{H(X)}{H(X) + \dots} = 0.90,$$

可以得到 $\dots = 0.28$

要求译码错误概率 10^{-6} , 由(3 2 6)式得

$$L \geq \frac{\sigma^2(X)}{2} = \frac{7.82}{0.28^2 \times 10^{-6}} = 9.8 \times 10^7 \sim 10^8$$

由此可见, 在对编码效率和译码错误概率的要求并不十分苛刻的情况下, 就需要 $L = 10^8$ 个信源符号一起进行编码, 这对存储或处理技术的要求太高, 目前还无法实现。

如果用 3 比特来对上述信源的 8 个符号进行定长二元编码, $L = 1$, 则 $\overline{K} = H(X) + \dots = 3$, 可以求得 $\eta = 0.45$ 。此时译码无差错, 即 $\epsilon = 0$ 。在这种情况下, (3 2 6)式就不适用了。

但此时编码效率只能为 $\eta = \frac{2.55}{3} = 85\%$ 。因此一般来说, 当 L 有限时, 高传输效率的定长码往往要引入一定的失真和错误, 它不像变长码那样可以实现无失真编码。

3.3 变长编码定理

在变长编码中,码长 K 是变化的,我们可根据信源各个符号的统计特性,如概率大的符号用短码,如例 3 2 1 中的 x_1, x_2 可用 1 或 2 比特,而对概率小的 x_7, x_8 用较长的码,这样在大量信源符号编成码后平均每个信源符号所需的输出符号数就可以降低,从而提高编码效率。下面分别给出单个符号 ($L = 1$) 和符号序列的变长编码定理。

单个符号变长编码定理: 若一离散无记忆信源的符号熵为 $H(X)$, 每个信源符号用 m 进制码元进行变长编码,一定存在一种无失真编码方法,其码字平均长度 \bar{K} 满足下列不等式

$$\frac{H(X)}{\log m} \leq \bar{K} < \frac{H(X)}{\log m} + 1 \quad (3.3.1)$$

离散平稳无记忆序列变长编码定理: 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,使平均信息率 \bar{K} 满足不等式

$$H_L(X) \leq \bar{K} < H_L(X) + \epsilon \quad (3.3.2)$$

其中 ϵ 为任意小正数。

可从 (3.3.1) 式推出 (3.3.2) 式。设用 m 进制码元作变长编码,序列长度为 L 个信源符号,则由 (3.3.1) 式可以得到平均码字长度 \bar{K}_L 满足下列不等式

$$\frac{LH_L(X)}{\log m} \leq \bar{K}_L < \frac{LH_L(X)}{\log m} + L$$

已知平均输出信息率为

$$\bar{K} = \frac{\bar{K}_L}{L} \log m$$

$$\text{则 } H_L(X) \leq \bar{K} < H_L(X) + \frac{\log m}{L}$$

当 L 足够大时,可使 $\frac{\log m}{L} < \epsilon$, 这就得到了 (3.3.2) 式。

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。从 (3.3.2) 式可得编码效率的下界:

$$\eta = \frac{H_L(X)}{\bar{K}} > \frac{H_L(X)}{H_L(X) + \frac{\log m}{L}} \quad (3.3.3)$$

例如用二进制, $m = 2, \log_2 m = 1$, 仍用前面的例 3 2 1, $H(X) = 2.55$ 比特/符号,若要求 $\eta > 90\%$, 则

$$\frac{2.55}{2.55 + \frac{1}{L}} = 0.9, L = \frac{1}{0.28} \approx 4$$

就可以了。

编码效率总是小于 1 的,我们可以用它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

$$= 1 - \frac{H_L(X)}{K} \tag{3.3.4}$$

例 3.3.1 设离散无记忆信源的概率空间为

$$\begin{matrix} X & x_1 & x_2 \\ P & \frac{3}{4} & \frac{1}{4} \end{matrix}$$

其信源熵为

$$H(X) = \frac{1}{4} \log_2 4 + \frac{3}{4} \log_2 \frac{4}{3} = 0.81 \text{ 比特/符号}$$

若用二元定长编码(0,1)来构造一个即时码： $x_1 = 0, x_2 = 1$ 。这时平均码长

$$\overline{K} = 1 \text{ 二元码符号/信源符号}$$

编码效率为

$$= \frac{H(X)}{\overline{K}} = 0.811$$

输出的信息率为

$$R = 0.811 \text{ 比特/二元码符号}$$

再对长度为 2 的信源序列进行变长编码,其即时码如表 3.3.1 所示。

表 3.3.1 L = 2 时信源序列的变长编码

序 列	序列概率	即 时 码	序 列	序列概率	即 时 码
$x_1 x_1$	9/ 16	0	$x_2 x_1$	3/ 16	110
$x_1 x_2$	3/ 16	10	$x_2 x_2$	1/ 16	111

这个码的码字平均长度

$$\overline{K}_2 = \frac{9}{16} \times 1 + \frac{3}{16} \times 2 + \frac{3}{16} \times 3 + \frac{1}{16} \times 3 = \frac{27}{16} \text{ 二元码符号/信源序列}$$

每一单个符号的平均码长

$$\overline{K} = \frac{\overline{K}_2}{2} = \frac{27}{32} \text{ 二元码符号/信源符号}$$

其编码效率

$$\eta_2 = \frac{32 \times 0.811}{27} = 0.961$$

$$R_2 = 0.961 \text{ 比特/二元码符号}$$

可见编码复杂了一些,但信息传输率有了提高。

用同样的方法可进一步将信源序列的长度增加, L = 3 或 L = 4, 对这些信源序列 X 进行编码,并求出其编码效率为

$$\eta_3 = 0.985$$

$$\eta_4 = 0.991$$

这时信息传输率分别为

$$R_3 = 0.985 \text{ 比特/二元码符号}$$

$$R_4 = 0.991 \text{ 比特/二元码符号}$$

如果对这一信源采用定长二元码编码,要求编码效率达到 96% 时,允许译码错误概率 10^{-5} 。则根据(3 2 6)式,自信息的方差

$$\sigma^2(X) = \sum_{i=1}^2 p_i (\log p_i)^2 - [H(X)]^2 = 0.4715$$

所需要的信源序列长度

$$L = \frac{0.4715}{(0.811)^2} \cdot \frac{(0.96)^2}{0.04^2 \times 10^{-5}} = 4.13 \times 10^7$$

很明显,定长码需要的信源序列长,使得码表很大,且总存在译码差错;而变长码要求编码效率达到 96% 时,只需 $L = 2$ 。因此用变长码编码时, L 不需要很大就可达到相当高的编码效率,而且可实现无失真编码。随着信源序列长度的增加,编码的效率越来越接近于 1,编码后的信息传输率 R 也越来越接近于无噪无损二元对称信道的信道容量 $C = 1$ 比特/二元码符号,达到信源与信道匹配,使信道得到充分利用。

至此讨论的编码定理,针对的是离散平稳无记忆信源,也就是说仅考虑了信源符号分布的不均匀性,并没有考虑符号之间的相关性。因相关性比较复杂,很难定量描述。但在一些具体的编码方法中都考虑了相关性,如预测编码、变换编码等,将在 4.4 节中讨论。

3.4 最佳编码

凡是能载荷一定的信息量,且码字的平均长度最短,可分离的变长码的码字集合都可称为最佳码。为此必须将概率大的信息符号编以短的码字,概率小的符号编以长的码字,使得平均码字长度最短。能获得最佳码的编码方法主要有:香农(Shannon)、费诺(Fano)、哈夫曼(Huffman)编码等。

3.4.1 香农编码方法

香农第一定理指出了平均码长与信源之间的关系,同时也指出了可以通过编码使平均码长达到极限值,这是一个很重要的极限定理。如何构造这种码?香农第一定理指出,选择每个码字的长度 K_i 满足下式

$$I(x_i) - K_i < I(x_i) + 1, \quad \forall i$$

就可以得到这种码。这种编码方法称为香农编码。

香农编码法多余度稍大,实用性不大,但有重要的理论意义。编码方法如下:

(1) 将信源消息符号按其出现的概率大小依次排列

$$p(x_1) \geq p(x_2) \geq \dots \geq p(x_n)$$

(2) 确定满足下列不等式的整数码长 K_i :

$$- \log_2 p(x_i) - K_i < - \log_2 p(x_i) + 1$$

(3) 为了编成唯一可译码,计算第 i 个消息的累加概率

$$P_i = \sum_{k=1}^{i-1} p(x_k)$$

(4) 将累加概率 P_i 变换成二进制数。

(5) 取 P_i 二进数的小数点后 K_i 位即为该消息符号的二进制码字。

例 3 4 1 设信源共 7 个符号消息,其概率和累加概率如表 3 4 1 所示。以 $i = 4$ 为例,

$$\begin{aligned} -\log_2 0.17 \quad K_4 < -\log_2 0.17 + 1 \\ 2.56 \quad K_4 < 3.56, K_4 = 3 \end{aligned}$$

累加概率 $P_4 = 0.57$, 变换成二进制为 0.1001..., 由于 $K_4 = 3$, 所以第 4 个消息的编码码字为 100。其他消息的码字可用同样方法求得, 如表 3 4 1 所示。该信源共有 5 个三位的码字, 各码字之间至少有一位数字不相同, 故是唯一可译码。同时可以看出, 这 7 个码字都不是延长码, 它们都属于即时码。这里 $L = 1, m = 2$, 所以信源符号的平均码长

$$\overline{K} = \sum_{i=1}^7 p(x_i) K_i = 3.14 \text{ 码元/符号}$$

平均信息传输率

$$R = \frac{H(X)}{\overline{K}} = \frac{2.61}{3.14} = 0.831 \text{ 比特/码元}$$

表 3 4 1 香农编码过程

信源消息符号 x_i	符号概率 $p(x_i)$	累加概率 P_i	$-\log_2 p(x_i)$	码字长度 K_i	码 字
x_1	0.20	0	2.34	3	000
x_2	0.19	0.2	2.41	3	001
x_3	0.18	0.39	2.48	3	011
x_4	0.17	0.57	2.56	3	100
x_5	0.15	0.74	2.74	3	101
x_6	0.10	0.89	3.34	4	1110
x_7	0.01	0.99	6.66	7	1111110

3.4.2 费诺编码方法

费诺编码属于概率匹配编码,但它不是最佳的编码方法。编码过程如下:

- (1) 将信源消息符号按其出现的概率大小依次排列: $p(x_1) \quad p(x_2) \quad \dots \quad p(x_n)$ 。
- (2) 将依次排列的信源符号按概率值分为两大组,使两个组的概率之和近于相同,并对各组赋予一个二进制码元“0”和“1”。
- (3) 将每一大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近于相同,并又赋予两个组一个二进制符号“0”和“1”。
- (4) 如此重复,直至每个组只剩下一个信源符号为止。
- (5) 信源符号所对应的码字即为费诺码。

下面再以上面的例子来求出费诺码。编码过程参见表 3 4 2。该费诺码的平均码长

$$\overline{K} = \sum_{i=1}^7 p(x_i) K_i = 2.74 \text{ 码元/符号}$$

信息传输速率

$$R = \frac{H(X)}{\overline{K}} = \frac{2.61}{2.74} = 0.953 \text{ 比特/码元}$$

显然费诺码要比上述香农码的平均码长小,消息传输速率大,说明编码效率高。

表 3 4 2 费诺编码过程

消息符号 x_i	各个消息 概率 $p(x_i)$	第 一 次 分 组	第 二 次 分 组	第 三 次 分 组	第 四 次 分 组	二 元 码 字	码 长 K_i	
X_1	0.20	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

3.4.3 哈夫曼编码方法

(1) 将 n 个信源消息符号按其出现的概率大小依次排列,

$$p(x_1) \quad p(x_2) \quad \dots \quad p(x_n)$$

(2) 取两个概率最小的字母分别配以 0 和 1 两码元, 并将这两个概率相加作为一个新字母的概率, 与未分配的二进符号的字母重新排队。

(3) 对重排后的两个概率最小符号重复步骤(2)的过程。

(4) 不断继续上述过程,直到最后两个符号配以 0 和 1 为止。

(5) 从最后一级开始, 向前返回得到各个信源符号所对应的码元序列, 即相应的码字。

我们还是用上面例子中的信源,哈夫曼编码过程如表 3 4 3 所示。

表 3-4-3 哈夫曼编码过程

信源符号 x_i	概率 $p(x_i)$	编 码 过 程	码字 W_i	码长 K_i
x_1	0.20		00	2
x_2	0.19		01	2
x_3	0.18		100	3
x_4	0.17		101	3
x_5	0.15		110	3
x_6	0.10		1110	4
x_7	0.01		1111	4

该哈夫曼码的平均码长为

$$\overline{K} = \frac{1}{7} \sum_{i=1}^7 p(x_i) K_i = 2.72 \text{ 码元/符号}$$

信息传输速率

$$R = \frac{H(\underline{X})}{K} = \frac{2.61}{2.72} = 0.9596 \text{ 比特/码元}$$

由此可见,哈夫曼码的平均码长最小,消息传输速率最大,编码效率最高。

哈夫曼编码方法得到的码并非唯一的。造成非唯一的原因如下：

每次对信源缩减时,赋予信源最后两个概率最小的符号,用 0 和 1 是可以任意的,所以可以得到不同的哈夫曼码,但不会影响码字的长度。

对信源进行缩减时,两个概率最小的符号合并后的概率与其它信源符号的概率相同时,这两者在缩减信源中进行概率排序,其位置放置次序是可以任意的,故会得到不同的哈夫曼码。此时将影响码字的长度,一般将合并的概率放在上面,这样可获得较小的码方差。

下面举例来说明。

例 3 4 2 设有离散无记忆信源

$$\begin{array}{rcccl} X & = & x_1 & x_2 & x_3 & x_4 & x_5 \\ P & = & 0.4 & 0.2 & 0.2 & 0.1 & 0.1 \end{array}$$

可有两种哈夫曼编码方法,如表 3 4 4 和表 3 4 5 所示,码树见图 3 4 1(a)和(b)所示。

表 3-4-4 哈夫曼编码方法一

信源符号 x_i	概率 $p(x_i)$	编 码 过 程	码字 W_i	码长 K_i
x_1	0.4		1	1
x_2	0.2		01	2
x_3	0.2		000	3
x_4	0.1		0010	4
x_5	0.1		0011	4

表 3-4-5 哈夫曼编码方法二

信源符号 x_i	概率 $p(x_i)$	编 码 过 程	码字 W_i	码长 K_i
x_1	0.4		01	2
x_2	0.2		10	2
x_3	0.2		11	2
x_4	0.1		000	3
x_5	0.1		001	3

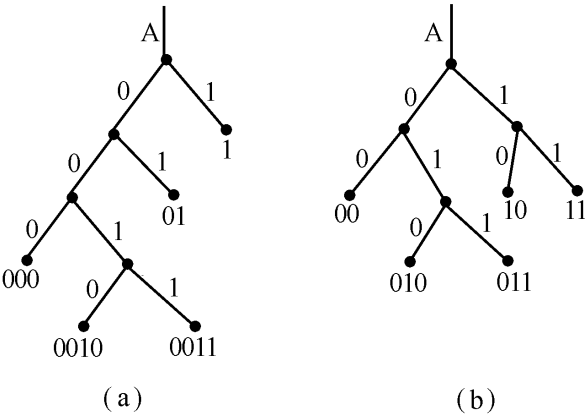


图 3 4 1 哈夫曼码树

由表 3 4 4 和表 3 4 5 给出的哈夫曼码的平均码长相等：

$$\overline{K} = \sum_{i=1}^5 p(x_i) K_i = 2.2 \text{ 码元/符号}$$

编码效率也相等:

$$= \frac{H(X)}{\overline{K}} = 0.965$$

但是两种码的质量不完全相同,可用码方差来表示:

$$\sigma_i^2 = E[(k_i - \overline{K})^2] = \sum_{i=1}^5 p(x_i) (k_i - \overline{K})^2$$

表 3 4 4 中哈夫曼码的方差为 $\sigma_{11}^2 = 1.36$; 表 3 4 5 中哈夫曼码的方差为 $\sigma_{12}^2 = 0.16$ 。

由此可见,第二种哈夫曼编码方法得到的码方差要比第一种哈夫曼编码方法得到的码方差小许多。故第二种哈夫曼码的质量要好。

从上述例子看出,进行哈夫曼编码时,为得到码方差最小的码,应使合并的信源符号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,充分利用短码。

哈夫曼码是用概率匹配方法进行信源编码。它有两个明显特点:一是哈夫曼码的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,充分利用了短码;二是缩减信源的最后二个码字总是最后一位不同,从而保证了哈夫曼码是即时码。

哈夫曼变长码的效率是相当高的,它可以单个信源符号编码或用 L 较小的信源序列编码,对编码器的设计来说也将简单得多。但是应当注意,要达到很高的效率仍然需要按长序列来计算,这样才能使平均码字长度降低。然而对于某一个信源符号而论,有时可能还会比定长码长。例如在上面的例子中,信源符号有 5 个,采用定长码方式可用 3 个二进制符号组成码字。而用变长码时,有的码字却长达 4 个二进制符号。所以编码简单化的代价是要有大量的存储设备来缓冲码字长度的差异,这也是码方差小的码质量好的原因。设一秒钟送一个信源符号,输出的码字有的只有一个二进符号,有的却有 5 个二进符号,若希望平均每秒输出 $\overline{K} = 2.61$ 个二进符号以压缩信息率(与 3 个符号的定长码相比),就必须先把编成的码字存储起来,再按 \overline{K} 的信息率输出,才能从长远来计算,输出和输入保持平衡。当存储量不够大时,就可能有时取空,有时溢出。例如信源常发出短码时,就会出现取空,就是说还没有存入就要输出。常发出长码时,就会溢出,就是存入太多,以致存满了还未取出就再要存入。所以应估计所需的存储器容量,才能使上述现象发生的概率小至可以容忍。

设 T 秒内有 N 个信源符号输出,信源输出符号速率 $S = N/T$, 若符号的平均码长为 \overline{K} , 则信道传输速率

$$R = S \overline{K} \quad (3.4.1)$$

时可以满足条件。

N 个码字的长度分别为 $K_i, i = 1, 2, \dots, N$, 即在此期间输入存储器 K_i 比特, 输出至信道 RT 比特, 则在存储器内还剩 X 比特,

$$X = \sum_{i=1}^N K_i - RT \quad (3.4.2)$$

已知 K_i 是随机变量,其均值和方差分别为

$$\overline{K} = E[K_i] = \sum_{j=1}^m p_j K_j \quad (3.4.3)$$

$$\sigma^2 = E[K_i^2] - \bar{K}^2 = \sum_{j=1}^m p_j K_j^2 - \bar{K}^2 \quad (3.4.4)$$

式中 m 是信源符号集的元数。当 N 足够大时, X 是许多同分布的随机变量之和;由概率论可知,它将近似于正态变量,其均值和方差分别为

$$E[X] = N\bar{K} - RT = (S\bar{K} - R)T$$

$$\sigma_x^2 = N\sigma^2$$

令

$$Y = \frac{X - E[X]}{\sigma_x} \quad (3.4.5)$$

它是标准正态变量,均值为 0,方差为 1。可得下列概率:

$$P(Y > A) = P(Y < -A) = Q(-A) \quad (3.4.6)$$

式中 $Q(-A)$ 是误差函数,可查表得其数值。

如果(3.4.1)式成立,则 $E[X] = 0$ 。设起始时存储器处半满状态,而存储器容量为 $2A_x$,可由(3.4.6)式求得溢出概率和取空概率;因 $Y > A$,即 $X > A_x$,存储器将溢出;而 $Y < -A$,即 $X < -A_x$,相应于存储器取空。这就是说,如果要求这些概率都小于 $Q(-A)$,存储器容量应大于 $2A_x$ 。例如要求溢出概率和取空概率都小于 0.001,查表得 A 应为 3.08,则存储器容量 C 应为

$$C > 6.16 N \quad (3.4.7)$$

当(3.4.1)式不成立时,存储器容量还要增加,在起始时存储器也不应处于半满状态。例如若 $R > S\bar{K}$,平均来说,输出大于输入,易被取空,起始状态可超过半满;反之,若 $R < S\bar{K}$,易于溢出,可不到半满。

由(3.4.7)式可见,时间 T 越长, N 越大,要求存储器的容量也越大。当容量已设定后,随着时间的增长,存储器溢出和取空的概率都将增大;当 T 很大时,几乎一定会溢出或取空,造成损失;即使(3.4.1)式成立,也是如此。由此可见,对于无限长的信息,很难采用变长码而不出现差错。一般来说,变长码只适用于有限长的信息传输;即送出一段信息后,信源能停止输出,例如传真机送出一张纸上的信息后就停止。对于长信息,在实际使用时可把长信息分段发送;也可检测存储器的状态,发现将要溢出就停止信源输出,发现将要取空就插入空闲标志在信道上传送,或加快信源输出。

变长编码可以无失真地译码,这是理想情况。如果这种变长码由信道传送时,有某一个符号错了。因为一个码字前面有某一个字母错了,就可能误认为是另一个码字而点断,结果后面一系列的码字也会译错,这常称为差错的扩散。当然也可以采用某些措施,使错了一段以后,能恢复正常的码字分离和译码,这一般要求在传输过程中差错很少,或者加纠错用的监督码位,但是这样一来又增加了信息率。

此外,当信源有记忆时,用单个符号编制变长码不可能使编码效率接近于 1,因为信息率只能接近一维熵 H_1 ,而 H 一定小于 H_1 。此时仍需要多个符号一起编码,才能进一步提高编码效率。但导致码表长、存储器多。

前面介绍的几种最佳编码都能实现,因为它们具体规定了编码的方法,能使无失真编码的效率非常接近于 1。其他如定长编码以及以后将要讨论的限失真信源编码和信道编码,都还没有具体的实用编法,只是证明了存在性和大致方法。所以在压缩信源信息率的实用

设备中,哈夫曼编码还是比较常用的。

习 题

3 1 将某六进信源进行二进编码如下表所示,求

符 号	概 率	C_1	C_2	C_3	C_4	C_5	C_6
u_1	$1/2$	000	0	0	0	1	01
u_2	$1/4$	001	01	10	10	000	001
u_3	$1/16$	010	011	110	1101	001	100
u_4	$1/16$	011	0111	1110	1100	010	101
u_5	$1/16$	100	01111	11110	1001	110	110
u_6	$1/16$	101	011111	111110	1111	110	111

- (1) 这些码中哪些是唯一可译码？
 - (2) 哪些码是非延长码(即时码)？
 - (3) 所有唯一可译码的平均码长和编码效率。
- 3 2 已知信源的各个符号分别为字母 A, B, C, D , 现用二进制码元对符号字母作信源编码, $A \rightarrow x_0 y_0, B \rightarrow x_0 y_1, C \rightarrow x_1 y_0, D \rightarrow x_1 y_1$, 每个二进制码元的长度为 5 ms 。
- (1) 若各个字母以等概率出现, 计算在无扰离散信道上的平均信息传输速率。
 - (2) 若各个字母的出现概率分别为 $P(A) = 1/5, P(B) = 1/4, P(C) = 1/4, P(D) = 3/10$, 再计算在无扰离散信道上的平均信息传输速率。
- 3 3 若上题中的字母符号改用四进制码元作信源编码, 码元幅度分别为 $0, 1, 2, 3 \text{ V}$, 码元长度为 10 ms 。重新计算上题中两种情况下的平均信息传输速率。
- 3 4 设信道的基本符号集合 $A = \{ a_1, a_2, a_3, a_4, a_5 \}$, 它们的时间长度分别为 $t_1 = 1, t_2 = 2, t_3 = 3, t_4 = 4, t_5 = 5$ (个码元时间)。用这样的信道基本符号编成符号序列, 且不能出现 $a_1 a_1, a_2 a_2, a_1 a_2, a_2 a_1$ 这四种符号相连的情况。
- (1) 求这种编码信道所能传输的信息速率；
 - (2) 若信源的符号集合 $X = \{ x_1, x_2, x_3, \dots, x_7 \}$, 它们的出现概率分别为 $p(x_1) = 1/2, p(x_2) = 1/4, p(x_3) = 1/8, p(x_4) = 1/16, p(x_5) = 1/32, p(x_6) = p(x_7) = 1/64$ 。试求按最佳编码原则利用上述信道来传输这些消息时的信息传输速率；
 - (3) 求上述信源编码的编码效率。
- 3 5 某信源有 8 个符号 $\{ u_1 \dots u_8 \}$, 概率分别为 $1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/128$, 编成这样的码: 000, 001, 010, 011, 100, 101, 110, 111。求
- (1) 信源的符号熵 $H(U)$ ；
 - (2) 出现一个“1”或一个“0”的概率；
 - (3) 这种码的编码效率；
 - (4) 相应的香农码和费诺码；
 - (5) 该码的编码效率。
- 3 6 设无记忆二元信源, 概率为 $p_0 = 0.005, p_1 = 0.995$ 。信源输出 $N = 100$ 的二元序列。

在长为 $N = 100$ 的信源序列中只对含有 3 个或小于 3 个“1”的各信源序列构成一一对应的一组定长码。

- (1) 求码字所需的最小长度。
 - (2) 考虑没有给予编码的信源序列出现的概率,该定长码引起的错误概率 P 是多少?
- 3 7 已知符号集合 $\{x_1, x_2, x_3, \dots\}$ 为无限离散集合,它们的出现概率分别为 $p(x_1) = 1/2$, $p(x_2) = 1/4$, $p(x_3) = 1/8$, $p(x_i) = 1/2^i, \dots$ 。
- (1) 用香农编码方法写出各个符号的码字。
 - (2) 计算码字的平均信息传输率。
 - (3) 计算信源编码效率。
- 3 8 某信源有 6 个符号,概率分别为 $3/8, 1/6, 1/8, 1/8, 1/8, 1/12$,试求三进码元(0,1,2)的费诺码,并求出编码效率。
- 3 9 若某一信源有 N 个符号,并且每个符号均以等概出现,对此信源用最佳哈夫曼二元编码,问当 $N = 2^i$ 和 $N = 2^i + 1$ (i 为正整数)时,每个码字的长度等于多少?平均码长是多少?
- 3 10 设有离散无记忆信源 $P(X) = \{0.37, 0.25, 0.18, 0.10, 0.07, 0.03\}$ 。
- (1) 求该信源符号熵 $H(X)$ 。
 - (2) 用哈夫曼编码编成二元变长码,计算其编码效率。
 - (3) 要求译码错误小于 10^{-3} ,采用定长二元码要达到(2)中哈夫曼编码的效率,问需要多少个信源符号一起编?
- 3 11 信源符号 X 有 6 种字母,概率为(0.32,0.22,0.18,0.16,0.08,0.04)。
- (1) 求符号熵 $H(X)$ 。
 - (2) 用香农编码编成二进变长码,计算其编码效率。
 - (3) 用费诺编码编成二进变长码,计算其编码效率。
 - (4) 用哈夫曼编码编成二进变长码,计算其编码效率。
 - (5) 用哈夫曼编码编成三进变长码,计算其编码效率。
 - (6) 若用单个信源符号来编定长二进码,要求能不出差错地译码,求所需要的每符号的平均信息率和编码效率。
 - (7) 当译码差错小于 10^{-3} 的定长二进码要达到(4)中哈夫曼的效率时,估计要多少个信源符号一起编才能办到?
- 3 12 已知一信源包含 8 个符号,其出现的概率为 $P(X) = \{0.1, 0.18, 0.4, 0.05, 0.06, 0.1, 0.07, 0.04\}$ 。
- (1) 该信源在每秒钟内发出 1 个符号,求该信源的熵及信息传输速率。
 - (2) 对这 8 个符号作哈夫曼编码,写出相应码字,并求出编码效率。
 - (3) 采用香农编码,写出相应码字,求出编码效率。
 - (4) 进行费诺编码,写出相应码字,求出编码效率。
- 3 13 有一 9 个符号的信源,概率分别为 $1/4, 1/4, 1/8, 1/8, 1/16, 1/16, 1/16, 1/32, 1/32$,用三进符号(a, b, c)编码。
- (1) 编出费诺码和哈夫曼码,并求出编码效率;
 - (2) 若要求符号 c 后不能紧跟另一个 c ,编出一种有效码,其编码效率是多少?

- 3 14 一信源可能发出的数字有 1、2、3、4、5、6、7, 对应的概率分别为 $p(1) = p(2) = 1/3$, $p(3) = p(4) = 1/9$, $p(5) = p(6) = p(7) = 1/27$, 在二进或三进无噪信道中传输, 二进信道中传输一个码字需要 1.8 元, 三进信道中传输一个码字需要 2.7 元。
- (1) 编出二进符号的哈夫曼码, 求其编码效率;
 - (2) 编出三进符号的费诺码, 求其编码效率;
 - (3) 根据(1)和(2)的结果, 确定在何种信道中传输可得到较小的花费?
- 3 15 有二元独立序列, 已知 $p_0 = 0.9$, $p_1 = 0.1$, 求这序列的平均符号熵。当用哈夫曼编码时, 以三个二元符号合成一个新符号, 求这种符号的平均代码长度和编码效率。设输入二元符号的速率为每秒 100 个, 要求三分钟内溢出和取空的概率均小于 0.01, 求所需的信道码率(bit/s)和存储器容量(比特数)。若信道码率已规定为 50 bit/s, 存储器容量将如何选择?

第 4 章 限失真信源编码

上一章所讲的信息源编码定理,都是针对无失真的情况。而在实际信息处理过程中,往往允许有一定的失真,例如 A/D 变换,就不可能完全不失真。人们的视觉和听觉都允许有一定失真,电影和电视就是利用了视觉残留,才没有发觉影片是由一张一张画面快速联结起来的。耳朵的频率响应也是有限的,在某些实际场合中只需保留信息的主要特征就够了。所以,一般可以对信源输出的信息进行失真处理。本章主要讨论限失真信源编码,从分析失真函数、信息率失真函数出发,给出限失真编码定理。最后在结束这两章信源编码理论之前简单介绍了一些其它常用的信源编码方法。

4.1 平均失真和信息率失真函数

在实际问题中,信号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。要规定失真限度,必须先有一个定量的失真测度,为此引入失真函数。

4.1.1 失真函数

假如某一信源 X 输出一个随机变量 $X = \{x_1, x_2, \dots, x_n\}$ 经信源编码后输出 $Y = \{y_1, y_2, \dots, y_m\}$ 。如果

$$x_i = y_j \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m \quad (4.1.1)$$

则认为没有失真。如果 $x_i \neq y_j$, 就产生了失真。失真的大小,用一个量来表示,即失真函数 $d(x_i, y_j)$, 以衡量用 y_j 代替 x_i 所引起的失真程度。一般失真函数定义为

$$d(x_i, y_j) = \begin{cases} 0 & x_i = y_j \\ a & a > 0 \quad x_i \neq y_j \end{cases} \quad (4.1.2)$$

将所有的 $d(x_i, y_j)$, $i = 1, 2, \dots, n; j = 1, 2, \dots, m$ 排列起来,用矩阵表示为

$$d = \begin{pmatrix} d(x_1, y_1) & \dots & d(x_1, y_m) \\ \dots & \dots & \dots \\ d(x_n, y_1) & \dots & d(x_n, y_m) \end{pmatrix} \quad (4.1.3)$$

称 d 为失真矩阵。

例 4.1.1 设信源符号 $X = \{0, 1\}$, 编码器输出符号 $Y = \{0, 1, 2\}$, 规定失真函数为

$$d(0, 0) = d(1, 1) = 0$$

$$d(0, 1) = d(1, 0) = 1$$

$$d(0, 2) = d(1, 2) = 0.5$$

求失真矩阵 d 。

解: 由(4.1.3)式得失真矩阵

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

失真函数 $d(x_i, y_j)$ 的函数形式可以根据需要任意选取, 例如平方代价函数、绝对代价函数、均匀代价函数等等。最常用的失真函数有:

$$\begin{aligned} \text{均方失真:} & \quad d(x_i, y_j) = (x_i - y_j)^2 \\ \text{绝对失真:} & \quad d(x_i, y_j) = |x_i - y_j| \\ \text{相对失真:} & \quad d(x_i, y_j) = |x_i - y_j| / |x_i| \end{aligned}$$

$$\text{误码失真:} \quad d(x_i, y_j) = \begin{cases} 0, & x_i = y_j \\ 1, & \text{其他} \end{cases}$$

前三种失真函数适用于连续信源, 后一种适用于离散信源。均方失真和绝对失真只与 $(x_i - y_j)$ 有关, 而不是分别与 x_i 及 y_j 有关, 在数学处理上比较方便; 相对失真与主观特性比较匹配, 因为主观感觉往往与客观量的对数成正比, 但在数学处理中就要困难得多。其实选择一个合适的失真函数, 要完全与主观特性匹配已是非常困难的, 更不用说还要易于数学处理。当然不同的信源应有不同的失真函数, 所以在实际问题中还可提出许多其他形式的失真函数。

失真函数的定义可以推广到序列编码情况, 如果假定离散信源输出符号序列 $X = \{x_1, x_2, \dots, x_i, \dots, x_n\}$, 其中 L 长符号序列 $x_i = [x_{i1}, x_{i2}, \dots, x_{iL}]$, 经信源编码后, 输出符号序列 $Y = \{y_1, y_2, \dots, y_j, \dots, y_m\}$, 其中 L 长符号序列 $y_j = [y_{j1}, y_{j2}, \dots, y_{jL}]$, 则失真函数定义为

$$d_L(x_i, y_j) = \frac{1}{L} \sum_{k=1}^L d(x_{ik}, y_{jk}) \quad (4.1.4)$$

式中 $d(x_{ik}, y_{jk})$ 是信源输出第 i 个 L 长符号序列 x_i 中的第 k 个符号 x_{ik} , 编码输出第 j 个 L 长符号序列 y_j 中的第 k 个符号 y_{jk} 的失真函数。

4.1.2 平均失真

由于 x_i 和 y_j 都是随机变量, 所以失真函数 $d(x_i, y_j)$ 也是随机变量, 限失真时的失真值, 只能用它的数学期望或统计平均值, 因此将失真函数的数学期望称为平均失真, 记为

$$\overline{D} = \sum_{i=1}^n \sum_{j=1}^m p(x_i, y_j) d(x_i, y_j) = \sum_{i=1}^n p(x_i) \sum_{j=1}^m p(y_j/x_i) d(x_i, y_j) \quad (4.1.5)$$

式中, $p(x_i, y_j)$, $i = 1, 2, \dots, n$, $j = 1, 2, \dots, m$ 是联合分布; $p(x_i)$ 是信源符号概率分布; $p(y_j/x_i)$, $i = 1, 2, \dots, n$, $j = 1, 2, \dots, m$ 是转移概率分布; $d(x_i, y_j)$, $i = 1, 2, \dots, n$, $j = 1, 2, \dots, m$ 是离散随机变量的失真函数; 平均失真 \overline{D} 是对给定信源分布 $p(x_i)$ 在转移概率分布为 $p(y_j/x_i)$ 的信源编码器时的失真的总体量度。

对于连续随机变量同样可以定义平均失真为

$$\overline{D} = \int \int p_{xy}(x, y) d(x, y) dx dy \quad (4.1.6)$$

式中, $p_{xy}(x, y)$ 是连续随机变量的联合概率密度; $d(x, y)$ 是连续随机变量的失真函数。

对于序列编码情况, 平均失真为

$$\overline{D}_L = \frac{1}{L} \sum_{k=1}^L E[d(x_{ik}, y_{jk})] = \frac{1}{L} \sum_{k=1}^L \overline{D}_k \quad (4.1.7)$$

式中 \overline{D}_k 是第 k 个符号的平均失真。

4.1.3 信息率失真函数 $R(D)$

如果信源输出的信息率大于信道的传输能力,就必须对信源进行压缩,使其压缩后的信息传输率小于信道传输能力,但同时要保证压缩所引入的失真不超过预先规定的限度 D 。因此信息压缩问题就是对于给定的信源,在满足平均失真

$$\overline{D} \leq D \quad (4.1.8)$$

的前提下,使编码后的信息率尽可能小。我们将有失真的信源编码器视作有干扰的信道,用分析信道传输的方法来研究限失真编码问题。满足(4.1.8)式条件的所有转移概率分布 p_{ij} 构成了一类假想的信道,称为 D 允许信道(也称为 D 允许的试验信道)。记为

$$P_D = \{ p(y/x) \mid \overline{D} \leq D \} \quad (4.1.9)$$

对于离散无记忆信道,相应地有

$$P_D = \{ p(y/x_i) \mid \overline{D} \leq D \quad i=1,2,\dots,n; j=1,2,\dots,m \}$$

在上述允许信道 P_D 中,可以寻找一个信道 $p(y/x)$,使给定的信源经过此信道传输后,互信息 $I(X;Y)$ 达到最小。该最小的互信息就称为信息率失真函数 $R(D)$,即

$$R(D) = \min_{P_D} I(X;Y) \quad (4.1.10)$$

由上式可知,当失真限度一定时,就要对 p_{ij} 加以限制。凡是满足(4.1.8)式的那些 p_{ij} 的集合称为 P_D 。在 P_D 中某一组 p_{ij} 使 $I(X;Y)$ 最小就是 $R(D)$ 。对于离散无记忆信源, $R(D)$ 函数可写成

$$R(D) = \min_{p_{ij}} \sum_{i=1}^n \sum_{j=1}^m p(x_i) p(y_j/x_i) \log \frac{p(y_j/x_i)}{p(y_j)} \quad (4.1.11)$$

其中 $p(x_i)$, $i=1,2,\dots,n$ 是信源符号概率分布; $p(y_j/x_i)$, $i=1,2,\dots,n, j=1,2,\dots,m$ 是转移概率分布; $p(y_j)$, $j=1,2,\dots,m$ 是接收端收到符号的概率分布。

例 4.1.2 已知编码器输入的概率分布为 $p(x) = \{0.5, 0.5\}$, 信道转移矩阵分别为

$$p_{ij} = \begin{bmatrix} 0.6 & 0.4 \\ 0.2 & 0.8 \end{bmatrix}, \quad p_{ij} = \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix}$$

求互信息。

解: 因为 $p(x_i y_j) = p(x_i) p(y_j/x_i)$, 用 p_{ij} 代入得

$$p(x_1 y_1) = 0.3, p(x_1 y_2) = 0.2, p(x_2 y_1) = 0.1, p(x_2 y_2) = 0.4$$

因为 $p(y_j) = \sum_i p(x_i y_j)$, 所以

$$p(y_1) = 0.4, p(y_2) = 0.6$$

又因为 $p(x_i/y_j) = \frac{p(x_i y_j)}{p(y_j)}$, 所以

$$p(x_1/y_1) = \frac{3}{4}, p(x_1/y_2) = \frac{1}{3}, p(x_2/y_1) = \frac{1}{4}, p(x_2/y_2) = \frac{2}{3}$$

根据互信息公式代入可得

$$I(X;Y) = \sum_{i,j} p(x_i y_j) \log_2 \frac{p(x_i/y_j)}{p(x_i)} = 0.125 \text{ 比特/符号}$$

用 p_{ij} 代入进行同样的运算可得

$$I(X;Y) = 0.397 \text{ 比特/符号}$$

可见,当 $p(x)$ 一定时, $I(X; Y)$ 随 p_{ij} 而变。从物理概念上看,这是明显的。因为 $p(x)$ 分布一定时,信道受干扰不同所能传递的信息量也是不同的。可以证明,当 $p(x)$ 一定时, $I(X; Y)$ 是关于 p_{ij} 的下凸函数。因此当改变 p_{ij} 时, $I(p_{ij})$ 有一极小值。

由互信息的表达式

$$I(X; Y) = H(Y) - H(Y/X) = H(X) - H(X/Y)$$

可理解为信源发出的信息量 $H(X)$ 与在噪声干扰条件下消失的信息量之差。应当注意,在这里讨论的是有关信源问题,一般不考虑噪声的影响。而是由于信息的存储和传输时需要去掉冗余,或者从某些需要出发认为可将一些次要成分去掉。也就是说,对信源的原始信息在允许的失真限度内进行了压缩。由于这种压缩损失了一定的信息,造成一定的失真。把这种失真等效成由噪声而造成的信息损失,看成一个等效噪声信道(又称为试验信道),因此信息率失真函数的物理意义是:对于给定信源,在平均失真不超过失真限度 D 的条件下,信息率容许压缩的最小值。此时的信道转移概率 p_{ij} 实际上指的是一种限失真信源编码方法。下面通过对一个信源处理的例子,进一步研究信息率失真函数的物理意义。

例 4 1 3 设信源的符号表为 $A = \{a_1, a_2, \dots, a_{2n}\}$, 概率分布为 $p(a_i) = 1/2n, i = 1, 2, \dots, 2n$, 失真函数规定为

$$d(a_i, a_j) = \begin{matrix} 1 & i \neq j \\ 0 & i = j \end{matrix}$$

即符号不发生差错时失真为 0,一旦出错,失真为 1,试研究在一定编码条件下信息压缩的程度。

由信源概率分布可求出信源熵为

$$H = \frac{1}{2n} \dots \frac{1}{2n} = \log_2 2n \text{ 比特/符号}$$

如果对信源进行不失真编码,平均每个符号至少需要 $\log_2 2n$ 个二进制码元。现在假定允许有一定失真,假设失真限度为 $D = 1/2$ 。也就是说,当收到 100 个符号时,允许其中有 50 个以下的差错。这时信源的信息率能减少到多少呢? 每个符号平均码长能压缩到什么程度呢? 设想采用下面的编码方案:

$$\begin{matrix} a_1 & a_1, a_2 & a_2, \dots, a_n & a_n, \\ a_{n+1} & a_n, a_{n+2} & a_n, \dots, a_{2n} & a_n \end{matrix}$$

用信道模型图表示,如图 4 1 1 所示。

按照上述关于失真函数的规定,求得平均失真 D 为

$$\begin{aligned} D &= \frac{1}{2n} \sum_{i=1}^{2n} p(a_i) p(a_j/a_i) d(a_i, a_j) \\ &= \frac{1}{2n} \sum_{i=n+1}^{2n} p(a_i) p(a_j/a_i) d(a_i, a_j) \\ &= \frac{1}{2} \end{aligned}$$

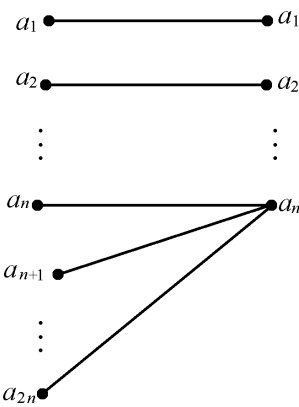


图 4 1 1 等效试验信道

由于上述编码相当于图 4 1 1 所示试验信道。由该信道模型不难看出,它是一个确定信道,所以

$$p_{ij} = 1(\text{或 } 0), H(Y/X) = 0$$

由互信息公式可得

$$I(X; Y) = H(Y) - H(Y/X) = H(Y)$$

信道输出概率分布为

$$p_1 = p_2 = \dots = p_{n-1} = \frac{1}{2n}$$

由于从 a_n 起,以后所有符号都编成 a_n ,所以概率分布为

$$p_n = \frac{1+n}{2n}$$

则输出熵 $H(Y)$ 为

$$H(Y) = H \underbrace{\frac{1}{2n} \dots \frac{1}{2n}}_{n-1 \text{ 个}} \cdot \frac{1+n}{2n} = \log 2n - \frac{n+1}{2n} \log(n+1)$$

由以上结果可知,经压缩编码以后,信源需要传输的信息率由原来的 $\log 2n$,压缩到 $\log 2n - [(n+1)/2n] \log(n+1)$ 。也就是说,信息率压缩了 $[(n+1)/2n] \log(n+1)$ 。这是采用了上述压缩编码方法的结果,所付出的代价是容忍了 $1/2$ 的平均失真。如果选取压缩更为有利的编码方案,压缩的效果可能更好。但一旦达到最小互信息这个极限值,就是 $R(D)$ 的数值(此处 $D=1/2$)。如果超过这个极限值,那么,失真就要超过失真限度。如果需要压缩的信息率更大,则可容忍的平均失真就要更大。

4.1.4 信息率失真函数的性质

1. $R(D)$ 函数的定义域

由于 D 是非负实数 $d(x, y)$ 的数学期望,因此 D 也是非负的实数,非负实数的下界是零。对应于无失真情况,相当于无噪声信道,此时信道传输的信息量等于信源熵,即

$$R(D) = R(0) = H(X) \quad (4.1.12)$$

对于连续信源来说,由于其信源熵只有相对意义,而真正的熵为 $H_c(x)$,当 $D=0$ 时相当于严格无噪声信道,通过无噪声信道的熵是不变的,所以

$$R(D) = R(0) = H_c(x) =$$

因为实际信道总是有干扰的,其容量有限,要无失真地传送这种连续信息是不可能的。当允许有一定失真时, $R(D)$ 将为有限值,传送才是可能的。

由于 $I(X; Y)$ 是非负函数,而 $R(D)$ 是在约束条件下的 $I(X; Y)$ 的最小值,所以 $R(D)$ 也是一个非负函数,它的下限值是零。取满足 $R(D)=0$ 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。因此可以得到 $R(D)$ 的定义域为 $D \in [0, D_{\max}]$ 。

D_{\max} 是这样来计算的。 $R(D)=0$ 就是 $I(X; Y)=0$,这时试验信道输入与输出是互相独立的,所以条件概率 $p(y_j/x_i)$ 与 x_i 无关。即

$$p_{ij} = p(y_j/x_i) = p(y_j) = p_j$$

这时平均失真为

$$D = \sum_{i=1}^n \sum_{j=1}^m p_i p_j d_{ij} \quad (4.1.13)$$

式中 $d_{ij} = d(x_i, y_j)$ 。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

$$D_{\max} = \min_{j=1}^m \sum_{i=1}^n p_j p_i d_{ij}$$

从上式观察可得:在 $j = 1, \dots, m$ 中,可找到 $\min_{i=1}^n p_i d_{ij}$ 值最小的 j , 当该 j 对应的 $p_j = 1$, 而其余 p_j 为零时, 上式右边达到最小, 可简化成

$$D_{\max} = \min_{j=1, 2, \dots, m} \sum_{i=1}^n p_i d_{ij} \quad (4.1.14)$$

例 4.1.4 设输入输出符号表为 $X = Y = \{0, 1\}$, 输入概率分布 $p(x) = \{1/3, 2/3\}$, 失真矩阵为

$$d = \begin{pmatrix} d(x_1, y_1) & d(x_1, y_2) \\ d(x_2, y_1) & d(x_2, y_2) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

求 D_{\max} 。

解: 由式(4.1.14)得

$$\begin{aligned} D_{\max} &= \min_{j=1, 2} \sum_{i=1}^2 p_i d_{ij} \\ &= \min_{j=1, 2} (p_1 d_{1j} + p_2 d_{2j}) \\ &= \min_{j=1, 2} \left(\frac{1}{3} \times 0 + \frac{2}{3} \times 1, \frac{1}{3} \times 1 + \frac{2}{3} \times 0 \right) \\ &= \min_{j=1, 2} \left(\frac{2}{3}, \frac{1}{3} \right) = \frac{1}{3} \end{aligned}$$

此时输出符号概率: $p(y_1) = 0, p(y_2) = 1$ 。

例 4.1.5 若输入输出符号表与输入概率分布同例 4.1.4, 失真矩阵为 $d = \begin{pmatrix} 1/2 & 1 \\ 2 & 1 \end{pmatrix}$

试求 D_{\max} 。

解: 由(4.1.14)式可得

$$\begin{aligned} D_{\max} &= \min_{j=1, 2} \sum_{i=1}^2 p_i d_{ij} \\ &= \min_{j=1, 2} \left(\frac{1}{3} \times \frac{1}{2} + \frac{2}{3} \times 2, \frac{1}{3} \times 1 + \frac{2}{3} \times 1 \right) \\ &= \min_{j=1, 2} \left(\frac{5}{3}, 1 \right) = 1 \end{aligned}$$

2. $R(D)$ 函数的下凸性和连续性

规定了定义域之后,再证明 $R(D)$ 在定义域内是下凸的。

令 $D = \lambda D_1 + (1 - \lambda) D_2, 0 \leq \lambda \leq 1$

$$R(D) = \min_{p_{ij} \in P_D} I(p_{ij}) = I(p_{ij})$$

其中 p_{ij} 是使 $I(p_{ij})$ 达到极小值的 p_{ij} , 且保证 $D = \lambda D_1 + (1 - \lambda) D_2$ 。同理:

$$R(D_1) = I(p_{ij_1})$$

令

$$p_{ij} = \lambda p_{ij_1} + (1 - \lambda) p_{ij_2}$$

先证明 p_{ij} 是 p_D 的元。已知

$$\begin{aligned}
D(p_{ij}) &= \sum_{i,j} p_i p_{ij} d_{ij} \\
&= \sum_{i,j} p_i [p_{ij} + (1 - \alpha) p_{ij}] d_{ij} \\
&= \sum_{i,j} p_j p_{ij} d_{ij} + (1 - \alpha) \sum_{i,j} p_i p_{ij} d_{ij} \\
&= D + (1 - \alpha) D = D
\end{aligned}$$

这是因为 p_{ij} 和 p_{ij} 分别是 P_D 和 P_D 中的元, 所以造成的失真必小于 D 和 D 。

利用 $I(p_{ij})$ 的下凸性, 可得

$$\begin{aligned}
R(D) &= \min_{p_{ij} \in P_D} I(p_{ij}) - I(p_{ij}) \\
&= I[p_{ij} + (1 - \alpha) p_{ij}] - I(p_{ij}) + (1 - \alpha) I(p_{ij}) \\
&= R(D) + (1 - \alpha) R(D)
\end{aligned}$$

这就证明了 $R(D)$ 的下凸性。

现在来证明 $R(D)$ 在定义域 $0 \sim D_{\max}$ 之间的连续性。

设 $D = D + \alpha$, 当 $\alpha \rightarrow 0$ 时, $P_D \rightarrow P_D$ 。由于 $I(p_{ij})$ 是 p_{ij} 的连续函数, 即当 $p_{ij} \rightarrow 0$, 有

$$I(p_{ij} + \alpha p_{ij}) \rightarrow I(p_{ij})$$

则

$$R(D) = \min_{p_{ij} \in P_D} I(p_{ij}) - \min_{p_{ij} \in P_D} I(p_{ij}) = R(D)$$

这就是连续性。

3. $R(D)$ 函数的单调递减性

$R(D)$ 的单调递减性可以作如下理解: 容许的失真度越大, 所要求的信息率越小。反之亦然。这一点可以由定义来证明。令

$$D > D$$

则

$$P_D \supset P_D$$

这一结果可以从(4.1.9)式 P_D 的定义式中得到。于是

$$R(D) = \min_{p_{ij} \in P_D} I(p_{ij}) - \min_{p_{ij} \in P_D} I(p_{ij}) = R(D)$$

上式中的不等式是因为 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, 所以 $R(D)$ 是非递增的函数。现在再证明上式中的等号不成立, 用反证法。

设有 $0 < D < D < D_{\max}$, 令

$$R(D) = I(p_{ij}) - p_{ij} \in P_D$$

$$R(D_{\max}) = I(p_{ij}) = 0 \quad p_{ij} \in P_{D_{\max}}$$

对于足够小的 α ($\alpha > 0$), 必有

$$D < (1 - \alpha) D + \alpha D_{\max} = D < D$$

令

$$p_{ij} = (1 - \alpha) p_{ij} + \alpha p_{ij}$$

则

$$\begin{aligned}
D(p_{ij}) &= (1 - \alpha) D(p_{ij}) + \alpha D(p_{ij}) \\
&= (1 - \alpha) D(p_{ij}) + \alpha D_{\max} = D
\end{aligned}$$

所以

$$p_{ij} \in P_D$$

$$\begin{aligned}
R(D) &= \min_{p_{ij} \in P_D} I(p_{ij}) - I(p_{ij}) - (1 - \alpha) I(p_{ij}) + \alpha I(p_{ij}) \\
&= (1 - \alpha) I(p_{ij}) < R(D)
\end{aligned}$$

可见 $R(D) \geq R(D)$ 。因此 $R(D)$ 是严格单调递减的。

综上所述,可以得出如下结论:

$R(D)$ 是非负的实数,即 $R(D) \geq 0$ 。其定义域为 $0 \sim D_{\max}$, 其值为 $0 \sim H(X)$ 。当 $D > D_{\max}$ 时, $R(D) = 0$ 。

$R(D)$ 是关于 D 的下凸函数,因而也是关于 D 的连续函数。

$R(D)$ 是关于 D 的严格递减函数。

由以上三点结论,对一般 $R(D)$ 曲线的形态可以画出来了,如图 4 1 2 所示。

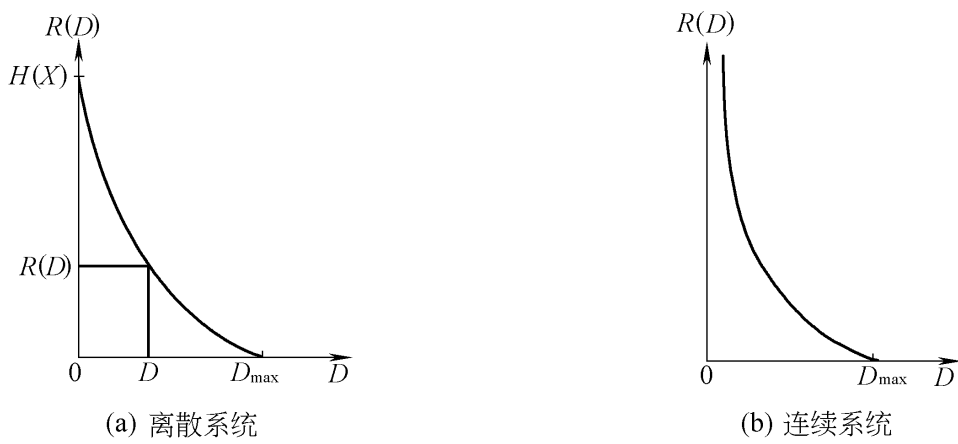


图 4 1 2 信息率失真曲线

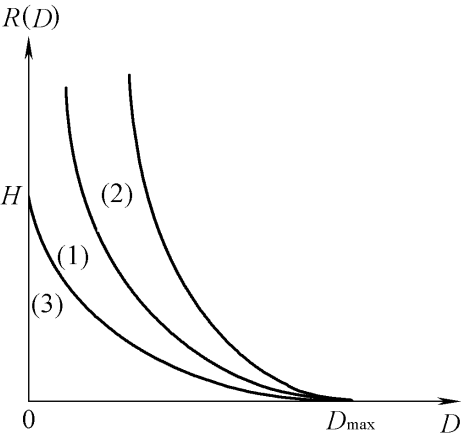
由上可知,当规定了允许失真 D , 又找到了适当的失真函数 d_{ij} , 就可以找到该失真条件下的最小信息率 $R(D)$, 这个最小信息率是一个极限数值。用不同方法进行数据压缩时(前提是都不能超过失真限度 D), 其压缩的程度如何, $R(D)$ 函数是一把尺子。由它可知是否还有压缩潜力, 潜力有多大。因此近年来引起很多学者对它的兴趣。

4.2 离散信源和连续信源的 $R(D)$ 计算

已给定信源概率 p_i 和失真函数 d_{ij} , 就可以求得该信源的 $R(D)$ 函数。它是在约束条件, 即保真度准则下, 求极小值的问题。但要得到它的显式表达式, 一般比较困难, 通常用参量表达式。即使如此, 除简单的情况外, 实际计算还是困难的, 只能用迭代逐级逼近的方法。

某些特殊情况下 $R(D)$ 的表示式为:

(1) 当 $d(x, y) = (x - y)^2$, $p(x) = \frac{1}{2 \exp - \frac{x^2}{2}}$ 时, $R(D) = \log \frac{1}{D}$



(2) 当 $d(x, y) = |x - y|$, $p(x) = \frac{1}{2} e^{-|x|}$ 时,

$$R(D) = \log \frac{1}{D}$$

(3) 当 $d(x, y) = (x, y)$, $p(x = 0) = p$, $p(x = 1) = 1 - p$ 时, $R(D) = H(p) - H(D)$

这些 $R(D)$ 可画成如图 4 2 1 中的三条曲线。它们都有一最大失真值 D_{\max} , 对应 $R(D) = 0$ 。当允许的平均失真 D 大于这最大值时, $R(D)$ 当然也是零, 也就是不用传送信息已能达到要求。上述三种情况的 D_{\max} 分

图 4 2 1 率失真函数 $R(D)$

别为 $1/2$, $1/2$ 和 p (若 $p < 1/2$, 不然就是 $1 - p$)。其实这是很好解释的。例如在均方失真和正态分布的第一种情况下, 不管信源符号是何值都用 $y = 0$ 来编码, 此时平均失真就是 $1/2$ 。
 Y 只有一个值, 当然不需要传送, 也不含有信息。其他两种情况也有类似的结果。当 $D < D_{\max}$ 时, $R(D)$ 就已不是零, 随着 D 的减小, $R(D)$ 单调地增加; 当 $D = 0$, 前两种情况下, $R(D)$ 趋于无限, 这就是说, 大于信息量无限大的连续信源符号, 无法进行无损编码, 除非信息率 R 趋向无限大。对于离散信源就不同, 在第三种情况下, $D = 0$ 时, $R(0) = H(p)$, 这就是无损编码时, 所需的信息率不能小于信源的符号熵。

下面将简单介绍参量表达式方法求解率失真函数 $R(D)$ 。具体推导过程从略(参见文献 1), 这里结合例子给出计算步骤。

例 4 2 1 设输入输出符号表为 $X = Y = \{0, 1\}$, 输入概率分布 $p(x) = \{p, 1 - p\}$, $0 < p < 1/2$, 失真矩阵为

$$d = \begin{pmatrix} d(x_1, y_1) & d(x_1, y_2) \\ d(x_2, y_1) & d(x_2, y_2) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

求信息率失真函数 $R(D)$ 。

解: 简记 $x_i = (x_i)$, $p_i = p(x_i)$, $y_j = p(y_j)$, $s = e^s$, $i, j = 1, 2$

(1) 按下式解方程

$$\sum_i (x_i) p(x_i) \exp[sd(x_i, y_j)] = 1, \quad j = 1, \dots, m$$

写成矩阵形式

$$\begin{pmatrix} p_1 & p_2 \end{pmatrix} \begin{pmatrix} 1 & s \\ s & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \end{pmatrix}$$

由此解得

$$p_1 = p_2 = \frac{1}{1 + s}, \quad s = \frac{1}{p(1 + s)}, \quad s = \frac{1}{(1 - p)(1 + s)}$$

(2) 按下式解方程

$$\sum_j p(y_j) \exp[sd(x_i, y_j)] = \frac{1}{(x_i)}, \quad i = 1, \dots, n$$

写成矩阵形式

$$\begin{pmatrix} 1 & s \\ s & 1 \end{pmatrix} \begin{pmatrix} 1 & s \\ s & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

解得

$$s = \frac{1}{1 - p^2} - \frac{1}{1 - p} = \frac{1}{1 - p} [p - (1 - p)]$$

$$s = \frac{1}{1 - p^2} - \frac{1}{1 - p} = \frac{1}{1 - p} (1 - p - p)$$

(3) 按下式得转移概率分布 p_{ij}

$$p_{ij} = (x_i) p(y_j) \exp[sd(x_i, y_j)], \quad i = 1, \dots, n; \quad j = 1, \dots, m$$

写成矩阵形式

$$P = \frac{1}{1 - \frac{p - (1 - p)}{1 - p}} = \frac{1 - p - p}{1 - p} = \frac{1 - 2p}{1 - p}$$

(4) 求 s ($s = \log$)

$$D = \sum_{ij} p_i p_{ij} d_{ij} = p_1 p_{11} d_{11} + p_1 p_{12} d_{12} + p_2 p_{21} d_{21} + p_2 p_{22} d_{22}$$

$$= \frac{1}{1 - \frac{p - (1 - p)}{1 - p}} [(1 - p - p) + (p - (1 - p))] = \frac{1}{1 + \frac{p - (1 - p)}{1 - p}}$$

$$D = \frac{1}{1 + \frac{p - (1 - p)}{1 - p}}, \quad \frac{D}{1 - D} = \frac{p}{1 - p}$$

$$s = \log D = \log D - \log(1 - D)$$

(5) 计算 $R(D)$, 将上面各式代入, 则有

$$R(D) = sD + \sum_i p_i \log \frac{1}{p_i}$$

$$= D \log \frac{D}{1 - D} + p \log \frac{1}{p(1 + \frac{p - (1 - p)}{1 - p})} + (1 - p) \log \frac{1}{(1 - p)(1 + \frac{p - (1 - p)}{1 - p})}$$

$$= D \log \frac{D}{1 - D} + H(p) - \log(1 + \frac{p - (1 - p)}{1 - p})$$

$$= D \log \frac{D}{1 - D} - \log \frac{1}{1 - D} + H(p)$$

$$= D \log D + (1 - D) \log(1 - D) + H(p)$$

结果得到如图 4 2 2 所示的曲线:

$$R(D) = \begin{cases} H(p) - H(D), & 0 \leq D \leq p \\ 0, & D > p \end{cases}$$

上述计算过程实质上第(1), (2)步是解简单的线性方程组, 第(3), (4), (5)步则是代入整理。

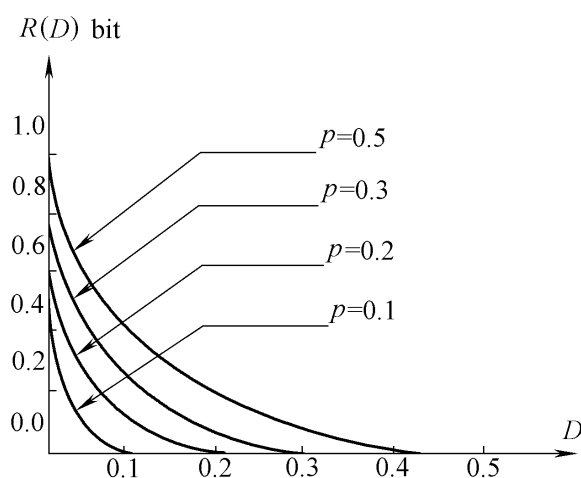


图 4 2 2 $R(D) = H(p) - H(D)$, p 为参数

4.3 限失真信源编码定理

信息率失真函数给出了失真小于 D 时所必须具有的最小信息率 $R(D)$; 只要信息率大

于 $R(D)$, 一定可以找到一种码, 以使译码后的失真小于 D 。

限失真信源编码定理: 设离散无记忆信源 X 的信息率失真函数 $R(D)$, 则当信息率 $R > R(D)$, 只要信源序列长度 L 足够长, 一定存在一种编码方法, 其译码失真小于或等于 $D + \epsilon$, ϵ 为任意小的正数; 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 。

如果是二元信源, 对于任意小的 $\epsilon > 0$, 每一个信源符号的平均码长满足如下公式

$$R(D) - \epsilon < \bar{K} < R(D) + \epsilon$$

该定理指出, 在失真限度内使信息率任意接近 $R(D)$ 的编码方法存在。然而, 要使信息率小于 $R(D)$, 平均失真一定会超过失真限度 D 。

对于连续平稳无记忆信源, 虽然无法进行无失真编码, 在限失真情况下, 有与该定理一样的编码定理。

该定理只能说明最佳编码是存在的, 而具体构造编码方法却一无所知, 因而就不能像无损编码那样从证明过程中引出概率匹配的编码方法。一般只能从优化的思路去求最佳编码。实际上, 迄今尚无合适的可实现的编码方法来接近 $R(D)$ 这个界。

4.4 常用信源编码方法简介

前面已经介绍了信源编码的两大定理, 实用的编码方法需要根据信源的具体特点。在编码理论指导下, 先后出现了许多性能优良的编码方法, 在此简要介绍一部分编码方法的基本原理。

哈夫曼码在实际中已有所应用, 但它仍存在一些分组码所具有的缺点。例如概率特性必须精确地测定, 它若略有变化, 还需更换码表; 对于二元信源, 常需多个符号合起来编码, 才能取得好的效果, 但当合并的符号数不大时, 编码效率提高不多, 尤其对于相关信源, 不能令人满意, 而合并的符号数增大时, 码表中的码字数很多, 设备将越来越复杂。因此在实用中常需作一些改进, 同时也就有研究非分组码的必要性。

4.4.1 游程编码

在二元序列中, 只有两种符号, 即“0”和“1”, 这些符号可连续出现, 连“0”这一段称为“0”游程, 连“1”这一段称为“1”游程。它们的长度分别称为游程长度 $L(0)$ 和 $L(1)$ 。“0”游程和“1”游程总是交替出现的。如果规定二元序列是以“0”开始, 第一个游程是“0”游程, 第二个必为“1”游程, 第三个又是“0”游程等等。对于随机的二元序列, 各游程长度将是随机变量, 其取值可为 $1, 2, 3, \dots$, 直到无限。将任何二元序列变换成游程长度序列, 这种变换是一一对应的, 也就是可逆的。例如有一个二元序列: 000101110010001... 可变换成游程序列: 3113213...

若已知二元序列是以“0”起始的, 从上面的游程序列很容易恢复成原来的二元序列, 包括最后一个“1”, 因为长度为3的“0”游程之后必定是“1”。游程序列已是多元序列, 各长度就可按哈夫曼编码或其他方法处理以达到压缩码率的目的。这种从二元序列转换成多元序列的方法, 在实现时比前面的并元法简单。因为游程长度的计数比较容易, 得到游程长度后就可从码表中找出码字输出, 同时去数下一个游程长度。此外, 在减弱原有序列的符号间的

相关性方面,采用游程变换一般也比并元法更有效。当然,要对二元序列进行哈夫曼编码时,应先测定“0”游程长度和“1”游程长度的概率分布,或由二元序列的概率特性去计算各种游程长度的概率。

对于多元序列也存在相应的游程序列。例如 m 元序列中,可有 m 种游程。连着出现符号 a_r 的游程,其长度 $L(r)$ 就是“ r ”游程长度。这也是一个随机变量。用 $L(r)$ 也可构成游程序列。但是这种变换必须再加一些符号,才能成为一一对应或可逆的,与二元序列变换所得的游程序列不同,这里每个“ r ”游程的前面和后面出现什么符号是不确定的,除 r 外的任何符号都是可能的,因此这一游程之后是何种符号的游程就无法确定,除非插入一个标志说明后一游程的类别。所以把多元序列变换成游程序列再进行压缩编码是没有多大意义的,因为上述的附加标志可能抵销压缩编码所得的好处,对原来的多元序列直接编码,或许会更有效一些。

游程编码仍是变长码,有其固有的缺点,即需有大量的缓冲和优质的信道。此外,由于游程长度可从 1 直到无限,这在码字的选择和码表的建立方面都有困难,实际应用时尚需采取某些措施来改进。

一般情况下,游程长度越大,其概率越小;这在以前的计算中也可看到,而且将随长度的增大渐趋向零。对于小概率的码字,其长度未达到概率匹配或较长,损失不会太大,也就是对平均码字长度影响较小。这样就可对长游程不严格按哈夫曼码步骤进行;在实际应用时,常采用截断处理的方法。

游程编码只适用于二元序列,对于多元信源,一般不能直接利用游程编码;但在下面介绍的冗余位编码,也可认为是游程编码在多元信源的一种应用。

在许多信源序列中,常有不少符号不携带信息,除了它的数目或所占时长外,完全可以不传送。例如在电话通信中,讲话时常有间隙,如字句间的停顿,听对方讲话而静默;又如图象信源中,背景基本上不变,并在图象中占相当大一部分,而其值为常量相当于平均亮度,一般也可以不传送;在数据信源序列中,信息包间的间歇或某种固定模式,也属于冗余性质。这些符号可称为冗余位,若能删除它们,可得较大的压缩比。

设有多元信源序列

$$x_1, x_2, \dots, x_{m1}, y, y, \dots, y, x_{m1+1}, x_{m1+2}, \dots, x_{m2}, y, y, \dots \quad (4.4.1)$$

其中 x 是含有信息的代码,取值于 m 元符号集 A ,可称为信息位; y 是冗余位,它们可为全零,即使未曾传送在收端也能恢复。这样的序列可用下列两个序列来代替:

$$111, \dots, 100, \dots, 000111, \dots, 111000 \text{ 和 } x_1, x_2, \dots, x_{m1}, x_{m1+1}, x_{m1+2}, \dots, x_{m2}, \dots \quad (4.4.2)$$

前一个序列中,用“1”表示信息位,用“0”表示冗余位;后一个序列是取消冗余位后留下的所有信息位。显然,从(4.4.1)式变换成(4.4.2)式中的两个序列是一一对应的,也就是可逆的。如果把(4.4.2)式中的两个序列传送出去,只要没有差错,在收端就可恢复(4.4.1)式中的多元信源序列。这样就把一个多元序列分解为一个二元序列和一个缩短了多元序列。它们可用不同的方法来编码以利于更有效地压缩码率。

4.4.2 算术编码

以上所讨论的无损编码,都是建立在符号和码字相对应的基础上的,这种编码通常称为

块码或分组码。此时信源符号应是多元的,而且不考虑符号相关性。要用于最常见的二元序列,需采用游程编码、分帧编码或合并符号等方法,转换成多值符号,而这些符号间的相关性也不予考虑。这就使信源编码的匹配原则不能充分满足,编码效率就有所损失。如果要较好地解除相关性,常需在序列中取很长的一段,而这将遇到采用等长码时的那种困难。

为了克服这种局限性,就需要跳出分组码的范畴,研究非分组码的编码方法,算术码即为其中之一。算术编码的基本思路是:从全序列出发,将各信源序列的概率映射到 $[0, 1]$ 区间上,使每个序列对应这区间内的一点,也就是一个二进制的小数。这些点把 $[0, 1]$ 区间分成许多小段,每段的长度等于某一序列的概率。再在段内取一个二进制小数,其长度可与该序列的概率匹配,达到高效率编码的目的。这种方法与香农编码法有些类似,只是它们考虑的信源序列对象不同,算术码中的信源序列长度要长得多。

如果信源符号集为 $A = \{ a_0, a_1, \dots, a_{m-1} \}$, 信源序列 $X = (x_0, x_1, \dots, x_l, \dots, x_{L-1})$, $x_l \in A$, 共有 m^L 种可能序列。由于考虑的是全序列,也许是整页纸上的信息作为一个序列,因而序列长度 L 很大。实用中很难得到对应序列的概率,只能从已知的信源符号概率 $P = \{ p(a_0), p(a_1), \dots, p(a_{m-1}) \} = \{ p_0, p_1, \dots, p_r, \dots, p_{m-1} \}$ 中递推得到。定义各符号的积累概率为

$$P_r = \sum_{i=0}^{r-1} p_i \quad (4.4.3)$$

显然,由上式可得 $P_0 = 0, P_1 = p_0, P_2 = p_0 + p_1$ 等,而且

$$p_r = P_{r+1} - P_r \quad (4.4.4)$$

由于 P_{r+1} 和 P_r 都是小于 1 的正数,可用 $[0, 1]$ 区间内的两个点来表示,则 p_r 就是这两点间的小区间的长度,如图 4.4.1 所示。不同的符号有不同的小区间,它们互不重叠,所以可将这种小区间内的任一个点作为该符号的代码。以后将计算这代码所需的长度,使之能与其概率匹配。

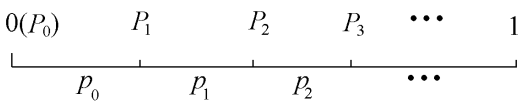


图 4.4.1 信源的符号概率和积累概率

例如有一序列 $S = 011$, 这种三个二元符号的序列可按自然二进数排列, 000, 001, 010,, 则 S 的积累概率为

$$P(S) = p(000) + p(001) + p(010) \quad (4.4.5)$$

如果 S 后面接一个“0”, 积累概率就成为

$$\begin{aligned} P(S0) &= p(0000) + p(0001) + p(0010) + p(0011) + p(0100) + p(0101) \\ &= p(000) + p(001) + p(010) = P(S) \end{aligned}$$

因为两个四元符号的最后一位是“0”和“1”时, 根据归一律, 它们的概率和应等于前 3 位的概率, 即 $p(0000) + p(0001) = p(000)$ 等。

如果 S 后面接一个“1”, 则其积累概率是

$$\begin{aligned} P(S1) &= p(0000) + p(0001) + p(0010) + p(0011) + p(0100) + p(0101) + p(0110) \\ &= P(S) + p(0110) \\ &= P(S) + p(S) p_0 \end{aligned}$$

由于单符号序列的积累概率为 $P_0 = 0, P_1 = p_0$, 所以上面两式可统一写作

$$P(Sr) = P(S) + p(S) P_r, \quad r = 0, 1$$

这样写的式子很容易推广到多元序列 ($m > 2$), 即可得一般的递推公式

$$P(Sa_r) = P(S) + p(S)P_r \quad (4.4.6)$$

以及序列的概率的公式

$$p(Sa_r) = p(S)p_r$$

从以上关于积累概率 $P(S)$ 的计算中可看出, $P(S)$ 把区间 $[0, 1)$ 分割成许多小区间, 每个小区间的长度等于各序列的概率 $p(S)$, 而这小区间内的任一点可用来代表这序列, 现在来讨论如何选择这个点。令

$$L = \left\lceil \log \frac{1}{p(S)} \right\rceil \quad (4.4.7)$$

式中 $\lceil \cdot \rceil$ 代表大于或等于的最小整数。把积累概率 $P(S)$ 写成二进位的小数, 取其前 L 位, 以后如果有尾数, 就进位到第 L 位, 这样得到一个数 C 。例如

$$P(S) = 0.10110001, p(S) = 1/17, \text{ 则 } L = 5, \text{ 得 } C = 0.10111$$

这个 C 就可作为 S 的码字。因为 C 不小于 $P(S)$, 至少等于 $P(S)$ 。又由 (4.4.7) 式可知 $p(S) \geq 2^{-L}$ 。令 $(S+1)$ 为按顺序正好在 S 后面的一个序列, 则

$$P(S+1) = P(S) + p(S) \geq P(S) + 2^{-L} > C$$

当 $P(S)$ 在第 L 位以后没有尾数时, $P(S)$ 就是 C , 上式成立; 如果有尾数时, 这尾数就是上式的左右两侧之差, 所以上式也成立。由此可见 C 必在 $P(S+1)$ 和 $P(S)$ 之间, 也就是在长度为 $p(S)$ 的小区间 (左闭右开的区间) 内, 因而是可以唯一地译码的。这样构成的码字, 编码效率是很高的, 因为已达到概率匹配, 尤其是当序列很长时。由 (4.4.7) 式可见, 对于长序列, $p(S)$ 必然很小, L 与概率倒数的对数已几乎相等; 也就是取整数所造成的差别很小, 平均代码长度将接近 S 的熵值。

实用中, 采用积累概率 $P(S)$ 表示码字 $C(S)$, 符号概率 $p(S)$ 表示状态区间 $A(S)$, 则有

$$\begin{aligned} C(Sr) &= C(S) + A(S)P_r \\ A(Sr) &= A(S)p_r \end{aligned} \quad (4.4.8)$$

对于二进制符号组成的序列, $r = 0, 1$ 。

实际编码过程是这样的: 先置两个存储器 C 和 A , 起始时可令

$$A(\emptyset) = 1, \quad C(\emptyset) = 0$$

其中 \emptyset 代表空集。每输入一个信源符号, 存储器 C 和 A 就按照 (4.4.8) 式更新一次, 直至程序结束, 就可将存储器 C 的内容作为码字输出。由于 $A(S)$ 是递增的, 而这增量随着序列的增长而减小, 因为这增量是序列的概率与信源符号序列的积累概率的乘积; 所以 C 的前面几位一般已固定, 在以后计算中不会被更新, 因而可以输出。只需保留后面几位用作更新。

译码也可逐位进行, 与编码过程相似。

例 4.4.1 有简单的四个符号 a, b, c, d 构成序列 $S = abda$, 各符号及其对应概率如表 4.4.1, 算术编解码过程如下:

设起始状态为空序列 \emptyset , 则 $A(\emptyset) = 1, C(\emptyset) = 0$ 。

表 4 4 1 各符号及其对元概率

符 号	符号概率 p_i	符号累积概率 P_j	符 号	符号概率 p_i	符号累积概率 P_j
a	0.100(1/ 2)	0.000	c	0.001(1/ 8)	0.110
b	0.010(1/ 4)	0.100	d	0.001(1/ 8)	0.111

递推得：

$$C(a) = C() + A() P_a = 0 + 1 \times 0 = 0$$
$$A(a) = A() p_a = 1 \times 0.1 = 0.1$$
$$C(ab) = C(a) + A(a) P_b = 0 + 0.1 \times 0.1 = 0.01$$
$$A(ab) = A(a) p_b = 0.1 \times 0.01 = 0.001$$
$$C(abd) = C(ab) + A(ab) P_d = 0.01 + 0.001 \times 0.111 = 0.010111$$
$$A(abd) = A(ab) p_d = 0.001 \times 0.001 = 0.000001$$
$$C(abda) = C(abd) + A(abd) P_a = 0.010111 + 0.000001 \times 0 = 0.010111$$
$$A(abda) = A(abd) p_a = 0.000001 \times 0.1 = 0.0000001$$

上述编码过程可用下列单位区间的划分来描述：

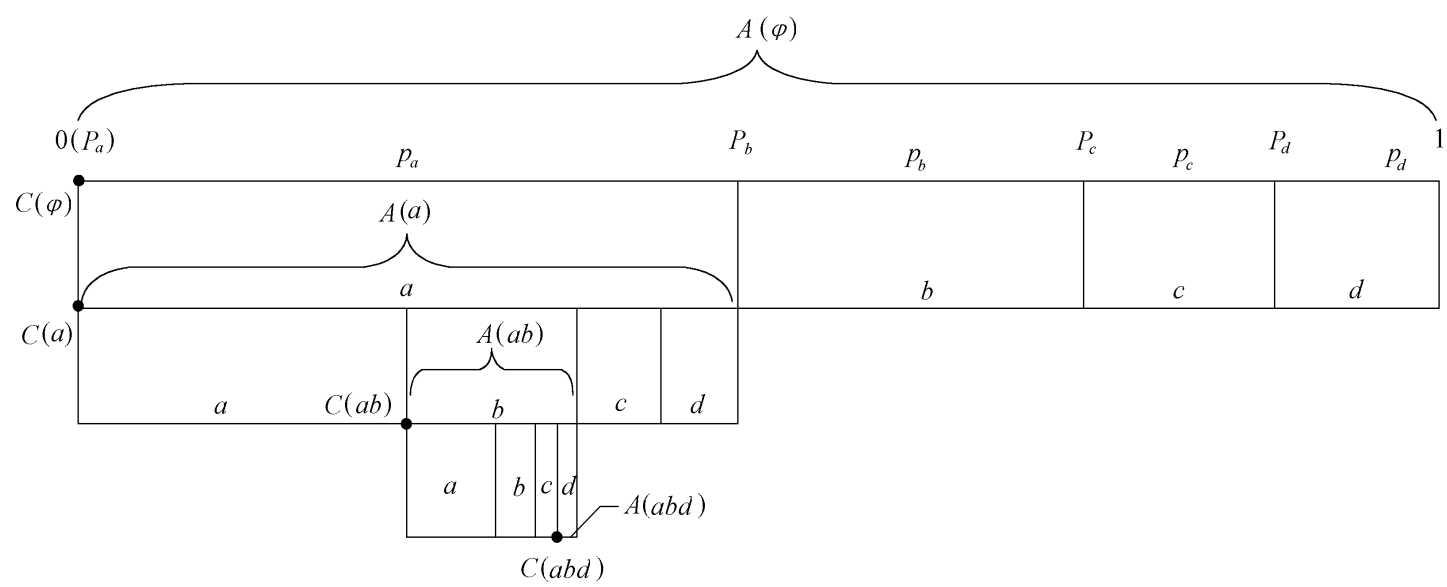


图 4 4 2 算术码编码过程

译码可通过对上述编码后的数值大小的比较来进行,即判断码字 $C(S)$ 落在哪一个区间就可以得出一个相应的符号序列。据递推公式的相反过程译出符号。具体译码顺序是后编的先译,故称为 *LIFO* 算术码,步骤如下：

$$C(abda) = 0.010111 < 0.1 \quad [0, 0.1)$$
$$\text{放大至}[0, 1)(\times P_a^{-1}): C(abda) \times 2^1 = 0.10111 \quad [0.1, 0.110)$$
$$\text{去掉累积概率 } P_b: 0.10111 - 0.1 = 0.00111$$
$$\text{放大至}[0, 1)(\times P_b^{-1}): 0.00111 \times 2^2 = 0.111 \quad [0.111, 1)$$
$$\text{去掉累积概率 } P_d: 0.111 - 0.111 = 0$$
$$\text{放大至}[0, 1)(\times P_d^{-1}): 0 \times 2^4 = 0 \quad [0, 0.1)$$

第一个符号为 a ;

第二个符号为 b ;

第三个符号为 d ;

第四个符号为 a 。

算术编码从性能上看具有许多优点,特别是由于所需的参数很少,不象哈夫曼编码那样需要一个很大的码表,常设计成自适应算术编码来针对一些信源概率未知或非平稳情况。

但是在实际实现时还有一些问题,如计算复杂性、计算的精度以及存储量等,随着这些问题的逐渐解决,算术编码正在进入实用阶段,但要扩大应用范围或进一步提高性能,降低造价,还需进一步改进。

4.4.3 矢量量化

连续信源进行编码的主要方法是量化,即将连续的样值 x 离散化成为 $y_i, i = 1, 2, 3, \dots, n$ 。 n 是量化级数, y_i 是某些实数。这样就把连续值转化为 n 个实数,可用 $0, 1, 2, \dots, n - 1$ 等 n 个数字来表示。离散信源也会涉及量化的问题,比如当提供的量化级数少于原来的量化级数时,也需要对该信源信号进行再次量化。在上述的这些量化中,由于 x 是一个标量,因此称为标量量化。在转化过程中将引入失真,量化时必须使这些失真最小。

设信源符号的取值区间为 (a_0, a_n) , 即 $a_0 < x < a_n$, a_0 可为负无限, a_n 可为正无限,所以上述假设不失一般性。量化就是将上面的区间分成 n 个小区间,每个区间内定一个量化值 y_i ,若各区间端点为 a_{i-1} 和 a_i ,必有

$$a_0 \quad y_1 \quad a_1 \quad y_2 \quad \dots \quad a_{n-1} \quad y_n \quad a_n$$

最佳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。此时的信息率为 $R = \log n$ 。

最佳标量量化在一般情况下是比较难解的。有时可采用递推算法获得近似解。从理论上说,只要递推次数足够多,就可得到相当精确的解,但是计算量很大,收敛速度不快。在某些特殊情况下可采用求解析表达式的方法,下面举一个有解析解的例子。

设 x 在 $(0, L)$ 均匀分布,即

$$p(x) = \begin{cases} 1/L & 0 < x < L \\ 0 & \text{其他} \end{cases}$$

当量化级数为 n , 失真函数为均方失真或绝对失真时,可将 $(0, L)$ 均匀分割成 n 个小区间,小区间的宽度或称量化级差为 $q = L/n$,以每个小区间的中点作为量化值,即

$$a_i = iL/n \\ y_i = (2i - 1)L/2n$$

则平均失真 D 为

$$D(n) = \int_{a_{i-1}}^{a_i} d(x, y_i) p(x) dx$$

当失真函数 d 是均方型时,这样量化的平均失真为

$$D = \int_{(i-1)L/n}^{iL/n} x - \frac{(2i-1)L}{2n} \frac{dx}{L} = \frac{L^2}{12n^2} = \frac{q^2}{12}$$

对于绝对失真,平均失真为

$$D = \int_{(i-1)L/n}^{iL/n} \left| x - \frac{(2i-1)L}{2n} \right| \frac{dx}{L} = \int_{(2i-1)L/2n}^{iL/n} x - \frac{(2i-1)L}{2n} \frac{dx}{L} = \frac{L}{4n} = \frac{q}{4}$$

上述两种情况定义的平均失真虽然不同,但结论是相同的:级差 q 越小,平均失真就越小, n 就越大,所需的信息率也就越大。

先计算均匀分布时 x 的方差 σ^2 和平均偏差 $1/n$ 。根据定义可得

$$\begin{aligned} \sigma^2 &= E(x - Ex)^2 = \int_0^L x - \frac{x dx}{L} \cdot \frac{2 dx}{L} = \int_0^L x - \frac{L}{2} \cdot \frac{2 dx}{L} = \frac{L^2}{12} \\ \sigma &= E|x - Ex| = \int_0^L \left| x - \frac{x dx}{L} \right| \frac{dx}{L} = 2 \int_{L/2}^L x - \frac{L}{2} \cdot \frac{dx}{L} = \frac{L}{4} \end{aligned}$$

利用以上各式,就可得均方失真时的信息率与平均失真的关系式

$$R_1 = \log n = \frac{1}{2} \log \frac{L^2}{12 D} = \frac{1}{2} \log \frac{1}{D} \quad (4.4.9)$$

而对于绝对失真则为

$$R_2 = \log n = \log \frac{L}{4 D} = \log \frac{1}{D} \quad (4.4.10)$$

均匀分布时的率失真函数尚未求得表达式;但信息论中已证明,对于均方失真,当方差一定时,正态分布的率失真函数 $R(D)$ 恒小于其它分布时的 $R(D)$;对于绝对失真,当平均偏差一定时,负指数分布的率失真函数 $R(D)$ 恒小于其它分布时的 $R(D)$ 。从上述结果可见,(4.4.9)式和(4.4.10)式的右侧分别是正态分布和负指数分布的率失真函数,所以,即使在最佳标量量化的情况下,仍未能达到率失真函数所规定的界。而且量化以后的各概率,通过计算可知都是 $1/n$,因而倘若各连续符号 x 相互独立,也就不能再用无损编码进一步压缩。由此可见,最佳标量量化在这里不是最佳限失真编码。

噪声、噪声功率、信噪比是研究信号质量中常用的一些概念,量化后引起的失真可与它们联系起来。当一个样值 x 经量化后所产生的误差为

$$z_i = x - y_i \quad \text{或} \quad y_i = x - z_i$$

也就是在信号值 x 上叠加了一个样值为 $-z_i$ 的噪声信号。这种噪声通常称为量化噪声。在实际问题中,信噪比常被用来表征信息的质量,虽然它并不能充分满足信宿的需要,也就是不能与人耳或人眼的主观特性完全一致,但一般还是常采用的这种参量。量化噪声既不是白噪声,也不是高斯噪声,而且还与信源特性有关,也就是不能认为它与信源信号相互独立。

下面以语音信号量化常用的脉码调制(PCM)为例。为了编译码简单,可采用均匀量化。设语音信号的准峰值为 L ,由于语音信号是双向性的,则其取值范围为 $[-L, L]$ 。量化级数为 n ,量化级差为 $2L/n$,用中心值作为量化值 y_i 。当 n 很大时,量化区间内样值可认为近似均匀分布。设语音信号样值 x 的概率密度是负指数分布

$$p(x) = \frac{1}{2} e^{-|x|}, \quad -\infty < x < \infty$$

这是常用的语音近似分布,它的均值为 $Ex = 0$,这说明语音信号的双向对称性;它的方差也就是信号的平均功率为

$$W = Ex^2 = \frac{2}{2}$$

在这种量化中将碰到两类失真:一般的量化失真和过载失真。前者是当样值的绝对值小于 L 时的失真,而后者是样值超过 L 的过载情况下的失真。两者之和为总的量化噪声。经过推导计算(见参考文献 2)可得信号功率与噪声功率之比即信噪比为

$$= -10 \lg \frac{1}{6} \frac{L^2}{n} + 1 + \frac{L}{n} + \frac{1}{3} \frac{L^2}{n} e^{-L} \quad \text{dB}$$

信噪比是表示语音质量的一种参数,量化级数 n 和准峰值 L 的选择常用它作为依据。实际上,语音信号虽可近似地具有负指数分布,但它的平均功率不是恒定的,而是随时间变化的,变化范围还相当大。这个范围通常称为动态范围,据实测可达 40dB,也就是可相差 10 000 倍。

要避免过大的过载噪声,必须取大功率时的准峰值,并使过载概率很小,同时为了保证小功率时的量化噪声不致过大,量化级数 n 应足够大,所以大功率时的量化噪声常远小于过载噪声,这时的信噪比为 27 dB。这个信噪比已可满足要求,因为认为语音信号的信噪比大于 26 dB 时,语音质量已能使人满意。倘若要更好些,就应提高准峰值 L 。

另一方面,小功率时过载概率已很小,过载噪声就可忽略不计,若用 s 比特来编码。即 $n = 2^s$ 。此时的信噪比为

$$= -10\lg \frac{1}{6} \left(\frac{L}{n} \right)^2 = 6s - 48 \text{ dB}$$

该式说明,每增加 1 比特来编码,信噪比可提高 6 dB,而准峰值每提高一倍,信噪比要下降 6dB。若要求信噪比达到 26 dB,由上式可计算每样值所需的比特数为 12。

这样编码的压缩率当然不是最佳的。用 12 比特来编码是为了满足小功率信号的要求,当大功率信号出现时,由量化噪声所引起的信噪比将远大于 26 dB,仍用前面的 40 dB 的动态范围,此时信噪比可达 66 dB。显然这是浪费的,为了综合考虑大、小功率的情况,将它们区别对待,即非均匀量化,小功率时量化级差小;大功率时量化级差大。这样就减小了小功率时的量化噪声,增大了大功率时的量化噪声。使得在较低的比特数编码时,既保证了小功率时的信噪比,又利用了大功率时信噪比的富余量。这就是我们现在所用的压扩技术,用来降低码率。在国际上有两种标准,在美洲通用 μ 律,而在欧洲和我国通用 A 律。

从上述例子可见,实际问题中的量化往往不是从最佳量化出发,也不一定用较符合主观特性的失真函数,而是必须考虑许多实际情况。不同的信源就要有不同的处理方法。例如电视信号一般也用均匀量化,但所取的区间是 $(0, L)$, 0 代表黑色电平, L 代表白色电平;又如载波技术中群信号的量化,也用均匀量化而不采取压扩变换,因为群信号的功率由于多路合并而趋于平稳,不像单路时变化很大。

要想得到性能好的编码,仅采用标量量化是不可能的。在前面的最佳编码中可看到将离散信源的多个符号联合编码可提高效率。连续信源也是如此,当把多个信源符号联合起来形成多维矢量,再对矢量进行标量量化时,自由度将更大,同样的失真下,量化级数可进一步减少,码率可进一步压缩。这种量化叫做矢量量化。

实验证明,即使各信源符号相互独立,多维量化常可压缩信息率,这就使人们对矢量量化感兴趣而成为当前连续信源编码的一个热点。可是当维数较大时,矢量量化尚无解析方法,只能求助于数值计算;而且联合概率密度也不易测定,还需采用训练序列等方法。一般来说,高维矢量联合是很复杂的,虽已有不少方法,但在实用时尚有不少困难,有待进一步研究。

4.4.4 预测编码

前面介绍的编码方法都是考虑独立的信源序列。哈夫曼码对于独立多值信源符号很有效;二元序列的游程编码实际上是为了把二值序列转化成多值序列以适应哈夫曼编码;多个

二元符号合并成一个符号的方法也有类似的情况。算术码对于独立二元信源序列是很有效的,对于相关信源虽然可采用条件概率来编码而达到高效率,但这样做所引起的复杂度,往往使之难以实现。由信息论可知,对于相关性很强的信源,条件熵可远小于无条件熵,因此人们常采用尽量解除相关性的办法,使信源输出转化为独立序列,以利于进一步压缩码率。

常用的解除相关性的两种措施是预测和变换。它们既适应于离散信源,也可用于连续信源。其实两者都是序列的变换。一般来说,预测有可能完全解除序列的相关性,但必需确知序列的概率特性;变换编码一般只解除矢量内部的相关性,但它可有许多可供选择的变换矩阵,以适应不同信源特性。这在信源概率特性未确知或非平稳时可能有利。

本节介绍预测的一般理论和方法。

预测就是从已收到的符号来提取关于未收到的符号的信息,从而预测其最可能的值作为预测值;并对它与实际值之差进行编码,达到进一步压缩码率的目的。由此可见,预测编码是利用信源的相关性来压缩码率的,对于独立信源,预测就没有可能。

预测的理论基础主要是估计理论。估计就是用实验数据组成一个统计量作为某一物理量的估值或预测值。最常见的估计是利用某一物理量在被干扰下所测定的实验值,这些值是随机变量的样值,可根据随机量的概率分布得到一个统计量作为估值。若估值的数学期望等于原来的物理量,就称这种估计为无偏估计;若估值与原物理量之间的均方误差最小,就称之为最佳估计。用来预测时,这种估计就成为最小均方误差的预测,所以也就认为这种预测是最佳的。

要实现最佳预测就需要找到计算预测值的预测函数。设有信源序列 $x_1, x_2, \dots, x_r, x_{r+1}, \dots$ 。 r 阶预测就是由 x_1, x_2, \dots, x_r 来预测 x_{r+1} 。可令预测值为

$$x_{r+1} = f(x_1, x_2, \dots, x_r)$$

式中 f 是待定的预测函数。要使预测值具有最小均方误差,必须确知 $r+1$ 个变量 ($x_1, x_2, \dots, x_r, x_{r+1}$) 的联合概率密度函数,这在一般情况下是困难的。因而常用线性预测的方法来达到次最佳的结果。线性预测就是预测函数为各已知信源符号的线性函数,即 x_{r+1} 的预测值

$$x_{r+1} = f(x_1, x_2, \dots, x_r) = \sum_{s=1}^r a_s x_s \tag{4 4 11}$$

并求均方误差

$$D = E(x_{r+1} - x_{r+1})^2 \tag{4 4 12}$$

最小时的各 a_s 值。可将(4 4 11)式代入(4 4 12)式,对各 a_s 取偏导并置零:

$$\frac{D}{a_s} = - E(x_{r+1} - \sum_{s=1}^r a_s x_s) x_s = 0$$

只需已知信源各符号之间的相关函数即可进行运算。

最简单的预测是令

$$x_{r+1} = x_r$$

这可称为零阶预测,常用的差值预测就属这类。高阶线性预测已在语音编码,尤其是声码器中广泛采用。如果信源是非平稳的或非概率性的,无法获得确切和恒定的相关函数,不能构成线性预测函数,则可采用自适应预测的方法。一种常用的自适应预测方法是设预测函数是前几个符号值的线性组合,即令预测函数为

$$X_t = \sum_{s=1}^r a_s X_{t-r-1-s}$$

再用已知信源序列来确定各系数 a_s , 使对该序列所造成的均方误差 D 最小。此时的各系数 a_s 并不能保证对该信源发出的所有序列都适用, 只有在平稳序列情况下, 这种预测的均方误差可逼近线性预测时的最小值。随着序列的延长, 各系数 a_s 可根据以后的 n 个符号值来计算, 因而将随序列的延长而变更, 也就是可不断适应序列的变化, 适用于缓变的非平稳信源序列。

利用预测值来编码的方法可分为两类: 一类是用实际值与预测值之差进行编码, 也叫差值编码。常用于相关性强的连续信源, 也可用于离散信源。在连续信源的情况下, 就是对此差值量化或取一组差值进行矢量量化。由于相关性很强的信源可较精确地预测待编码的值, 这差值的方差将远小于原来的值, 所以在同样失真要求下, 量化级数可明显地减少, 从而较显著地压缩码率。对于离散信源也有类似的情况。

另一类方法是根据差值的大小, 决定是否需传送该信源符号。例如, 可规定某一可容许值, 当差值小于它时可不传送。对于连续函数或相关性很强的信源序列, 常有很长一串符号可以不送而只需传送这串符号的个数, 这样能大量压缩码率。这类方法一般是按信宿要求设计的, 也就是失真应能满足信宿需求。

4.4.5 变换编码

变换是一个广泛的概念。在通信系统中, 常希望把信号进行变换以达到某一目的。信源编码实际上就是一种变换, 使之能在信道中更有效地传送。这里将讨论的变换是数学意义上的一一对应变换。变换编码就是经变换后的信号的样值能更有效地编码, 也就是通过变换来解除或减弱信源符号间的相关性, 再将变换后的样值进行标量量化, 或采用对于独立信源符号的编码方法, 以达到压缩码率的目的。

首先讨论变换的一般原理, 即连续函数的变换。

设有函数 $f(t)$, $0 < t < T$, 则

$$\int_0^T f^2(t) dt < \infty \quad (4.4.13)$$

这函数是希尔伯特空间 $L^2(0, T)$ 的一个矢量, 其维数是可数无限, 它的坐标系可用一完备正交函数系来表征。

设有一完备正交归一函数系 (ϕ_i, t) , $i = 0, 1, 2, \dots$ 。正交性就是

$$\int_0^T \phi_i(t) \phi_j(t) dt = 0, \quad i \neq j \quad (4.4.14)$$

归一性就是

$$\int_0^T \phi_i^2(t) dt = 1 \quad (4.4.15)$$

可把 $f(t)$ 展开为

$$f(t) = \sum_{i=0}^{\infty} a_i \phi_i(t) \quad (4.4.16)$$

式中 a_i 是待定系数,可用有限项逼近时的均方误差最小准则来求,即

$$D_n = \int_0^T f(t) - \sum_{i=0}^{n-1} a_i \phi(i, t) dt$$

$$\frac{D_n}{a_i} = \int_0^T f(t) \phi(i, t) dt - 2 \sum_{i=0}^{n-1} a_i \int_0^T \phi(i, t) \phi(i, t) dt$$

利用函数 $\phi(i, t)$ 的正交归一性(4 4 14)式和(4 4 15)式,可得

$$a_i = \int_0^T f(t) \phi(i, t) dt \quad (4 4 17)$$

如果 $\lim_n D_n = 0$,则称上述正交函数系是完备的,此时(4 4 16)式才成立,否则就是不完备的,因而(4 4 17)式也就不成立。

与欧氏空间类比,可见(4 4 16)式实际上就是把函数矢量分解成各坐标分量,(4 4 17)式就相当于内积运算,把函数 $f(t)$ 投影到 $\phi(i, t)$ 上去。

通过上述变换,就把函数 $f(t)$ 变换成一系列离散的系数 a_i ,已给这些系数,就可用(4 4 16)式恢复函数 $f(t)$ 而不产生误差;所以这种变换是可逆的。如果只取有限个系数,恢复时就会引入误差。

傅氏变换具有正交归一性函数系,但从解除相关性的意义上说,傅氏变换不是一种很好的变换。要有效地解除相关性,正交函数系必须根据信源的相关函数来选择。

按均方误差最小准则来推算,有一种正交变换叫做 K L 变换(Karhunen Loeve Transform),可使变换后的随机变量之间互不相关。一般认为 K L 变换是压缩编码的最佳变换,评价其它变换时,常与它进行比较。K L 变换的最大缺点是计算复杂,除了需测定相关函数和解积分方程外,变换时的运算也十分复杂,尚无快速算法可用。

以上的变换是在时间上连续的信源输出 $x(t)$ 中取一段 $(0, T)$ 进行积分运算,得到一系列系数 $a_i, i = 0, 1, 2, \dots$,截取有限个 (n) ,即 $i = 0, 1, 2, \dots, n - 1$ 并对各 a_i 进行量化,达到信源编码的目的。这种方法在实际编码时较少应用,因为积分运算一般地说是比较困难的,而且除了量化各系数将引入失真外,截断取有限个系数也会引入失真。要保持失真在某一限度内,可能量化级数要有一定的增多,使码率有所上升。

另一种方法是先对信源输出 $x(t)$ 取样,得到一系列离散值 $x(i), i = 0, 1, 2, \dots$,然后取 N 个样值形成一个 N 维矢量,对这矢量用矩阵进行变换,成为另一域内的 N 维矢量,以解除或减弱矢量内各分量的相关性。再对后一个分量进行标量量化或对矢量进行矢量量化来完成信源编码。此时的变换已不用积分运算而是矩阵运算。若变换所用的矩阵选得恰当,就可达到压缩码率的要求。用矩阵来变换常称为离散变换。

其实取样也是一种把连续函数变换成时间上离散的一系列值的变换。此时变换所用的正交函数是单元脉冲函数 $\delta(t - i), i = 0, 1, 2, \dots$ 。是取样间隔。单元脉冲函数系的正交性和完备性是明显的;但这里已不是截取一段 $(0, T)$ 信源输出而是连续进行取样运算。要使变换能一一对应,也就是能无失真地恢复原来的连续函数,信源输出必须是限频的。若其最高频率是 f_m ,则取样间隔必须小于 $1/2f_m$,才能使变换可逆。不然将引入失真。实际连续信源常是限频的,尤其对信宿来说,频率大于一定值的含量,信宿已不感兴趣或已不能感受,语音和图像对人耳与人眼都有这种情况,所以限频的要求常是能满足的。这样既避

免了积分运算,也不致引入额外失真,因此实际上常采用离散变换。

上面提到的傅氏变换就有其相应的离散变换,只需将以前的正交函数取样即得。令取样点为 $t = kT/N, k = 0, 1, 2, \dots, N - 1$, 变换矩阵的元为

$$a_{ik} = w^{ik}/N$$

其中 $w = e^{j2\pi/N}$, 变换和反变换写成矩阵形式分别为

y_0		1	1	1	...	1	x_0
y_1		1	w	w^2	...	w^{N-1}	x_1
y_2	$= \frac{1}{N}$	1	w^2	w^4	...	$w^{2(N-1)}$	x_2
...	
y_{N-1}		1	w^{N-1}	$w^{2(N-1)}$...	$w^{(N-1)^2}$	x_{N-1}

(4 4 18)

x_0		1	1	1	...	1	y_0
x_1		1	w^*	w^{*2}	...	$w^{*(N-1)}$	y_1
x_2	$= \frac{1}{N}$	1	w^{*2}	w^{*4}	...	$w^{*2(N-1)}$	y_2
...	
x_{N-1}		1	$w^{*(N-1)}$	$w^{*2(N-1)}$...	$w^{*(N-1)^2}$	y_{N-1}

(4 4 19)

经傅氏变换后的输出各分量间的相关系数将与原输入过程的相关函数有关。一般来说,输入过程的相关系数越接近 1, 输出各分量间的相关函数越小,也就是说傅氏变换对强相关的信源是有效的。此外各输出分量的方差将不同,有大有小,即经变换后能量有所集中,这对压缩码率也是有利的。

离散傅氏变换虽有快速算法(FDFT 或 FFT)可减少计算量,但运算将在复数域内进行,这是不方便的,在实用中常用离散余弦变换(DCT)。尤其是对视频图像信号,其统计特性接近一阶马氏链,离散余弦变换的正交矢量近似于相应的 K L 变换的正交矢量。

余弦变换的完备正交归一函数系是

$$\begin{aligned} (0, t) &= \frac{1}{N} \\ (i, t) &= \frac{2}{N} \cos \frac{(2i+1)t}{2T}, \quad t \in (0, T) \end{aligned}$$

对这些函数在(0, T)内取 N 个样值,即得离散余弦变换矩阵的元

$$\begin{aligned} a_{0k} &= 1/\sqrt{N} \\ a_{ik} &= \sqrt{2/N} \cos[(2k+1)i/N] \end{aligned}$$

变换和反变换的矩阵形式分别为

y_0		$\frac{1}{2}$	$\frac{1}{2}$...	$\frac{1}{2}$	x_0
y_1	$= \frac{2}{N}$	$\cos \frac{1}{2N}$	$\cos \frac{3}{2N}$...	$\cos \frac{2N-1}{2N}$	x_1
...	
y_{N-1}		$\cos \frac{N-1}{2N}$	$\cos \frac{3(N-1)}{2N}$...	$\cos \frac{(2N-1)(N-1)}{2N}$	x_{N-1}

$$\begin{array}{ccccccc}
x_0 & & \frac{1}{2} & \cos \frac{N-1}{2N} & \cdots & \cos \frac{N-1}{2N} & y_0 \\
x_1 & = \frac{2}{N} & \frac{1}{2} & \cos \frac{3}{2N} & \cdots & \cos \frac{3(N-1)}{2N} & y_1 \\
\cdots & & \cdots & \cdots & \cdots & \cdots & \cdots \\
x_{N-1} & & \frac{1}{2} & \cos \frac{2N-1}{2N} & \cdots & \cos \frac{(2N-1)(N-1)}{2N} & y_{N-1}
\end{array}$$

在离散变换中,最佳变换也是 K L 变换。其正交矢量系和变换矩阵可根据输入矢量各分量间的相关系数来求,而不用解积分方程,只需求相关矩阵的特征值和特征矢量。容易验证经过 K L 变换后输出矢量的相关系数为零,即它能完全解除输出矢量间的线性相关性,且各分量的方差就是各特征值,它们各不相等,下降很快。在实际编码时,后面几个分量,方差已很小,往往可以不传送,有利于压缩编码。

还有很多离散变换,如正反变换矩阵都相同的离散哈尔变换和离散沃尔什变换;由有限维正交矢量系导出的广泛用于电视信号编码的斜变换和多重变换;可把信号分割成多个窄带以解除或减弱信号样值间相关性的子带编码和小波变换等等。在实际应用中,需要根据信源特性来选择变换方法以达到解除相关性、压缩码率的目的。另外还可以根据一些参数来比较各种变换方法间的性能优劣,如反映编码效率的编码增益、反映编码质量的块效应系数等。当信源的统计特性很难确知时,可用各种变换分别对信源进行变换编码,然后用实验或计算机仿真来计算这些参数。

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 设输入符号与输出符号为 $X = Y \in \{0, 1, 2, 3\}$, 且输入信号的分布为

$$p(X = i) = 1/4, i = 0, 1, 2, 3$$

设失真矩阵为

$$d = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

求 D_{\max} 和 D_{\min} 及 $R(D_{\max})$ 和 $R(D_{\min})$, 以及相应的转移概率。

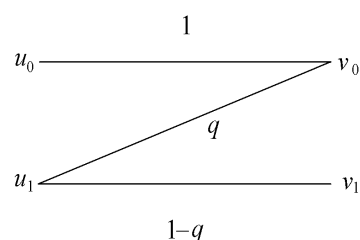
4 3 具有符号集 $U = \{u_0, u_1\}$ 的二元信源, 信源发生概率为:

$$p(u_0) = p, p(u_1) = 1 - p (0 < p < 1/2).$$

Z 信道如图题 4 3 所示, 接收符号集 $V = \{v_0, v_1\}$, 转移概率为:

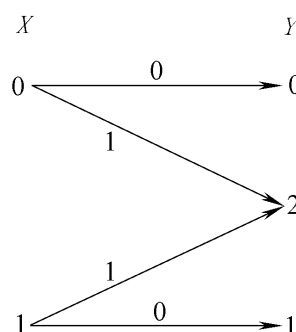
$$p(v_0/u_0) = 1, p(v_1/u_1) = 1 - q.$$

发出符号与接收符号的失真: $d(u_0, v_0) = d(u_1, v_1) = 0, d(u_1, v_0) = d(u_0, v_1) = 1$ 。



图题 4 3

- (1) 计算平均失真 D ;
- (2) 率失真函数 $R(D)$ 的最大值是什么? 当 q 为何值时可达到该最大值? 此时平均失真 D 是多大?
- (3) 率失真函数 $R(D)$ 的最小值是什么? 当 q 为何值时可达到该最小值? 此时平均失真 D 是多大?
- (4) 画出 $R(D) \sim D$ 的曲线。
- 4 4 已知信源的符号集合 $X = \{0, 1\}$, 它们以等概率出现, 信宿的符号集合 $Y = \{0, 1, 2\}$, 失真函数如图题 4 4 所示, 其中连线旁的值为失真函数, 无连线表示失真函数为无限大, 即 $d(0, 1) = d(1, 0) = \infty$, (同时有 $p(y_1/x_0) = p(y_0/x_1) = 0$), 求 $R(D)$ 。
- 4 5 三元信源的概率分别为 $p(0) = 0.4$, $p(1) = 0.4$, $p(2) = 0.2$, 失真函数 d_{ij} 为: 当 $i = j$ 时, $d_{ij} = 0$; 当 $i \neq j$ 时, $d_{ij} = 1$ ($i, j = 0, 1, 2$), 求信息率失真函数 $R(D)$ 。
- 4 6 利用 $R(D)$ 的性质, 画出一般 $R(D)$ 的曲线并说明其物理意义? 试问为什么 $R(D)$ 是非负且非增的?
- 4 7 在电视信号中, 亮度信号的黑色电平为 0, 白色电平为 L 。用均匀分割来量化其样值, 要求峰功率信扰比大于 50 dB, 求每样值所需的量化比特数。



图题 4 4

第 5 章 信道编码

5.1 信道模型和信道容量

5.1.1 信道模型

回忆通信系统模型,在信道编码器和信道解码器之间相隔着许多其他部件,如调制解调、放大、滤波、均衡等器件以及各种物理信道。信道遭受各类噪声的干扰,使有用信息遭受损伤。从信道编码的角度,我们对信号在信道中具体如何传输的物理过程并不感兴趣,而仅对传输的结果感兴趣:送入什么信号、得到什么信号,如何从得到的信号中恢复出送入的信号,差错概率是多少。为了集中注意力研究以上问题,我们把信道编、解码器之间的所有部件看成是一个“黑箱”(blackbox),像研究多端口网络那样把问题归结为输入、输出和转移概率矩阵三个要素,如图 5 1 1 所示。图中, $X = \{x_0, x_1, \dots, x_{q-1}\}$ 是包含 q 个元素的输入符号集, $Y = \{y_0, y_1, \dots, y_{Q-1}\}$ 是包含 Q 个元素的输出信号集。由 q 和 Q 等于 2、大于 2 还是趋于 ∞ ,可区分出如下一些信道模型。

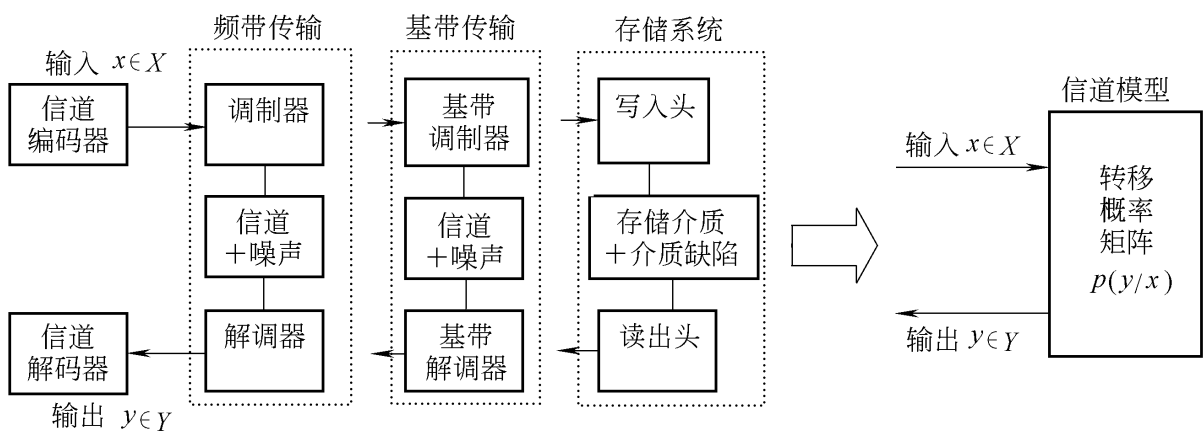


图 5 1 1 信道模型

1. 二进制离散信道

二进制离散信道模型有一个允许输入值的集合 $X = \{0, 1\}$ 和可能输出值的集合 $Y = \{0, 1\}$, 以及一组表示输入、输出关系的条件概率(转移概率)组成。如果信道噪声和其他干扰导致传输的二进序列发生统计独立的差错,且条件概率对称,即

$$\begin{aligned} p(Y=0 \mid X=1) &= p(Y=1 \mid X=0) = p \\ p(Y=1 \mid X=1) &= p(Y=0 \mid X=0) = 1 - p \end{aligned} \tag{5.1.1}$$

则这种对称的二进制输入、二进制输出信道叫做二进制对称信道,简称为 BSC 信道,其信道模型如图 5 1 2 所示。由于这种信道的输出符号仅与对应时刻的一个输入符号有关而与以

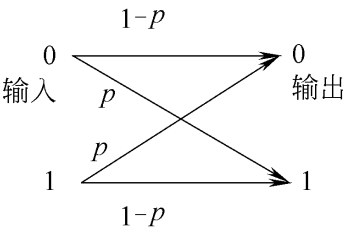


图 5 1 2 二进制对称信道(BSC)

前的输入无关,所以这种信道是无记忆的。当图 5 1 1 中采用二进制调制、检测器实行门限硬判决且信道对称时,就构成了 BSC 信道。BSC 信道是研究二元编解码最简单、也是最常用的信道模型。

2 . 离散无记忆信道

BSC 可视为是一种更广义的离散输入、离散输出信道的一个特例。假设信道编码器的输入是 q 元符号即输入符号集由 q 个元素 $X = \{ x_0, x_1, \dots, x_{q-1} \}$ 构成,而检测器的输出是 Q 元符号即信道输出符号集由 Q 个元素 $Y = \{ y_0, y_1, \dots, y_{Q-1} \}$ 构成,且信道和调制过程是无记忆的,那么图 5 1 1 所示信道模型黑箱的输入 输出特性可以用一组共 qQ 个条件概率来描述

$$p(Y = y_j / X = x_i) = p(y_j / x_i) \quad (5 1 2)$$

式中, $i = 0, 1, \dots, q - 1$; $j = 0, 1, \dots, Q - 1$ 。这样的信道称为离散无记忆信道 (DMC: Discrete Memoryless Channel), 其示意图如图 5 1 3 所示。

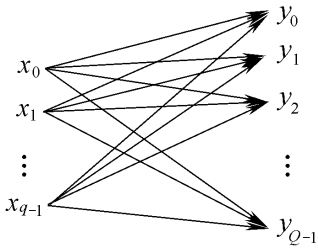


图 5 1 3 离散无记忆信道 (DMC)

若 DMC 信道的输入、输出是一个由 n 个符号组成的序列,其中输入序列的 n 个符号 u_1, u_2, \dots, u_n 选自符号集 X 即 $u_i \in X$, 相应输出序列的 n 个符号 v_1, v_2, \dots, v_n 选自符号集 Y 即 $v_i \in Y$, 则联合条件概率是

$$p(Y_1 = v_1, Y_2 = v_2, \dots, Y_n = v_n / X = u_1, \dots, X = u_n) = \prod_{k=1}^n p(Y_k = v_k / X = u_k) \quad (5 1 3)$$

这个表达式正是满足无记忆条件的数学表述。

通常,决定 DMC 特点的条件概率 $\{ p(y_j / x_i) \}$ 可以写成矩阵的形式 $P = [p_{ij}]$ 。根据定义,式中的 $p_{ij} = p(y_j / x_i)$, P 称作是信道的转移概率矩阵。

$$\begin{aligned} P &= \begin{bmatrix} p(y_0/x_0) & p(y_1/x_0) & \dots & p(y_{Q-1}/x_0) \\ p(y_0/x_1) & p(y_1/x_1) & \dots & p(y_{Q-1}/x_1) \\ \dots & \dots & \dots & \dots \\ p(y_0/x_{q-1}) & p(y_1/x_{q-1}) & \dots & p(y_{Q-1}/x_{q-1}) \end{bmatrix} \\ &= \begin{bmatrix} p_{00} & p_{01} & \dots & p_{0, Q-1} \\ p_{10} & p_{11} & \dots & p_{1, Q-1} \\ \dots & \dots & \dots & \dots \\ p_{q-1, 0} & p_{q-1, 1} & \dots & p_{q-1, Q-1} \end{bmatrix} \end{aligned} \quad (5 1 4)$$

在信道输入为 x_i 的条件下,由于干扰的存在,信道输出不是一个固定值而是概率各异的一组值,这种信道就叫有扰离散信道。显然,输入 x_i 时各可能输出值 y_j 的概率之和必定等于 1,即

$$\sum_{j=0}^{Q-1} p(y_j / x_i) = 1 \quad i = 0, 1, \dots, q - 1 \quad (5 1 5)$$

如果信道转移概率矩阵的每一行中只包含一个“1”,其余元素均为“0”,说明信道无干扰,叫无扰离散信道。

3. 离散输入、连续输出信道

假设信道输入符号选自一个有限的、离散的输入字符集 $X = \{x_0, x_1, \dots, x_{q-1}\}$, 而信道(检测器)输出未经量化 ($Q = \infty$), 这时的译码器输入可以是实轴上的任意值, 即 $Y = \{-\infty, \infty\}$ 。定义这样的信道模型为离散时间无记忆信道, 它的特性由离散输入 X 、连续输出 Y 以及一组条件概率密度函数 $p(y/X = x_i)$, $i = 0, 1, \dots, q-1$ 来决定。这类信道中最重要的一种是加性高斯白噪声(AWGN)信道, 对它而言

$$Y = X + G \quad (5.1.6)$$

式中 G 是一个零均值、方差为 σ^2 的高斯随机变量, $X = x_i$, $i = 0, 1, \dots, q-1$ 。

当 X 给定后, Y 是一个均值为 x_i 、方差为 σ^2 的高斯随机变量

$$p(y/x_i) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y-x_i)^2}{2\sigma^2}} \quad (5.1.7)$$

4. 波形信道

波形信道是这样一种信道模型: 其输入是模拟波形, 其输出也是模拟波形。假设输入该信道的是带限信号 $x(t)$, 相应的输出是 $y(t)$, 那么

$$y(t) = x(t) + n(t) \quad (5.1.8)$$

这里 $n(t)$ 代表加性噪声过程的一个样本函数。为了定义一组能体现信道特征的转移概率, 一般的做法是把 $x(t)$, $y(t)$ 和 $n(t)$ 展开成一个正交函数的完备集, 得到与展开式对应的一套系数 $\{x_i\}$, $\{y_i\}$ 和 $\{n_i\}$, 然后利用展开式中的系数来描述信道特征。

在分析问题时选用以上的何种信道模型完全取决于我们的目的。如果我们的兴趣在于设计和分析离散信道编、解码器的性能, 从工程角度出发, 最常用的是 DMC 信道模型或其简化形式 BSC 信道模型, 若分析性能的理论极限, 则多选用离散输入、连续输出信道模型。另一方面, 如果我们是想要设计和分析数字调制器和解调器的性能, 则可采用波形信道模型。本书的主题是编、解码, 因此 DMC 信道模型使用最多。

5.1.2 信道容量

1. DMC 信道的容量

考虑一个 DMC 信道, 其输入字符集是 $X = \{x_0, x_1, \dots, x_{q-1}\}$, 输出字符集是 $Y = \{y_0, y_1, \dots, y_{Q-1}\}$, 转移概率 $p(y_j/x_i)$ 如(5.1.2)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 $I(X; Y)$

$$I(X; Y) = \sum_{i=0}^{q-1} \sum_{j=0}^{Q-1} p(x_i) p(y_j/x_i) \log \frac{p(y_j/x_i)}{p(y_j)} \quad (5.1.9)$$

式中 $p(y_j)$ 可利用下式计算得到。

$$p(y_j) = p(Y = y_j) = \sum_{i=0}^{q-1} p(x_i) p(y_j/x_i) \quad (5.1.10)$$

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

$$C = \max_{p(x_i)} I(X; Y) = \max_{p(x_i)} \sum_{i=0}^{q-1} \sum_{j=0}^{Q-1} p(x_i) p(y_j/x_i) \log \frac{p(y_j/x_i)}{p(y_j)} \quad (5.1.11)$$

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(bits/ symbol 或 bits/ channel use)。当然以上 $I(X; Y)$ 值的最大化是在下列限制条件下进行的,

$$\begin{aligned} p(x_i) &= 0 \\ \sum_{i=0}^{q-1} p(x_i) &= 1 \end{aligned} \quad (5.1.12)$$

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(nats/ symbol)。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(bits/ s)或奈特/秒(nats/ s)为信道容量单位。

转移概率矩阵 P 已知后,由(5.1.11)式计算 DMC 信道容量的关键是找出能使 $I(X; Y)$ 最大的 $p(x_i) (i=0, \dots, q-1)$ 的概率分布。若将 $P_x = [p(x_0), p(x_1), \dots, p(x_{q-1})]$ 定义为输入符号的概率矢量 P_x ,由式(5.1.11)及关系式

$$I(X; Y) = H(X) - H(X/Y) = H(Y) - H(Y/X) \quad (5.1.13)$$

可得:

$$C = \max_{P_x} I(X; Y) = \max_{P_x} [H(X) - H(X/Y)] = \max_{P_x} [H(Y) - H(Y/X)] \quad (5.1.14)$$

这里存在两个问题,一是 $I(X; Y)$ 的最大值是否存在? 二是如果最大值存在,怎样才能找到它? 第一个问题即信道容量存在性问题,可以用以下的存在性定理来说明。

定理: 给定转移概率矩阵 P 后,平均互信息 $I(X; Y)$ 是概率矢量 P_x 的上凸函数。(证明略)

用 $I(P_x)$ 表示 I 是 P_x 的函数,则在 $I(P_x)$ 曲线上凸点所对应的输入符号概率矢量 P_x 上, $I(P_x)$ 取得了极大值,这个极大值就是信道容量。第二个问题是信道容量的计算问题。

(1) 对称 DMC 信道的容量

如果转移概率矩阵 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 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 DMC 为对称的 DMC 信道。

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

可以证明,有扰的对称 DMC 信道具有如下性质:

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

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

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,信道输入符号必定也是等概分布。

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

$$C = \log Q - H(Y/X) = \log Q + \sum_{j=1}^Q p_{ij} \log p_{ij} \quad (5.1.15)$$

由于对称信道的条件熵 $H(Y/X)$ 与信道输入符号的概率分布无关,式(5.1.14)化成

$$\begin{aligned} \max_{p_x} [H(Y) - H(Y/X)] &= \max_{p_x} [H(Y) - H(Y/x_i)] \\ &= \max_{p_x} [H(Y)] - H(Y/x_i) \end{aligned}$$

于是问题就简化为求 $\max_{p_x} [H(Y)]$ 。由信息论原理,当输出符号集的各符号等概出现时可得最大信源熵,即

$$H(Y) = \log Q \quad \text{或者} \quad \max [H(Y)] = \log Q \quad (5.1.16)$$

这就是(5.1.15)式的来历。

(2) BSC 信道的容量

BSC 信道是 DMC 对称信道的特例,因此对于转移概率为 $p(0/1) = p(1/0) = p$ 及 $p(0/0) = p(1/1) = 1 - p$ 的 BSC 信道而言,当输出概率 $p(y_0) = p(y_1) = 0.5$ 时其平均互信息最大。所以,BSC 的信道容量是

$$\begin{aligned} C &= p(x_0) p(0/0) \log [p(0/0)/0.5] + p(x_0) p(1/0) \log [p(1/0)/0.5] \\ &\quad + p(x_1) p(0/1) \log [p(0/1)/0.5] + p(x_1) p(1/1) \log [p(1/1)/0.5] \\ &= p \log 2p + (1 - p) \log 2(1 - p) \end{aligned} \quad (5.1.17)$$

C 随 p 变化的曲线如图 5.1.4 所示。由图可知, $p = 0$ 时的信道容量是 1 比特每符号 (1 bit/symbol); 当 $p = 1/2$ 时,从输出得不到关于输入的任何信息,互信息为 0 即信道容量是零。对于 $1/2 < p < 1$ 的情况,可在 BSC 的输出端颠倒 0 和 1,导致信道容量以 $p = 1/2$ 点为中心对称。

从信息论的角度看,平均的条件自信息即条件熵 $H(X/Y)$ 可以解释为由于信道干扰和噪声所造成的平均信息量的损伤。如果 BSC 信道中 $p(0/1) = p(1/0) = p = 0$,即无误码概率,那么从接收的 Y 可完全确定发送的 X ,信道的介入没有产生任何损伤或模糊度,因此条件熵 $H(X/Y) = 0$ 。由式(5.1.13),若 $H(X/Y) = 0$,必有 $I(X; Y) = H(X)$,互信息等于输入符号的信息熵。换言之,信道上传送的信息量正是输入信号的全部信息量,相当于在图 5.1.4 中信道容量为 1。

当 X 和 Y 统计独立时,接收的 Y 完全与发送的 X 无关,此时 $p = 0.5$ 及 $H(X/Y) = H(X)$,说明损失的信息达到与输入符号信息熵相等的程度。由式(5.1.13)或图 5.1.4 可得 $I(X; Y) = 0$ 或 $C = 0$,即信道上没能传送任何信息。

在一般情况下,根据香农不等式,

$$I(X; Y) = H(X) - H(X/Y) \geq 0 \quad (5.1.18)$$

应有 $I(X; Y) > 0$ 或 $H(X) > H(X/Y)$,即总有信息传递过来而使输出端的不确定度小于输入端的信息量。

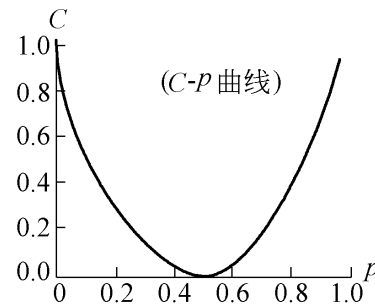


图 5.1.4 二进制信道的信道容量

(3) 准对称 DMC 信道的容量

如果转移概率矩阵 P 是输入对称而输出不对称, 即转移概率矩阵 P 的每一行都包含同样的元素而各列的元素可以不同, 则称该矩阵是准对称 DMC 信道。例如, 矩阵

$$P = \begin{pmatrix} 0.3 & 0.2 & 0.2 & 0.3 \\ 0.2 & 0.3 & 0.2 & 0.3 \end{pmatrix}$$

就是准对称的 DMC 信道。

可以证明, 准对称 DMC 信道的容量

$$C = \log Q - \frac{1}{Q} \sum_{j=1}^Q p_{ij} \log p_{ij} \quad (5.1.19)$$

当信道输入符号等概分布时, 准对称 DMC 信道达到其信道容量 C 。

例 5.1.1 已知一个信道的信道转移矩阵为 $P = \begin{pmatrix} 0.5 & 0.3 & 0.2 \\ 0.3 & 0.5 & 0.2 \end{pmatrix}$, 求该信道的容量。

解: 由 p 可看出信道的输入符号有两个, 可设 $p(x_1) = \frac{1}{2}$, $p(x_2) = 1 - \frac{1}{2}$ 。信道的输出符号有三个, 用 y_1, y_2, y_3 表示。由 $p(x_i, y_j) = p(x_i) p(y_j | x_i)$ 得联合概率的矩阵为

$$\begin{pmatrix} 0.5 & 0.3 & 0.2 \\ 0.3(1 - \frac{1}{2}) & 0.5(1 - \frac{1}{2}) & 0.2(1 - \frac{1}{2}) \end{pmatrix}$$

由 $p(y_j) = \sum_i p(x_i, y_j)$ 得

$$\begin{aligned} p(y_1) &= 0.5 + 0.3(1 - \frac{1}{2}) = 0.3 + 0.2 \\ p(y_2) &= 0.3 + 0.5(1 - \frac{1}{2}) = 0.5 - 0.2 \\ p(y_3) &= 0.2 + 0.2(1 - \frac{1}{2}) = 0.2 \end{aligned}$$

其中 $p(y_3)$ 恒定, 与 x_i 的分布无关。

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

由 $\frac{\partial I(X; Y)}{\partial p(x_1)} = 0$ 得 $0.2 \ln(0.3 + 0.2) - 0.2 + 0.2 \ln(0.5 - 0.2) + 0.2 = 0$

解得 $p(x_1) = 1/2$, 即输入符号分布等概率时, $I(X; Y)$ 达到极大值。所以信道容量为

$$C = \max I(X; Y) = 0.036 \text{ 比特/符号}$$

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

事实上该信道叫做二元对称删除信道, 当 $p(x_1) = p(x_2) = 1/2$ 时, 可达到信道容量 $C = \max I(X; Y)$, 因为 $p(y_3)$ 恒定为 0.2, 则 y_1, y_2 应等概分布, 即 $p(y_1) = p(y_2) = 0.4$ 。

(4) 一般 DMC 信道的容量

以输入符号概率矢量 P_x 为自变量求函数 $I(P_x)$ 极大值即信道容量的问题, 从数学上看是一个规划问题, 这个问题已经解决。目前常用的方法是 1972 年由 R. Blahut 和 A. Arimoto 分别独立提出的一种算法, 现在称为 Blahut Arimoto 算法。一般地说, 为使 $I(X; Y)$ 最大化以便求取 DMC 容量, 输入概率集 $\{p(x_i)\}$ 必须满足的充分和必要条件是

$$\begin{aligned} I(x_i; Y) &= C \quad \text{对于所有满足 } p(x_i) > 0 \text{ 条件的 } i \\ I(x_i; Y) &\leq C \quad \text{对于所有满足 } p(x_i) = 0 \text{ 条件的 } i \end{aligned} \quad (5.1.20)$$

这里

$$I(x_i; Y) = \sum_{j=0}^{Q-1} p(y_j | x_i) \log \frac{p(y_j | x_i)}{p(y_j)}$$

2. 离散时间无记忆信道的容量

如 DMC 信道的输入字符集有限而输出字符集 $Y = \{y_0, y_1, \dots, y_{Q-1}\}$ 中 $Q \rightarrow \infty$ 时, 信道就不再是离散信道而是离散输入、连续输出的离散时间无记忆信道。离散时间无记忆信道的容量, 可视作 DMC 信道软判决译码时的容量极限, 具有研究价值。这类信道中最重要的一种是加性高斯白噪声(AWGN)信道, 对它而言, 离散输入 $X = \{x_0, x_1, \dots, x_{q-1}\}$ 和模拟输出 $Y = \{y_0, y_1, \dots, y_{q-1}\}$ 之间的最大平均互信息即信道容量由下式给出(单位是比特/符号):

$$C = \max_{P_x} \sum_{i=0}^{q-1} p(x_i) \log_2 \frac{p(y | x_i)}{p(y)} \quad (5.1.21)$$

式中

$$p(y) = \sum_{i=0}^{q-1} p(y | x_i) p(x_i)$$

作为特例, 对于一个二进制输入的 AWGN 无记忆信道, 若 $X = \{x_0, x_1\} = \{A, -A\}$, 输入概率矢量 $P_x = (0.5, 0.5)$ 即等概输入时, 平均互信息 $I(X; Y)$ 最大而达到信道容量, 以比特/符号为单位是

$$C = \frac{1}{2} \int_{-\infty}^{\infty} p(y | A) \log_2 \frac{p(y | A)}{p(y)} dy + \frac{1}{2} \int_{-\infty}^{\infty} p(y | -A) \log_2 \frac{p(y | -A)}{p(y)} dy \quad (5.1.22)$$

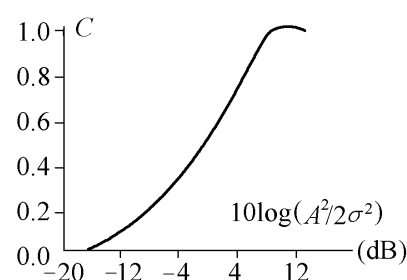
式中, $p(y | A)$, $p(y | -A)$ 和 $p(y)$ 均与信道中的噪声方差有关; C 作为 $A^2/2\sigma^2$ 比值函数的关系曲线如图 5.1.5 所示。我们注意到, 当比值增大时, C 从 0 到 1 比特/符号单调地增大。

3. 带限波形信道的容量

一个受加性高斯白噪声干扰的带限波形信道的容量, 已由香农(1948)正式定义为

$$C = \lim_{T \rightarrow \infty} \max_{P_x} \frac{1}{T} I(X; Y) \quad (5.1.23)$$

图 5.1.5 AWGN 无记忆信道二元输入时
作为 $A^2/2\sigma^2$ 函数的信道容量



若把输入、输出和噪声波形 $x(t)$ 、 $y(t)$ 和 $n(t)$ 展开成一个正交函数的完备集, 可得到与展开式对应的一组系数 $\{x_i\}$ 、 $\{y_i\}$ 和 $\{n_i\}$, 然后利用展开式中的系数来描述信道特征。令 $X_N = [x_1, x_2, \dots, x_N]$ 及 $Y_N = [y_1, y_2, \dots, y_N]$, 这里 $N = 2WT$, $y_i = x_i + n_i$, 则 AWGN 信道 X_N 和 Y_N 之间的平均互信息是

$$I(X_N; Y_N) = \sum_{i=1}^N p(y_i | x_i) p(x_i) \log \frac{p(y_i | x_i)}{p(y_i)} \quad (5.1.24)$$

式中

$$p(y_i | x_i) = \frac{1}{\sqrt{N_0}} e^{-\frac{(y_i - x_i)^2}{2N_0}} \quad (5.1.25)$$

当 $\{x_i\}$ 是统计独立、零均值的高斯随机变量时, 即

$$p(x_i) = \frac{1}{\sqrt{2\pi\sigma_x^2}} e^{-\frac{x_i^2}{2\sigma_x^2}} \quad (5.1.26)$$

式中 σ_x^2 是各 x_i 的方差, 则对于已知的输入 pdf 值 $p(x_i)$, 可求出 $I(X; Y)$ 的最大值。

由(5 1 24)式

$$\max_{P_x} I(X_N; Y_N) = \sum_{i=1}^N \frac{1}{2} \log \left(1 + \frac{\sigma_x^2}{N_0} \right) = \frac{1}{2} N \log \left(1 + \frac{\sigma_x^2}{N_0} \right) = WT \log \left(1 + \frac{\sigma_x^2}{N_0} \right) \quad (5 1 27)$$

假如我们对 $x(t)$ 的平均功率加以限制, 即

$$P_{av} = \frac{1}{T} \int_0^T E[x^2(t)] dt = \frac{1}{T} \sum_{i=1}^N E(x_i^2) = \frac{N \sigma_x^2}{T} \quad (5 1 28)$$

于是有

$$\sigma_x^2 = \frac{TP_{av}}{N} = \frac{P_{av}}{2W} \quad (5 1 29)$$

将上式代入(5 1 27)式, 并将结果除以 T , 就得到单位时间的信道容量

$$C = W \log \left(1 + \frac{P_{av}}{WN_0} \right) = W \log(1 + \text{SNR}) \quad (5 1 30)$$

这就是带限 AWGN 波形信道在平均功率受限条件下信道容量的基本公式, 也就是有名的香农公式。根据香农公式, 带宽一定时, 信道容量随 SNR 的增加而单调增加, 因此增大信号功率、减小信道噪声可以增加信道容量。另一方面, 如果 SNR 固定, 信道容量随着带宽的增加而增加。利用关系式 $\ln(1+x) \approx x$ (x 很小时), 可得极限情况即 $W \rightarrow \infty$ 时, $C \rightarrow C_\infty$

$$C_\infty = \frac{P_{av}}{N_0 \ln 2} \quad (5 1 31)$$

C_∞ 称为无限带宽 AWGN 波形信道的容量。

如果以最大速率即信道容量 C 来传递信息, 每传输 1 比特信息所需的能量为 E_b , 总的信号功率是

$$P_{av} = CE_b \quad (5 1 32)$$

代入(5 1 30)式, 等式两边再同除 W , 得

$$\frac{C}{W} = \log_2 \left(1 + \frac{CE_b}{WN_0} \right) \quad (5 1 33)$$

式中, C/W 代表归一化的信道容量即单位带宽的信道容量。为了说明归一化信道容量 C/W 与达到该容量所需信噪比 E_b/N_0 的关系, 将(5 1 33)式改写为

$$\frac{E_b}{N_0} = \frac{2^{C/W} - 1}{C/W} \quad (5 1 34)$$

当 $C/W = 1$ 时(每赫兹传 1 比特), $E_b/N_0 = 1$ 即要求的信噪比为 0 dB。

当 $C/W \rightarrow \infty$ 时

$$\frac{E_b}{N_0} = \frac{2^{C/W}}{C/W} \exp \left[\frac{C}{W} \ln 2 - \ln \frac{C}{W} \right] \quad (5 1 35)$$

说明 $C/W \rightarrow \infty$ 时对 E_b/N_0 的要求是指数增加的。

另一方面, 当 $C/W \rightarrow 0$ 时,

$$\frac{E_b}{N_0} = \lim_{C/W \rightarrow 0} \frac{2^{C/W} - 1}{C/W} = \ln 2 = 1.6 \text{ dB} \quad (5 1 36)$$

说明当 E_b/N_0 为 -1.6 dB 时,归一化信道容量 $C/W=0$,即信道完全丧失了通信能力。把 -1.6 dB 称作香农限,是一切编码方式所能达到的理论极限。

从上述香农公式还可以看出,对于给定的信道容量 C ,带宽 W 和信噪比 S/N 存在着互换的关系,即若减小带宽则必须发送较大的信号功率,若有较大的传输带宽,则在同样信道容量的情况下能够用较小的信号功率来传送,这表明宽带系统具有较好的抗干扰性。扩频通信就是利用这个原理,将所需传送的信号扩频,使之远远大于原始信号带宽,以增强抗干扰的能力。

至此,已推导了三种重要信道模型的信道容量。第一种是 DMC 信道,BSC 信道是它的一个特例。第二种是离散输入、连续输出的无记忆加性高斯白噪声信道。可以利用这两种信道模型构成标准平台,用以衡量数字通信系统分别在硬、软判决译码时的编码性能。第三种是波形信道,在这种信道中假设信道带宽受限、信号受到加性高斯噪声的损伤且发送的平均功率受限,在这些约束条件下,推导出了(5.1.30)式的香农公式,接着又推导了(5.1.36)式的香农限。上述信道容量公式的主要意义是:为了在噪声信道中可靠通信,可以用它来确定信息传输速率的上限值。

5.2 有扰离散信道的编码定理

在高效的信息传输中,编码调制的设计有两条基本途径。一条是代数途径,即运用编、解码技术来设计特定种类的码,比如分组码、卷积码等。第二条途径是采用概率方法,在给定信道特性的条件下对编码信号的性能作统计分析,求出差错概率的上下限边界,其中最优码所能达到的差错概率的上界,称作随机码界,对指导编码技术具有特别重要的理论价值。下面先介绍随机编码方法,再引入有扰离散信道的编码定理,最后讨论差错控制与信道编码的基本原理。

5.2.1 随机编码

考虑一个 q 元入、 Q 元出的 DMC 离散无记忆信道,其输入字符集是 $X = \{x_0, x_1, \dots, x_{q-1}\}$,输出字符集是 $Y = \{y_0, y_1, \dots, y_{Q-1}\}$,转移概率集合为 $\{p(y_j/x_i)\}$ 。再考虑一个 (N, K) 分组码编码器,如图 5.2.1 所示。该编码器介于信源输出和信道输入之间,对由 K 个 q 进制符号组成的消息组 $m = (m_1, m_2, \dots, m_K)$ 实行编码,生成由 N 个 q 进制符号(也称码元)组成的码字 $c = (c_1, c_2, \dots, c_N)$,其中,码元 $c_1, \dots, c_N \in X$ 。码字 $c = (c_1, c_2, \dots, c_N)$ 可视为是一个 N 重矢量,如果设想有一个 N 维的矢量空间 X^N ,每个码字想像为 N 维矢量空间 X^N 中的一个点。由于每维有 q 个取值(q 进制),因此 N 维矢量空间 X^N 中共有 q^N 个点。然而消息组 m 由 K 个 q 进制符号组成,总共只有 $M = q^K$ 个可能的组合。如果消息组与码字构成一一对应的映射关系,那么可能的码字只能有 $M = q^K$ 个。把所有码字的集合称为码集,显然,构成码集的 M 个码字均是 N 维矢量空间 X^N 中的点,即码集是矢量空间 X^N 的一个子集。从 N 维矢量空间 X^N 的 q^N 个点中选择 $M = q^K$ 点的一个子集有许多选法,通常, (N, K) 分组码的研究是借助近世代数理论寻找最佳的 $m \rightarrow c$ 映射规律,并设计出最好的编、译码器实现之。然而本节不是从最优而是从随机的角度来分析问题,通过随机编码随机地选择码集,分析其统计规律来确定其性能。

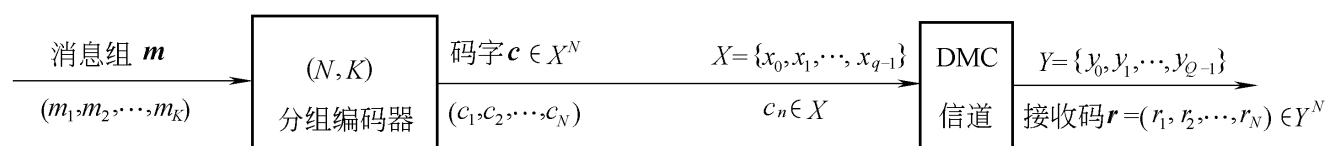


图 5 2 1 分组编码与随机编码

随机编码情况下,允许各消息组 m 随机地对应到矢量空间 X^N 的任意点,不一定是一一对应。在这种情况下,从 N 维矢量空间 X^N 的 q^N 个点中选择消息组所对应的 M 点子集(码集),总共有 q^{NM} 种选法。在这些选法中,有的码集“好”些(码字间距离大,差错概率 P_e 小),有的码集“差”些。代数编码的任务是找出那些好码,而随机编码的任务是找出统计规律,求取平均的差错概率 $\overline{P_e}$ 及它的上下界。

码集点数 M 占矢量空间 X^N 总点数的比例是

$$F = q^K / q^N = q^{-(N-K)} \quad (5.2.1)$$

显然,当 K 和 N 的差值拉大即富余的空间点数增加时,平均而言码字的分布将变得稀疏,码字间的平均距离将变大,平均差错概率 $\overline{P_e}$ 将变小。现在提出这样一个问题:当 $(N-K)$

即 $F \rightarrow 0$ 时,能否让平均差错概率 $\overline{P_e} \rightarrow 0$?

假如 M 个码集是随机地从 q^{NM} 个候选码集当中选取的,那么其中第 m 个码集(记作 $\{c\}_m$)被随机选中的概率是

$$p(\{c\}_m) = q^{-NM} \quad (5.2.2)$$

假设与这种选择相对应的条件差错概率是 $P_e(\{c\}_m)$,那么全部码集的平均差错概率是

$$\overline{P_e} = \frac{1}{q^{NM}} \sum_{m=1}^{q^{NM}} P_e(\{c\}_m) p(\{c\}_m) = q^{-NM} \sum_{m=1}^{q^{NM}} P_e(\{c\}_m) \quad (5.2.3)$$

显然,必定存在某些码集的差错概率大于平均值即 $P_e(\{c\}_m) > \overline{P_e}$,也必定存在某些码集的差错概率小于平均值。合乎逻辑的结论是,如果我们算出了 $\overline{P_e}$ 的上边界,必然有一批码集的 $P_e(\{c\}_m)$ 小于这个 $\overline{P_e}$ 的上边界;如果我们能证明在 $F \rightarrow 0$ 时 $\overline{P_e} \rightarrow 0$,就必然存在一批码集的 $P_e(\{c\}_m) \rightarrow 0$,那时我们就可以下结论说,差错概率趋于零的好码一定存在。

Gallager 在 1965 年推出了 $\overline{P_e}$ 的上边界,证明了这个上边界是按指数规律收紧的。他的推导过程是这样的:假设 X^N 中某码集 $\{c\}_m$ 的某个码字 $c_k = (c_{k1}, c_{k2}, \dots, c_{kN})$, $c_{k1}, \dots, c_{kN} \in X$, $c_k \in X^N$, 经 DMC 信道传输后变成接收码字 $r = (r_1, r_2, \dots, r_N)$, $r_1 \dots r_N \in Y$, $r \in Y^N$ 。由于 r 未必等于 c_k 原样,接收端根据 r 译码时产生的差错概率是

$$P_e(c_k) = \sum_{r \in Y^N} p(r|c_k) I_k(r) \quad (5.2.4)$$

这里, $I_k(r)$ 是示性函数,定义为

$$I_k(r) = \begin{cases} 0, & \text{if } i = k, p(r|c_k) > p(r|c_i) \\ 1, & \text{if } i = k, p(r|c_k) \leq p(r|c_i) \end{cases} \quad (5.2.5)$$

示性函数 $I_k(r)$ 的意思是:当发码字 c_k 而收到 r 的概率大于发任何其他码字 c_i 而收到 r 的概率(c_k 具有最大后验概率)时,令 $I_k(r) = 0$,因为满足这个条件时通过最优译码可以无差错地正确译码;当发码字 c_k 而收到 r 的概率小于等于发任何其它码字 c_i 而收到 r 的概率

时,令 $I_k(r) = 1$, 因为在这种情况下无法正确译码, 将发生差错, 应计入总的差错概率。 $I_k(r)$ 必定满足不等式

$$I_k(r) = \frac{p(r|c_i)^{\frac{1}{1+\epsilon}}}{p(r|c_k)^{\frac{1}{1+\epsilon}}} \quad (5.2.6)$$

这是因为当(5.2.5)式满足条件(a)时, (5.2.6)式左边等于 0, 而右边由于概率的非负性总是大于等于 0 的; 当(5.2.5)式满足条件(b)时, (5.2.6)式左边等于 1, 而右边至少有一个 $p(r|c_i) > p(r|c_k)$, 即分子大于等于分母而整个式子 ≥ 1 。式中, ϵ 是人为加入的一个修正因子, $0 < \epsilon < 1$ 。

将(5.2.6)式代入(5.2.4)式, 得

$$P_e(c_k) = \sum_{r \in Y^N} p(r|c_k) \frac{p(r|c_i)^{\frac{1}{1+\epsilon}}}{p(r|c_k)^{\frac{1}{1+\epsilon}}} = \sum_{r \in Y^N} p(r|c_k)^{\frac{1}{1+\epsilon}} \sum_{i \neq k} p(r|c_i)^{\frac{1}{1+\epsilon}} \quad (5.2.7)$$

不等式(5.2.7)就叫 Gallager 界, 它指出了码字的误码上界。

5.2.2 编码定理

为了找到有扰信道差错概率的规律, 在不等式(5.2.7)的两边对 X^N 的所有码字取平均, 由于码字及码集均是等概分布, 求得的平均值应该就是平均差错概率 $\overline{P_e}$

$$\overline{P_e} = \sum_{r \in Y^N} E \left[p(r|c_k)^{\frac{1}{1+\epsilon}} \sum_{i \neq k} p(r|c_i)^{\frac{1}{1+\epsilon}} \right] \quad (5.2.8)$$

由于各码字互相独立, 总的平均等于各项的平均, 式(5.2.8)变为

$$\overline{P_e} = \sum_{r \in Y^N} E \left[p(r|c_k)^{\frac{1}{1+\epsilon}} \right] \left\{ \sum_{i \neq k} E \left[p(r|c_i)^{\frac{1}{1+\epsilon}} \right] \right\} \quad (5.2.9)$$

又由于各码字等概, (5.2.9)式的第一项

$$E \left[p(r|c_k)^{\frac{1}{1+\epsilon}} \right] = E \left[p(r|c_i)^{\frac{1}{1+\epsilon}} \right] = \sum_c p(c) p(r|c)^{\frac{1}{1+\epsilon}}$$

其中 \sum_c 表示对某码集所有求和。 x 在 0 ~ 1 区间是上凸函数。利用 Jensen 不等式, 函数运算后取平均一定小于或等于求平均后的函数运算, 即

$$E[f(x)] \leq f[E(x)] \quad (5.2.10)$$

于是(5.2.9)式变为

$$\begin{aligned} \overline{P_e} &= \sum_{r \in Y^N} \sum_c p(c) p(r|c)^{\frac{1}{1+\epsilon}} \cdot \sum_{i \neq k} p(c) p(r|c)^{\frac{1}{1+\epsilon}} \\ &= \sum_{r \in Y^N} \sum_c p(c) p(r|c)^{\frac{1}{1+\epsilon}} \cdot (M-1) \sum_c p(c) p(r|c)^{\frac{1}{1+\epsilon}} \\ &= (M-1) \sum_{r \in Y^N} \sum_c p(c) p(r|c)^{\frac{1}{1+\epsilon}} \quad (5.2.11) \end{aligned}$$

由于信道无记忆, 码字概率等于组成该码字的各码元概率之积, 有

$$p(c) = \prod_{i=1}^N p(c_i) \text{ 及 } p(r|c) = \prod_{i=1}^N p(r_i|c_i)$$

所以(5 2 11)式变为

$$\begin{aligned} \overline{P_e} &= (M-1) \dots \dots p(c_1) p(r_1/c_1)^{\frac{1}{1+\dots}} \dots p(c_N) p(r_N/c_N)^{\frac{1}{1+\dots}} \\ &< M \dots \dots p(c_1) p(r_1/c_1)^{\frac{1}{1+\dots}} \dots \dots p(c_N) p(r_N/c_N)^{\frac{1}{1+\dots}} \\ &= M \dots \dots (c) p(r/c)^{\frac{1}{1+\dots}} \dots \dots \end{aligned} \quad (5.2.12)$$

式中 $c = X = \{x_0, x_1, \dots, x_{q-1}\}$ 及 $r = Y = \{y_0, y_1, \dots, y_{q-1}\}$ 分别代表信道发送和接收的码元符号, 仅与信道有关而与如何编码无关, 换言之, $\overline{P_e}$ 的上界仅与信道有关而与编码方式无关。(5 2 12)式可写作

$$\begin{aligned} \overline{P_e} &= M \prod_{j=1}^{Q-1} \prod_{i=1}^{q-1} p(x_i) p(y_j/x_i)^{\frac{1}{1+\dots}} \\ &= \exp \left[\ln M + N \ln \prod_{j=1}^{Q-1} \prod_{i=1}^{q-1} p(x_i) p(y_j/x_i)^{\frac{1}{1+\dots}} \right] \\ &= \exp \left[-N - \frac{\ln M}{N} - \ln \prod_{j=1}^{Q-1} \prod_{i=1}^{q-1} p(x_i) p(y_j/x_i)^{\frac{1}{1+\dots}} \right] \end{aligned} \quad (5.2.13)$$

定义码率为

$$R = \ln M / N \quad (5.2.14)$$

式中, $M = q^K$ 是可能的信息组合数; N 是每码字的码元数; R 表示每码元携带的信息量, 所以称作码率(也有人称之为传信率), 单位是每符号奈特(nat/symbol)。

又定义函数

$$E_0(\lambda, P_x) = - \ln \prod_{j=1}^{Q-1} \prod_{i=1}^{q-1} p(x_i) p(y_j/x_i)^{\frac{1}{1+\lambda}} \quad (5.2.15)$$

式中的 $E_0(\lambda, P_x)$ 是以修正因子 λ 及输入符号概率矢量 P_x 为自变量、与信道容量有关系的函数。 P_x 一定时, $E_0(\lambda, P_x)$ 和 λ 的关系如图 5 2 2 所示。从图中可知, 当 λ 由 0 变到 1 时, $E_0(\lambda, P_x)$ 是单调上升的凸函数, 其值由 $E_0(0, P_x) = 0$ 变到最大值 $E_0(1, P_x)$ 。

将(5 2 14)式、(5 2 15)式代入(5 2 13)式, 可得

$$\begin{aligned} \overline{P_e} &< \exp \{ -N [-R + E_0(\lambda, P_x)] \} \\ &< \exp \{ -N \max_{P_x} [-R + E_0(\lambda, P_x)] \} \\ &< \exp \{ -N E(R) \} \end{aligned} \quad (5.2.16)$$

式中 $E(R)$ 定义为

$$E(R) = \max_{P_x} [-R + E_0(\lambda, P_x)] \quad (5.2.17)$$

当最优的 λ 及 P_x 选定之后, $E(R)$ 仅与信道有关, 是 R 的函数。 $E(R)$ 值越大则 $\overline{P_e}$ 越小, 即可靠性越高, 所以 $E(R)$ 称作 DMC 的可靠性函数, 也称误差指数。

如果保持最优输入符号概率矢量 P_x 不变(一般是等概分布), $E(R)$ 在区间 0 ~ 1 上的极大值点位于对 λ 的偏导数等于零的地方, 即满足

$$\frac{E(R)}{\lambda} = \frac{[-R + E_0(\lambda, P_x)]}{\lambda} = -R + \frac{E_0(\lambda, P_x)}{\lambda} = 0$$

此时的码率

$$R = \frac{E_0(\rho, P_x)}{\rho}$$

(5 2 18)

极值点处 R 与 ρ 的关系如图5 2 3所示。由图可知, $\rho = 0$ 时, $E(R)$ 极值的位置在 $R = C$ 处, 这里的 C 是信道容量, 对应于 $E_0(\rho, P_x) \sim \rho$ 曲线 $\rho = 0$ 处的斜率。而在 $\rho = 1$ 时, $E(R)$ 极值的位置在 $R = R_0$ 处, R_0 叫做临界速率, 对应于图 5 2 2 曲线在 $\rho = 1$ 处的斜率, 且 $R_0 < C$ 。

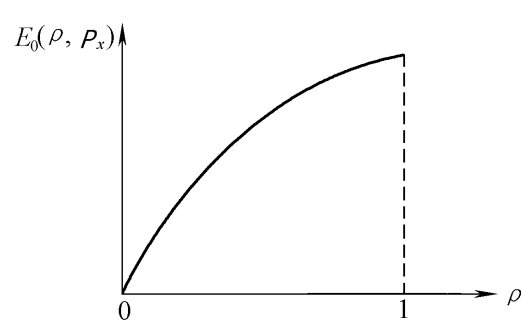


图 5 2 2 $E_0(\rho, P_x)$ 和 ρ 的关系曲线

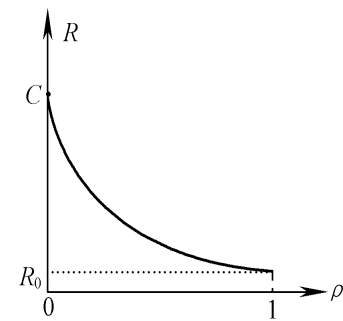


图 5 2 3 R 和 ρ 的关系曲线

如果 ρ 固定而考察 $E(R)$ 和 R 的关系, 由式(5 2 17)可得

$$\frac{E(R)}{R} = -$$

(5 2 19)

可见 ρ 固定时, $E(R) \sim R$ 关系曲线是一条斜率为 $-\frac{E(R)}{R}$ 的直线; ρ 变化时, $E(R) \sim R$ 关系曲线的斜率为 $-\frac{E(R)}{R}$, 完整的 $E(R) \sim R$ 曲线如图 5 2 4 所示。从图中看, R 在 $[0, R_0]$ 区间时(此时恒有 $\rho = 1$, 见图 5 2 3), $E(R) \sim R$ 曲线是斜率为 -1 (-45°) 的直线; R 在 $[R_0, C]$ 区间时(此时 ρ 从 $1 \rightarrow 0$), 曲线斜率从 $-1 \rightarrow 0$; 而当 $R = C$ 时(此时 $\rho = 0$), $E(R) = 0$ 。

(5 2 16) 式左边的 \bar{P}_e 是平均差错概率, 而随机码中总有某些码集, 它们的差错概率 P_e 小于平均差错概率 \bar{P}_e 。所以可以断言: 一定存在某种编码方式, 满足

$$P_e < e^{-NE(R)}$$

(5 2 20)

(5 2 20) 式正是有扰离散信道的信道编码定理。用文字叙述其内涵, 就是: 只要传信率 R 小于信道容量 C , 总存在一种信道码(及解码器), 可以以所要求的任意小的差错概率实现可靠的通信。

后来 Fano 推导了一个 Fano 不等式, 并利用它推出了信道编码逆定理: 信道容量是可靠通信系统传信率 R 的上边界, 如果 $R > C$, 就不可能有任何一种编码能使差错概率任意小。

这两个定理常被写在一起统称为有扰或噪声信道的信道编码定理。

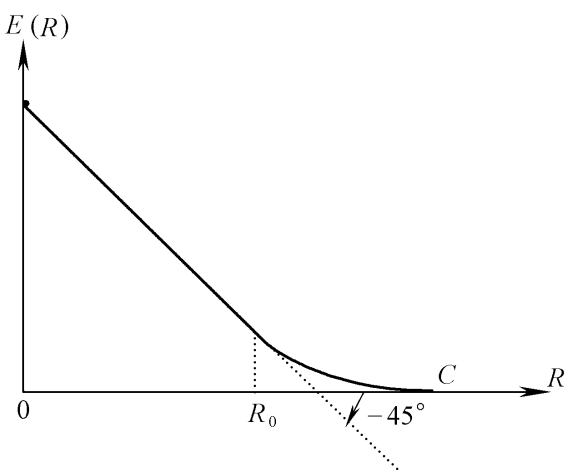


图 5 2 4 $E(R)$ 和 R 的关系曲线

5.3 差错控制与信道编译码的基本原理

5.3.1 差错控制的途径

下面从公式和概念两条途径来论述差错控制与信道编译码的基本原理。

1. 途径一

从信道编码定理的公式出发,可知减小差错概率应增大码长 N 或增大可靠性函数 $E(R)$, 而想增大 $E(R)$ 就要加大信道容量 C 或减小码率(传信率) R 。从图 5.3.1 可以看出:

对于同样的码率,信道容量大者其可靠性函数

$E(R)$ 也大;

对于同样的信道容量,码率减小时其可靠性函数

$E(R)$ 增大。

鉴于上面的分析,可采取以下措施减小差错概率。

(1) 增大信道容量 C

根据信道容量公式(5.1.30),信道容量 C 与带宽 W 、信号平均功率 P_{va} 和噪声谱密度 N_0 有关。为此,可以

扩展带宽。比如开发新的宽带媒体,有线通信从明线(150 kHz)、对称电缆(600 kHz)、同轴电缆(1 GHz)

到光纤(25 THz),无线由中波、短波、超短波到毫米波、微米波。又比如采取信道均衡措施如加感、时/频域的自适应均衡器等。

加大功率。如提高发送功率、提高天线增益、将无方向的漫射改为方向性强的波束或点波束、分集接收等。

降低噪声。如采用低噪声器件、滤波、屏蔽、接地、低温运行等。

在纠错编码技术发展之前,通信系统设计者传统上主要就是靠增大 C 来提高通信可靠性的。

(2) 减小码率 R

对于二进制(N, K)分组码(K 位二进符号编成由 N 位符号组成的码字),码率是 $R = K/N$ 比特/符号;对于 q 进制(N, K)分组码(K 个 q 元符号编成 N 个 q 元符号),码率是 $R = K \log_2 q / N$ 。所以降低码率的方法有

q, N 不变而减小 K ,这意味着降低信息源速率,每秒少传一些信息。

q, K 不变而增大 N ,这意味着提高符号速率(波特率),占用更大带宽。

N, K 不变而减小 q ,这意味着减小信道的输入、输出符号集,在发送功率固定时提高信号间的区分度,从而提高可靠性。

在一定的通信容量下减小 R ,等效于拉大 C 和 R 之差,因此说这是用增加信道容量的冗余度来换取可靠性。从 50 年代到 70 年代,主要的纠错编码方法都是以这种冗余度为基础的。

(3) 增加码长 N

如要保持码率 R 不变,增加码长 N 的同时应增大信息位 K ,以保持 K/N 之比不变。

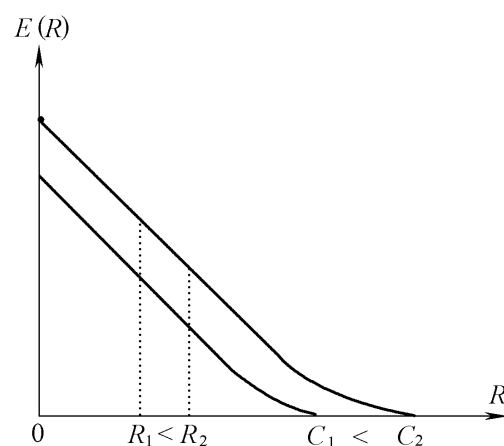


图 5.3.1 增大 $E(R)$ 的途径

在 C 和 R 固定情况下加大 N 并没有增加信道容量的冗余度,它是利用了随机编码的特点:随着 N 增大,矢量空间 X^N 以指数量级增大,从统计角度而言码字间距离也将加大,从而可靠性提高。另外,码长 N 越大,其实际差错概率就越能符合统计规律。比如投掷一个硬币记录其正面向上的比例,理论值应是 0.5,如果比例降到 0.4 以下或 0.6 以上就算差错,则投掷 10 次就统计比例较之投掷 100 万次再统计比例,其差错概率要大得多。可以断言,投掷 100 万次而正面向上比例在 $[0.4, 0.6]$ 区间之外的概率几乎是零,增加码长 N 的作用与增加投掷次数的作用类似。增加码长 N 所带来的好处同样需要付出某种代价才能换得,代价就是码长越长,编解码算法就越复杂,编解码器也越昂贵。所以,虽然香农早在 1948 年就已指出增大 N 的途径,但 70 年代前由于器件水平不允许编解码器做得太复杂,实用的纠错码主要还是靠牺牲功率、带宽效率来取得可靠性。80 年代后随着 VLSI 的发展,编解码器可以做得越来越复杂,很多编解码算法可在 ASIC 或数字信号处理专用芯片 DSP 上实现,因此码长允许设计得很长。当前,通过增加码长 N 来提高可靠性已成为纠错编码的主要途径之一,它实际上是以设备的复杂度换取可靠性,从这个意义上说,妨碍数字通信系统性能提高的真正限制因素是设备的复杂性。

2. 途径二

从概念上分析纠错编码的基本原理,可以把纠错能力的获取归结为两条,一条是利用冗余度,另一条是噪声均化(随机化)。

(1) 利用冗余度

冗余度就是在信息流中插入冗余比特,这些冗余比特与信息比特之间存在着特定的相关性。这样,在传输过程中即使个别信息比特遭受损伤,也可以利用相关性从其他未受损的冗余比特中推测出受损比特的原貌,保证了信息的可靠性。举例来说,如果用 2 比特表示 4 种意义,那么无论如何也不能发现差错,因为如有一信息 01 误成 00,根本无法判断这是在传输过程中由 01 误成 00,还是原本发送的就是 00。但是,如果用 3 比特来表示 4 种意义,那就有可能发现差错,因为 3 比特的 8 种组合能表示 8 种意义,用它代表 4 种意义尚剩 4 种冗余组合,如果传输差错使收到的 3 比特组合落入 4 种冗余组合之一,就可断言一定有差错比特发生了。至于加多少冗余、加什么样的相关性最好,这正是纠错编码技术所要解决的问题,但必须有冗余,这是纠错编码的基本原理。

为了传输这些冗余比特,必然要动用冗余的资源。这些资源可以是:

时间。比如一个比特重复发几次,或一段消息重复发几遍,或根据收端的反馈重发受损信息组,如 ARQ (automatic repeat request) 系统。

频带。插入冗余比特后传输效率下降,若要保持有用信息的速率不变,最直接的方法就是增大符号传递速率(波特率),结果就占用了更大的带宽。比如采用二进码(1 比特/符号),编成(8,4)分组码后使符号速率增大一倍,所占带宽也增大一倍。

功率。采用多进制符号,比如用一个八进制 ASK 符号代替一个四进制 ASK 符号来传送 2 比特信息,可腾出位置另传 1 冗余比特。但为了维持信号集各点之间的距离不变,八进制 ASK 符号的平均功率肯定比四进制时要大,这就是动用冗余的功率资源来传输冗余比特。

设备复杂度。加大码长 N ,采用网格编码调制(TCM),是在功率、带宽受限信道中实施纠错编码的有效方法,代价是算法复杂度的提高,需动用设备资源。

(2) 噪声均化

纠错编码的第二条基本原理是噪声均化,或者说让差错随机化,以便更符合编码定理的条件从而得到符合编码定理的结果。噪声均化的基本思想是设法将危害较大的、较为集中的噪声干扰分摊开来,使不可恢复的信息损伤最小。这是因为噪声干扰的危害大小不仅与噪声总量有关,而且与它们的分布有关。举例来说,(7,4)汉明码能纠一个差错,假设噪声在14比特(两个码字)上产生2个差错,那么差错的不同分布将产生不同后果。如果2个差错集中在前7比特(一个码字上),该码字将出错。如果在前7比特出现一个差错、后7比特也出现一个差错,则每码字中差错比特的个数都没有超出其纠错能力范围,这两个码字将全部正确解码。由此可见,集中的噪声干扰(称之为突发差错)的危害甚于分散的噪声干扰(称之为随机差错)。噪声均化正是将差错均匀分摊给各码字,达到提高总体差错控制能力的目的。

噪声均化的方法主要有3条。

增加码长 N 。前面已从编码公式角度提到过这种方法,这里想通过一个具体例子从概念上理解它。如图 5 1 2 所示的某 BSC 信道误码概率 $P_e = 0.01$,假如编码后的纠错能力是 10%,即长度 N 的码字中,只要差错码元个数少于等于 N 的 10%,就可以通过译码加以纠正。若码长 $N = 10$,则码字中多于 1 个码元出错时就会产生译码差错,差错概率为

$$P = 1 - \sum_{m=0}^1 \frac{10}{m} P_e^m (1 - P_e)^{10-m} = 4.27 \times 10^{-3}$$

如果保持码率 R 不变,将码长增加到 $N = 40$,那么当码字中多于 4 个码元出错时就会产生译码差错,差错的概率为

$$P = 1 - \sum_{m=0}^4 \frac{40}{m} P_e^m (1 - P_e)^{40-m} = 4.92 \times 10^{-5}$$

从本例看到,只要将码长由 10 增加到 40,译码误差的概率就可以下降二个数量级。增加码长可使译码误差减小的原因在于:码长越大,具体每个码字中误码元的比例就越接近统计平均值,换言之,噪声按平均数被均摊到各码字上。而如果真的均摊了,译码就不会发生任何差错,因为信道的差错概率($P_e = 1\%$)远远小于编码后的纠错能力 10%。

卷积。上面的例子都是把信息流分割成 K 位一组,每组再编成 N 长的码字,也就是说相关性仅限于加在各个码字内,而码字之间是彼此无关的。后来卷积码的出现改变了这种状况,卷积码在一定约束长度内的若干码字之间也加进了相关性,译码时不是根据单个码字、而是一串码字来作判决。如果再加上适当的编译码方法,就能够使噪声分摊到码字序列而不是一个码字上,达到噪声均化目的。

交错(或称交织),是对付突发差错的有效措施。突发噪声使码流产生集中的、不可纠的差错,若能采取某种措施,对编码器输出的码流与信道上的符号流作顺序上的变换,则信道噪声造成的符号流中的突发差错,有可能被均化而转换为码流上随机的、可纠正的差错。加了交错器的传输系统如图 5 3 2 所示。

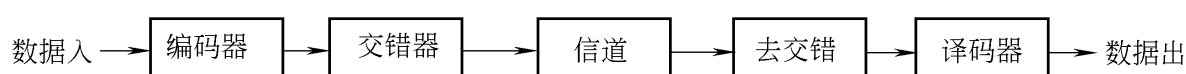


图 5 3 2 带交错器的传输系统

交错的效果取决于信道噪声的特点和交错方式。最简单的交错器是一个 $n \times m$ 的存

储阵列,码流按行输入后按列输出。图 5 3 3 是一个适用于码长 $N = 7$ 的 5×7 行列交错器的示意图,从图中看到,码流的顺序 $1, 2, 3 \dots, 7, 8, \dots$ 经交错器后变为 $1, 8, 15, 22, 29, 2, 9, \dots$ 。现假设信道中产生了 5 个连续的差错,如果不交错,这 5 个差错集中在 1 个或 2 个码字上,很可能就不可纠。采用交错方法,则去交错后差错分摊在 5 个码字上,每码字仅 1 个。

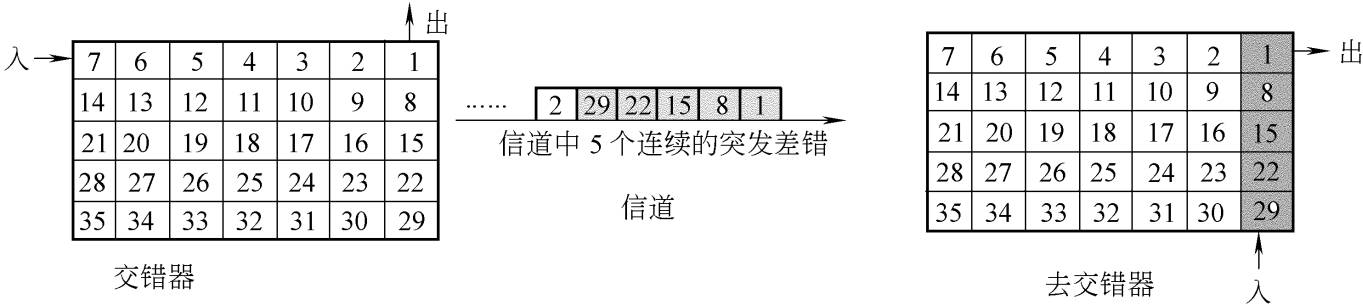


图 5 3 3 5×7 行列交错器工作原理示意图

5 3 2 码距与纠、检错能力

在 5 2 1 节关于随机编码的论述中,码字 $\mathbf{c} = (c_1, c_2, \dots, c_N)$ 可视作一个 N 重矢量,每个码字与 N 维矢量空间 X^N 中的一个点对应,全部码字所对应的点的集合构成矢量空间里的一个子集,该子集所包含的点只是全部 N 维空间点的一部分。当传输无误时,接收到的 N 重矢量一定是码字,在矢量空间中一定对应到该子集相应的点上。但当出现差错时,接收的 N 重矢量有两种可能:一种是不再对应到该子集,而是对应到与子集点相邻的另一个空间点上;第二种是仍然对应到该子集,却对应到该子集的另一点上。前一种情况下尚能发现对应点不在子集上从而判断出差错的存在,而后一种情况下根本无法判断是传输发生差错还是原本发送的就是另一个码字。子集的任意两点间都存在一定距离,设子集两点间的最小距离是 d_{\min} ,那么,一个能使空间点位置偏移 d_{\min} 的差错组合(称之为重量 d_{\min} 的差错图案)有可能把接收矢量所对应的空间点位置从子集的一个点偏移到另一个点,导致从一个码字译成另一个码字。在这种情况下(就是上面所说的第二种情况)下,就说是产生了一个“不可检的差错”。另一方面,如果差错数小于 d_{\min} ,就不可能从子集一个点偏移到子集另一个点,就可以检测出差错的存在。显然,对于 (n, k) 分组码而言,我们有能力检测出 $d_{\min} - 1$ 个差错。

码的纠错能力也同样取决于最小距离。为了确定 (n, k) 分组码的纠错能力,一种方便的办法是把 2^k 个码字看作是位于 n 维空间的点。如果以每个码字为球心,以汉明距离 t 为半径作 2^k 个球体,那么使它们之中任意一对球体两两不相交(包括不相切)的 t 的最大取值是 $t = \text{INT}[(d_{\min} - 1)/2]$,这里 $\text{INT}[\cdot]$ 表示取整。在每一个球内,含有与该码字距离小于等于 t 的所有可能的接收码字。译码时,所有落在球内的接收码字都被译为位于球心的那个码字。这就意味着:最小距离为 d_{\min} 的 (n, k) 分组码有能力纠正 $t = \text{INT}[(d_{\min} - 1)/2]$ 个差错,纠错能力总是小于检错能力。图 5 3 4 是码字和球的两维示意图。

如上所述,分组码若单独考虑检错或单独考虑纠错,则可检 $d_{\min} - 1$ 个差错或纠 $t = \text{INT}[(d_{\min} - 1)/2]$ 个差错,然而若将两者放在一起统一考虑,则情况有所变化。能纠 t 个差错显然能检 t 个差错,但若想检 $d_{\min} - 1$ 个差错就必须对码的纠错能力有所抑制。例如,假设两码字 A, B 相距 7 ($d_{\min} = 7$),若码字 A 发生 3 个差错时能够纠正过来,若发生 4 个差

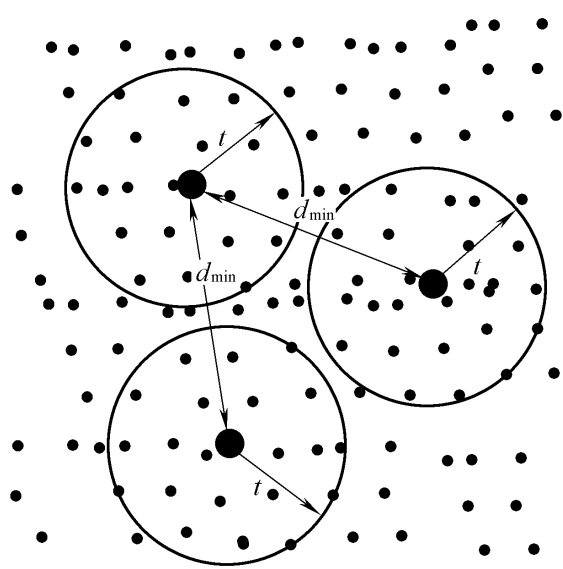


图 5 3 4 以码字为球心、半径 $t = \text{INT}[(d_{\min} - 1)/2]$ 的差错控制球体的示意图

错以致对应点与码 A 距离 4 而与码 B 距离 3 时(从 A 球范围落入 B 球范围),译码器就会认为接收点是由码 B 发生 3 个差错而来的,译码输出将是码 B ,而并不认为是码 A 发生了 4 个差错,换言之,此时译码器的检错能力也只有 3。若要有检测 4 个差错的能力,只能把球半径从 3 减为 2,这样,差错数不大于 2 的接收码可纠,差错数在 3、4 之间时可检,而 4 个以上的差错可能使接收点落入别的球中,将是不可检的差错。同理,如 $d_{\min} = 7$ 不变,也可以让它检 5 个差错、纠 1 个差错。一般性的结论是,若最小距离 d_{\min} 的码同时能检 e_d 个、纠 e_c 个差错,则必有

$$e_d + e_c = d_{\min} - 1 \quad (5.3.1)$$

及

$$e_c = e_d \quad (5.3.2)$$

至此已从概念上定性地说明了码的纠错能力取决于码的距离。对于线性码来说,码距特性也就是码的重量特性(码重可视为该码与全零码的距离)。如果有 2^k 个码字,就存在 $2^k(2^k - 1)/2$ 个距离,这些距离是大小不一的,码的总体性能取决于这些距离的分布特性(重量谱),而纠错能力取决于其中的最小者 d_{\min} 。正如各符号等概时熵最大一样,从概念上可以联想到:当所有码距相等时码的性能应该最好,或者退一步,当各码距相差不大时性能应该较好。事实也确实是如此,但距离分布与性能之间的确切定量关系对于大部分码而言尚在进一步研究之中。

对于转移概率为 p 的 BSC 信道(如图 5.1.2),由于无记忆,各比特差错的发生是独立的,如果不加纠错编码, k 位信息组出错的概率是

$$P_M = 1 - (1 - p)^k \quad (5.3.3)$$

如果采用纠错能力为 t 的 (n, k) 分组码,当差错个数在 $(t + 1)$ 到 n 之间时该码可能出错,因此差错概率的上限为

$$P_M \leq \sum_{m=t+1}^n \binom{n}{m} p^m (1 - p)^{n-m} \quad (5.3.4)$$

5.3.3 最优译码与最大似然译码

译码器的任务是从受损的信息序列中尽可能正确地恢复出原信息。作为译码器的输入,译码算法的已知条件是实际接收到的码字序列 $\{r\}$, $r = (r_1, r_2, \dots, r_N)$ 。发端所采用的编码算法和该算法产生的码集 X^N , 满足 $G_i = (c_{i1}, c_{i2}, \dots, c_{iN}) \in X^N$ 。信道模型及信道参数。其中 G_i 是必要条件,而 X^N 尽管为译码算法提供了分析、选择的依据,但并非所有译码过程都直接用到它。译码器译码时,先根据接收序列 $\{r\}$ 解得发送码字序列 $\{c_i\}$ 的估值序列 $\{\hat{c}_i\}$,再实行编码的逆过程从码字估值序列 $\{\hat{c}_i\}$ 还原出消息序列 $\{m_i\}$,如图 5.3.5 所

示。 $\{r\}$ $\{\hat{c}_i\}$ $\{\hat{m}_i\}$ 的过程是从功能角度描述的,具体实现时可综合到译码算法中一次完成。由于从 $\{\hat{c}_i\}$ 可唯一地解得 $\{\hat{m}_i\}$,所以还原的消息正确与否取决于 $\{\hat{c}_i\}$ 是否等于 $\{c_i\}$ 。

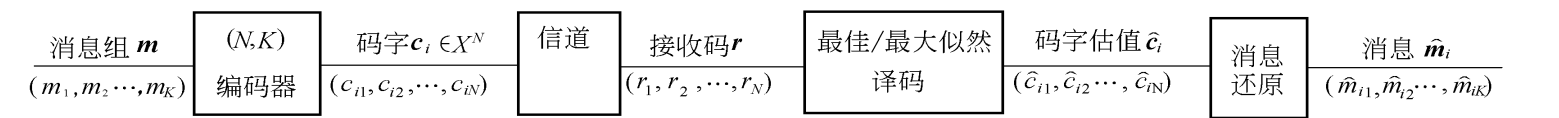


图 5 3 5 译码过程

在已知 r 的条件下找出可能性最大的发码 c_i 作为译码估值 \hat{c}_i , 即令

$$\hat{c}_i = \max p(c_i/ r) \tag{5 3 5}$$

这种译码方法叫做最佳译码,也叫最大后验概率译码(MAP: Maximum Aposteriori),它是一种通过经验与归纳由收码推测发码的方法,是我们认为的最优译码算法。但在实际译码时,后验概率的定量确定是很困难的。比如在图 5 1 2 的 BSC 信道或图 5 1 3 的 DMC 信道模型里,只告诉我们信道的前向(发 收)转移概率即先验概率,却并没有告诉我们信道的后向(收 发)转移概率即后验概率。在已知 r 的条件下使先验概率最大的译码算法叫最大似然(ML: Maximum Likelihood)译码,即令

$$\hat{c}_i = \max p(\uparrow c_i) \tag{5 3 6}$$

因此 $p(\uparrow c_i)$ 也叫似然函数。根据贝叶斯公式可以建立先验概率和后验概率之间的关系

$$p(c_i/ r) = \frac{p(c_i) p(\uparrow c_i)}{p(r)} \qquad i = 1, 2, \dots, q^k \tag{5 3 7}$$

式中, $p(c_i)$ 是发送码字 c_i 的概率, $p(r)$ 是接收码为 r 的概率, $p(\uparrow c_i)$ 是先验概率, $p(c_i/ r)$ 是后验概率。

如果

构成码集的 q^k 个码字以相同概率发送,满足 $p(c_i) = 1/ q^k, i = 1, 2, \dots, q^k$;

$p(r)$ 对于任何 r 都有相同的值,满足 $p(r) = 1/ q^N$ 。

则 $p(c_i/ r)$ 的最大等效于 $p(\uparrow c_i)$ 的最大,此时最大后验概率译码等同最大先验概率译码,或者说最佳译码等同最大似然译码。理论上,通过信源编码算法的改进及扰码、交织的采用可使发码 c_i 等概化,令信道对称均衡而使收码 r 也等概化,从而可用最大似然译码替代最佳译码。实践上尽管不能做到 c_i 、 r 两者的完全等概,但最大似然译码仍是可行的最好、最常用方法。

对于无记忆信道,码字的似然函数 $p(\uparrow c_i)$ 等于组成该码字的各码元的似然函数之积(联合概率),码字的最大似然也就是各码元似然函数之积的最大化,

即若

$$r = (r_1, r_2, \dots, r_N), c_i = (c_{i1}, c_{i2}, \dots, c_{iN})$$

则

$$\max p(\uparrow c_i) = \max_{j=1}^N p(r_j/ c_{ij}) \tag{5 3 8}$$

为了将乘法运算简化为加法运算,取似然函数的对数,称作对数似然函数。由于对数的单调性,似然函数最大时对数似然函数也最大。于是,码字对数似然函数最大化等效于各码元对数似然函数之和的最大化,即

$$\max \log p(\mathbf{r} | \mathbf{c}_i) = \max_{j=1}^N \log p(r_j | c_{ij}) \quad (5.3.9)$$

上式的对数可以 e 为底 (自然对数), 也可以 2 或 10 为底。

作为一个特例, BSC 信道的最大似然译码可以简化为最小汉明距离译码。这是因为当我们逐比特地比较发码和收码时, 仅存在两种可能性: 相同或不同, 两种情况发生的概率分别是

$$p(r_j | c_{ij}) = \begin{cases} p & (c_{ij} \neq r_j \text{ 时}) \\ 1 - p & (c_{ij} = r_j \text{ 时}) \end{cases} \quad (5.3.10)$$

如果 \mathbf{r} 中有 d 个码元与 \mathbf{c}_i 的码元不同, 我们说 \mathbf{r} 与 \mathbf{c}_i 的汉明距离是 d 。显然, d 代表 \mathbf{c}_i 在 BSC 信道传输过程中的码元差错个数, 也就是 \mathbf{r} 与 \mathbf{c}_i 模 2 加后的重量

$$d = \text{dis}(\mathbf{r}, \mathbf{c}_i) = W(\mathbf{r} \oplus \mathbf{c}_i) = \sum_{j=1}^N r_j \oplus c_{ij} \quad (5.3.11)$$

此时的似然函数是

$$p(\mathbf{r} | \mathbf{c}_i) = \prod_{j=1}^N p(r_j | c_{ij}) = p^d (1 - p)^{N-d} = \frac{p^d}{1 - p} (1 - p)^N \quad (5.3.12)$$

$(1 - p)^N$ 是常数而 $p^d (1 - p)^{N-d}$, d 越大则似然函数 $P(\mathbf{r} | \mathbf{c}_i)$ 越小, 因此求最大似然函数 $\max p(\mathbf{r} | \mathbf{c}_i)$ 的问题转化成求最小汉明距离 d 的问题。

汉明距离译码是一种硬判决译码。只要在接收端将收码 \mathbf{r} 与发码 \mathbf{c}_i 的各码元逐一作比较, 选择其中汉明距离最小的码字作为译码估值 $\hat{\mathbf{c}}_i$ 。由于 BSC 信道是对称的, 只要发送的码字独立、等概, 汉明距离译码也就是最佳译码。

5.4 线性分组码

5.4.1 线性分组码基本概念

分组码是由一组固定长度 n 、称之为码字的矢量构成的, 正如已在 5.2.1 节中描述的那样。每个矢量元素就是一个符号 (symbol), 也称一个码元, 其值选自 q 个元素组成的字符集 $X = \{x_0, x_1, \dots, x_{q-1}\}$ 。当 $q = 2$ 比如 $X = \{0, 1\}$ 时, 该码就是二进制码; 当 $q > 2$ 时, 该码为 q 进制码或 q 元码。对二元码而言, 符号、码元、比特可以混用, 尽管在严格意义上比特与码元代表不同概念。除此之外在实用中也常取 q 等于 3 或 q 等于 2 的幂次。当 $q = 2^b$ (b 是正整数) 时, 每个 q 进制码元可以用对应的、包含 b 比特的二进制码来表示, 即分组长度为 N 的 q 元码可以映射成分组长度为 $n = bN$ 的二进制分组码。

长度为 n 的二进制分组码有 2^n 种可能的组合, 选择其中的 2^k 种 ($k < n$) 构成一个许用码的码集 C 。编码就是将 k 比特信息组一一对应地映射到许用码码集, 不同的编码算法对应不同的映射方法, 把这样得到的分组码称为 (n, k) 码, 并定义 $k/n = R_c$ 为码率。上述概念可推广到一般, 对长度 n 的 q 元码而言, 有 q^n 种可能的组合, 选择其中的 q^k 种构成一个码集来传送长度为 k 的信息分组。

决定分组码纠错能力的一个重要参数是码字的重量 (组成该码字的非零元素的个数)。通常, 各码字重量不等, 对码集的所有码字进行统计可得该码的重量分布。如果码集的所有码字都具有相同的重量, 这种码就叫做恒重码。5 码元的国内电报码和 7 码元的国际电报

码都是采用恒重码,接收端通过检测非零元素的个数来判断码字是否出错。

k 比特信息组对于码集的映射是通过乘、加运算决定的,运算的规则服从元素所在字符集代数域的惯例,比如二进制字符的运算服从二元域的布尔代数。推广为一般,在由一个元素集合、两种算术运算构成的域 $F(\text{field})$ 中,算术运算必须满足以下规则:

(1) 加法

集合 F 在加法运算下是封闭的,即如有 $a, b \in F$,必有 $a + b \in F$ 。

满足加法结合律,即如有 $a, b, c \in F$,则 $a + (b + c) = (a + b) + c$ 。

满足加法交换律,即 $a + b = b + a$ 。

集合中一定包含一个零元素,满足条件 $a + 0 = a$ 。

集合中的每个元素都有其对应的逆元素。若 b 是一个元素,那么它的逆元素记作 $-b$ 。两元素的减法,比如 $a - b$,被定义为 $a + (-b)$ 。

(2) 乘法

集合 F 在乘法运算下是封闭的,即如有 $a, b \in F$,必有 $ab \in F$ 。

满足乘法结合律,即 $a(bc) = (ab)c$ 。

满足乘法交换律,即 $ab = ba$ 。

满足乘法分配律,即 $(a + b)c = ac + bc$ 。

集合中一定包含一个单位元 $I(\text{identity})$,使对于任何 $a \in F$,满足条件 $aI = a$ 。

除零元素外,集合 F 的每个元素都有逆元素。因此,若 $b \in F (b \neq 0)$,那么它的逆元素记作 b^{-1} 。两元素的除法,比如 $a \div b$,被定义为 ab^{-1} 。

近世代数已对群、环、域的概念作了详细描述。大家所熟悉的实数域和复数域含有无穷个元素,叫无限域。 $0, 1$ 构成的比特世界是二元域,符合上述运算规则的 q 个元素构成的是 q 元域。由有限个元素构成的域叫有限域,有限域通常称为伽罗华域 (Galois field),用 $GF(q)$ 来表示。每个域必须要有一个零元和一个单位元(也称幺元),最简单的域是二元域 $GF(2)$ 。通常,只有当 q 是素数或素数的幂时才能构成有限域 $GF(q)$,其域元素是 $\{0, 1, \dots, q-1\}$,与该 q 元域对应的加、乘运算为模 q 运算。例如 $GF(5)$ 是由元素 $\{0, 1, 2, 3, 4\}$ 构成的 5 元域,该域中的加法和乘法运算举例如下:

$$3 + 4 = 2(\text{mod}5), 3 + 2 = 0(\text{mod}5), 3 + 4 = 2(\text{mod}5), 3 + 2 = 1(\text{mod}5)$$

如果 $q = p^m$ (p 是素数, m 是任意正整数),则有可能将 $GF(q)$ 域扩展成 $GF(q^m)$,称为 $GF(q)$ 的扩域,扩域中元素的乘、加运算也是基于模 q 运算。

编码除了可划分为二进制或非二进制码以外,也可用是否线性来描述它。假设 C_i, C_j 是某 (n, k) 分组码的两个码字; c_1, c_2 是码元字符集里的任意两个元素,那么当且仅当 $c_1 C_i + c_2 C_j$ 也是码字时,才称该码是线性码,现实中采用的分组码绝大多数是线性的。作为加法零元的 0 是不可缺少的域元素,当然可以选 $c_1 = 0$ 和 $c_2 = 0$,由此可导出这样一个结论:凡线性码必须包含全零码字。据此推断,恒重码是非线性的。若令 c_1, c_2 各等于单位元,则 $C_i + C_j$ 也是码字,这就体现了线性分组码的封闭性,即码字的组合仍然是码字(封闭在码集 C 中),在这种情况下

$$d = \text{dis}(C_i, C_j) = W(C_i \oplus C_j) = W(C_k) = \text{dis}(C_k, 0) \quad (5.4.1)$$

其中 0 表示全零码。(5.4.1)式说明

两码字的距离必定等于某一码字的重量,因此,线性分组码的最小距离 d_{\min} 等于码集中非零码字的最小重量,即

$$d_{\min} = \min_{\substack{C_i \\ C_i \neq 0}} W\{C_i\} \quad (5.4.2)$$

研究两两码字间的距离特性,可用各码与全零码的距离,或各码自身的重量来代替。

最小距离 d_{\min} 决定了分组码的纠、检错能力,正如 5.3.2 节分析过的那样,分组码的最大检错能力是 $(d_{\min} - 1)$,最大纠错能力是 $t = \text{INT}[(d_{\min} - 1)/2]$ 。

线性分组码的理论基础是近世代数,因此分组码也被称作代数码。后来发现有的分组码也可以从几何角度来分析,所以又有代数码、几何码之分。无论从什么角度,线性代数中的许多基本概念,特别是矢量空间的概念,是非常有用的。矢量空间理论指出:以 n 个线性无关的矢量为基底,它们的全部线性组合可构成一个 n 维 n 重矢量空间 S ,这里 n 重(n tuples)指 n 个 $\text{GF}(q)$ 域元素的有序排列。如果从 n 个基底中选出一组 k ($k < n$) 个基底,则它们所有的线性组合也构成一个集合,这个集合是 S 的一个 k 维子集,称为 k 维 n 重子空间,记作 S_c 。举例来说,以 $(1,0,0), (0,1,0), (0,0,1)$ 为 3 个基底可构成一个 3 维 3 重空间,以 $(1,0,0), (0,1,0)$ 为 2 个基底可构成一个 2 维 3 重空间,以 $(0,0,1)$ 为基底可构成一个 1 维 3 重空间。对于二元域 $\text{GF}(2)$ 而言,2 维 3 重空间包含 $(0,0,0), (1,0,0), (0,1,0), (1,1,0)$ 四个矢量,1 维 3 重空间包含 $(0,0,0), (0,0,1)$ 两个矢量。每个矢量可对应一个码长 $n=3$ 的码字,所以也可把矢量说成是一个码矢,一个码字,或一个 n 重。我们注意到:每个子空间都包含全 0 码矢,这与上面“凡线性码必须包含全零码字”的结论出于同一道理。 n 维空间必定有 n 个基底,但这 n 个基底不是唯一的。比如在二元域 $\text{GF}(2)$ 中,用 $(1,0), (0,1)$ 可以张成一个 2 维 2 重空间,用 $(1,1), (0,1)$ 同样可以张成一个二元域的 2 维 2 重空间,因为 $(1,1)$ 和 $(0,1)$ 是线性无关的,同样可用作基底。这个道理,正如以水平、垂直单位矢量为基可张成一个直角坐标平面,而把两单位矢量作同样旋转、位移后仍然能张成一个直角坐标平面一样。如果两矢量的点积(内积)为零,则称之为矢量正交。如果一矢量空间中的任一矢量都与另一矢量空间中的任一矢量正交,则称这两矢量空间正交。以互相正交的基底组张成的两个矢量空间一定正交,这两个空间称为对偶空间(dual space),其中一个空间是另一个空间的零空间(null space,也称零化空间)。若把 n 维 n 重矢量空间中互相正交的 n 个基底分成两组,一组 k 个基底,另一组 $n-k$ 个基底,则它们分别张成 k 维 n 重和 $(n-k)$ 维 n 重两个正交的对偶空间。

用空间的概念表达二进制分组码,我们说在 n 维 n 重空间 S 中含有 2^n 个长度 n 的矢量(二进制 n 重),其中至少存在一组 n 个矢量 B_1, B_2, \dots, B_n 可用作基底, n 维 n 重空间中的所有矢量都是这些基底的线性组合

$$B = \sum_{i=1}^n b_i B_i \quad b_i \in \{0,1\}, i=1,2,\dots,n \quad (5.4.3)$$

如果除了系数取全零($b_1 = b_2 = \dots = b_n = 0$)外不能使它等于 0,则称 B_1, B_2, \dots, B_n 线性无关,否则就是线性相关。若把上述 n 个基底分成 k 和 $n-k$ 个两组,可分别张成 k 维 n 重、 $(n-k)$ 维 n 重两个对偶空间,其中 k 维 n 重空间的 2^k 个矢量由 k 个基底的线性组合形成:

$$B = \sum_{i=1}^k b_i B_i \quad b_i \in \{0,1\}, i=1,2,\dots,k$$

把 k 维 n 重空间用作码空间 C ,将 $(n-k)$ 维 n 重空间用作校验空间 H 。分组编码器的作用,是把 k 维 k 重信息组空间的 2^k 个矢量一一对应到 k 维 n 重码空间 C ,如图 5.4.1 所示。

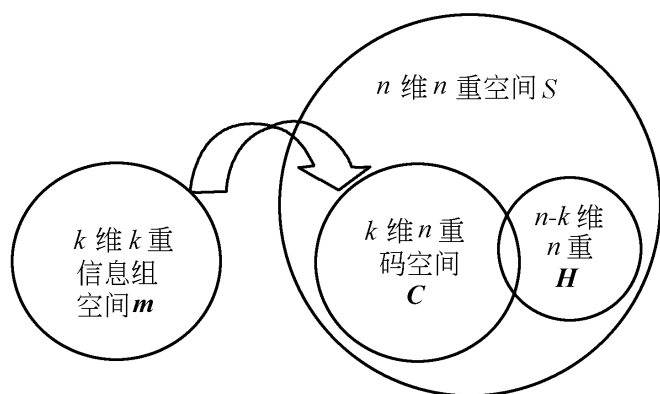


图 5 4 1 码空间与映射

因此,编码算法的核心问题是:

如何确定码空间,也就是如何选择 B_1, B_2, \dots, B_k 的 k 个基底。

k 维 k 重信息组空间的 2^k 个矢量以什么法则一一对应到 k 维 n 重码空间 C 。

不同的基底选择方法,不同的矢量映射规则,就形成了形形色色不同的编码方法。基底的排列可化为矩阵,矩阵运算隐含着映射规则,由此,引出了生成矩阵和校验矩阵的概念。

5 4 2 生成矩阵和校验矩阵

设 m_1, m_2, \dots, m_k 是一组 k 个信息比特,将它对应到 k 维 k 重信息组空间 m ,可写成行矢量的形式 $m = (m_1 \ m_2 \ \dots \ m_k)$ 。编码器输出的码字是 k 维 n 重码空间 C 中的矢量,记为 $C_i = (c_{i1} \ c_{i2} \ \dots \ c_{in})$ 。对于二进制线性分组码,编码运算可以用一组 n 个如下形式的方程表示

$$c_{ij} = m_1 g_{1j} + m_2 g_{2j} + \dots + m_k g_{kj}, \quad j = 1, 2, \dots, n \quad (5.4.4)$$

式中 $g_{ij} \in \{0, 1\}$, $m_i g_{ij}$ 表示 m_i 和 g_{ij} 的乘积。线性方程组(5.4.4)也可以用矩阵形式表示:

$$C_i = (c_{i1} \ c_{i2} \ \dots \ c_{in}) = mG = (m_1 \ m_2 \ \dots \ m_k) \cdot (g_1 \ g_2 \ \dots \ g_k)^T \quad (5.4.5)$$

式中, G 称为该码的生成矩阵,是 $k \times n$ (k 行 n 列)矩阵:

$$G = (g_1 \ g_2 \ \dots \ g_k)^T = \begin{pmatrix} g_{11} & \dots & g_{1n} \\ \dots & & \dots \\ g_{k1} & \dots & g_{kn} \end{pmatrix} \quad (5.4.6)$$

而 $g_i = [g_{i1} \ g_{i2} \ \dots \ g_{in}]$, ($i = 1, \dots, k$) 是 G 中第 i 行的行矢量。

由(5.4.5)式看出:任何码字都是 G 的行矢量 $\{g_i\}$ 的线性组合,即

$$C_i = m_1 g_1 + m_2 g_2 + \dots + m_k g_k \quad (5.4.7)$$

与(5.4.3)式对照,可知 G 的 k 个行矢量 g_i 正对应于码空间 C 的 k 个基底。想要保证 (n, k) 线性分组码能够构成 k 维 n 重子空间, G 的 k 个行矢量必须是线性无关的,只有这样才能作为基底的条件。由于基底不是唯一的,所以 G 也就不是唯一的。此外,由于子空间是 k 维的,因此 G 的秩是 k 。 G 的 k 个行矢量 g_i 既是一个基底,也是一个码字。

(n, k) 码的任何生成矩阵都可以通过行运算(以及列置换)简化成“系统形式”

$$G = (I_k \ P) = \begin{pmatrix} 1 & 0 & \dots & 0 & p_{11} & p_{12} & \dots & p_{1(n-k)} \\ 0 & 1 & \dots & 0 & p_{21} & p_{22} & \dots & p_{2(n-k)} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & p_{k1} & p_{k2} & \dots & p_{k(n-k)} \end{pmatrix} \quad (5.4.8)$$

这里 I_k 是 $k \times k$ 单位矩阵, P 是 $k \times (n - k)$ 矩阵。编码时,信息组 m 乘以系统形式的 G 所得的码字,其前 k 位由单位矩阵 I_k 决定,一定与信息组各比特相同,等于把信息组原封不动搬到码字的前 k 位;而其余的 $n - k$ 位叫冗余比特或一致校验位,是前 k 个信息位的线性

组合。这样生成的 (n, k) 码叫做系统码。

生成矩阵如不具备(5 4 8)式所示的系统形式,则该码叫做非系统码。对于两个生成矩阵,如果一个矩阵能通过行运算和列置换得到另一个矩阵的话,这两生成矩阵称为是等效的。从这个意义上说,非系统码的生成矩阵可以通过运算转变为另一个系统码的生成矩阵。由两个等效的生成矩阵所生成的两个 (n, k) 线性码也是等效的,这样,每个 (n, k) 线性码都可以和一个系统的 (n, k) 线性码等效。

与任何一个 (n, k) 线性码的码空间 C 相对应,一定存在一个对偶空间 H 。事实上,码空间基底数 k 只是 n 维 n 重空间全部 n 个基底的一部分,若能找出另外 $n - k$ 个与之正交的基底,也就找到了对偶空间 H 。 C 和 H 是对偶的,既然用 k 个基底能产生一个 (n, k) 线性码,那么也能用 $n - k$ 个基底产生一个有 2^{n-k} 个码矢的 $(n, n - k)$ 线性码,称 $(n, n - k)$ 线性码是 (n, k) 线性码的对偶码,反之亦然。由于 C 的基底和 H 的基底正交,空间 C 和空间 H 也正交,它们互为零空间,因此, (n, k) 码的任意一个码字 C_i 均正交于其对偶码的任意一个码字。将 H 空间的 $n - k$ 个基底排列起来可构成一个 $(n - k) \times n$ 矩阵,称为校验矩阵 H 。 H 是 $(n, n - k)$ 对偶码的生成矩阵,它的每一行是一个基底,也是一个码字。因此, (n, k) 码的任意一个码字均正交于校验矩阵 H 的任意一个行矢量,即

$$C_i H^T = 0 \quad (5\ 4\ 9)$$

式中,0 代表 $1 \times (n - k)$ 全零行矢量, C_i 是 (n, k) 码的一个码字。因为(5 4 9)式对 (n, k) 码的每个码字都成立,于是有

$$GH^T = 0 \quad (5\ 4\ 10)$$

这里,0 代表一个由全零元素组成的 $k \times (n - k)$ 矩阵。

假定 (n, k) 线性码是系统码,其生成矩阵 G 符合(5 4 8)的形式,那么由于 $GH^T = 0$,必有

$$H = (-P^T \quad I_{n-k}) \quad (5\ 4\ 11)$$

上式中的负号在二进码情况下可省略,因为模 2 减法和模 2 加法是等同的。

例 5 4 1 考虑一个 $(7, 4)$ 码,其生成矩阵是

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} = (I_4 \quad P) \quad (5\ 4\ 12)$$

(1) 对于信息组 $m = (1 \ 0 \ 1 \ 1)$,编出的码字是什么?

(2) 设计一个 $(7, 4)$ 分组编码器原理图。

(3) 若接收到一个 7 位码 $r = (1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1)$,它是否是码字?

解:

(1) 设输入 4 信息比特组是 $m = (m_1 \ m_2 \ m_3 \ m_4)$,编码所得码字为 $C_i = (C_{i1} \ C_{i2} \ C_{i3} \ C_{i4} \ C_{i5} \ C_{i6} \ C_{i7})$ 。

一般编码算法是 $C_i = mG$,但由于是系统码,一个典型的码字可以表示为

$$C_i = (m_1 \ m_2 \ m_3 \ m_4 \ C_{i5} \ C_{i6} \ C_{i7})$$

如图 5 4 2 所示

根据生成矩阵 G 中 P 的 0,1 分布, 上式

3 个校验位可由下式求得:

$$\begin{aligned} c_{i5} &= m_1 + m_2 + m_3 \\ c_{i6} &= m_2 + m_3 + m_4 \quad (5\ 4\ 13) \\ c_{i7} &= m_1 + m_2 + m_4 \end{aligned}$$

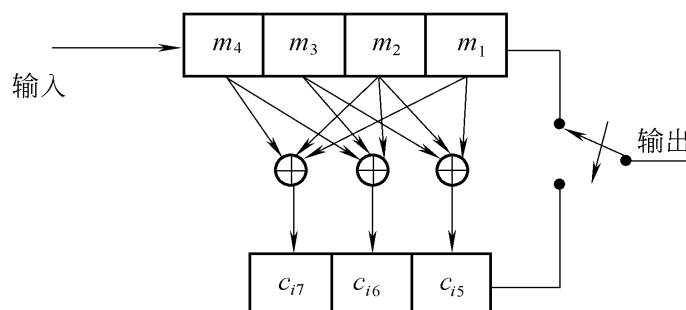


图 5 4 2 $(7,4)$ 二进制线性码编码器

式中“+”指模 2 加。

将 $m_1 = 1, m_2 = 0, m_3 = 1, m_4 = 1$ 代入, 得 $c_{i5} = 0, c_{i6} = 0, c_{i7} = 0$ 。于是码字为

$$C_i = (1011000)$$

(2) 一个二进制 (n, k) 系统线性分组码的编码器可用 k 级移存器和连接到移存器适当位置(由 G 决定)的 $n - k$ 个模 2 加法器组成。加法器生成校验位后按顺序暂存在另一个长度为 $n - k$ 的移存器中。 k 比特信息组移位输进 k 级移存器, 加法器计算 $n - k$ 校验比特, 然后先是 k 位信息、紧接是 $n - k$ 位校验比特从两个移存器中移位输出。

(3) 根据(5 4 11)式, 可得 H 矩阵如下:

$$H = (P^T \quad I_{n-k}) = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \quad (5\ 4\ 14)$$

由(5 4 9)式, $C_i H^T$ 的乘积可产生以下三个方程:

$$\begin{aligned} c_{i1} + c_{i2} + c_{i3} + c_{i5} &= m_1 + m_2 + m_3 + c_{i5} = 0 \\ c_{i2} + c_{i3} + c_{i4} + c_{i6} &= m_2 + m_3 + m_4 + c_{i6} = 0 \\ c_{i1} + c_{i2} + c_{i4} + c_{i7} &= m_1 + m_2 + m_4 + c_{i7} = 0 \end{aligned}$$

式中“+”指模 2 加。

如果 r 是某个码字 C_i , 必有 $rH^T = 0$; 反之, 如 $rH^T \neq 0$, r 必定不是码字。

将 $r = (r_1 \ r_2 \ r_3 \ r_4 \ r_5 \ r_6 \ r_7) = (1001101)$ 代入上式, 方程 和 都不能成立, 说明 r 与 H 不正交, r 不是码字。

从(5 4 13)式看到: 系统码的后 $n - k$ 位是前 k 位的线性组合; 又从(5 4 15)式看到: 通过验证系统码的后 $n - k$ 位是否是前 k 位的线性组合, 我们可判断是否是码字。因此, 系统码的后 $n - k$ 位称为校验位, 而矩阵 H 称为 (n, k) 码的一致校验矩阵(简称校验矩阵)。

校验矩阵 H 除了用来校验码字外, 还与码的最小距离、进而与码的纠错能力发生一定关系。 H 是 $(n - k) \times n$ 矩阵, 既可将它看作是 $n - k$ 个行矢量的排列, 也可将它看作是 n 个列矢量的排列, 写成

$$H = \begin{pmatrix} h_{11} & h_{12} & \dots & h_{1n} \\ h_{21} & h_{22} & \dots & h_{2n} \\ \dots & \dots & \dots & \dots \\ h_{n-k,1} & h_{n-k,2} & \dots & h_{n-k,n} \end{pmatrix} = (h_1 \ h_2 \ \dots \ h_n) \quad (5\ 4\ 16)$$

式中 $h_j (j = 1, \dots, n)$ 是 $(n - k) \times 1$ 列矢量。由(5 4 9)式

$$C_i H^T = (c_{i1} \ c_{i2} \ \dots \ c_{in}) (h_1 \ h_2 \ \dots \ h_n)^T = c_{i1} h_1^T + c_{i2} h_2^T \ \dots + c_{in} h_n^T = 0 \quad (5\ 4\ 17)$$

可见, $C_i H^T$ 代表了 n 个 $1 \times (n - k)$ 矢量 h_j^T 的线性组合, 并且由于 $C_i H^T = 0$, n 个矢量 h_j^T 是线性相关的。

如果分组码的最小距离等于 d_{\min} , 说明码集里重量最小的那个码字有 d_{\min} 个非零码元。若将该最小重量的码字代入 (5 4 17) 式, 那么等式左边将有 d_{\min} 个 h_j^T 项, 右边为 0。由此可以断言: H 矩阵的列矢量至少要有 d_{\min} 个才能线性相关, 而 $d_{\min} - 1$ 个列矢量必定是线性无关的, 因为如果 $d_{\min} - 1$ 个 H 的列矢量线性相关, 必然存在一个重量为 $d_{\min} - 1$ 的码字, 这与最小距离 d_{\min} 的假设相悖。

由于 H 是 $(n - k) \times n$ 矩阵, 其秩至多是 $n - k$, 即最多有 $n - k$ 个列矢量线性无关。在寻找“好”码时我们希望 d_{\min} (或 $d_{\min} - 1$) 越大越好, 等效于上面线性无关的列矢量越多越好, 而线性无关的列矢量至多 $n - k$ 个, 于是得出这样一个关系式:

$$(d_{\min} - 1) \leq n - k \text{ 即 } d_{\min} \leq n - k + 1 \quad (5\ 4\ 18)$$

这就是说: 二进制 (n, k) 线性码最小距离 d_{\min} 的上边界是 $n - k + 1$ 。如果设计的 (n, k) 线性码的 d_{\min} 达到了 $n - k + 1$, 那么就是达到了设计性能的极点。因此, $d_{\min} = n - k + 1$ 的码称为极大最小距离码 (MDC: Maximized minimum Distance Code), 这从 d_{\min} 角度看是最好的码, 当然是所追求的码。

如果给每个码字添加一位奇偶校验位 $c_{i(n+1)}$ 来对码字的所有比特进行校验, 使满足 $c_{i1} + c_{i2} \dots + c_{in} + c_{i(n+1)}$ 等于 0 或 1, 就构成了一个二进制 $(n + 1, k)$ 线性码, 称为扩展码。在偶校验的情况下, 若原来码字中 1 的个数为偶数, 则添加的校验位为 0; 若原来码字中 1 的个数为奇数, 则添加一个校验位 1。因此, 如果某码原来的最小重量 d_{\min} 是奇数, 那么添加了偶校验位后使最小距离变为 $d_{\min} + 1$, 检错能力增 1。此时它的校验矩阵 H_e 是

$$H_e = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \quad 1 \quad \dots \quad 1 \quad 1 \end{bmatrix} \quad H$$

式中 H 是原码的校验矩阵。

(n, k) 系统码同样可以被缩短。在码集的全部码字 $[m_1 \ m_2 \ \dots \ m_k \ c_{i(k+1)} \ \dots \ c_{in}]$ 中, 如果分布是均匀的, 码字第一位 $m_1 = 0$ 的概率应是 0.5, 即一半码字的第一位是零。把第一位为 1 的所有码字舍去, 剩下另一半第一位为 0 的码字舍去它们的第一位后组成一个新的 $(n - 1, k - 1)$ 系统码, 称为缩短码, 表示码长和信息位长度均缩短了。 $(n - 1, k - 1)$ 缩短码共包含 $2^k \div 2 = 2^{k-1}$ 个码字。由于原先保留的均是第一位为 0 的码, 舍去它们的第一位不会改变码的最小重量, 因此缩短码与原码具有同样的 d_{\min} 。在缩短后的 $(n - 1, k - 1)$ 码中又有一半码字的第一位 $m_2 = 0$, 重复上述过程, 可得 $(n - 2, k - 2)$ 缩短码。推广到一般, 可以把一个 (n, k) 线性码缩短 1 位, 由 2^{k-l} 个码字构成一个 $(n - l, k - l)$ 缩短码而保持 d_{\min} 不变。若 (n, k) 线性码的编码运算为 $G = mG$, 由于缩短码信息组的前 l 位是 0, 因此缩短码的编码运算为

$$G_s = m_s G \quad (5\ 4\ 20)$$

式中, m_s 是 $k - l$ 位信息组, G_s 是 G ($k \times n$ 矩阵) 去掉最左边 l 列及最上边 l 行后剩下的那

个 $(n - l) \times (k - l)$ 矩阵。译码时, $n - k$ 个校验位可仍按原码(缩短前)的方法计算。至于校验矩阵,原码为 $(n - k) \times n$ 矩阵 H , 缩短后为 $(n - k) \times (n - l)$ 矩阵 H_s , 可见矩阵 H 和 H_s 行数一样, 将矩阵 H 最左边的 l 列去掉就得矩阵 H_s 。

5 4 3 伴随式与译码

设 k 位信息编成 n 位线性分组码 $C = (c_1, c_2, \dots, c_n)$ 后送入信道, 经传输, 接收端的收码是 $R = (r_1, r_2, \dots, r_n)$ (参见图 5 3 5)。显然, 发码与收码之差异是由信道干扰产生的, 定义差错图案 E 为

$$E = (e_1, e_2, \dots, e_n) = (r_1 - c_1, r_2 - c_2, \dots, r_n - c_n) = R - C \quad (5\ 4\ 21)$$

差错图案是信道上干扰图案的反映。对于下面将要讨论的二进制码, 模 2 加与模 2 减是等同的, 因此有

$$E = R + C \text{ 及 } R = C + E \quad (5\ 4\ 22)$$

根据(5 4 9)式的结论, 在一定的差错范围内, 可以通过如下运算, 利用 $CH^T = 0$ 的特点来判断收码 R 是否等于发码 C :

$$RH^T = (C + E)H^T = CH^T + EH^T = EH^T \quad (5\ 4\ 23)$$

如果收码无误, 必有 $R = C$ 即 $E = 0$ 及 $EH^T = 0$, 此时上式 $RH^T = 0$ 。如果信道中产生差错即 $E \neq 0$, 必有 $RH^T = EH^T \neq 0$ 。在 H^T 固定的前提下, RH^T 仅仅与差错图案 E 有关, 而与发送码 C 是什么样无关。为此定义伴随式 S 为

$$S = (s_1, s_2, \dots, s_{n-k}) = RH^T = EH^T \quad (5\ 4\ 24)$$

伴随式 S 是伴随接收码 R 的一个 $(n - k)$ 重矢量。从物理意义上看, 伴随式 S 并不反映发送的码字是什么, 而只是反映信道对码字造成怎样的干扰。此外还看到: 差错图案 E 是 n 重矢量, 共有 2^n 个可能的组合, 而伴随式 S 是 $(n - k)$ 重矢量, 只有 2^{n-k} 个可能的组合, 因此不同的差错图案可能有相同的伴随式。

在接收端我们并不知道发码 C 是什么, 但可以知道 H^T 和 R 是什么, 并通过伴随式译码找到 C 的估值, 其过程是:

$$RH^T = S \quad E \quad C = R + E \quad (5\ 4\ 25)$$

即先算出 S , 再由 S 算出 E , 最后令 $C = R + E$ 而求得 C 。这里最关键的一步是如何从 S 找出 E , 只要 E 正确, 译出的码也就是正确的。

可以通过解线性方程求解 E , 由(5 4 24)式,

$$S = (s_1, s_2, \dots, s_{n-k}) = EH^T = (e_1, e_2, \dots, e_n) \begin{pmatrix} h_{11} & \dots & h_{1n} \\ \dots & \dots & \dots \\ h_{n-k,1} & \dots & h_{n-k,n} \end{pmatrix}^T \quad (5\ 4\ 26)$$

展开成线性方程组形式, 为

$$\begin{aligned} s_1 &= e_1 h_{11} + e_2 h_{12} + \dots + e_n h_{1n} \\ s_2 &= e_1 h_{21} + e_2 h_{22} + \dots + e_n h_{2n} \\ &\vdots \\ s_{n-k} &= e_1 h_{n-k,1} + e_2 h_{n-k,2} + \dots + e_n h_{n-k,n} \end{aligned} \quad (5\ 4\ 27)$$

(5 4 27)式的方程组中有 n 个未知数 e_1, e_2, \dots, e_n , 却只有 $n - k$ 个方程, 可知方程组有多解。在有理数或实数域中, 少一个方程就可能导致无限多个解, 而在二元域中, 少一个方程

导致两个解,少两个方程四个解,以此类推,少 $n - (n - k) = k$ 个方程导致每个未知数有 2^k 个解。因此,由 RH^T 确定 S 后,对应的差错图案 E 可以有 2^k 个解。到底取哪一个作为附加在收码 R 上的差错图案 E 的估值呢?有一种概念上很简单但计算效率不高的译码方法,叫概率译码,它把所有 2^k 个解的重量(差错图案 E 中 1 的个数)作比较,选择其中最轻者作为 E 的估值。这种算法的理论根据是:若 BSC 信道的差错概率是 p ,则长度 n 的码中错 1 位(对应于 E 中有一个 1 或 E 的重量为 1)的概率是 $p(1 - p)^{n-1}$,错 2 位的概率是 $p^2(1 - p)^{n-2} \dots$,以此类推。由于 $p < 1$,必有 $p(1 - p)^{n-1} > p^2(1 - p)^{n-2} > \dots > p^{n-1}(1 - p) > p^n$,所以 S 对应最小重量 E 的可能性最大。由于 $E = R + C$ 即收、发码之间的汉明距离, E 重量最小就是 R, C 的距离最小,所以概率译码实际上体现了最小距离译码法则,也就是最大似然译码。

上述的概率译码,如每接收一个码 R 就要解一次线性方程,那就太麻烦了。好在伴随式的数目是有限的 2^{n-k} 个,如果 $n - k$ 不太大,我们可以预先把不同 S 下的方程组解出来,把各种情况下的最大概率译码输出列成一个码表,这样,在实时译码时就不必再去解方程,而只要象查字典那样查一下码表就可以了。下面我们来讨论在一般情况下构造标准阵列译码表的方法。

将没有任何差错时的收码 R 放在第一行,此时收码等于发码 $R = C(C = C_i, i = 1, 2, \dots, 2^k)$,差错图案为全零 $E_1 = (0, 0, \dots, 0)$,伴随式为全零 $S_1 = (0, 0, \dots, 0)$ (由(5 4 24)式)。由于有 2^k 个码字,码表有 2^k 列。接着,在第 2 到第 $n + 1$ 的 n 行中我们填上所有重量为 1 的差错图案(共 n 个)。如果 $(1 + n) < 2^{n-k}$,接着再在下面 $\frac{n}{2}$ 行写出全部带有 2 个差错的图案(共 $\frac{n}{2}$ 个)。如果总行数 $(1 + n + \frac{n}{2})$ 仍然小于 2^{n-k} ,再列出带有 3 个差错的图案,以此类推,直到放满 2^{n-k} 行,每行一个 E_j 对应一个不同的伴随式 S_j 。这样,表的行数 2^{n-k} 正等于伴随式的数目。最后,在码表的第 j 行、第 i 列填入 $C_i + E_j$,如表 5 4 1 所示。

表 5 4 1 标准阵列译码表

$S_1 \quad E_1$	$E_1 + C_1 = 0 + 0 = 0$	$E_1 + C_2 = C_2$...	$E_1 + C_i = C_i$...	$E_1 + C_{2^k} = C_{2^k}$
$S_2 \quad E_2$	$E_2 + C_1 = E_2$	$E_2 + C_2$...	$E_2 + C_i$...	$E_2 + C_{2^k}$
...
$S_j \quad E_j$	$E_j + C_1 = E_j$	$E_j + C_2$...	$E_j + C_i$...	$E_j + C_{2^k}$
...
$S_{2^{n-k}} \quad E_{2^{n-k}}$	$E_{2^{n-k}} + C_1 = E_{2^{n-k}}$	$E_{2^{n-k}} + C_2$...	$E_{2^{n-k}} + C_i$...	$E_{2^{n-k}} + C_{2^k}$

表中有 2^{n-k} 行,每行是一个陪集,每陪集的第一个元素(位于第一列)叫陪集首。同一陪集中的所有元素对应共同的一个伴随式。第一行陪集的陪集首是全零伴随式 S_1 所对应的全零差错图案 E_1 (无差错),而第 j 行陪集的陪集首是伴随式 S_j 所对应的重量最小的差错图案 E_j 。

表中有 2^k 列,每列是一个子集,每子集的第一个元素(位于第一行)叫子集头。同一子集中的所有元素对应同一个码字,第一列子集的子集头是全零码字 C_1 ,而第 i 列子集的子

集头是码字 C_i 。

例 5 4 2 某一个 $(5, 2)$ 系统线性码的生成矩阵是 $G = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \end{pmatrix}$, 设收码是 $R = (10101)$, 请先构造该码的标准阵列译码表, 然后译出发码的估值 C 。

解 (1) 构造标准阵列译码表。

分别以信息组 $m = (00), (01), (10), (11)$ 及已知的 G 代入式 (5 4 5), 求得 4 个许用码字为 $C_1 = (00000), C_2 = (10111), C_3 = (01101), C_4 = (11010)$ 。

由式 (5 4 14), 求出校验矩阵

$$H = [P^T \quad I_3] = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} & h_{14} & h_{15} \\ h_{21} & h_{22} & h_{23} & h_{24} & h_{25} \\ h_{31} & h_{32} & h_{33} & h_{34} & h_{35} \end{pmatrix}$$

$$\begin{aligned} \text{由 (5 4 27)} \quad s_1 &= e_1 h_{11} + e_2 h_{12} + e_3 h_{13} + e_4 h_{14} + e_5 h_{15} = e_1 + e_2 + e_3 \\ s_2 &= e_1 h_{21} + e_2 h_{22} + e_3 h_{23} + e_4 h_{24} + e_5 h_{25} = e_1 + e_4 \\ s_3 &= e_1 h_{31} + e_2 h_{32} + e_3 h_{33} + e_4 h_{34} + e_5 h_{35} = e_1 + e_2 + e_5 \end{aligned}$$

伴随式有 $2^3 = 8$ 种组合, 而差错图案中代表无差错的有一种, 代表一个差错的图案有 $\binom{5}{1} = 5$ 种, 代表两个差错的图案有 $\binom{5}{2} = 10$ 种。要把 8 个伴随式对应到 8 个最轻的差错图案, 无疑应先选上那 1 种无差错的图案和 5 种一个差错的图案。对于两个差错的图案, 若不选它们则多出 2 个伴随式, 若全部选上则缺少 8 个伴随式。我们于是只能挑选其中的两个, 至于挑选方法可有若干种, 不是唯一的。先将 $E_i = (00000), (10000), (01000), (00100), (00010), (00001)$ 代入上面的线性方程组, 解得对应的 S_i 分别是 $(000), (111), (101), (100), (010), (001)$ 。剩下的伴随式中, (011) 所对应的差错图案是 2^k 个即 $(00011), (10100), (01110), (11001)$, 其中 (00011) 和 (10100) 并列重量最轻, 任选其中一个比如 (00011) ; 同样可得伴随式 (110) 所对应的最轻差错图案之一是 (00110) 。至此, 已求得了所需的数据, 可画出标准阵列如表 5 4 2 所示。

表 5 4 2 标 准 阵 列

$S_1 = 000$	$E_1 + C_1 = 00000$	$C_2 = 10111$	$C_3 = 01101$	$C_4 = 11010$
$S_2 = 111$	$E_2 = 10000$	00111	11101	01010
$S_3 = 101$	$E_3 = 01000$	11111	00101	10010
$S_4 = 100$	$E_4 = 00100$	10011	01001	11110
$S_5 = 010$	$E_5 = 00010$	10101	01111	11000
$S_6 = 001$	$E_6 = 00001$	10110	01100	11011
$S_7 = 011$	$E_7 = 00011$	10100	01110	11001
$S_8 = 110$	$E_8 = 00110$	10001	01011	11100

(2) 对于收码 $R = (10101)$, 可选以下三种方法之一译码。

直接搜索码表,查得(10101)所在列的子集头是(10111),因此译码输出取为(10111)。

可以先求伴随式找到行数, $RH^T = (10101) \cdot H^T = (010) = S_5$,再搜索码表的第5行找到(10101),最后顺着该列向上找出码字(10111)。

先求出伴随式 $RH^T = (010) = S_5$,在表中查出对应的差错图案是 $E_5 = (00010)$,通过计算得到码字 $C = R + E_5 = (10101) + (00010) = (10111)$ 。

从方法 到方法 ,查表的时间下降而所需计算量增大,可针对不同情况选用。

如对上例作进一步分析,还可以看到,该(5,2)码的 $d_{\min} = 3$,纠错能力是 $t = \text{INT}[(3-1)/2] = 1$ 。因此,译码阵列中只有前6行具有唯一性、可靠性,而第7,8行的差错图案(00011)和(00110)中包含两个“1”,已超出了 $t=1$ 的纠错能力,译码已不可靠。比如,当收码 $R = (10100)$ 时,根据码表译出的码字是(10111),与收码 R 的汉明距离是2,然而收码 R 与全零码字(00000)的汉明距离也是2,为什么不能译成(00000)呢?事实上,码表的第7,8行本身就不是唯一的。注意在码表计算过程中,伴随式(011)所对应的4个差错图案中有两个并列重量最轻,如果当时选的不是(00011)而是(10100),那么码表第7行就不是现在这样了。

由此想到,伴随式的个数 2^{n-k} 应该与 n, k 及纠错能力 t 之间有一定的关系,这就导致了完备码的概念。

任何一个二元(n, k)线性分组码都有 2^{n-k} 个伴随式,假如该码的纠错能力是 t ,则对于任何一个重量小于等于 t 的差错图案,都应有一个伴随式与之对应,也就是说,伴随式的数目满足条件

$$2^{n-k} = \sum_{i=0}^t \binom{n}{i} \quad (5.4.28)$$

上式称作汉明限,任何一个纠 t 码都应满足上述条件。

如果某二元(n, k)线性分组码能使式(5.4.28)的等号成立,即该码的伴随式数目不多不少恰好和不大于 t 个差错的图案数目相等,相当于在标准阵列中能将所有重量不大于 t 的差错图案选作陪集首而没有一个陪集首的重量大于 t ,这时的校验位得到最充分的利用。把满足方程

$$2^{n-k} = \sum_{i=0}^t \binom{n}{i} \quad (5.4.29)$$

的二元(n, k)线性分组码称为完备码(Perfect code)。

从多维矢量空间的角度来看完备码(参见图 5.3.4),假定我们围绕每一个码字 C 放置一个半径为 t 的球,每个球内包含了与该码字汉明距离小于、等于 t 的所有收码 R 的集合,这样在半径为 $t = [(d_{\min} - 1)/2]$ 的球内的收码数是 $\sum_{i=0}^t \binom{n}{i}$ 。因为有 2^k 个可能发送的码字,也就有 2^k 个不相重叠的半径为 t 的球。包含在 2^k 个球中的码字总数不会超过 2^n 个可能的接收码字。于是一个纠 t 差错的码必然满足不等式

$$2^k \cdot \sum_{i=0}^t \binom{n}{i} \leq 2^n \quad \text{即} \quad 2^{n-k} \geq \sum_{i=0}^t \binom{n}{i} \quad (5.4.30)$$

如果满足(5.4.29)式的条件,(5.4.30)式中等号成立,表示所有的收码都落在 2^k 个球内

而球外没有一个码,这就是完备码。完备码具有下述特性:围绕 2^k 个码字、汉明距离为 $t = [(d_{\min} - 1)/2]$ 的所有球都是不相交的,每一个接收码字都落在这些球中之一,因此接收码离发码的距离至多为 t ,这时所有重量 t 的差错图案都能用最佳(最小距离)译码器得到纠正,而所有重量 $t + 1$ 的差错图案都不能纠正。能够满足(5 4 29)式条件的完备码并不多见,迄今发现的完备码有 $t = 1$ 的汉明码, $t = 3$ 的高莱码,以及长度 n 为奇数、由两个码字组成、满足 $d_{\min} = n$ 的任何二进制码,还有三进制 $t = 3$ 的(11,6)码。

汉明码(Hamming Code)不是指一个码,而是代表一类码。汉明码的纠错能力 $t = 1$,既有二进制的,也有非二进制的,下面先讨论二进制的。汉明码码长 n 和信息位 k 服从以下规律

$$(n, k) = (2^m - 1, 2^m - 1 - m) \tag{5 4 31}$$

其中 $m = n - k$, 是正整数。当 $m = 3, 4, 5, 6, 7, 8, \dots$ 时, 有 $(7, 4), (15, 11), (31, 26), (63, 57), (127, 120), (255, 247), \dots$ 汉明码。汉明码是完备码,因为它满足(5 4 29)式。

$$\sum_{i=0}^1 \binom{n}{i} = 1 + n = 1 + (2^m - 1) = 2^m = 2^{n-k}$$

汉明码的校验矩阵 H 具有特殊的性质,使我们能以相对简单的方法来描述该码。一个 (n, k) 码的校验矩阵有 $n - k$ 行和 n 列,二进制时 $n - k$ 个码元所能组成的列矢量总数(全零矢量除外)是 $2^{n-k} - 1$, 恰好和校验矩阵的列数 $n = 2^m - 1$ 相等。只要排列所有列,通过列置换将矩阵 H 转换成系统形式,就可以进一步得到相应的生成矩阵 G 。

例 5 4 3 构造一个 $m = 3$ 的二元(7,4)汉明码。

解: 所谓构造就是求一个(7,4)汉明码的生成矩阵。先利用汉明码的特性构造一个校验矩阵 H , 再通过列置换将它变为系统形式;

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \xrightarrow{\text{列置换}} \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} = (P^T \mid I_3)$$

由(5 4 8), (5 4 11)式,得生成矩阵 G 为

$$G = (I_4 \mid P) = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

由于生成矩阵 G 中包含了单位阵 I_4 , 矩阵的秩是 4, 所以矩阵 G 的 4 行是 4 个线性无关的基底, 可以张成一个包含 $2^4 = 16$ 码字的码空间。

必须指出, 完备码是标准阵列最规则因而译码最简单的码, 但并不一定是纠错能力最强的码。完备码强调了 n, k, t 的关系, 保证 d_{\min} 至少等于 3 (即 $t = 1$), 但并未强调 d_{\min} 最大化即达到极大最小距离码 MDC $d_{\min} = n - k + 1$ 的程度。换言之, 完备码未必是 MDC 码。但如果在遵守 n, k, t 关系的基础上再服从 MDC 码的设计规则, 那么完备码又可以同时是 MDC 码。比如(63,57)码, 按汉明码要求可保证 $t = 1$, 如进一步按 MDC 码要求设计可得 $t = \text{INT}[(63 - 57 + 1)/2] = 3$ 。

如果给 (n, k) 汉明码添加一位奇偶校验位, 可得到一个 $d_{\min} = 4$ 的 $(n + 1, k)$ 扩展汉明码。反之, 在生成矩阵 G 中删除 1 行, 或等效地在校验矩阵 H 中删除 1 列, 汉明码可以缩短

为 $(n - l, k - l)$ 码。

(n, k) 汉明码码字的重量分布是已知的, 可用一个称为重量估值算式 (weight enumerating polynomial) 的 z 的多项式来表达, z^i 项的系数 A_i 表示重量为 i 的码字的数目:

$$A(z) = \sum_{i=0}^n A_i z^i = \frac{1}{n+1} (1+z)^n + n(1+z)^{(n-1)/2} (1-z)^{(n+1)/2} \quad (5.4.32)$$

将等式右边展开即可得到 A_i 值。

二进制汉明码的概念也可扩展到多进制, 推出 $GF(q)$ 域上的汉明码。在 q 进制中, 一个码元上的差错位置就可以有 $(q - 1)$ 种, n 个码元上的差错位置有 $n(q - 1)$ 种。而 $(n - k)$ 个校验位可以表达 q^{n-k} 个不同的意思, 由汉明码定义, 它应该恰好等于所有的单个差错图案加上 1 (无差错), 即 $q^{n-k} = n(q - 1) + 1$ 。令 $n - k = m$, 则 q 进制汉明码的 n, k 应服从 $n = (q^m - 1)/(q - 1)$ 及 $k = (q^m - 1)/(q - 1) - m$ 。

高莱 (Golay) 码是二进制 $(23, 12)$ 线性码, 其最小距离 $d_{\min} = 7$, 纠错能力 $t = 3$ 。由于满足 (5.4.29) 式, 即

$$2^{23-12} = 2048 = 1 + \frac{23}{1} + \frac{23}{2} + \frac{23}{3}$$

因此它也是完备码。在 $(23, 12)$ 码上添加一位奇偶位即得二进制线性 $(24, 12)$ 扩展高莱码, 其最小距离 $d_{\min} = 8$ 。

5.4.4 循环码

循环码是线性码的一个子集, 它满足下列循环移位特性: 码集 C 中任何一个码字的循环移位仍是码字。一般 (n, k) 线性分组码的 k 个基底之间不存在规则的联系, 因此我们需用 k 个基底组成生成矩阵来表示一个码的特征, 而循环码的 k 个基底可以是同一个基底循环 k 次得到, 因此用一个基底就足以表示一个码的特征。既然只有一个基底, 就无需矩阵, 只要用多项式作为数学工具就足够了。为此, 我们把码字 $C = [c_{n-1} \ c_{n-2} \ \dots \ c_1 \ c_0]$ 与一个不大于 $n - 1$ 次 (degree $n - 1$) 的码多项式 $C(x)$ 联系起来。码多项式 $C(x)$ 定义为:

$$C(x) = c_{n-1} x^{n-1} + c_{n-2} x^{n-2} + \dots + c_1 x + c_0 \quad (5.4.33)$$

对于二进制码, $c_i \in \{0, 1\}$, $i = 0, \dots, n - 1$ 。

注意, 这里码元序号采用 $0 \sim n - 1$ 而不用 $1 \sim n$ 是为了在以后多项式运算中系数序号与 x 幂次一致, 同样, 多项式按降幂排列还是升幂排列, 发送时是高位先发还是低位先发, 这些都仅是表达上的差异, 并无本质区别。

根据循环码的定义, n 重形式的码字的循环移位可表示为

$$(c_{n-1} \ c_{n-2} \ \dots \ c_1 \ c_0) \xrightarrow{\text{循环移 1 位}} (c_{n-2} \ \dots \ c_1 \ c_0 \ c_{n-1}) \quad (5.4.34)$$

与之对应的多项式的变化为

$$C_0(x) = c_{n-1} x^{n-1} + c_{n-2} x^{n-2} + \dots + c_1 x + c_0$$

循环移 1 位

$$C_1(x) = c_{n-2} x^{n-1} + c_{n-3} x^{n-2} + \dots + c_0 x + c_{n-1}$$

比较循环移位的前后可用如下的多项式运算来表达循环移位:

$$\begin{aligned}
\text{移 1 位:} \quad C_1(x) &= xC_0(x) \quad \text{mod}(x^n + 1) \\
\text{移 2 位:} \quad C_2(x) &= xC_1(x) = x^2 C_0(x) \quad \text{mod}(x^n + 1) \\
&\dots \quad \dots \quad \dots \\
\text{移 } n-1 \text{ 位:} \quad C_{n-1}(x) &= xC_{n-2}(x) = x^{n-1} C_0(x) \quad \text{mod}(x^n + 1)
\end{aligned} \tag{5 4 35}$$

依此类推。因码字 $C_0(x)$ 移 n 位后又回到码字 $C_0(x)$, 一个码字的移位最多能得到 n 个码字, 因此“循环码字的循环仍是码字”并不意味着循环码集可以从一个码字循环而得。

根据码空间的封闭性, 码字的线性组合仍是码字。对(5 4 35)式中各码作线性组合, 组合的结果仍应是码字

$$\begin{aligned}
C(x) &= a_0 C_0(x) + a_1 x C_0(x) + a_2 x^2 C_0(x) \dots + a_{n-1} x^{n-1} C_0(x) \\
&= (a_0 + a_1 x + a_2 x^2 \dots + a_{n-1} x^{n-1}) C_0(x) = A(x) C_0(x) \quad \text{mod}(x^n + 1)
\end{aligned} \tag{5 4 36}$$

式中, $C_0(x)$ 是一个码多项式, 而 $A(x)$ 是次数不大于 $n-1$ 的任意多项式。对于二进制码, $a_i \in \{0, 1\}$, $i = 0, \dots, n-1$ 。

从近世代数观点看, $GF(2)$ 上次数小于 n 的多项式在模 2 加、模 $(x^n + 1)$ 乘法运算下构成了一个交换环, 从多项式环的性质出发, 有如下结论(证明略)

(1) 一个 (n, k) 循环码的码多项式是模 $(x^n + 1)$ 乘运算下多项式交换环的一个主理想子环, 反之, 多项式交换环的一个主理想子环一定可以产生一个循环码。而主理想子环中的所有码多项式都可以由其中一个元素(码多项式)的倍式组成, 这个元素称为该主理想子环的生成元, 或称它为对应循环码的生成多项式。生成多项式不是唯一的, 但总有一个是次数最低的。

(2) $GF(2)$ 上 (n, k) 循环码中, 存在着唯一的一个次数最低即 $(n-k)$ 次的首一码多项式 $g(x)$

$$g(x) = x^{n-k} + g_{n-k-1} x^{n-k-1} + \dots + g_2 x^2 + g_1 x + 1 \tag{5 4 37}$$

使得所有码多项式都是 $g(x)$ 的倍式即 $C(x) = m(x) g(x)$, 且所有小于 n 次的 $g(x)$ 的倍式都是码多项式。这里所说的首一, 指多项式的零次项(常数项)为 1。

(3) (n, k) 循环码的生成多项式 $g(x)$ 一定是 $(x^n + 1)$ 的因子即 $g(x) | (x^n + 1)$, 这里的“ $|$ ”表示“整除”, 或写成 $(x^n + 1) = g(x) h(x)$ 。相反, 如果 $g(x)$ 是 $(x^n + 1)$ 的 $(n-k)$ 次因子, 则 $g(x)$ 一定是 (n, k) 循环码的生成多项式。

以上面三个结论为基础, 我们可以找到构造 (n, k) 循环码的步骤:

对 $(x^n + 1)$ 作因式分解, 找出其 $(n-k)$ 次因式。

以该 $(n-k)$ 次因式为生成多项式 $g(x)$, 与不高于 $(k-1)$ 次的信息多项式 $m(x)$ 相乘, 即得码多项式 $C(x) = m(x) g(x)$, $C(x)$ 的次数不高于 $(k-1) + (n-k) = (n-1)$ 次。

可以这样来验证所得码的循环性: 令 $C_1(x) = xC(x) = xm(x) g(x) \text{ mod}(x^n + 1)$, 由于 $g(x)$ 本身也是码多项式(次数最低), 而 $xm(x)$ 是不高于 k 次的多项式, 由式(5 4 36), $C_1(x)$ 一定是码字, 即码字的循环也是码字, 所以确实是循环码。

例 5 4 4 研究一个长度 $n=7$ 的循环码的构成方法。

解：(1) 对 $(x^7 + 1)$ 作因式分解, 得 $x^7 + 1 = (x + 1)(x^3 + x^2 + 1)(x^3 + x + 1)$, 因此 $(x^7 + 1)$ 有如下因式:

- 1 次因式 1 种: $(x + 1)$
- 3 次因式 2 种: $(x^3 + x^2 + 1)$ 或 $(x^3 + x + 1)$
- 4 次因式 2 种: $(x + 1)(x^3 + x^2 + 1) = x^4 + x^2 + x + 1$ 或 $(x + 1)(x^3 + x + 1) = x^4 + x^3 + x^2 + 1$
- 6 次因式 1 种: $(x^3 + x^2 + 1)(x^3 + x + 1) = x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$

(2) 若以 $(n - k)$ 次因式为生成多项式, 可供选取的有 $(n - k) = 1, (n - k) = 3, (n - k) = 4, (n - k) = 6$ 。在 $n = 7$ 的情况下, 可生成 1 种 $(7, 6)$ 、2 种 $(7, 4)$ 、2 种 $(7, 3)$ 以及 1 种 $(7, 1)$ 循环码。比如要想得到 $(7, 4)$ 循环码, 可选 $7 - 4 = 3$ 次多项式 $(x^3 + x^2 + 1)$ 或 $(x^3 + x + 1)$ 为生成多项式。

现以 $g(x) = (x^3 + x^2 + 1)$ 生成 $(7, 4)$ 码为例说明得到的确是循环码。设信息的多项式为 $m = (m_3 \ m_2 \ m_1 \ m_0) = (0 \ 1 \ 1 \ 0)$ 对应信息多项式 $m(x) = m_3 x^3 + m_2 x^2 + m_1 x + m_0 = x^2 + x$ 循环编码后所得码多项式是

$$C(x) = m(x) g(x) = (x^2 + x)(x^3 + x^2 + 1) = x^5 + x^3 + x^2 + x \quad \text{对应码字} \quad (0101110)$$

在 GF(2) 中, $(m_3 \ m_2 \ m_1 \ m_0)$ 共有 16 种可能的组合, 对应 16 个码字。利用信息多项式 $m(x)$ 与生成多项式 $g(x)$ 的乘法运算, 可得所有 $(7, 4)$ 循环码的码字如表 5 4 3 所示。

表 5 4 3 (7, 4) 循环码 (生成多项式 $g(x) = 1 + x^2 + x^3$)

信息比特 $m_3 \ m_2 \ m_1 \ m_0$	码字 (循环 1) $c_6 \ c_5 \ c_4 \ c_3 \ c_2 \ c_1 \ c_0$	信息比特 $m_3 \ m_2 \ m_1 \ m_0$	码字 (循环 2) $c_6 \ c_5 \ c_4 \ c_3 \ c_2 \ c_1 \ c_0$	信 息 比 特 $m_3 \ m_2 \ m_1 \ m_0$	码字 (循环 3 和 4) $c_6 \ c_5 \ c_4 \ c_3 \ c_2 \ c_1 \ c_0$
0001	0001101	0011	0010111	0000	0000000
0010	0011010	0110	0101110		
0100	0110100	1100	1011100	1011	1111111
1000	1101000	0101	0111001		
1101	1010001	1010	1110010		
0111	0100011	1001	1100101		
1110	1000110	1111	1001011		

由表看出, 任何码字的循环仍是码字, 而整个码集有 4 组码字循环。

一般情况下, 多项式 $x^n + 1$ 可因式分解为 $x^n + 1 = g(x) h(x)$ 的形式, 比如例 5 4 3 中, $x^7 + 1 = (x + 1)(x^3 + x^2 + 1)(x^3 + x + 1)$, 若取 $g(x) = (x^3 + x^2 + 1)$, 则有 $h(x) = (x + 1)(x^3 + x + 1)$ 。在 GF(2) 上不能再分解的因式叫最小多项式, 生成多项式 $g(x)$ 可以是最小多项式如 $(x^3 + x^2 + 1)$, 也可以不是最小多项式如 $g(x) = (x^3 + x^2 + 1)(x + 1)$ 。如果 $g(x)$ 代表 (n, k) 循环码的生成多项式, 则 $h(x)$ 代表该循环码的一致校验多项式, 其阶次为 k 。 $h(x)$ 的校验作用表现在: 任何码多项式 $C(x)$ 与 $h(x)$ 的模 $x^n + 1$ 乘积一定等于 0, 而非码字与 $h(x)$ 的乘积必不为 0。这时因为

$$C(x) h(x) = m(x) g(x) h(x) = m(x) (x^n + 1) = 0 \quad \text{mod}(x^n + 1)$$

在 $x^n + 1 = g(x)h(x)$ 分解中, $g(x)$ 和 $h(x)$ 处于同等地位。既然可以用 $g(x)$ 生成一个循环码, 也就可以用 $h(x)$ 生成一个循环码, 此时, $h(x)$ 用作生成多项式, 而 $g(x)$ 用作一致校验多项式。由 $g(x)$ 生成的 (n, k) 循环码和由 $h(x)$ 生成的 $(n, n - k)$ 循环码互为对偶码, $(n, n - k)$ 对偶码构成 (n, k) 循环码的零空间, 反之亦然。

另外, 例 5 4 4 中的 $(x^3 + x^2 + 1)$ 和 $(x^3 + x + 1)$ 互为反多项式。设 $f(x)$ 是 k 次多项式, 则 $f(x)$ 的反多项式定义为

$$\begin{aligned} x^k f(x^{-1}) &= x^k (x^{-k} + f_{k-1} x^{-k+1} + f_{k-2} x^{-k+2} + \dots + f_1 x^{-1} + 1) \\ &= 1 + f_{k-1} x + f_{k-2} x^2 + \dots + f_1 x^{k-1} + x^k \end{aligned} \quad (5\ 4\ 38)$$

可以证明: 在二元域中, 若 $f(x)$ 是 $x^n + 1$ 的一个因式, 那么它的反多项式也是 $x^n + 1$ 的一个因式。

在前面 5.4.2 节里, 用生成矩阵描述线性分组码, 本节则用生成多项式来描述循环码。循环码是线性分组码的子类, 所以不一定能用生成多项式描述一般的线性分组码, 却一定可以用生成矩阵描述循环码。下面来说明从 (n, k) 循环码生成多项式获得生成矩阵的方法。正如前面已说明的那样, (n, k) 循环码的生成矩阵可以用码空间中任何一组 k 个线性无关的码字作为基底来构成, 所以不是唯一的。但当循环码生成多项式 $g(x)$ 给定后, 最简单的方法是取 $g(x)$ 本身加上移位 $k - 1$ 次所得的 $k - 1$ 个码字作为 k 个基底。因为根据上页结论 2, 任何一个次数小于等于 $n - 1$ 、能被 $g(x)$ 整除的多项式都可以用这组多项式的线性组合来表达, 所以这组多项式确能构成 k 维循环码空间的基底。因此, 若循环码生成多项式 $g(x) = x^{n-k} + g_{n-k-1}x^{n-k-1} + \dots + g_2x^2 + g_1x + 1$, 则生成矩阵是

$$\begin{array}{l} x^{k-1}g(x) \\ x^{k-2}g(x) \\ \dots \\ xg(x) \\ g(x) \end{array} \quad G = \begin{array}{cccccccccccc} 1 & g_{n-k-1} & \dots & g_2 & g_1 & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & g_{n-k-1} & \dots & g_2 & g_1 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & g_{n-k-1} & \dots & g_2 & g_1 & 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & 1 & g_{n-k-1} & \dots & g_2 & g_1 & 1 \end{array} \quad (5\ 4\ 39)$$

注意, 通过这种方法获得的生成矩阵不是系统形式的。如要构造一个系统形式 $G = (I_k \ P)$ 的循环码生成矩阵, G 的第 l 行应是多项式 $x^{n-l} + R_l(x)$, $l = 1, 2, \dots, k$, 这里的 $R_l(x)$ 是 x^{n-l} 除以 $g(x)$ 的余式, 其次数小于 $n - k$ 。之所以取这种形式, 原因是 $R_l(x)$, x^{n-l} , $g(x)$ 三者关系为

$$x^{n-l} = Q_l(x)g(x) + R_l(x) \quad (5\ 4\ 40)$$

这里 $Q_l(x)$ 是商, 移项后得 $x^{n-l} + R_l(x) = Q_l(x)g(x)$ (二元域中“+”“ \times ”等效)。由于 $Q_l(x)g(x)$ 一定是循环码的一个码字 (结论 2), 所以 G 的第 l 行的多项式 $x^{n-l} + R_l(x)$ 也一定是循环码的一个码字且各行的码字不相关, 可以作基底。

受 (5 4 40) 式的启发, 想到由生成多项式 $g(x)$ 也可直接产生系统码。如果将消息多项式 $m(x) = m_{k-1}x^{k-1} + m_{k-2}x^{k-2} + \dots + m_1x + m_0$ 乘以 x^{n-k} , 得

$$x^{n-k}m(x) = m_{k-1}x^{n-1} + m_{k-2}x^{n-2} + \dots + m_1x^{n-k+1} + m_0x^{n-k}$$

在系统码中, 这个多项式充填了码字的前 k 个位置, 还必须在其后再加上一个代表校验比特的、次数低于 $n - k$ 的多项式。如果将 $x^{n-k}m(x)$ 除以 $g(x)$, 得商 $Q(x)$ 及余式 $r(x)$,

写作

$$x^{n-k}m(x) = Q(x)g(x) + r(x) \quad \text{或} \quad x^{n-k}m(x) + r(x) = Q(x)g(x) \quad (5.4.41)$$

式中, $g(x)$ 是 $n - k$ 次多项式; $r(x)$ 的次数一定低于 $n - k$ 次。显然, $Q(x)g(x)$ 是循环码的码字, 而 $x^{n-k}m(x) + r(x)$ 是系统形式, 因此系统循环码可通过如下步骤得到:

- 将消息多项式 $m(x)$ 乘以 x^{n-k}
- 将 $x^{n-k}m(x)$ 除以 $g(x)$ 得到余式 $r(x)$
- 将 $r(x)$ 加在 $x^{n-k}m(x)$ 后面

例 5.4.5 (7,4) 循环码的生成多项式为 $g(x) = x^3 + x + 1$, 求

- (1) 该循环码系统形式的生成矩阵。
- (2) 用生成多项式 $g(x)$ 编出消息 (1001) 的系统循环码字。
- (3) 用带反馈的移寄存器实现上述运算。

解: (1) 生成矩阵为

$$G = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

系 统 化

分别将 x^6, x^5, x^4, x^3 除以 $g(x)$, 得余式为:
 $R_1(x) = x^2 + 1, R_2(x) = x^2 + x + 1, R_3(x) = x^2 + x, R_4(x) = x + 1$

或

通过矩阵运算: 将矩阵第 3, 4 行加到第 1 行;
将矩阵第 4 行加到第 2 行

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

(2) $(m_3 \ m_2 \ m_1 \ m_0) = (1001)$ 对应消息多项式 $m(x) = m_3x^3 + m_2x^2 + m_1x + m_0 = x^3 + 1$

$x^{n-k}m(x) = x^6 + x^3$ 除以 $x^3 + x + 1$, 得商 $x^3 + x$ 和余式 $r(x) = x^2 + x$, 因此码多项式是 $C(x) = x^{n-k}m(x) + r(x) = x^6 + x^3 + x^2 + x \quad (1001110)$

(3) 循环码编码电路实现时的硬件结构图

如图 5.4.3 所示。带反馈的移寄存器构成一个除以 $g(x) = x^3 + x + 1$ 的除法电路, 反馈线的位置与 $g(x)$ 的项对应, 从左到右分别对应 1, x , x^2 和 x^3 。正常做除法时, 消息 $m(x)$ 应从除法器的最左端(对应 $g(x)$ 常数项 1)进入。如消息 $m(x)$ 右移一位, 则应从 $g(x)$ 一次项 x 的位置进入, 相当于作 $xm(x)$ 运算后再去做除法。本题 $m(x)$ 从 $x^{n-k} = x^3$ 的位置进入, 相当于作 $x^{n-k}m(x)$ 运算后再去除以 $g(x)$ 。每编一个码需化 $n = 7$ 拍时间。前 4 拍时开关 k_1, k_2 在位置 1, 消息位先 m_3 再 m_2, m_1, m_0 依次输入除法器做 $x^{n-k}m(x)/g(x)$ 运算, 同时依次将该 4 个码元输出。到第 4 拍完成时, 除法器移寄存器里的数据就是余式系数。后 3 拍消息停止输入(空 3 拍), 开关 k_1, k_2 倒向位置 2, 移寄存器断开反馈线后不再起除法器而仅起一般移寄存器作用, 其中的数据分 3 拍依次移出, 作为第 5 到第 7 循环码校验位的输出。

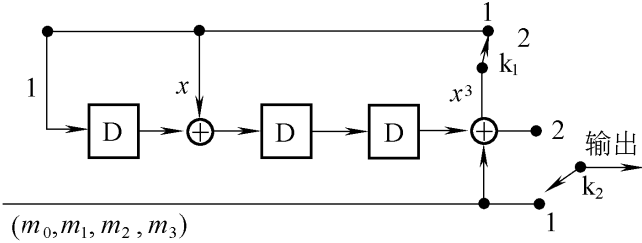


图 5.4.3 用除法器实现(7,4)循环码编码器

循环码将生成矩阵简化为生成多项式, 从而将与编码矩阵对应的硬件阵列(平面型)简化为带反馈的移寄存器(直线型)。针对循环码的特点, 在译码上也出现了许多高效的算法, 如

捕错译码、大数逻辑译码等,限于篇幅,这里不再讨论译码问题。

一个码可以兼有许多特点,循环特征仅是其中之一。前面讲到过的汉明码也可以兼有循环特征,这类码就叫作循环汉明码,其分组长度是 $n = 2^m - 1$, 校验位 $n - k = m$, 而任何码字的循环依然是码字。同样,兼有循环特征的高莱码叫作循环高莱码,比如用生成多项式 $g(x) = x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1$ 产生的线性(23, 12)高莱码就是循环高莱码。在无线信道上应用最广泛的 BCH 码、RS 码也是循环码,它们在具有循环特性的基础上又兼有另外一些特点。总之,当前实用的线性分组码几乎都是循环码。下面,我们对最常用的一些循环码作一个简要介绍,详细内容可参见纠错编码的有关书籍。

CRC 码

并不是任何 n, k 的取值都能产生循环码的,因为 x^{n+1} 的因式数目有限,它们能够组合出来的多项式阶数也有限。为了满足实践中对 n, k 取值的多样性要求,循环码作为线性分组码的子类同样也可以采用缩短的办法。缩短循环码就是在 (n, k) 循环码的 2^k 个码字中挑选出前 i 位均为 0 的所有码字,组成一个新的 $(n - i, k - i)$ 缩短循环码码集,该码集是原循环码码集的一个子集,子集里所有码多项式的阶数均小于 $n - i$ 且能够被生成多项式 $g(x)$ 整除。反之,次数小于 $n - i$ 的所有 $g(x)$ 倍式一定包含于该子集中,是 $(n - i, k - i)$ 缩短循环码的一个码多项式。

既然 $(n - i, k - i)$ 缩短循环码由前 i 位均为 0 的码字截去前 i 位构成,在这过程中码的重量没有变,校验位的数量也没有变 $(n - i - (k - i) = n - k)$, 因此 $(n - i, k - i)$ 缩短循环码的纠错能力与原 (n, k) 循环码完全一样,只是码率 R 下降了,由 k/n 变为 $(k - i)/(n - i)$ 。另外,循环码的外部特征在缩短循环码中已不复存在,缩短码码字的循环未必仍是码字;但循环码的内部特征仍然存在,即所有码多项式一定能够被 $g(x)$ 整除。这样,缩短循环码的编、译码可以借用循环码的方法,将消息多项式 $m(x)$ 乘以 x^{n-k} 后除以 $g(x)$ 得到余式 $r(x)$,再将 $r(x)$ 加在 $x^{n-k}m(x)$ 后面即可,所不同的是这里的消息多项式 $m(x)$ 由 $(k - i)$ 项组成而不是 k 项组成。缩短循环码用于检错时也和循环码一样,只要将接收序列除以 $g(x)$ 后检查其余式即可,余式全 0(除尽)则表示接收的是码字($g(x)$ 的倍式),否则就不是码字。

缩短循环码的最大应用在于帧校验,这就是在数据通信中大家所熟悉的循环冗余校验码(CRC: Cyclic Redundancy Check)。在数据通信中,信息都是先划分成小块再组装成帧后(或叫分组、包、信元,仅名称不同而已)在线路上统计复用传送或存入共同物理介质的,帧尾一般都留有 8, 12, 16 或 32 位用作差错校验。如把一帧视为一个码字,则其校验位长度 $n - k$ 不变而信息位 k 和码长 n 是可变的,正符合 $(n - i, k - i)$ 缩短循环码的特点。只要以一个选定的 (n, k) 循环码为基础,改变 i 值,就能适用于任何信息长度的编码。

例 5 4 6 用于 HDLC, X. 25, ISDN 和 7 号信令的 CRC ITU T 循环冗余校验码的生成多项式为 $g(x) = x^{16} + x^{12} + x^5 + 1$, 说明其编码过程及检错原理。

解: 帧结构如图 5 4 4(a)所示, CRC 编码器如图 5 4 4(b)所示。

根据循环码定义, $g(x)$ 必定是 $x^n + 1$ 的因式,而 $g(x) = x^{16} + x^{12} + x^5 + 1$ 是本原多项式,它所能整除的 $x^n + 1$ 中 n 的最小值是 $2^{16} - 1 = 65535$, 又因 $n - k = 16$, 所以原 (n, k) 循环码是(65535, 65519) 循环码。

将该码所有前 i 位为零的码字去除前 i 位后集合在一起就构成 $(n - i, k - i)$ 缩短循

环码, 表现为 $k-i$ 信息位加上 16 位 CRC 组成长度 $(n-i)$ 的一个帧。由于 i 可变, 因此帧长可变, 但不能超过原循环码 $n=65535$ 的长度。

实施 CRC 编码时, 信息在输出的同时由位置 x^{16} 输入移寄存器, 相当于信息移 16 位后即 $x^{16}m(x)$ 再除以 $g(x)$ 。输完 $k-i$ 信息位后断开移寄存器的反馈线, 将此时移寄存器内的数据 (余式, 即 CRC 校验位) 移位输出即可。

接收端校验时, 只要将整个帧除以 $g(x)$ 检查余式是否为零即可。如为零, 说明传输无误; 如余式非零, 说明该帧有差错, 须反馈重发或丢弃。除法器的结构与图 5 4 4 (b) 所示相同, 只是接收序列由最左端而不是从 x^{16} 处输入而已。

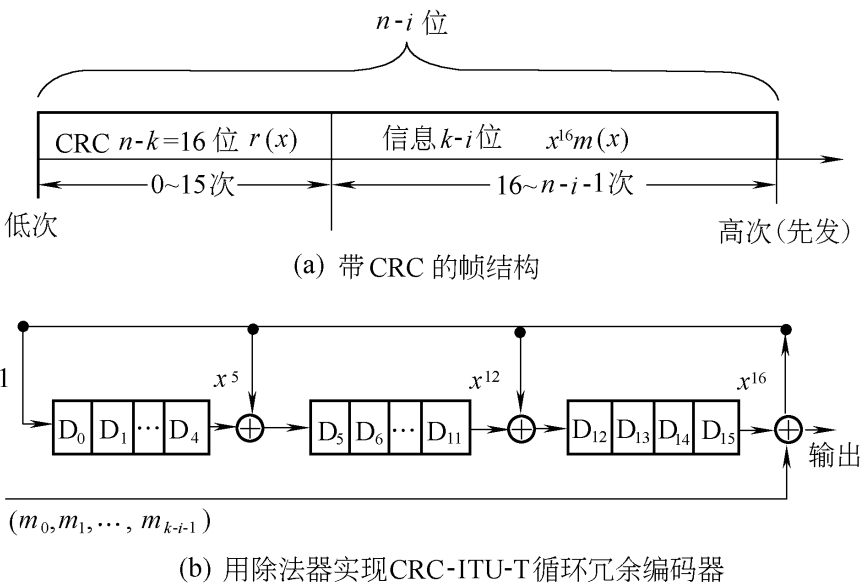


图 5 4 4 CRC 码编码器

$g(x)$ 本身也是码字, 而且是最轻的码字, 所以本码的 $d_{\min}=4$, 能纠一个差错 ($t=1$) 或检 3 个差错。另外, 由 $g(x)$ 生成的码字的重量一定是偶数, 所以它能检出所有奇数个差错。

BCH 码

BCH(Bose: Chaudhuri Hocquenghem) 码是循环码中的一大子类, 它可以是二进制码, 也可以是非二进制码。二进制本原 BCH 码具有下列参数:

$$\begin{aligned} n &= 2^m - 1 \\ n - k &= mt \\ d_{\min} &= 2t + 1 \end{aligned} \tag{5 4 42}$$

式中 $m(m \geq 3)$ 和纠错能力 $t(t \leq 2^{m-1})$ 是任意正整数。

BCH 码的基本特点是其生成多项式 $g(x)$ 包含 $2t$ 个连续幂次的根。由上面关于循环码的论述可知, 若在二元域 $GF(2)$ 上把 x^n+1 分解为 l 个最小多项式 $m_i(x)$, $i=1, 2, \dots, l$ 之积, 其中 l_1 个组成 $g(x)$ 而剩余的组成 $h(x)$, 则包含于 $g(x)$ 中的最小多项式一定满足

$$m_i(x) \mid g(x) \mid C(x) \mid (x^n + 1) \tag{5 4 43}$$

式中“ \mid ”表示整除, $C(x)$ 表示码多项式。由近世代数可进一步得知, 在二元扩域 $GF(2^m)$ 上可把 x^n+1 分解为如下 n 个根的乘积

$$(x^n + 1) = (x + \alpha^0)(x + \alpha^1)(x + \alpha^2) \dots (x + \alpha^{n-1}) \tag{5 4 44}$$

式中, α 是 $GF(2^m)$ 上本原元, $n = (2^m - 1)$ 。若对于每个 j , $j=1, 2, \dots, 2t$, 均有

$$(x + \alpha^j) \mid g(x) = \prod_{i=1}^{l_1} m_i(x) \quad (5.4.45)$$

换言之, $g(x)$ 包含 $2t$ 个连续幂次的根, 则由该 $g(x)$ 生成的循环码就是纠错能力不小于 t 的 BCH 码。BCH 的出现为通信系统设计者们在纠错能力、码长和码率的灵活设计上提供了很大的选择余地, 加上其构码方法带来的译码特点, 使之可以用伯利坎普 (Berlekamp) 迭代译码等通用、高效的译码算法, 以致 BCH 码从 70 年代起已成为线性分组码的主流。

RS 码

RS 码 (Reed Solomon 里德 索罗门码) 属于 BCH 码的一个子类, 是一种 q 进制 ($q \geq 2$) 的 BCH 码, 其码字的每个码元取值于符号集 $\{0, \alpha^0, \alpha^1, \dots, \alpha^{q-2}\}$, 实用时通常选取 q 为 2 的幂次 ($q = 2^m$), 以便一组 m 个信息比特可以一一映射到 q 个符号之一, 也便于与 4, 8, 16, 32, ... 信号点集的 PSK 或 QAM 调制相适应。近年来也常采用 $m = 8$ 即 256 进制, 以便将整个 8 比特字节变为 RS 码的一个码元。如用 N 表示 RS 码的码长, K 表示信息符号长度, $N - K$ 表示校验符号长度, 则 RS 码的参数可用以下式子来表达

$$\begin{aligned} N &= q - 1 = 2^m - 1 \\ K &= 1, 2, 3, \dots, N - 1 \\ D_{\min} &= N - K + 1 \end{aligned} \quad (5.4.46)$$

式中 D_{\min} 是码的最小距离, 对照 (5.4.18) 式可知, RS 码一定是极大最小距离 (MDC) 码。这种码可确保纠正 t 个符号差错, t 为

$$t = \text{INT}[(D_{\min} - 1)/2] = [(N - K)/2] \quad (5.4.47)$$

当然, 这种码也可以扩展或者缩短, 所用方法与前面描述过的二进制码的做法一样。

RS 码的重量分布是已知的。重多项式第 i 次项的系数 (重量为 i 的码字个数) 是

$$A_i = \frac{n}{i} (q - 1) \sum_{j=0}^{i-D_{\min}} (-1)^j \binom{i-1}{j} q^{i-j-D_{\min}} \quad i \geq D_{\min} \quad (5.4.48)$$

RS 码之所以重要, 原因之一是该码的距离特性好, 是 MDC 码。第二是因为存在一种有效的硬判决译码算法, 使得在许多需要长码的应用场合, 该码能够被实现。第三是 q 进制 RS 码的二进衍生码具有良好的抗突发差错能力。对于一个编好的 $q = 2^m$ 进制 (N, K) RS 码, 如果不以 q 进制调制发送 (1 符号间隔发 1 码元), 而是将每码元对应为 m 比特后以二进制发送 (用 m 符号间隔发 1 码元), 这实际上就是把 q 进制 (N, K) RS 码化作了 (mN, mK) 二进衍生码, 这样的二进衍生码特别适用于纠突发差错, 下面举例说明之。

例 5.4.7 某八进制 (7, 3) RS 码的符号集与对应的 3 比特组及多项式如表 5.4.4 所示。

表 5.4.4 (7, 3) RS 二进衍生码

符号集	多项式	3 比特组	符号集	多项式	3 比特组
0	0	000	3	$\alpha + 1$	011
$\alpha^0 = 1$	1	001	4	$\alpha^2 + 1$	110
$\alpha^1 = \alpha$		010	5	$\alpha^2 + \alpha + 1$	111
2	2	100	6	$\alpha^2 + \alpha$	101

注: 关系式 $\alpha^7 = \alpha^0 = 1, \alpha^i + \alpha^i = 0, \alpha^3 = \alpha + 1$

$C_{RS2}(111, 011, 010, 011, 110, 110, 010)$ 长度 5 的突发差错影响的符号数与位置有关。

该表用本原多项式 $P(x) = x^3 + x + 1$ 的根 的各次幂及零元素组成 8 个符号,用关系式 $x^7 = 1$ 和 $x^3 + x + 1 = 0$ 找出各次幂与 3 比特组、多项式的关系,详细过程略。若八进制信息符号为 $5\ 3$,生成多项式是

$$G = \begin{matrix} & 1 & 0 & 0 & & 1 & & \\ & & & & 3 & & & 3 \\ 0 & 1 & 0 & & 6 & 6 & 1 & 2 \\ & & & & 5 & 4 & 1 & 4 \\ & 0 & 0 & 1 & & & & \end{matrix}$$

求编出的八进制 RS 码及其二进衍生码各是什么？纠突发差错能力如何？

解： $C = (\begin{smallmatrix} 5 & 3 \end{smallmatrix}) G$

$$= (\begin{smallmatrix} 5 & 3 & & & 5 & 3 & + & 3 & 6 & + & 5 & , & 5 & + & 3 & 6 & + & 4 & , & 5 & + & 3 & + & , \\ & 5 & 3 & + & 3 & 2 & + & 4 \end{smallmatrix})$$

$$= (\begin{smallmatrix} 5 & 3 & , & , & + & 2 & + & 6 & , & 6 & + & 2 & + & 5 & , & 5 & + & 3 & + & , & + & 5 & + & 5 \end{smallmatrix})$$

$$= (\begin{smallmatrix} 5 & 3 & , & , & 3 & , & 4 & , & 4 & , & \end{smallmatrix})$$

由表 5 4 4 可得二进衍生码为 $C_{RS2} = (111,011,010,011,110,110,010)$

由于 RS 码是 MDC 码,必有 $D_{min} = N - K + 1 = 7 - 3 + 1 = 5$, $t = INT[(5 - 1)/ 2] = 2$

所以八进制 RS 码能纠每码字两个符号的差错。对于其二进衍生码,若以二进制信号在信道上传送,突发差错长度 4 比特时最多影响到两个八进制符号,可纠正;若突发差错长度等于 5 时,可能只影响两个八进制符号,也可能跨三个八进制符号,就不一定能纠正了。从上例可以得到一般的结论:若 q 进制 ($q = 2^m$) RS 码的纠错能力是 t 个 q 进制符号,那么它的二进衍生码能纠正的二进制突发差错长度 b 为

$$b = (t - 1) m + 1 \tag{5 4 49}$$

由于 RS 码纠突发差错能力强,现已被广泛应用于无线通信和广播,如 GSM 蜂窝移动通信中采用(224,184)RS 码,欧洲数字视频广播 DVB(S/ T/ C) 统一采用(204,188)RS 码。

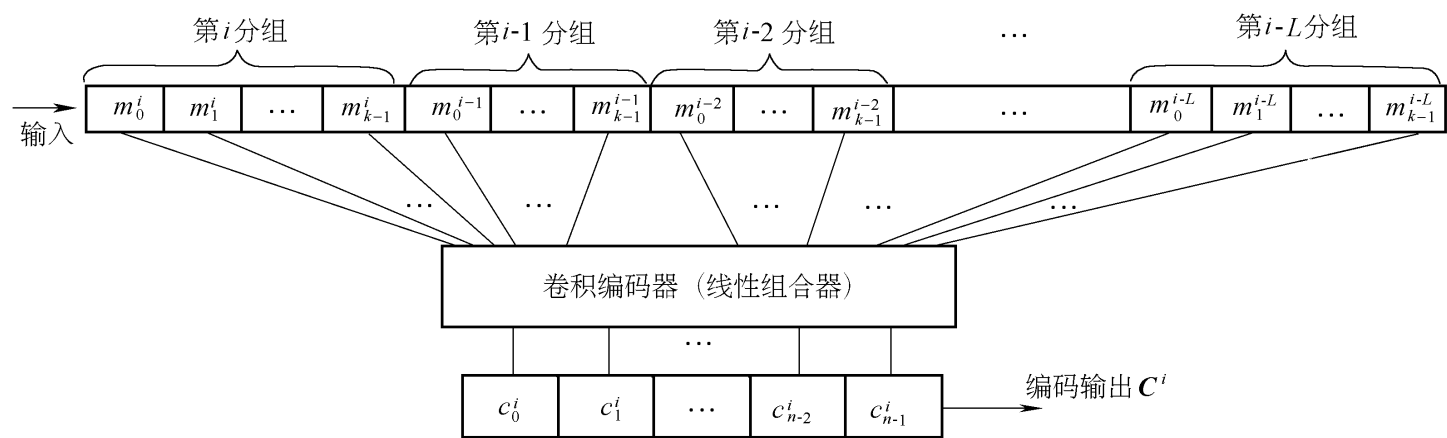
5.5 卷 积 码

前面研究过的各种分组码都是将序列切割成分组后孤立地进行编译码,分组与分组之间没有任何联系。从信息论的角度看,这样做忽略了各信息分组之间的联系,必然丧失一部分相关信息,且信息序列切割得越碎(码字越短),丧失的信息就越多。我们自然想到可把分组码长 n 尽量搞大,但译码复杂度的指数上升使这条路走不远。于是想到,在码长 n 有限时,能否将有限个分组的相关性信息添加到码字里从而等效地增加码长?译码时能否利用前后码字的相关性将前面的译码信息反馈到后面供作译码参考?这些想法导致了卷积码的产生。卷积码最早由埃利斯(Elias,1955)提出,它的主要问题在于译码。

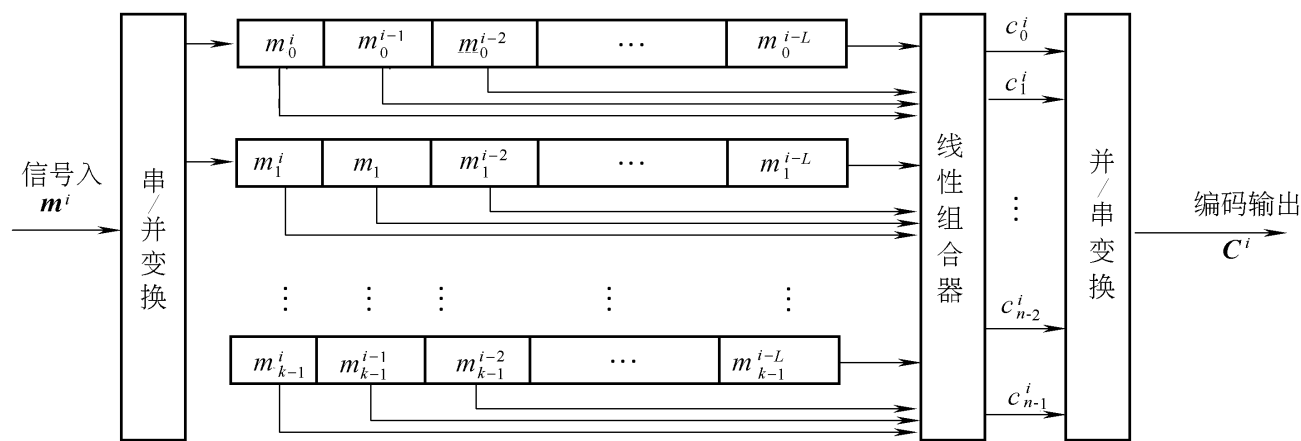
5.5.1 卷积码的基本概念和描述方法

卷积码是一个有限记忆系统,它也将信息序列切割成长度 k 的一个个分组,与分组码不同的是在某一分组编码时,不仅参看本时刻的分组而且参看本时刻以前的 L 个分组。我们把 $L + 1$ 称为约束长度。 L 是卷积码的重要参数,为了突出特征参数常把卷积码写成 (n, k, L) 卷积码,其编码原理示意如图 5 5 1(a), (n, k, L) 卷积编码器的一般结构如图 5 5

1 (b)。



(a) 卷积编码示意



(b) 卷积编码器的一般结构

图 5 5 1 卷积编码器

由图可知,卷积码将信息序列串/并变换后存入由 k 个 $L + 1$ 级移存器构成的 $k \times (L + 1)$ 记忆阵列中,然后按一定规则对记忆阵列中的数据进行线性组合,编出当前的码元 $c_j^i, j = 0, \dots, n - 1$,最后并/串变换后将当前码字输出。

图 5 5 1(b)记忆阵列中的每一存储单元都有一条连线将数据送到线性组合器,但实际上无需每个单元都有连接。这是因为二元域线性组合时的系数只能选“0”或者“1”,选“0”时表示该项在线性组合中不起作用,对应存储单元就不需要连接到线性组合器。从图上看到,每一个码元都是 $k \times (L + 1)$ 个数据线性组合的结果,需要有 $k \times (L + 1)$ 个系数来描述组合规则,于是每一个码字需用 $n \times k \times (L + 1)$ 个系数才能描述。显然,只有将这些系数归纳为矩阵才能理顺它们的关系和便于使用。下面先给出一个具体例子,再推广到一般。

例 5 5 1 某二进制(3,2,1)卷积编码器如图 5 5 2 所示。若本时刻 $i = 1$ 信息比特组是 $m^1 = (m_0^1, m_1^1) = (01)$, 上一时刻 $i = 0$ 的信息比特组 $m^0 = (m_0^0, m_1^0) = (10)$, 试用矩阵表示该编码器,并计算输出码字。

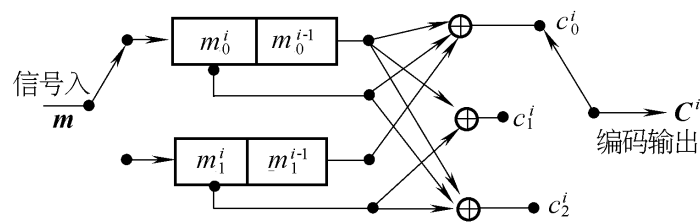


图 5 5 2 二进制(3,2,1)卷积编码器

解：用 g_{kj}^l 表示记忆阵列第 k 行 ($k=0,1$) 第 l 列 ($l=0,1$) 对第 j 个 ($j=0,1,2$) 码元的影响。令参与组合(指有连线接到模 2 加法器)者相应的系数 $g_{kj}^l=1$, 否则 $g_{kj}^l=0$ 。由图中的接线可以得到如下 $n \times k \times (L+1) = 3 \times 2 \times 2$ 个系数：

$$\begin{aligned} g_{00}^0 &= 1, & g_{01}^0 &= 0, & g_{02}^0 &= 1, & g_{00}^1 &= 1, & g_{01}^1 &= 1, & g_{02}^1 &= 1, \\ g_{10}^0 &= 0, & g_{11}^0 &= 1, & g_{12}^0 &= 1, & g_{01}^1 &= 1, & g_{11}^1 &= 0, & g_{12}^1 &= 0. \end{aligned}$$

设编码发生在本时刻 $i=1$, 记忆矩阵内已存有过本时刻和上一时刻的信息组 $m^1=(01)$ 和 $m^0=(10)$, 用 k 行 n 列 (2×3) 系数矩阵

$$G_0 = \begin{pmatrix} g_{00}^0 & g_{01}^0 & g_{02}^0 \\ g_{10}^0 & g_{11}^0 & g_{12}^0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \quad \text{及} \quad G_1 = \begin{pmatrix} g_{00}^1 & g_{01}^1 & g_{02}^1 \\ g_{10}^1 & g_{11}^1 & g_{12}^1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

分别描述本时刻 ($i=0$) 和上一时刻 ($i=1$) 的信息比特组 m^{i-0} 、 m^{i-1} 对编码的影响,

则本时刻 $i=1$ 的编码输出是：

$$\begin{aligned} C^0 &= (c_0^0, c_1^0, c_2^0) = m^1 G_0 + m^0 G_1 \\ &= (01) \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} + (10) \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix} = (011) + (111) = (100) \end{aligned}$$

上例中, 系数矩阵 G_0 、 G_1 的设定具有一般性。以时刻 i 为基准(一般可将 i 理解为编码时刻或当前时刻, 这个时刻在编码过程中是不断向前滑移的), 把 i 以前的第 l 个信息比特组 $m^l = (m_0^{i-l}, m_1^{i-l}, \dots, m_{k-1}^{i-l})$ 对线性组合的影响用一个 $k \times n$ 的生成子矩阵 G_l 来表示：

$$G_l = \begin{pmatrix} g_{00}^l & g_{01}^l & \dots & g_{0(n-1)}^l \\ g_{10}^l & g_{11}^l & \dots & g_{1(n-1)}^l \\ \dots & \dots & g_{kj}^l & \dots \\ g_{(k-1)0}^l & g_{(k-1)1}^l & \dots & g_{(k-1)(n-1)}^l \end{pmatrix} \quad (5.5.1)$$

其中 g_{kj}^l 表示图 5.5.1(b) 记忆阵列第 k 输入行 ($k=0,1,\dots,k-1$) 第 l 时延列 ($l=0,1,\dots,L$) 对第 j 个 ($j=0,1,\dots,n-1$) 输出码元的影响, $g_{kj}^l \in (0,1)$ 。

设编码器的初始状态为零(记忆阵列全体清 0), 则随着一个个 k 比特信息组 ($m^0, m^1, \dots, m^L, m^{L+1}, \dots$) 的输入, 编码器源源不断地输出码字 ($C^0, C^1, \dots, C^L, C^{L+1}, \dots$),

$$\begin{aligned} \text{在时刻 } i=0 \text{ 时,} & \quad C^0 = m^0 G_0 \\ i=1 \text{ 时,} & \quad C^1 = m^1 G_0 + m^0 G_1 \\ \dots & \quad \dots \\ i=L \text{ 时,} & \quad C^L = m^L G_0 + m^{L-1} G_1 + \dots + m^0 G_L \\ i=L+1 \text{ 时,} & \quad C^{L+1} = m^{L+1} G_0 + m^L G_1 + \dots + m^1 G_L \\ \dots & \quad \dots \end{aligned}$$

或等效地写成如下半(单边)无限矩阵的形式

$$C = (C^0 \ C^1 \ C^2 \ \dots) = mG = \begin{pmatrix} m^0 & m^1 & m^2 & \dots \end{pmatrix} \begin{pmatrix} G_0 & G_1 & \dots & G_L & 0 & 0 & 0 \\ 0 & G_0 & G_1 & \dots & G_L & 0 & 0 \\ 0 & 0 & G_0 & G_1 & \dots & G_L & 0 \\ 0 & 0 & 0 & \dots & \dots & \dots & \dots \end{pmatrix} \quad (5.5.2)$$

定义 G 为卷积码的生成矩阵,它是半无限的,因为输入的信息序列本身是半无限的。还可以把它写成如下的数学表达式

$$C^i = \sum_{l=0}^L m^{i-l} G^l \quad (5.5.3)$$

(5.5.3)式可视作一个无限长矩阵序列 m^i 与一个有限长矩阵序列 G_l 的卷积运算 $m^i * G_l$,这也就是卷积码名称的来历。

以上把 $k \times n$ 个系数 g_{kj}^l 归结为一个时延 l 的生成子矩阵 G_l ,再把 $L+1$ 个生成子矩阵 G_l 归并为一行,再将该行向右、向下延伸得到生成矩阵 G ,这样做较好地描述了卷积码,但还是需要用 $L+1$ 个 $k \times n$ 矩阵。考虑到子矩阵 G_l 和 G_{l+1} 内同一位置上的两系数 g_{kj}^l 和 g_{kj}^{l+1} 都表示记忆阵列第 k 行对第 j 输出码元的影响,两者仅差一列(一个时延 D),完全可以表达成 $g_{kj}^l D^l + g_{kj}^{l+1} D^{l+1}$ 的形式。这样,可用一个多项式矩阵来表示一个卷积码

$$\begin{aligned} G(D) &= G_0 + G_1 D + \dots + G_L D^L \\ &= \begin{pmatrix} g_{00}(D) & g_{01}(D) & \dots & g_{0(n-1)}(D) \\ g_{10}(D) & g_{11}(D) & \dots & g_{1(n-1)}(D) \\ \dots & \dots & \dots & \dots \\ g_{(k-1)0}(D) & g_{(k-1)1}(D) & \dots & g_{(k-1)(n-1)}(D) \end{pmatrix} \end{aligned} \quad (5.5.4)$$

多项式矩阵 $G(D)$ 中的每一个元素都是多项式,其通式是

$$g_{kj}(D) = g_{kj}^0 + g_{kj}^1 D + g_{kj}^2 D^2 + \dots + g_{kj}^L D^L = \sum_{l=0}^L g_{kj}^l D^l \quad (5.5.5)$$

从后面的叙述将看到,借助信号流图法可直接得到多项式矩阵 $G(D)$,所以可按信号流图法中的名称,把矩阵 $G(D)$ 定义为转移函数矩阵,其 k 行 j 列元素 $g_{kj}(D)$ 是第 k 行并行输入支路与第 j 个并行输出支路之间的转移函数。

一旦转移函数矩阵 $G(D)$ 给定,卷积编码器的结构也就给定,请看下例。

例 5.5.2 某二元 $(3, 1, 2)$ 卷积码的转移函数矩阵 $G(D) = (1, 1 + D, 1 + D + D^2)$,试画出编码器结构图。

解: 根据转移函数矩阵,

$$\begin{aligned} g_{00}(D) &= g_{00}^0 + g_{00}^1 D + g_{00}^2 D^2 = 1 \\ g_{01}(D) &= g_{01}^0 + g_{01}^1 D + g_{01}^2 D^2 = 1 + D \\ g_{02}(D) &= g_{02}^0 + g_{02}^1 D + g_{02}^2 D^2 = 1 + D + D^2 \end{aligned}$$

得

$$\begin{aligned} g_{00}^0 &= 1, & g_{00}^1 &= 0, & g_{00}^2 &= 0, \\ g_{01}^0 &= 1, & g_{01}^1 &= 1, & g_{01}^2 &= 0, \\ g_{02}^0 &= 1, & g_{02}^1 &= 1, & g_{02}^2 &= 1. \end{aligned}$$

编码器应有一行($k=1$)、3列($L+1=3$)的记忆阵列,记忆阵列在线性组合中的作用由系数规定,而系数来自转移函数矩阵中各转移函数的各次幂系数,根据系数可画出卷积编码器结构如图 5.5.3 所示。

以上的卷积码描述方法将矩阵、多项式与编码器结构的关系揭示得清清楚楚,但并没能揭示卷积码的内在特性。在这点上,状态图和网格图提供了很好的描述工具。

从图 5.5.1(b)看到,卷积编码器在 i 时刻编出的码字不仅取决于本时刻的输入信息组

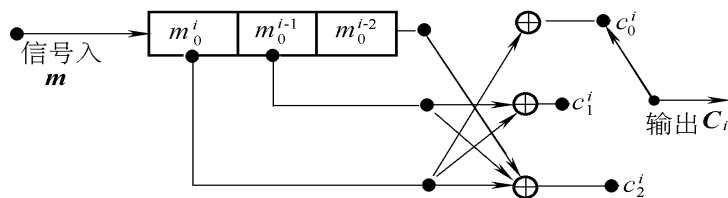


图 5 5 3 二元(3,1,2)卷积编码位

m^i , 而且取决于 i 之前存入记忆阵列的 L 个信息组, 换言之取决于记忆阵列的内容或称它为编码器的状态, 用函数形式表示是

$$C^i = f(m^i, m^{i-1}, \dots, m^{i-L}) = f(m^i, S^i) \quad (5.5.6)$$

式中

$$S^i = h(m^{i-1}, \dots, m^{i-L}) \text{ 或写成} \\ S^{i+1} = h(m^i, m^{i-1}, \dots, m^{i-L+1}) = h(m^i, S^i) \quad (5.5.7)$$

(5.5.6)式和(5.5.7)式的涵义是: 本时刻输入信息组 m^i 和编码器状态 S^i 一起决定了编码输出 C^i 和下一状态 S^{i+1} 。由于编码器状态和信息组花样都是有限数量的, 所以可以用一个信息组 m 触发的状态转移图来描述一个卷积码。

例 5.5.3 同上例的(3,1,2)卷积码, 转移函数矩阵 $G(D) = (1, 1 + D, 1 + D + D^2)$, 编码器结构如图 5.5.3。试用状态流图来描述该码。假如输入信息序列是 10110..., 输出码字是什么?

解: 本题 $n = 3, k = 1, L = 2$, 记忆阵列为 1 行 3 列, 其中第 1 列是本时刻输入信息 m_0^i , 第 2, 3 列是记忆信息 m_0^{i-1}, m_0^{i-2} , 它们的 4 种组合决定了编码器的 4 个状态。 m_0^i 和 m_0^{i-1}, m_0^{i-2} 一起又决定了编出的码字和编码器的下一个状态。把各种可能的情况汇总列表如表 5.5.1 所示:

表 5.5.1(a)

编码器状态的定义

状态	$m_0^{i-1} m_0^{i-2}$
S_0	00
S_1	01
S_2	10
S_3	11

表 5.5.1(b)

不同状态与输入时编出的码字

状态 \ 输入	输入	
	$m_0^i = 0$	$m_0^i = 1$
S_0	000	111
S_1	001	110
S_2	011	100
S_3	010	101

表 5.5.1(c) 不同状态 S^i 与

输入时的下一状态 S^{i+1}

状态 \ 输入	输入	
	$m_0^i = 0$	$m_0^i = 1$
S_0	S_0	S_2
S_1	S_0	S_2
S_2	S_1	S_3
S_3	S_1	S_3

比表更为简练和直观的方法是采用编码矩阵和状态流图:

$$\begin{array}{c} \begin{array}{cccc} & S_0 & S_1 & S_2 & S_3 \\ \begin{array}{c} S_0 \\ S_1 \\ S_2 \\ S_3 \end{array} & \begin{array}{c} 000 \\ 001 \\ \cdot \\ \cdot \end{array} & \begin{array}{c} \cdot \\ \cdot \\ 011 \\ 010 \end{array} & \begin{array}{c} 111 \\ 110 \\ \cdot \\ \cdot \end{array} & \begin{array}{c} \cdot \\ \cdot \\ 100 \\ 101 \end{array} \end{array} \\ \text{编码矩阵 } C = \end{array}$$

编码矩阵第 i 行第 j 列的元素表示由状态 S_{i-1} 转移到下一状态 S_{j-1} 时发送的码字, 若

矩阵元素是“.”,说明这种状态转移是不可能事件。比如从状态 S_0 转移到下一状态 S_1 就不可能,因为输入比特只有 0 或 1 两种可能,只能对应两种转移,从表 5 5 1(c)看出状态 S_0 只能转移到状态 S_0 或 S_2 。

图 5 5 4 是状态流图,圆圈代表状态,箭头代表转移,与箭头对应的标注,比如 0/010,表示输入信息 0 时编出码字 010。每个状态都有两个箭头发出,对应输入分别是 0、1 两种情况下的转移路径。假如输入信息序列是 10110...,从状态流图可以容易地找到输入/ 输出和状态的转移。我们可从状态 S_0 出发,根据输入找到相应箭头,随箭头在状态流图上移动,得以下结果: $S_0 \xrightarrow{1/111} S_2 \xrightarrow{0/011} S_1 \xrightarrow{1/110} S_2 \xrightarrow{1/100} S_3 \xrightarrow{0/010} S_1$ 如图 5 5 4 上粗线所示。

从上例看到,编码矩阵 C 很好地展示了状态转移规律,而状态图则为用信号流图研究卷积码奠定了基础。但美中不足的是不能记录下状态转移的轨迹,因为还缺少一根时间轴。网格图(也有人称之为格栅图、格子图、篱笆图)弥补了这一个缺点,它以状态为纵轴,以时间(周期 T 采样)为横轴,将状态转移展开于时间轴上,从而使编码全过程跃然纸上。网格图有助于发现卷积码的性能特征,有助于译码算法的推导,是分析研究卷积码的最得力工具之一。

网格图分成两部分,一部分是对编码器的描述,告诉人们从本时刻的各状态可以转移到下一时刻的哪些状态,伴随转移的输入信息/ 输出码字是什么。另一部分是对编码过程的记录,一根半无限的水平线(纵轴上的常数)标志某一个状态,一个箭头代表一次转移,每隔时间 T (相当于图 5 5 1(b)移存器一位时延 D)转移一次,转移的轨迹称为路径。两部分可以合画在一起,也可单独画,比如在描述卷积编码器本身而并不涉及具体编码时,只需第一部分网格图就够了。当状态很多、转移线很密时,网格图上难以标全伴随所有转移的输入/ 输出码字信息,此时,利用编码矩阵可看得更清楚些。

例 5 5 4 同上例的(3,1,2)卷积码,编码器结构如图 5 5 3。试用网格图来描述该码。假如输入信息序列是 10110...,输出码字是什么 ?

解: 参见例 5 5 3 所得的编码矩阵和状态流图,可得网格图和编码轨迹,如图 5 5 5 所示。

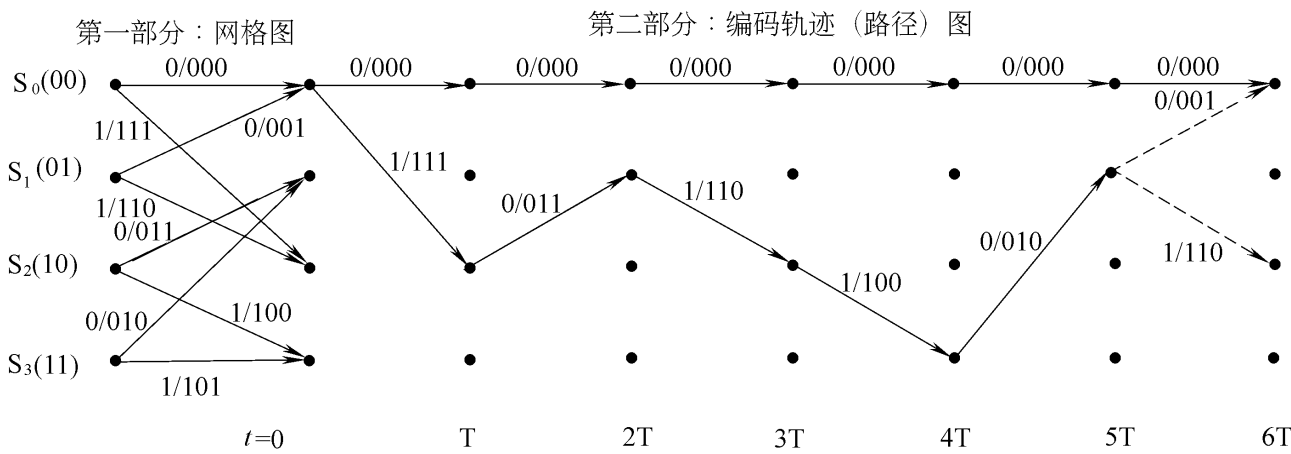


图 5 5 5 (3,1,2)卷积码网格图

由图 5 5 5 看到,当输入 5 位信息 10110 时,输出码字和状态转移是

$$S_0 \xrightarrow{1/111} S_2 \xrightarrow{0/011} S_1 \xrightarrow{1/110} S_2 \xrightarrow{1/100} S_3 \xrightarrow{0/010} S_1$$

如果继续输入第 6 位信息,信息为 0 或 1 时,状态将分别转移到 S_0 或 S_2 ,而不可能转移到 S_1 或 S_3 。网格图顶上的一条路径代表输入全 0 信息/输出全 0 码字时的路径,这条路径在卷积码分析时常被用作为参考路径。

上例中,从某一状态出发只能转移到 4 个状态之中的 2 个,可能进入到每个状态的分支也只有两条,由此可见在网格图里的编码路径并不是随意的。推广为一般结论,我们说一个码率 k/n 、约束长度 $L+1$ 的卷积码,其网格图和状态图中的状态数不会大于 2^{kL} 个;从每个状态发出的分支有 2^k 条,可能进入到每个状态的分支也有 2^k 条。

对于无限长的信息序列,每一个 k 位信息组产生一个 n 位的码字,与分组码一样。但对于有限长的信息如分帧的信息,情况就不同了。设信息序列长度为 M 个 k 位分组,由于记忆效应,编码器在输出 M 个码字后将继续输出 L 个码字才能将记忆阵列中的内容完全移出,这就导致卷积码码率下降为

$$R_c = \frac{kM}{n(M+L)} \quad (5.5.8)$$

可见,卷积码约束长度 $L+1$ 越长,信息组数 M 越短,则编码效率越低,而当 $M \rightarrow \infty$ 时,码率 $R_c = k/n$ 。从这点来看,对于短的突发信息,卷积码约束长度也应设计得短些。

5.5.2 卷积码的最大似然译码——维特比算法

卷积码的性能取决于卷积码距离特性和译码算法,其中距离特性是卷积码自身本质的属性,它决定了该码潜在的纠错能力,而译码算法是个如何将潜在纠错能力转化为实际纠错能力的问题。为此,了解卷积码距离特性是必要的。

描述距离特性的最好方法是利用网格图。设序列 $C^{(1)}$ 、 $C^{(2)}$ 是同一时刻从同一状态出发的任意两个不同的二进制码字序列,不失一般性不妨设 0 时刻从 0 状态出发。序列距离定义为两序列 $C^{(1)}$ 和 $C^{(2)}$ 在对应时刻的码字的汉明距离之和,即两序列模 2 加后的重量。由于线性卷积码的封闭性,若 $C^{(1)} \oplus C^{(2)} = C$,则 C 也是一个码字序列,有以下关系式

$$d(C^{(1)}, C^{(2)}) = W(C^{(1)} \oplus C^{(2)}) = W(C) = W(C \oplus 0) = d(C, 0) \quad (5.5.9)$$

(5.5.9)式的含义是:任意两序列的距离一定等于某一序列与全零序列的距离,等效地等于某一序列的重量。因此与研究分组码距离特性一样,可以通过研究序列重量或与全零序列的距离来研究卷积码距离特性,两序列的最小距离等于最轻序列的重量。

序列距离还与序列的长度有关。长度为一个码字的两序列,距离不可能超过码长 n ;两个码字长度的两序列,距离不可能超过 $2n$;而当序列长度趋于无穷时,距离可能趋于无穷。为此,我们定义长度 l (码字)的任意两序列的最小距离为 l 阶列距离,记作 $d_c(l)$,

$$d_c(l) = \min\{d(C^{(1)}, C^{(2)}) : C^{(1)} \oplus C^{(2)} = C\} = \min\{W(C) : C \neq 0\} \quad (5.5.10)$$

式中,下标 l 表示序列长度。当 $l \rightarrow \infty$ 时,任意两序列的最小距离叫做自由距离 d_f

$$d_f = \lim_{l \rightarrow \infty} d_c(l) = \min\{d(C^{(1)}, C^{(2)}) : C^{(1)} \oplus C^{(2)} = C\} = \min\{W(C) : C \neq 0\} \quad (5.5.11)$$

现在也有人直接把自由距离叫做最小距离,写成 d_m 。根据定义,自由距离在网格图上就是

0 时刻从 0 状态与全零路径分叉 (C=0)、经若干分支后又回到全零路径 (与全零序列距离不再继续增大) 的所有路径中, 重量最轻 (与全零序列距离最近) 的那条路径的重量。

例 5 5 5 同上例的 (3, 1, 2) 卷积码, 编码器结构如图 5 5 3。试计算该码的自由距离 d_f 。

解: 分析 0 时刻从 0 状态与全零路径分叉、又回到全零路径的所有可能的路径如图 5 5 6 所示, 其中伴随每个转移所标的数字是对应码字与全零码的距离。图中, 0 时刻分叉后的第一次转移只有一条非零分支, 列距离 $d_c(1) = 3$ 。第二次转移后 (时刻 $2T$) 有 $S_0 S_2 S_1$ 和 $S_0 S_2 S_3$ 两条路径, 重量分别是 $d[(111011), (000000)] = 5$ 和 $d[(111100), (000000)] = 4$, 选其中小者为列距离, 得 $d_c(2) = 4$ 。以此类推, 可得各阶列距离如图底部所标。比较各值, 发现 l 在 $[4, \infty]$ 范围内列距离不变, 即得自由距离 $d_f = \lim_l d_c(l) = 6$, 而具有该自由距离的路径有两条:

$S_0 S_2 S_1 S_0 S_0 \dots$
 $S_0 S_2 S_3 S_1 S_0 S_0 \dots$

$\lim_l d_c(l) = W(111, 011, 001, 000, 000 \dots) = 6$
 $\lim_l d_c(l) = W(111, 100, 010, 001, 000 \dots) = 6$

列距离不再增加的原因是序列一旦重新与全零序列汇合, 后面重合部分与全零序列的距离永远为零, 整个序列的重量也就不再增加。

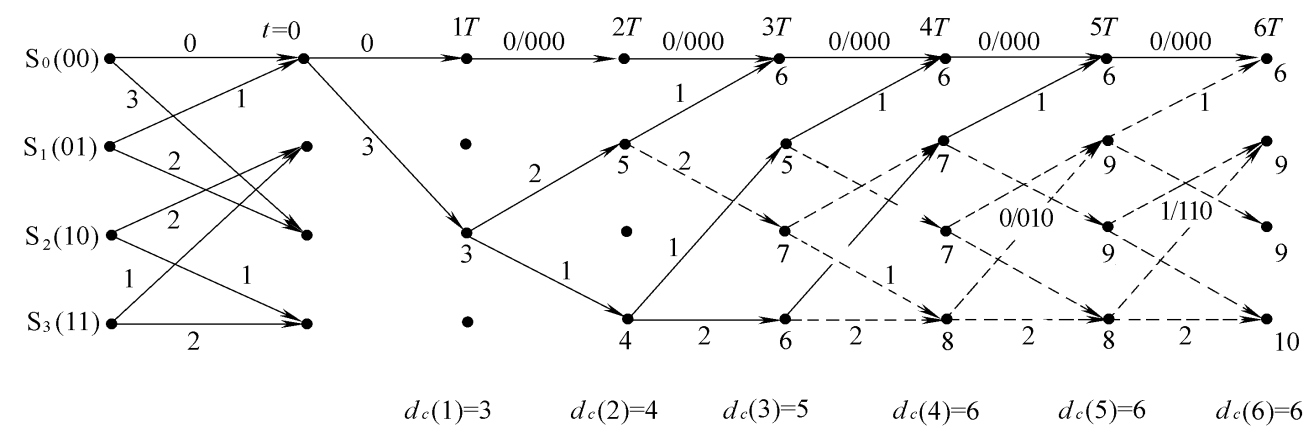


图 5 5 6 (3, 1, 2) 卷积码的自由距离 d_f

前面已经学过, 分组码的纠错能力取决于码的最小距离, 分组码的最大似然译码实际上就是最小距离译码。这些准则同样也适用于卷积码, 不同之处在于, 分组码是孤立地考虑各分组的距离, 而卷积码是通盘考虑整个序列的距离。正因为序列距离决定了卷积码特性, 衡量序列距离最主要的参数——自由距离 d_f 就成了卷积码的主要性能指标。卷积码自由距离 d_f 的计算有很多方法, 简单卷积码如上例可以直接在网格图上推得; 稍微复杂一些的卷积码可采用信号流图法, 它也最具理论价值; 而最实用的方法还是靠编程利用计算机来搜索。

信号流图可用来计算任何一个以支路为基础线性累积的物理量。如果希望这个量不是以“积”而是以“和”的形式累积, 可将这个物理量写作某个基底的幂次。状态流图就是一种信号流图, 一个状态对应一个节点, 一次转移对应一条支路, 两状态间一条路径的重量对应于信号流图两节点间一条路径的增益, 而两节点间的生成函数 (或叫转移函数) $T(D)$ 代表所有路径增益之和。解信号流图可以利用 Mason 的增益公式, 也可根据有向图列出线性状态方程从而把解图化为解方程, 还可通过图论中的等效变化解图。若由信号流图法解得 T

态方程从而把解图化为解方程, 还可通过图论中的等效变化解图。若由信号流图法解得 $T(D)$, 则不但自由距离可知, 而且有助于从理论上分析卷积码的差错控制能力。下面举例说明 $T(D)$ 和 d_f 的关系, 至于求解 $T(D)$ 的详细方法请参考有关书籍。

例 5 5 6 同上例的 $(3, 1, 2)$ 卷积码, 其状态流图如图 5 5 4 所示。试用信号流图法计算生成函数 $T(D)$, 并得出该码的自由距离 d_f 。

解: 由于自由距离是由零状态出发又回到零状态的最轻序列的重量, 我们把零状态拆分成两个节点, 一个为发点, 一个为收点, 如图 5 5 7(a)。这样, 沿着任意一条由发点到收点的路径都有一个对应的路径增益, 增益最小的路径就是最轻路径, 生成函数 $T(D)$ 就是所有路径增益之和。利用图 5 5 7(b) 的等效变化, 将 5 5 7(a) 的流图变为最简形式后求得 $T(D)$, 如图 5 5 7(c) 所示。

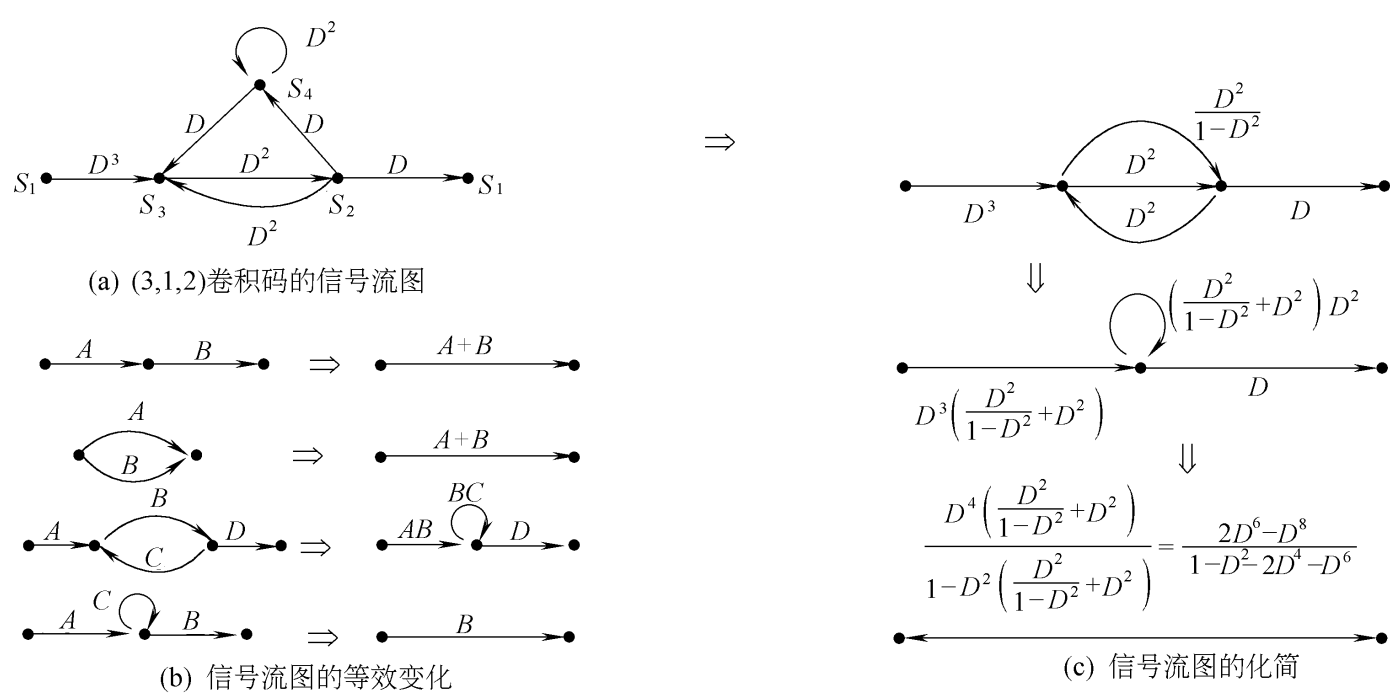


图 5 5 7 信号流图

根据化简的结果, 得生成函数 $T(D)$, 再用长除法将其展开, 得

$$T(D) = \frac{2D^6 - D^8}{1 - D^2 - 2D^4 - D^6} = 2D^6 + D^8 + 5D^{10} + \dots$$

生成函数 $T(D)$ 的每一项对应网格图上的一条非零路径, 该项的幂次指示对应非零路径的重量。因此本题的 $T(D)$ 告诉我们: 从零状态出发又回到零状态的非零路径有无数条, 其中有两重量为 6 的路径, 1 重量为 8 的路径, 5 重量为 10 的路径……。显然, 最低幂次 6 就是自由距离 d_f , 最低次项的系数就是重量等于 d_f 的路径的条数。对照图 5 5 6, 可知计算结果是正确的。

上例的生成函数虽然是针对具体问题计算的, 但其结果具有一般性。对于给定的信号流图, 解出的生成函数 $T(D)$ 均可写成以下形式

$$T(D) = \sum_{d=d_f} A_d D^d \tag{5 5 12}$$

式中 d 次项系数 A_d 代表重量为 d 从零状态出发又回到零状态的非零路径的条数, 上例的具体数据是 $A_6 = 2, A_8 = 1, A_{10} = 5, \dots$ 。

例 5 5 6 中, 我们感兴趣的物理量是与各转移对应的码字的重量, 比如一个转移对应码

字 111, 重量是 3, 又因想让它以和的形式积累, 故将 3 写成 D 的指数, 写作 D^3 , 底 D 只是载体, 本身并无特殊意义。除了距离特性之外, 利用生成函数还可以给出更多的信息。假如想统计每条路径到底包含多少次转移, 不妨再在所有转移分支上引入一个因子 N , 比如原来是 D^3 的变为 $D^3 N$, 此时生成函数也改写作 $T(D, N)$ 。那么每当路径横越一个分支时 N 的指数累积值就增加 1, 最后从 $T(D, N)$ 各项 N 的幂次可看出每条路径包含的转移次数。可以想象到, 上例结果的前两项将是 $T(D, N) = D^6 N^3 + D^6 N^4 + \dots$ 。

某些卷积码表现出一种特别的性质叫做恶性差错传播。当具有这种特性的卷积码用于二进制对称信道时, 就可能因为有限数量的信道差错而引起无限数量的译码差错。这种码可从它的状态图看出来。它含有一条从某个非零状态返回同一状态的零距离的路径(因子 $D^0 = 1$ 的路径), 这意味着可以沿着这条零距离路径环绕无限多次, 而与全零路径之间的距离并不增大。但是, 如果这条自环对应于传送“1”时, 则译码器将产生无穷多个差错。因此, 在实用中应注意识别和避免恶性卷积码。需指出的是: 系统卷积码一定是非恶性的, 但系统卷积码通常并不是性能最好的码。

对于编码器编出的任何码字序列, 在网格图上一定可以找到一条连续的路径与之对应。但在译码端, 一旦传输、存储过程中出现差错, 输入到译码器的接收码字流在网格图上就找不出一条对应的连续路径, 最多只有若干不确定的、断续的路由可供作译码参考。而译码输出的码字流必须对应一条连续路径, 否则肯定是译码差错。卷积码最小距离译码的思路是: 以断续的接收码流为基础, 逐个计算它与其它所有可能出现的、连续的网格图路径的距离, 选出距离最小者作为译码估值输出。在二进制硬判决译码情况下, 最小距离就是最小汉明距离; 在二维调制(PSK, QAM)和软判决情况下, 最小距离一般指最小欧氏(Euclidean)距离。这种以序列为基础的译码叫序列译码, 在编码理论发展过程中曾出现过多种序列译码方法, 如 Wozencraft 和 Reiffen 1961 年提出的序列译码算法, 这种算法后来由范诺于 1963 年作了修改和完善, 现在称为范诺算法, 以及齐盖吉洛(Zigangirov)1966 年和杰林克 1969 年设计出的堆栈算法等, 但这些都不是最佳译码。

卷积码本质上是一个有限状态机, 它的最佳译码器应该与有记忆信号的最佳解调器类似是一个最大似然序列估计器。所以, 卷积码的译码就是要搜遍网格图找出最可能的序列。根据译码器之前的解调器执行的是软判决还是硬判决, 搜寻网格图时所用的相似性量度可以是汉明距离, 也可以是欧氏距离, 这种最小距离准则的译码算法叫卷积码的最大似然译码。在加性高斯白噪声、 $p_n = 1/2$ 的二进制对称信道中, 这种算法的差错概率最小, 因此也是最佳译码。当前最流行的卷积码译码算法是 A.J. Viterbi 于 1967 提出的维特比算法。该算法提出的两年后, 小村(Omura)指出维特比算法等价于在一个加权图上求最短路径。1973 年福尼(G.D. Forney)又证明了维特比算法实质上就是卷积码的最大似然译码。由于最优的特性和相对适中的复杂度, 使维特比算法在 $K = 10$ 的卷积码译码中成为最普遍采用的算法。下面结合具体例子来说明维特比算法的执行过程。

例 5 5 7 同上例的 $(3, 1, 2)$ 卷积码, 其网格图如图 5 5 8 所示。设发送的码字序列是 $C = (000, 111, 011, 001, 000, 000, \dots)$, 接收的码字序列是 $R = (110, 111, 011, 001, 000, 000, \dots)$, 试用维特比算法译码。

解: (1) 用数组描述 5 5 8 网格图结构:

$$\begin{aligned}
p(1,1) &= 1, \quad c(1,1) = 000, \quad p(1,2) = 2, \quad c(1,2) = 001, \\
p(2,1) &= 3, \quad c(2,1) = 011, \quad p(2,2) = 4, \quad c(2,2) = 010, \\
p(3,1) &= 1, \quad c(3,1) = 111, \quad p(3,2) = 2, \quad c(3,2) = 110, \\
p(4,1) &= 3, \quad c(4,1) = 100, \quad p(4,2) = 4, \quad c(4,2) = 101.
\end{aligned}$$

其中, $p(4,1) = 3$, $p(4,2) = 4$ 表示到达第 4 状态的第 1、第 2 个前状态(predecessors)分别是状态 3 和 4, 对应的码字分别是 $c(4,1) = 100$ 和 $c(4,2) = 101$, 其他类推。

(2) 计算第 l 时刻接收码 R_l 相对于各码字的相似度, 称作分支量度 BM(Branch Metric)。在软判决情况下, BM 一般指欧氏距离。在二进制硬判决情况下, BM 即汉明距离

$$BM^l(i, j) = W[c(i, j) \quad R_l] \quad (5.5.13)$$

式中 $BM^l(i, j)$ 表示第 l 时刻接收码 R_l 与到达第 i 状态的第 j 个转移所对应的码字的距离。本题 $R_1 = 110$, $R_2 = 111$, $R_3 = 011$, $R_4 = 001$, $R_5 = 000, \dots$, 时刻 3 的分支量度 (见图 5.5.9) 分别是 $BM^3(1,1) = W[c(1,1) \quad R_3] = W[000 \quad 011] = 2$, 以及 $BM^3(1,2) = 1$, $BM^3(2,1) = 0$, $BM^3(2,2) = 1$, $BM^3(3,1) = 1$, $BM^3(3,2) = 2$, $BM^3(4,1) = 3$, $BM^3(4,2) = 2$ 。

(3) 计算第 l 时刻到达状态 i 的最大似然路径之相似度即路径量度 (Path Metric) $PM^l(i)$, 它是将上一时刻的路径量度 PM^{l-1} 与本时刻分支量度 BM 累加后选择其中相似度最大的一个, 对于二进制硬判决就是选汉明距离最小的一个

$$PM^l(i) = \min_j \{ PM^{l-1}[p(i, j)] + BM^l(i, j) \} \quad (5.5.14)$$

初始时, 除全零状态的 $PM^0(1) = 0$ 外, 其余 PM^0 均置为 1。

图 5.5.9 中, 时刻 3 到达状态 1 的路径可以来自状态 1 和 2 两处, 该两处前时刻的路径量度分别是 $PM^2(1) = 5$ 和 $PM^2(2) = 2$, 本时刻的分支量度分别是 $BM^3(1,1) = 2$ 和 $BM^3(1,2) = 1$, 因此时刻 3 状态 1 的路径量度

$$\begin{aligned}
PM^3(1) &= \min\{ PM^2[p(1,1)] + BM^3(1,1), PM^2[p(1,2)] + BM^3(1,2) \} \\
&= \min\{ 5 + 2, 2 + 1 \} = 3.
\end{aligned}$$

以上计算路径量度的过程实际上就是挑选到达状态 1 的最大似然路径的过程。有两条路径可达, 一条与接收码的汉明距离为 $5 + 2$, 另一条的汉明距离为 $2 + 1$, 距离越小则似然度越大, 所以取 $PM^3(1) = 3$ 隐含了选择路径 $S_1 \rightarrow S_3 \rightarrow S_2 \rightarrow S_1$ 为到达状态 1 的最大似然路径。同理, 到达其他各状态最大似然路径的 PM 分别是

$$\begin{aligned}
PM^3(2) &= \min\{ 2 + 0, 3 + 1 \} = 2, \\
PM^3(3) &= \min\{ 5 + 1, 2 + 2 \} = 4, \\
PM^3(4) &= \min\{ 2 + 3, 3 + 2 \} = 5.
\end{aligned}$$

再将时刻 3 各状态的 PM 进行比较, 显然, 到达状态 2 的路径最似然。

(4) 译码输出以及更新第 l 时刻、状态 i 对应的留存路径 (Survivor) $S^l(i)$ 。留存路径是与最大似然路径对应的码字序列, 每状态一个, 长度为 D 。留存路径每时刻按以下步骤更新一次: 设到达状态 i 的最大似然路径的前状态是 j , 则令 j 状态前时刻的留存路径作为本时刻本状态 i 的留存路径, 即 $S^l(i) = S^{l-1}(j)$ 。选择具有最小(最似然)PM 那个状态的留存路径最左边 (D 时刻之前进入)的码字作为译码输出。将各状态留存路径最左

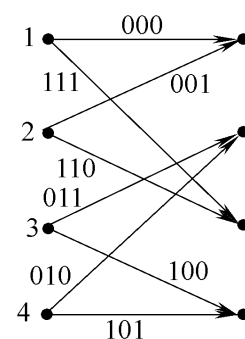


图 5.5.8 (3,1,2)卷积码
网格图结构

边的码字从各移寄存器移出,再将到达各状态的最大似然路径在时刻 l 所对应的码字从右面移入留存路径 $S^l(i)$ 。

比如在图 5 5 9 中,时刻 $l=3$ 到达状态 2 的最大似然路径来自状态 3,而前时刻状态 3 的留存路径是 $S^2(3)=000,000,000,111$ (长度 $D=4$)。比较各状态的 $PM^3(i)$,发现状态 2 是最大似然路径,其前时刻在状态 3,于是取 $S^2(3)$ 最左边的码字 000 作为译码输出。接着,将 $S^2(2)$ 最左边(最旧)的码字 000 移出,将时刻 3 到达状态 2 的转移所对应的码字 011 从右边移入,得更新后状态 2 的留存路径 $S^3(2)=000,000,111,011$ 。同理可得 $S^3(1)$, $S^3(3)$, $S^3(4)$ 。

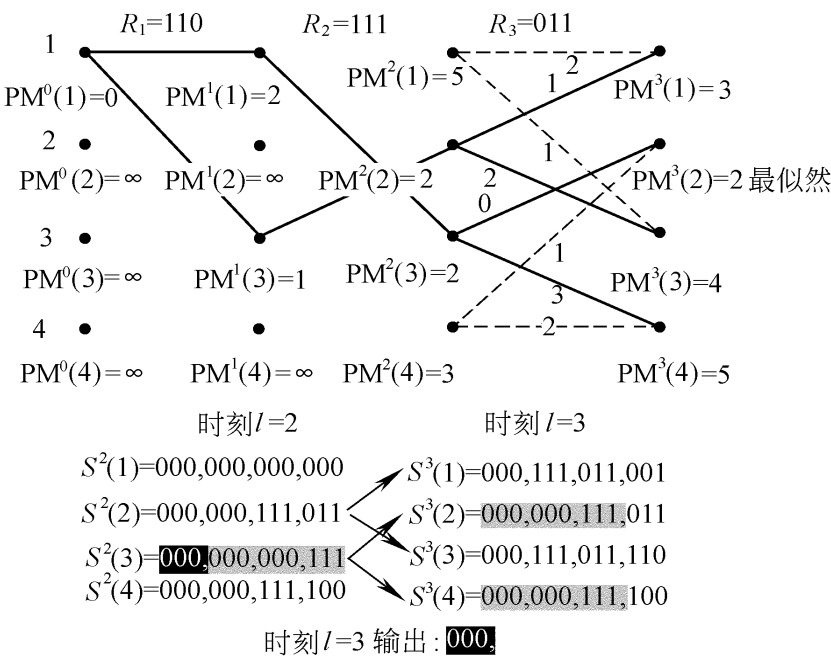


图 5 5 9 $l=3$ 时的 $BM^l(ij)$, $PM^l(i)$, $S^l(i)$ 和网格图

重复步骤(2)~(4),将维特比算法持续下去,如图 5 5 10 所示。
最后结果是

发码: 000,111,011,001,000,000,...
收码: 110,111,011,001,000,000,...
译码: 000,000,000,000,000,111,011,001,000,000,...

可见,经时延 $D=4$ 后,维特比译码克服了收码中一个码字的差错,正确译码输出。

从上例看到:

每个状态都有自己的留存路径和路径量度,但最后只有其中一个被采纳作为译码估值序列的输出。在硬判决时,支路量度 BM 表示一次转移的差错数,路径量度 PM 表示一条路径上差错数的累计,而留存路径是到达该状态差错累计数最少的那条路径所对应的码字序列片断(长度 D)。

引入适当时延能提高译码器的纠错能力。网格图上正确路径只有一条,它和其他的路径量度 PM 虽然都在持续增大,但造成增大的原因不同,统计特性也不同。正确路径的 PM 是由于码字差错造成的,增大速率取决于差错概率;而其他路径是由于路径差异造成的,PM 持续增大且上升速度快。当信道中产生突发差错时,会导致正确路径的 PM 突然增大而暂时超过其他路径,但只要突发差错长度在一定限度之内,那么经过一段时间后正确路径的 PM 总会恢复为最小。因此,引入时延就是按统计特性而不是逐码字去判决,可提高译

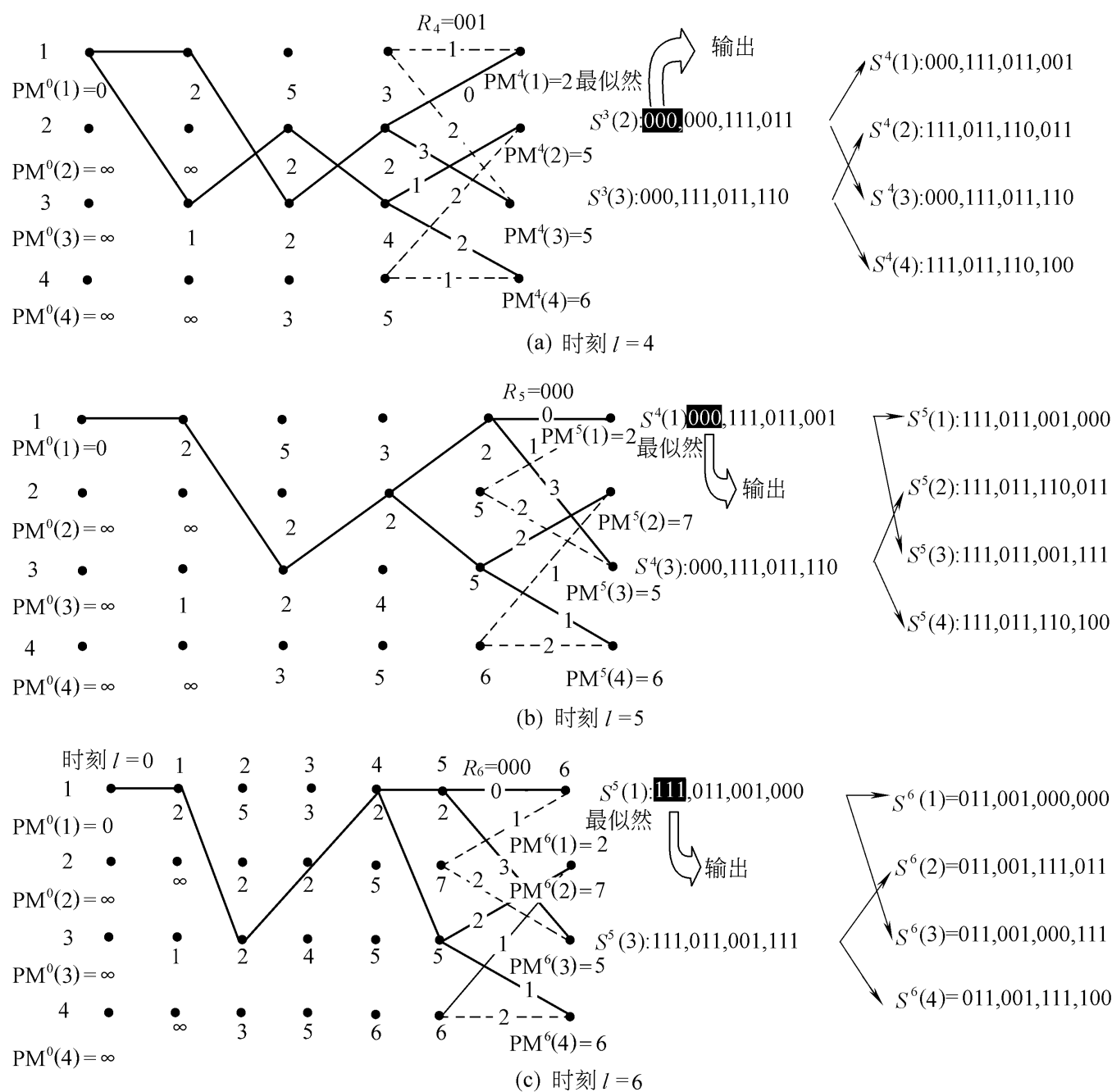


图 5 5 10 $l=4,5,6$ 时的 $BM^i(i, j)$, $PM^i(i)$, $S^i(i)$ 和网络图

码正确率。时延 D 的长度一般取为卷积码状态数的 5 倍。

各状态的留存路径有合并为一条的趋势。比较图 5 5 10(a) 和 5 5 10(c), 我们看到在时刻 $l=0$ 到 $l=4$ 的留存路径已合为一条, 这不是偶然的, 但需要一定条件, 那就是时延足够。

PM 是单调增大的, 如不处理总会趋于无穷, 所以要定期处理比如各状态 PM 同时减去同一个数。由于最大似然译码仅对各状态 PM 的相对大小进行比较, 所以同减一数对算法没影响。

一般来说, 若用维特比算法对具有 2^M 个状态的 (n, k) 卷积码进行译码, 就有 2^M 个路径量度和 2^M 条留存路径。在网格图每一时刻的每一节点, 有 2^k 条路径汇合于该点, 其中每一条路径都要计算其量度并最后比大小, 因此每个节点要计算 2^k 个量度, 这样, 在执行每一级的译码中, 计算量将随 k 和 M 成指数地增加, 这就将维特比算法的应用局限于 k 和 M 值较小的场合。

以上举的例子是硬判决维特比算法。软判决维特比算

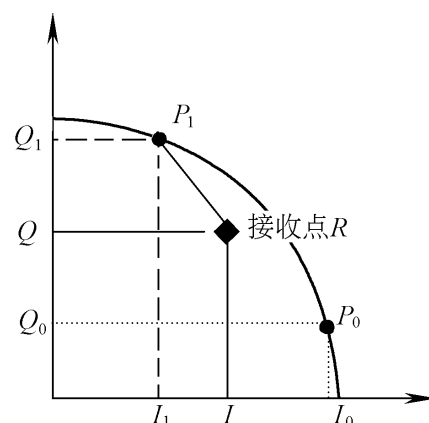


图 5 5 11 欧氏距离示意图

法的步骤与硬判决完全一样,不同点只是似然度 BM 的定义,上例的似然度是汉明距离,而软判决似然度是欧氏距离。图 5 5 11 表示 8 PSK 调制下似然度 BM 的计算。

接收点 R 离信号点 P_1 的欧氏距离的平方是

$$\begin{aligned} & (I_1 - I)^2 + (Q_1 - Q)^2 \\ &= (I_1^2 + Q_1^2) + (I^2 + Q^2) - 2(I I_1 + Q Q_1) \end{aligned}$$

同理, R 离信号点 P_0 的欧氏距离的平方是

$$(I_0 - I)^2 + (Q_0 - Q)^2 = (I_0^2 + Q_0^2) + (I^2 + Q^2) - 2(I I_0 + Q Q_0)$$

由于 $(I_1^2 + Q_1^2) = (I_0^2 + Q_0^2)$,上面前两项 $(I_i^2 + Q_i^2) + (I^2 + Q^2)$ 在比大小时不起作用,而第三项 $I I_i + Q Q_i$ 越大则欧氏距离的平方越小,说明接收点越靠近该 i 点,所以定义接收点与第 i 个信号点的支路量度 BM 为

$$BM = I I_i + Q Q_i$$

维特比算法中只要用以上定义的 BM 代替汉明距离的 BM 作为相似度,PM 是 BM 的累计,并取 PM 最大者(而不像汉明距离时取最小者)为最似然路径,算法的其余部分就都是一样的了。

5.5.3 卷积码的性能限与距离特点

卷积码的性能限由编码方法决定,而实际能否达到该性能限还与译码方法有关。在各序列等概的情况下,维特比最大似然译码等效于最佳译码,因此,当讨论卷积码性能时总是以维特比算法为基础的。

估计卷积码性能的常用方法有： 计算机模拟。如误码率不是很小比如大于 10^{-6} 时可采用,当误码率太小时可能会因耗时太多而无法实施,这与计算机运算能力有关。 推导出近似公式来计算性能限。 估算出性能的渐近线公式,信噪比越大时实际性能离渐近线越近,误差越小。

分组码的一个差错只影响一个码字,而卷积码的一个差错却要影响一个序列,为此,在讨论差错概率之前,有必要先讨论一下差错事件。简单地说,发码序列与收码序列不相同就是差错事件。但发码序列与收码序列几乎有无限多个,不可能一一讨论,为此必须利用卷积码的线性特性把问题简化。不失一般性,假定发送的是一个全零序列,则正确译码序列的轨迹应是网格图顶部水平的那条全零路径,称之为正确路径。任何偏离这条正确路径的译码估值序列都是错误路径。确切地说,把某时刻 i 从正确路径分岔出去,经若干步后在 j 时刻又合并回正确路径的这段过程定义为差错事件,相应的路径就是差错路径,如图 5 5 12 所示。

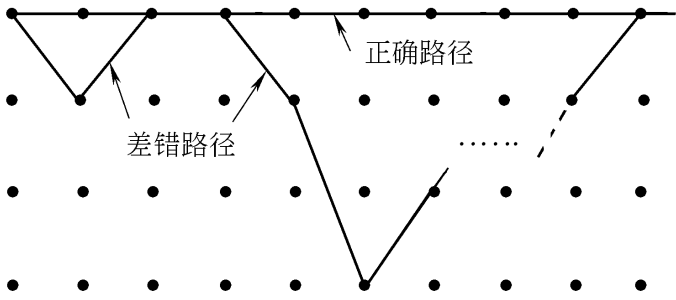


图 5 5 12 差错事件示意图

对照‘ 差错路径 ’和‘ 自由距离 ’的定义,可知差错路径与正确路径之间的距离必定大于或等于自由距离,至于大多少则不一定。由于差错事件并没有一个固定长度,计算差错事件概率只能利用起始概率(时刻 i 从正确路径分岔的概率)或者终结概率(时刻 j 汇合到全零序列的概率)来推导,两者应是等效的。下面来推导 BSC 信道(硬判决)条件下的差错概率。

维特比算法中,假如 j 时刻与全零路径汇合的某条差错路径与接收序列的距离 $CM^{(1)}$ 小于正确(全零)路径与接收序列的距离 $CM^{(0)}$,那么译码器就会选择差错路径作为最大似然路径,译码就出错了。设差错路径的重量是 d (路径上有 d 个“1”而其余为“0”),此时接收序列的重量必定大于 $d/2$ 。显然,重量为 d 的差错事件概率就是接收序列重量大于或小于 $(d+1)/2$ 而小于或等于 $d/2$ 的概率

$$P(E, d) = \sum_{l=(d+1)/2}^d \binom{d}{l} p^l (1-p)^{d-l} \quad (5.5.15)$$

式中, p 是 BSC 信道的转移概率; l 是差错个数。

利用组合公式 $\sum_{l=0}^d \binom{d}{l} = 2^d$, 设 d 为奇数, 经不等式的放大, 由(5.5.15)式得

$$\begin{aligned} P(E, d) &< \sum_{l=(d+1)/2}^d \binom{d}{l} p^{d/2} (1-p)^{d/2} < p^{d/2} (1-p)^{d/2} \sum_{l=0}^d \binom{d}{l} \\ &= 2^d p^{d/2} (1-p)^{d/2} = (4p(1-p))^{d/2} \end{aligned} \quad (5.5.16)$$

同理可证,当 d 为偶数时(5.5.16)式也成立。接着取 d 的不同值,于是总的差错事件概率是

$$P(E) = \sum_{d=d_f} A_d P(E, d) < \sum_{d=d_f} A_d (4p(1-p))^{d/2} \quad (5.5.17)$$

式中 A_d 是正整数,表示重量为 d 的差错路径的条数。

将(5.5.17)式与(5.5.12)式生成函数 $T(D)$ 相比较,可得

$$P(E) < T(D) \Big|_{D=4p(1-p)} \quad (5.5.18)$$

(5.5.18)式说明:差错事件概率 $P(E)$ 不大于 $T(D) \Big|_{D=4p(1-p)}$ 。由此可见,由信号流图算出的生成函数 $T(D)$ 不但表明了自由距离,还可以用来计算卷积码的性能限。

当 BSC 转移概率 p 很小时(一般都是这样), (5.5.17)式的值主要由第一项($d = d_f$)决定,式子可简化为

$$P(E) \approx A_{d_f} (4p(1-p))^{d_f/2} = A_{d_f} 2^{d_f} p^{d_f/2} \quad (5.5.19)$$

从通信角度讲,最终的质量指标是误信息比特率 $P_b(E)$,为此还需寻找从差错事件概率 $P(E)$ 推导误比特率 $P_b(E)$ 的方法。一个重量为 d 的差错路径包含 d 个差错比特,对于系统卷积码而言,这些差错比特有的是信息比特,有的并不是信息比特。我们定义所有(A_d 条)重量为 d 的差错路径所对应的信息序列(有别于码字序列)的重量之和为 B_d ,由于正确信息序列的重量为 0,显然 B_d 越大误信息比特率也就越大。某一时刻差错事件的概率实质上就是译码(以码字为单位)差错的概率,对于一个(n, k)卷积码而言,一个码字含 k 比特信息。用 B_d 取代(5.5.17)式中的 A_d 并除以 k (分摊到每个信息位),就得到信息比特的差错概率

$$P_b(E) < \sum_{d=d_f} \frac{B_d}{k} (4p(1-p))^{d/2} \quad (5.5.20)$$

上式称为契尔诺夫上边界(Chernoff)。当 $p \ll 1$ 时取上式的首项,得

$$P_b(E) < \frac{B_{d_f}}{k} 2^{d_f} p^{d_f/2} \quad (5.5.21)$$

这就是 BSC 信道的误信息比特率。

在加性高斯白噪声 AWGN 信道, 信噪比与硬判决误码率(可视为 BSC 中的 p) 的关系是

$$p = \frac{1}{2} \operatorname{erfc} \sqrt{\frac{E}{N_0}} = \frac{1}{2} e^{-E/N_0} \tag{5.5.22}$$

式中, $\operatorname{erfc}(\cdot)$ 是误差补函数; E 是每码元的能量, N_0 是单边噪声功率谱密度。

当 $p \ll 1$ 时, 将(5.5.22)式代入(5.5.21)式, 得

$$P_b(E) = \frac{B_{df}}{k} 2^{d_f/2} e^{-\frac{E}{N_0} \cdot \frac{d_f}{2}} \tag{5.5.23}$$

令码率 $R = k/n$, 将每一码元的能量折合成一信息比特的能量

$$E = E_b R \tag{5.5.24}$$

代入(5.5.23)式得

$$P_b(E) = \frac{B_{df}}{k} 2^{d_f/2} \exp \left[-\frac{E_b}{N_0} \cdot \frac{R d_f}{2} \right] \tag{5.5.25}$$

与不编码的情况相比较, 不编码时一个码元就是一个比特即 $E = E_b$, 由(5.5.22)式得

$$P_b(E)_{\text{不编码}} = \frac{1}{2} e^{-E_b/N_0} \tag{5.5.26}$$

将编码(5.5.25)式与不编码(5.5.26)式时的误比特率 $P_b(E)$ 作比较, 忽略 $\exp(\cdot)$ 的系数而注意起主导作用的指数项, 定义两者指数之比(分贝)为渐近编码增益

$$= 10 \lg(R \cdot d_f/2) \text{ dB} \tag{5.5.27}$$

渐近编码增益 的物理意义是指 E_b/N_0 时, 在同样的信息速率和同样的误比特率条件下, 采用硬判决维特比译码较之不编码的信息传输所要求的信噪比 E_b/N_0 可以降低的分贝数。正因是渐近, 所以实际的编码增益总是小于 。

用类似的方法, 可以求得 DMC 信道软判决时的各项结果。连同上面的结果一起, 已列在表 5.5.2 中。

表 5.5.2 硬、软判决下的误比特率和渐近编码增益

	硬 判 决	软 判 决
BSC 信道的误比特率	$P_b(E) = \frac{B_{df}}{k} 2^{d_f/2} p^{d_f/2}$	$P_b(E) = \frac{B_{df}}{k} \prod_{j=1}^Q p(j 0) p(j 1)^{d_f}$
AWGN 信道误比特率	$P_b(E) = \frac{B_{df}}{k} 2^{d_f/2} \exp \left[-\frac{E_b}{N_0} \cdot \frac{R d_f}{2} \right]$	$P_b(E) = \frac{B_{df}}{k} \exp \left[-\frac{E_b}{N_0} \cdot R d_f \right]$
渐近编码增益	$= 10 \lg(R \cdot d_f/2) \text{ dB}$	$= 10 \lg(R \cdot d_f) \text{ dB}$

值得注意的是: 软判决与硬判决的渐近编码增益相差一个因子 $\lg 2$ 即 3 dB, 这从理论上证明了软判决优于硬判决, 能多产生 3 dB 增益。

从以上分析可知: 卷积码的自由距离 d_f 是最关键的参数, 它与码率一起决定了编码增益。卷积码设计的目标就是要取得尽可能大的自由距离, 而这个最大自由距离是存在着上

限的。当码率和约束长度给定后,设计的卷积码如果能够得到最大的 d_f ,则从距离角度看该码是最佳的。不同码率和约束长度下最佳卷积码的生成多项式和相应的 d_f 值已经用计算机搜索方法得到,可参见 Odenwalder(1970),Larsen (1973),Paaske(1974)和 Daut(1982)等文章。

海勒(Heller)推导了一种简单的、计算码率 $1/n$ 卷积码自由距离上边界的公式

$$d_f = \min_{l=1} \text{INT} \frac{2^{l-1}}{2^l - 1} (m + l - 1) n \tag{5.5.28}$$

式中 $\text{INT}[\cdot]$ 表示取整, m 是约束长度。道特(Daut)等人于 1982 年又对海勒边界作了改进。

5.6 网格编码调制与级联码简介

5.6.1 网格编码调制

在纠错编码技术中,任何纠错能力的获取都是以资源的冗余度为基础,需要付出代价的。比如由于校验位的插入, (n, k) 卷积码或分组码或使传信率降低,或使信道带宽增加,或使信号能量变大。如果增加码长 n ,则处理设备变得复杂、昂贵。

1982 年, Ungerboeck.G. 提出了一种将编码和调制结合在一起,利用状态记忆和分集映射来增大编码序列之间距离的方法,称之为网格编码调制(TCM: Trellis Coded Modulation)。这种方法无需增大带宽和功率,而是利用信号集空间的冗余度,是一种“信号集空间码”(signal space code),非常适合用于频带、功率均受限的信道如卫星、深空、微波、电话等信道。

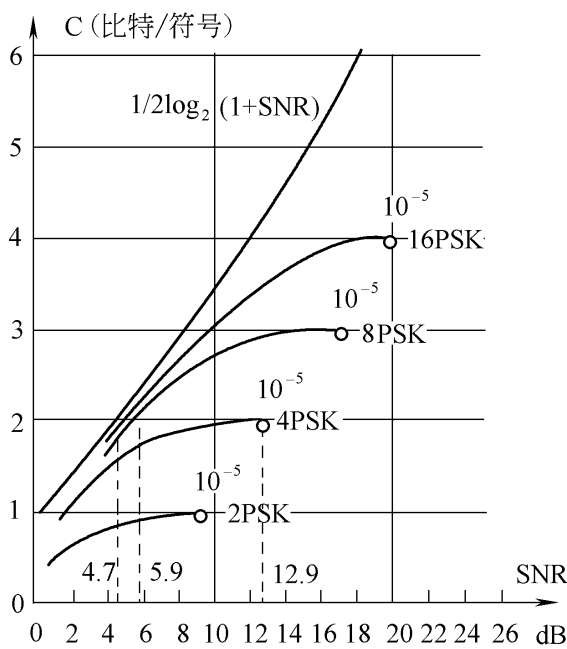


图 5.6.1 带限 AWGN 信道 PSK 调制时信道容量与 SNR 的关系曲线

信号集空间冗余度的概念可从信道容量与信噪比的关系曲线,图 5.6.1 看出。

图中左上边的一条线是根据香农信道容量公式 $C = (1/2) \log_2 (1 + \text{SNR})$ 得出的,被视为理论极限。右面的几条线是带限 AWGN 信道中采用 PSK 调制时携带的信息量与所需信噪比的关系曲线。从图中看到,用 4PSK 调制、以 10^{-5} 误比特率、每符号传递 2 比特信息时所要求的信噪比是 12.9 dB,如果不用 4PSK 而是用 8PSK 调制,则每符号传递 2 比特信息仅要求信噪比 5.9 dB。以往我们总是理所当然地认为 8PSK 就是每符号携带 3 比特信息,但如果换一种思路,用 8PSK 传 2 比特信息,则可减小所需信噪比达 7 dB 之多。进一步,如果采用 16PSK, 32PSK, ... 传 2 比特信息,信噪比还可减小,但总不能超过香农公式规定的 4.7 dB 的极限。这就是

摘自 Ungerboeck, IEEE T IT, 1982, p57

说,用冗余信号集传 2 比特信息至多可产生 8.2 dB 编码增益,而用 8PSK 代替 4PSK 已经可以取得其中的绝大部分(7 dB),再增大信号集将使设备变得更复杂,代价大而收益小。因此,TCM 的码率 R 一般写成 $m/(m+1)$,表示每符号用 2^{m+1} 点信号集传送 m 比特信息。

网格编码调制一般由三个部分组成：差分编码,与后面的合理映射相结合,解决接收端解调时信号集的相位混淆问题。卷积编码器,将 m 比特编码到 $m+1$ 比特。分集映射,将 $m+1$ 比特的 2^{m+1} 种组合——对应到 2^{m+1} 点信号集(也称为星座 constellation)。为了直观地了解 TCM 的原理,下面以 4 状态 8PSK 网格编码调制器(图 5 6 2)为例加以说明。

差分编码后的两位信息是 X_n^1 和 X_n^2 ,其中 X_n^2 不参与编码直接送到映射器即 $X_n^2 = Y_n^2$,另一位 X_n^1 进入 $R = 1/2$ 的卷积编码器,编出两位系统卷积码 $Y_n^1 Y_n^0$ 。 $Y_n^2 Y_n^1 Y_n^0$ 含 2 比特信息,却有 8 种可能的组合与 8PSK 对应,所以星座点不能简单地与 2 比特信息——对应,而必须加上卷积码状态转移信息作为参考。根据编码器的构造,可以推导出编码器的状态转移和输出码字的规律如表 5 6 1 所示,进而画出网格图如图 5 6 3。

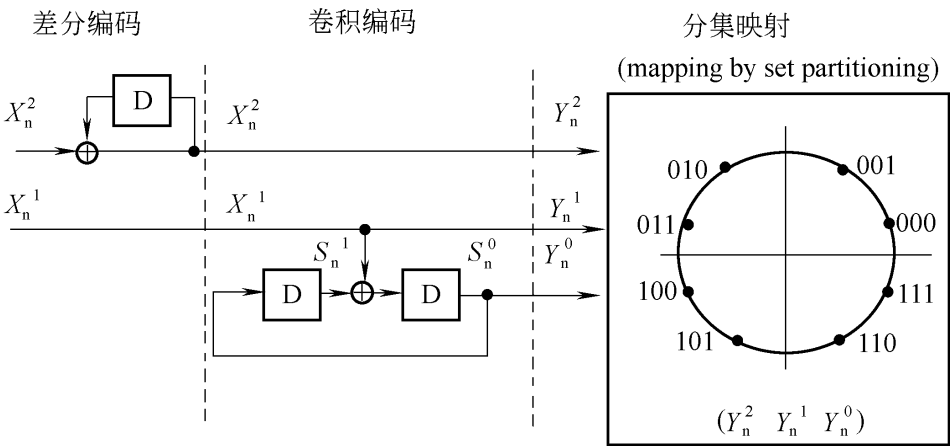


图 5 6 2 4 状态 8PSK 网络编码调制器

表 5 6 1a 状态转移表

$S_{n+1}^1 S_{n+1}^0 \backslash S_n^1 S_n^0$	1	0
0 0	0 1	0 0
0 1	1 1	1 0
1 0	0 0	0 1
1 1	1 0	1 1

$$S_{n+1}^1 = S_n^0$$

$$S_{n+1}^0 = S_n^1 \oplus X_n^1$$

表 5 6 1b 输出码字表

$Y_n^2 \backslash Y_n^1 Y_n^0$	$x_n^2 1$	$x_n^2 0$
0 0	$X_n^2 10$	$x_n^2 00$
0 1	$x_n^2 11$	$x_n^2 01$
1 0	$x_n^2 10$	$x_n^2 00$
1 1	$x_n^2 11$	$x_n^2 01$

$$Y_n^2 = X_n^2$$

$$Y_n^1 = X_n^1$$

$$Y_n^0 = S_n^0$$

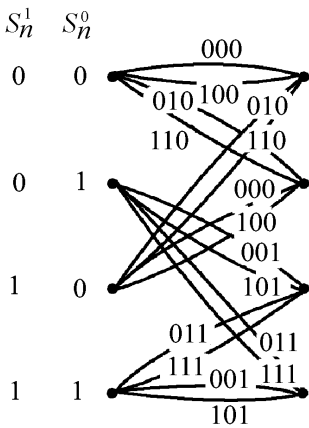


图 5 6 3 网络图

从网格图上看,一个状态转移到另一个状态不止一根线而是两条线,称为并行转移。并行转移的起因在于有的输入信息没有参与编码,从图 5 6 2 看,由移存器 S_n^0 、 S_n^1 决定的编码器状态仅仅与 X_n^1 有关,卷积编码输出也只占整个码字 $Y_n^2 Y_n^1 Y_n^0$ 中的两位 $Y_n^1 Y_n^0$,而 Y_n^2 究竟是 1 还是 0 与状态转移无关,所以状态之间都有 $1 Y_n^1 Y_n^0$ 和 $0 Y_n^1 Y_n^0$ 两条并

行转移。换个说法,由于每次输入两位信息 $X_n^2 X_n^1$, 共有 $2^2 = 4$ 种组合, 其中只一位影响到状态转移, 只有 $2^1 = 2$ 种组合, 所以一种状态转移必然对应 2 种 $X_n^2 X_n^1$ 组合即 2 码字。

并行转移影响了卷积码的自由距离。图 5 6 3 网格图上从零状态分叉又回到零状态、与全 0 路径距离最小的路径(自由距离的定义)不可能大于并行转移的距离, 因为并行转移(码字 100)本身就是“ 从零状态分叉又回到零状态 ”的一条路径。正因为如此, 并行转移对应的一组码字应距离越大越好, 对于两维调制, 就是欧氏距离越大越好。为此, 我们将 8PSK 星座对半又对半地划分成子集(set partitioning), 使每级子集具有逐级增大的距离, 然后把并行转移的一组码字映射到点数相符的同一子集上, 以保证并行转移具有最大的距离。这个过程叫作分集映射(mapping by set partitioning), 见图 5 6 4。

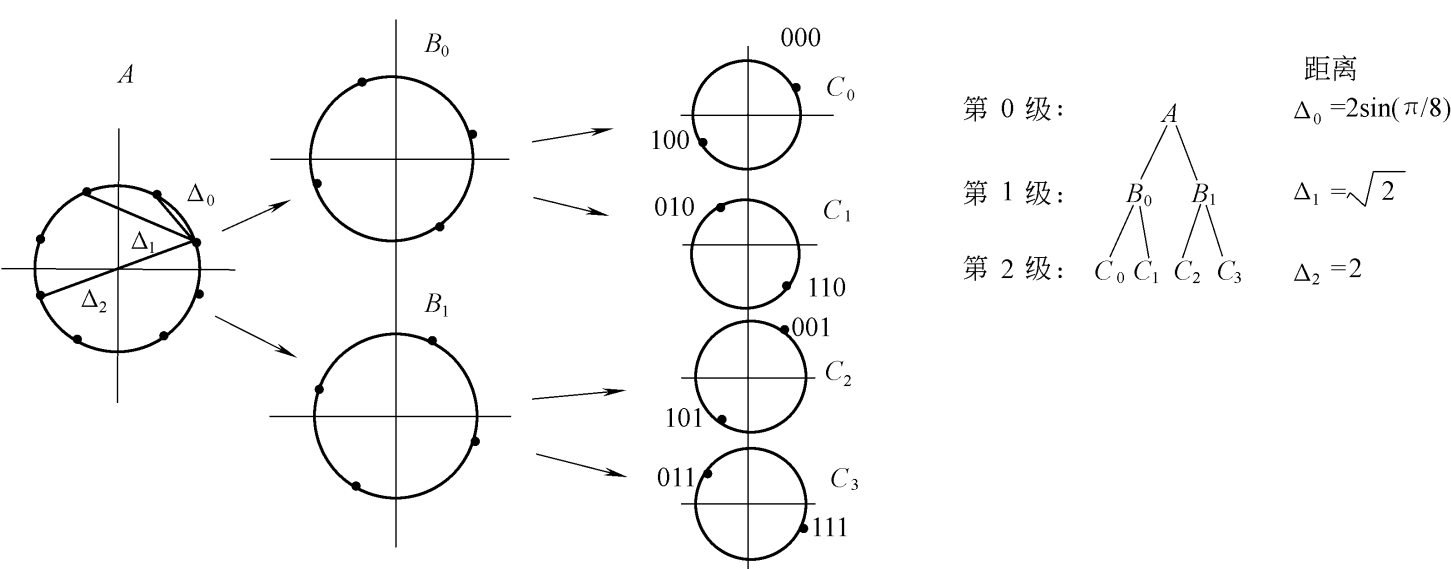


图 5 6 4 8 PSK 星座的子集分割

分集的结果产生了 4 个子集 $C_0 \sim C_3$, 每子集与一组并行转移对应, 对应的原则是:
从某一状态发出的子集源于同一个上级子集, 比如 C_0 、 C_1 就是源于同一上级子集 B_0 。

到达某一状态的子集源于同一个上级子集。

各子集在编码矩阵中出现的次数相等, 并呈现出一定的对称性。

另外, 由于接收端载波恢复时会造成不同程度的相位不定度, 比如对于 8PSK, 一般的载波提取可产生 45° 、 90° 、 135° 、 $180^\circ \dots$ 等相位不定度, 如采用判决反馈情况会好些, 但还存在 180° 的相位混淆。为此, 在将码字对应到星座点时还应遵照如下原则:

采用差分编码。如存在 180° 相位混淆需一位差分编码; 如存在 90° 、 180° 、 270° 相位混淆则需两位差分编码。

未差分编码的码元, 应选择使得不受相位混淆的影响, 即相位混淆时其值不变。

按上述准则, 得各子集信号点与码字的对应分配关系如图 5 6 4, 其编码矩阵如 (5 6 1) 式。

$$\begin{array}{ccccccc}
000 & 110 & & & & & \\
& & & & & & \\
100 & 010 & & & & & \\
& & & & & & \\
& & 101 & 111 & C_0 & C_1 & \cdot & \cdot \\
& & & & & & & \\
& & 001 & 011 & \cdot & \cdot & C_2 & C_3 \\
C = & 110 & 000 & & = & C_1 & C_0 & \cdot & \cdot \\
& & & & & & & & \\
& & & & & & \cdot & \cdot & C_3 & C_2 \\
010 & 100 & & & & & & \\
& & & & & & & \\
& & 111 & 101 & & & & \\
& & & & & & & \\
& & 011 & 001 & & & &
\end{array} \quad (5.6.1)$$

从编码矩阵看, 每一行、每一列的子集都具有相同的上级子集, $C_0 \sim C_3$ 同等地各出现了 2 次, 分布很规则, 符合上面所述原则。从图上看, 凡是相差 180° 的两星座点, 比如 C_0 的 000, 100, 它的后两位 $Y_n^1 Y_n^0$ 总是相同的, 180° 相位差对它们无影响; 而第一位 Y_n^2 采用了差分编码, 可以抗 180° 相位混淆。

网格编码调制将编码和调制统一加以考虑, 好处是编码序列的自由距离得以提高。下面以 4 状态 8PSK 为例定量分析如下:

用子集代替并行转移, 网格图 5.6.3 可简化为图 5.6.5 左边的图案。分析与全零路径分叉又回到全零路径的所有路径, 距离最近的、真正离开零状态的一条路径如图 5.6.5 所示。另外, 还有一条这样的路径就是与 000 的并行转移 100。为了加以区分, 把前者称为序列距离, 记作 d_{seq} , 而把并行转移距离称为并联距离, 记作 d_{par} 。显然, 自由距离应该是其中最小者

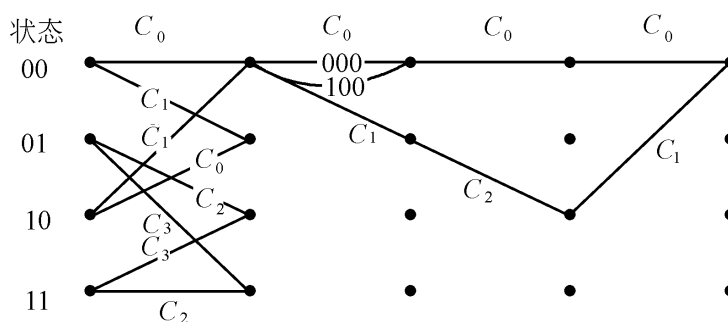


图 5.6.5 8PSK TCM 码的自由距离

$$d_f = \min(d_{\text{seq}}, d_{\text{par}}) \quad d_{\text{par}} \quad (5.6.2)$$

具体到本例, 由于

$$\begin{aligned}
d_{\text{seq}}^2 &= \text{dis}^2(C_0, C_1) + \text{dis}^2(C_0, C_2) + \text{dis}^2(C_0, C_1) = 1^2 + 0^2 + 1^2 \\
&= (2)^2 + (2\sin(\pi/8))^2 + (2)^2 = 4.586 \\
d_{\text{par}}^2 &= 2^2 = 2^2 = 4
\end{aligned}$$

所以

$$d_f^2 = \min(d_{\text{seq}}^2, d_{\text{par}}^2) = d_{\text{par}}^2 = 4$$

在功率一定的两维调制下, 总是欧氏距离越大差错概率越小。这里所谓距离, 不编码情况下是指信号点集之间的最小距离, 编码情况下是指自由距离。为了定量说明编码前后的变化, 定义编码增益为

$$= 10 \log \frac{d_f^2 / E_c}{d_{un}^2 / E_{un}} \quad (5.6.3)$$

式中, d_{un}^2 是不编码时信号点集的最小距离; E_c, E_{un} 分别是编码、不编码条件下信号集的平均能量。

本例中, 不编码时无需信号点集冗余度, 只要 4PSK 即可传送 2 比特/符号信息, 4PSK 的最小距离是 $d_{un}^2 = 1^2 = 1$, 而 4PSK 和 8PSK 的平均能量相同, 于是得编码增益

$$= 10 \log (d_f^2 / d_{un}^2) = 10 \log (4 / 1) = 6 \text{ dB}$$

由此可知, 通过简单的 4 状态 TCM 编码即已获得了 3dB 的编码增益。这里有一点需要说明: 并不是任何 4 状态编码都能得到 3 dB 增益, 结构不好的码达不到 3 dB。事实上图 5.6.2 的网格编码调制器是能找到的 4 状态的最优码, 任何其他码都不可能给出更大的自由距离。

可以想像, 如果进一步增加编码器的复杂度, 使 TCM 具有 8 状态、16 状态、32 状态……, 一定可以得到更大的编码增益。实际情况确是如此, 通过计算机模拟发现, 码率 $m' / (m + 1)$ 的 TCM 码, 8 状态时最大可得 3.97dB 编码增益(理论值), 而 16, 32, 64, 128 状态时的最大编码增益分别是 4.39 dB, 5.11 dB, 5.44 dB 和 6.02 dB。

以上分析所揭示的一些基本规律可以推广到一般, 即设计 TCM 码的关键在于编码、分集、映射三者的完美结合。

分集总是按距离逐级增大的原则进行, PSK 是如此, QAM 也是如此, 图 5.6.6 给出的就是 16QAM 星座分集的情况。

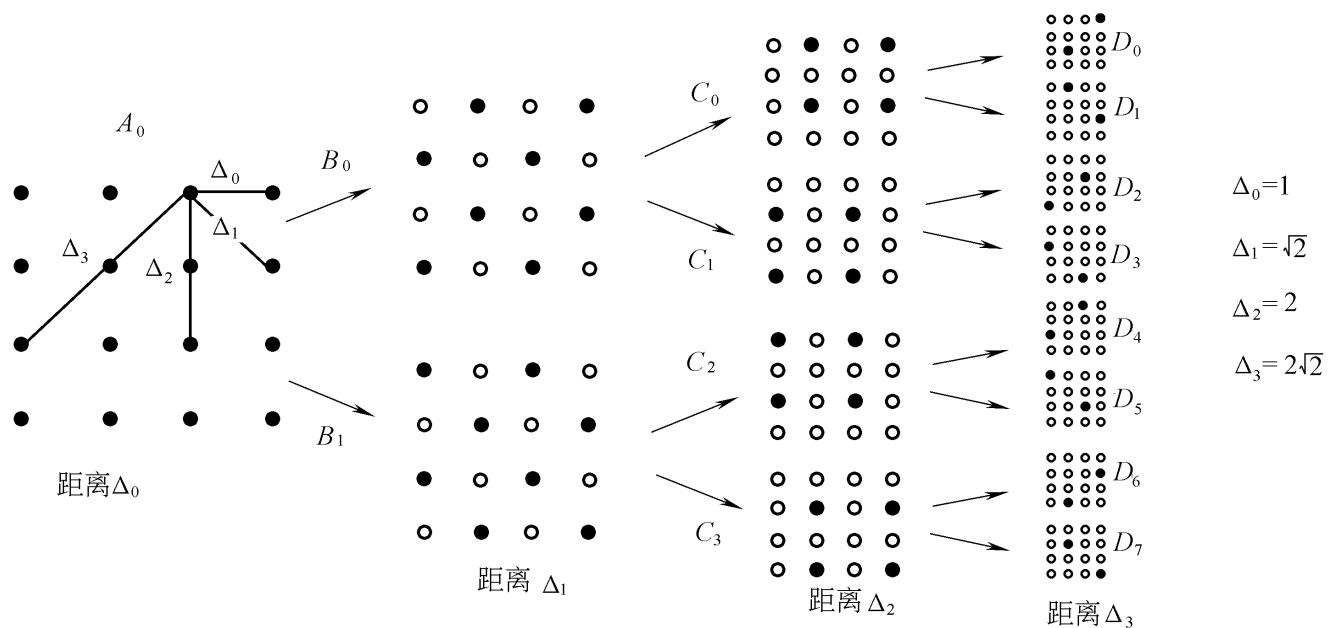


图 5.6.6 16QAM 星座的分集

以上 PSK 和 QAM 分集的过程都是进行到每子集仅包含二个信号点为止, 但一般并不一定需要这样。例如, 16QAM 的信号星座也可以只进行两级分集, 产生 4 个子集, 每子集 4 个信号点。

信号究竟分集到什么程度取决于编码器结构, $m' / (m + 1)$ TCM 编码器的一般结构如图 5.6.7 所示。一个 m 比特信息块的一部分, 设 k 比特参与 $k' / k + 1$ 卷积编码, 输出 $k + 1$ 比特, 而其余比特不参与编码。这样, 不参与编码的 $m - k$ 比特导致 $2^{m - k}$ 条并行转移, 即每子集应包含 $2^{m - k}$ 点, 而子集总数应有 $2^{m + 1} / 2^{m - k} = 2^{k + 1}$ 个。于是用编码后的 $k + 1$ 比特

选取 2^{k+1} 个子集之一,用不参与编码的 $m - k$ 比特在各子集的 2^{m-k} 个点里选取其中之一。
 当 $k = m$ 即所有信息比特都参与编码时就不必再分集。

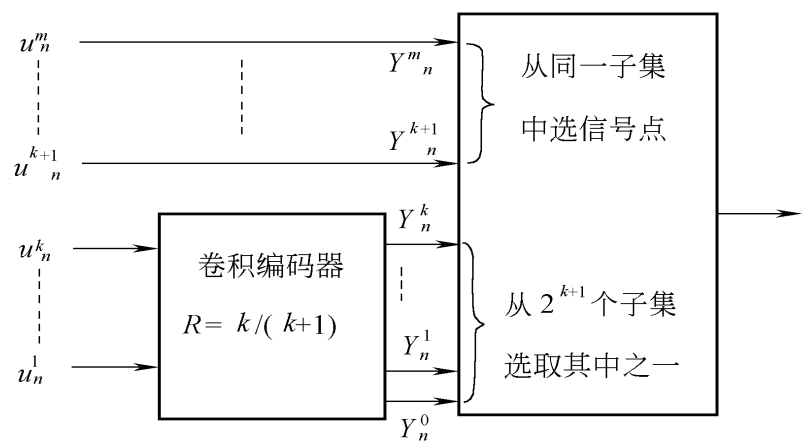


图 5 6 7 $m/(m+1)$ TCM 编码器的一般结构

由于 $d_f \neq d_{par}$, 并行转移的存在限制了自由距离,但网格图上存在并行转移并不一定是坏事,这是因为并行转移破坏了网格图的“连接性”,使序列距离 d_{seq} 变大。举例来说,如果上面 4 状态 8PSK TCM 中两位信息比特都参与编码,那么状态转移就由 2 比特(4 种组合)而不是 1 比特(2 种组合)决定,从某一状态出发的下一状态就可能是 4 个状态中的任何一个。换言之,从零状态分叉出去的任何路径在下一步都可回到零状态,序列距离就不可能大,如图 5 6 8 所示。事实上,目前发现的 TCM 好码无一例外都存在并行转移。究竟多少并行转移为好取决于并行距离 d_{par} 与序列距离 d_{seq} 的相对大小,在图 5 6 9 中我们看到:随着编码比特数的增加(即子集数增加而每子集点数减少),序列距离下降而并行距离上升,两者交点就是自由距离的最大值,此时应有 $d_{par} = d_{seq}$ 。因此,TCM 码设计时通过不断核算 d_{par} 和 d_{seq} 的相对大小来调整编码与不编码的比特数目,直到取得最大的自由距离。

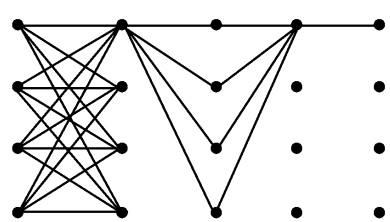


图 5 6 8 连接性越好自由距离越小

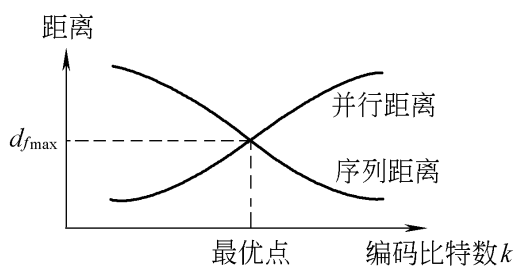


图 5 6 9 d_{par} 、 d_{seq} 和 d_f 三者关系

以上 TCM 的例子中编码都是采用卷积码,但这并非说 TCM 一定要用卷积码。事实上无论是分组码还是卷积码都可以与信号星座的分集相结合,一般来说,同样码率下分组码能得到的编码增益卷积码都能得到,并因适于使用维特比算法作软判决译码而使译码比较简单。正因如此,实用中以卷积码居多。

在存在加性高斯噪声的情况下,前节用于计算卷积码差错概率的方法也同样适用于网格编码调制。回想一下计算卷积码差错概率的方法,即先计算所有各种可能的差错事件的概率,然后把这些差错事件概率相加,忽略次要项而计算主项(距离为 d_f 的差错事件概率)的联合边界。因此在高 SNR 时,主项差错概率可用下式近似:

$$P_e \approx \frac{1}{2} N_{free} \operatorname{erfc} \frac{d_f}{2 \sqrt{E_b/N_0}} \tag{5 6 4}$$

式中, N_{free} 是与正确路径距离为 d_f 的差错路径的条数, $\text{erfc}(\cdot)$ 是误差补函数; σ^2 是噪声均方差。

从(5 6 4)式看到, TCM 码的差错概率在信噪比一定时取决于 N_{free} 和 d_f , 而计算 N_{free} 和 d_f 的方法与卷积码时的计算方法相同, 只是将距离的概念从汉明距离换成星座图上的欧氏距离即可。

5.6.2 级联码简介

1. 主导的思路：增加码长 n

Shannon 1948 年提出的信道编码定理虽然仅是一个存在性定理, 但却指出了纠错码的研究方向。该定理包含了两方面的含义, 一是每个信道都有一定的信道容量 C , 只要传信率 $R < C$, 则当 $n \rightarrow \infty$ 时, 都存在着使差错概率 $P_e \rightarrow 0$ 的好码(又称渐近好码或 shannon 码), 由此也就给出了信道编码增益的理论上限, 或传输每一信息比特所需信噪比的下限; 二是为了达到理论限, 必须使用最大似然译码。最近几十年来, 纠错编码理论和实践的发展正是沿着这二条主线, 即构造码长 $n \rightarrow \infty$ 的渐近好码以及在可接受的译码复杂度范围内实现最大似然译码这样两个方向展开的。

2. 无限长码的译码是不可实现的, 可行的办法是用短码组合成为长码

从理论上讲, 几乎所有的码都可以是渐近好码, 比如可以把 $(7, 4)$ 汉明码变为 $(14, 8)$, $(21, 12), \dots, (7n, 4n)$, $n \rightarrow \infty$ 码。但到目前为止, 构造出真正意义上的 shannon 码还有相当长的距离。纠错码的理论与实践包含两方面课题: 编码与译码。构码理论的难度主要体现在编码上, 要从理论上找到好码特别是渐近好码是一个异常困难的问题, 目前还没有完全解决。但从工程上讲, 一旦规则或方法确定后, 编码实现起来却相当容易, 其复杂度仅在 k 或 $n - k$ 数量级, 写作 $O(k)$ 或 $O(n - k)$, 这里 k 是信息位数。与此相反, 在高斯白噪声信道条件下实现差错概率最小的最佳译码的方法在理论上早已解决, 这就是最大后验概率(MAP, APP)译码, 在码字等概发送下也就是最大似然译码(MLD)或最小汉明距离译码(MHDD)或相关译码。但是, 从工程实现上要做到这些最佳译码却相当复杂, 其译码复杂性是 $O(2^k)$ 或 $O(2^{n-k})$, 因此对长码实现最佳译码几乎不可能。到目前为止, 能真正达到最佳译码的只有维特比一种算法, 它是一种 MLD, 但仅适合约束度较小的卷积码和低纠错能力的短分组码。

构造长码的一个比较自然而又有效的方法, 是 1966 年由 Forney 提出的、利用两个短码构造一个长码的思想, 其结构如图 5 6 10 所示。这样的码叫串行级联码, 简称级联码, 码长为 Nn , 信息位为 Kk , 码率为 $R_c = R_i R_o$, 这里 R_i 和 R_o 分别是内码和外码的码率。

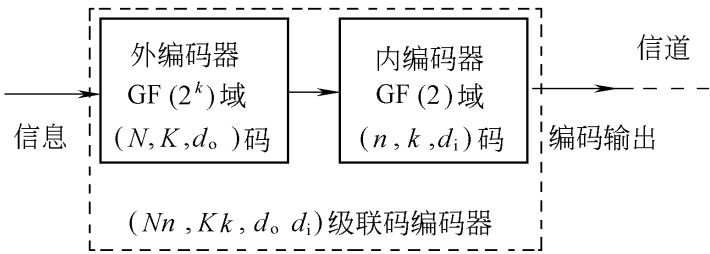


图 5 6 10 串行级联码

内码一般是软判决信号入、硬判决译码出, 用于纠错或检错。外码则是硬判决入、硬判

决出,用于二次纠错。由于级联码具有很强的纠错能力,且译码也不很复杂,特别是它展现了构造 shannon 码的美好前景,故以后不少学者对这种编码方法进行了推广和扩展,如用多级级联以及并行级联等等。由于软判决维特比最大似然译码算法适合于约束度较小的卷积码,因此级联码的内码常用卷积码。外码可以是 BCH 码或 RS 码,但由于维特比译码是序列译码,内码一旦出现译码差错就是一个序列的差错,相当于一个长度为约束长度左右的突发差错,因此具有良好纠突发差错能力的 RS 码成为首选的外码。如果卷积内码是 (n, k, L) , L 为约束长度,RS 外码是 $GF(q)$ 域上的 (N, K, d) 码,其中 $q = 2^J$,则根据 RS 码的特点,必有 $N = 2^J - 1$, $K = 2^J - 1 - 2t$, $d = 2t + 1$ 。由于卷积码最可能的差错序列长度是 $L + 1$,而 RS 二进衍生码纠突发差错的能力是 $(t - 1)J + 1$,因此原则上应有 $(t - 1)J + 1 \geq L + 1$,使卷积码译码差错的大部分能被 RS 码纠正。比如,当外码采用 $(255, 233)$ RS 码,内码采用 $(2, 1, 7)$ 卷积码且维特比译码时,与不编码相比可产生约 7dB 的编码增益,特别适用于高斯白噪声信道如卫星通信和宇航通信中。

3. 软输出译码:使外码也能实现软判决译码

内码译码的硬输出不但使外码失去了软判决译码的机会,也丧失了一定数量的信息量,于是人们提出了软输出译码。软输出译码器的输出不仅应包含硬判决值,而且要包括作出这种判断的可靠程度。若接收码字为 r ,定义对数似然比(LLR)为

$$L(u/r) = \log \frac{P_r\{u = 1/r\}}{P_r\{u = 0/r\}}$$

$L(u/r)$ 的正负符号正是硬判决值

$$u = \begin{cases} 1 & \text{当 } L(u/r) \geq 0 \text{ 时} \\ 0 & \text{当 } L(u/r) < 0 \text{ 时} \end{cases}$$

而 $L(u/r)$ 的绝对值代表硬判决的可靠度,绝对值越大,表明判决的结果越可信。此时,如内码输出用 $L(u/r)$ 代替单一的硬判决值送入外码,则除了硬判决信息(符号)外, $L(u/r)$ 的绝对值还为外码提供了判决可信程度的信息,这是一种额外的参考信息,叫外信息(extrinsic information),或称边信息、软信息。

人们提出的各种软输出算法中,以 Bahl 的算法最有代表性,应用也最广。这是一种对具有有限状态马尔可夫特性的码及离散无记忆特性的信道提供逐符号或逐比特似然值的最优算法。与其他的最大似然算法不同,它是一种递推算法,每符号运算量不随总码长变化。软输出译码算法的提出使得级联码不仅在内码、而且在外码也可采用译码简单的卷积码,即先用 Bahl 算法得到内码的软译码输出,解交织后再进行外码的软判决维特比译码,从而进一步改善了性能。不过此时的思路仍是简单的级联,其性能分析也仍然是级联式的。

4. 反馈(迭代)译码:将外码译码信息反馈到内码以便多次使用

进一步的观察使人们提出这样一个问题:既然 Bahl 算法可使内码的译码产生软输出而为下面的外码译码提供了软输入,那么能否对外码也进行软输出译码,并将其软输出反馈回内码、作为软输入进行再一轮译码而提高总体性能呢?这种反馈的概念在信号处理、自动化过程中屡见不鲜,但用于译码信息的处理还是一种新事物。困难的问题在于现有的编码结构是难以实现反馈的,原因主要有两点:

(1) 串行级联码(图 5 6 11)的编码关系为

$$\text{外码 } C_1 = f(x), \quad \text{内码 } C_2 = g(C_1) = g(f(x))$$

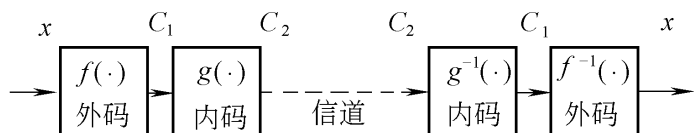


图 5 6 11 级联码编码

由于内码译码器仅处理 C_2 和 C_1 的关系而不直接处理 C_2 和 x 的关系,因此外码译码输出的关于 x 的信息并不能直接作为输入软信息提供给内码译码器使用。

(2) 简单的反馈必然存在正反馈和负反

馈两种可能,负反馈使差错越来越小,正反馈使差错越来越大。控制反馈性质使算法收敛到正确解处实在不是一件容易的事。

为了解决第一个问题,总是希望信息符号 x 能直接反映到内码 C_2 上去,这就要求两层码均为系统码。对于第二个问题,则要求在第二次内码译码时用到的反馈软信息中不包含上次译码相同的码符号时用过的信息。从严格意义上看,这种要求是不可能实现的。但我们注意到,如果两次译码之间加上交织处理,则本次译码用到的反馈信息(一个连续的码符号序列)来源于上一次译码时分散、扰乱了的码符号。交织长度越长,交织方法产生的随机化越好,相邻两次译码反馈符号的相关性就越低。此时只要从反馈符号似然信息中去除已用过的关于该符号本身的部分,就可以基本清除正反馈,实现迭代译码。

基于以上思路,Berrou 等 1993 年提出了并行级联的 Turbo 码,之后又研究了串行级联码的可能性。模拟结果表明:当 $E_b/N_0 = 7\text{dB}$ 时, $\text{BER} = 10^{-5}$ 。这个结果与 Shannon 限 ($E_b/N_0 = 0\text{dB}$) 仅差 0.7dB (迭代次数 18 次,交织器大小为 $256 \times 256 = 65536$),这一优异性能立即在编码界引起轰动。自那时起,编码领域掀起了研究 Turbo 码的热潮。

5. Turbo 码编码器

Turbo 编码器的构造如图 5 6 12 所示。由图可见:输入信息分成三支分别处理。第一支经时延后直接送入复合器,时延的目的是为了在时间上使未经处理的信息与经交织、编码处理的信息匹配。第二支经时延、编码产生校验位,收缩后送入复合器,编码方式可以是卷积码,也可以是分组码。第三支与第二支的处理相同,只是将延时改为交织。编码器 2 的编码方式可以与编码器 1 相同,也可以不同。交织的目的是为了改变码重分布,如交织前 d_k 对应一个轻码,那么期望交织后的 d_n 能对应一个重码。最后,信息 X_k 和校验位 Y_{1k}, Y_{2k} 复合后合为一个信息流发出。

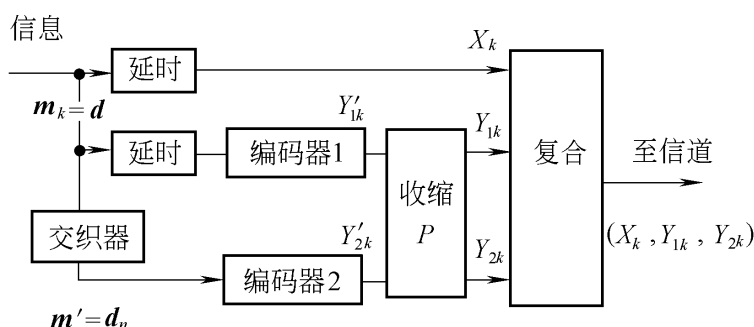


图 5 6 12 Turbo 编码器

6. 收缩(Puncture)

收缩是通过删余(删除校验位)来调整码率。比如采用 $R = 1/2$ 的系统卷积码编码时,将第一支的信息位与第二、第三支的校验位复合,将产生 $R = 1/3$ 的码流。但如果令第二支的校验流乘一个收缩矩阵 $P_1 = [1\ 0]^T$ 而让第三支的校验流乘以收缩矩阵 $P_2 = [0\ 1]^T$,那就产生了在第二、第三支轮流取值(相当于每支一个隔一个取值)的效果,使码率调整为

$R = 1/2$ 。一般地,若第二、第三支编码器输出(对于系统码可以是去除信息位后的校验位)为 $(2 \times N)$ 矩阵 $[mG \ mG]^T$ (mG, mG 分别是 $1 \times N$ 矢量),则收缩矩阵 P 为 $(N \times 2)$ 矩阵 $(P_1 \ P_2)$,其中 P_1, P_2 均为 $N \times 1$ 列矢量,由 0,1 值组成,分别体现对第二、三支校验位的选择。

借助缩短码,可用较简单的编、译码器(比如 $1/2$ 卷积码)实现较高码率(比如 $R = 6/7$)的编、译码。比如, $1/2$ 卷积码的格栅图上从一个状态出发或到达一状态仅有两种可能(两条线),维特比算法只需作两条路径的比较。而 $6/7$ 卷积码从一个状态出发或到达一状态最多可有 $2^6 = 64$ 条线,维特比算法最多时需作 64 条路径的比较。一般来说, $R = k/n$ 编码器的每一状态要进行 2^k 次比较。可以想象,如果我们用 $1/2$ 编码器产生 $6/12$ 码,然后将它缩短到 $6/7$ 码,当然要比用 $6/7$ 编码器直接编译码要容易些。这就是缩短码广泛使用的原因。

7. 递归的系统卷积码 (RSC:Recursive Systematic Convolutional)

实用卷积码绝大多数是非递归(前馈)的非系统卷积码(NSC),这是因为 Forney 等已证明过:对于前馈型卷积码而言,非系统码比系统码有更大的自由距离。但是对递归型卷积码而言,系统码与非系统码相比毫不逊色。Turbo 码既然要求采用系统码,理所当然应选用递归型系统卷积码(RSC 码)。一些文献证明了:在收缩码形式以及小信噪比情况下,RSC 比 NSC 具有更好的重量谱分布和更佳的误码率特性,并且在码率越高、信噪比越低时其优势越明显。

RSC 码可以由 NSC 码转化而来。比如某 $R = 1/2$ NSC 码的生成函数矩阵是

$$G(D) = (1 + D^2, 1 + D + D^2)$$

可将信息序列 $U(D)$ 乘以 $1/(1 + D^2)$ 进行预编码,再让它通过原编码器,于是就成为 RSC 编码器了,如图 5 6 13 所示。

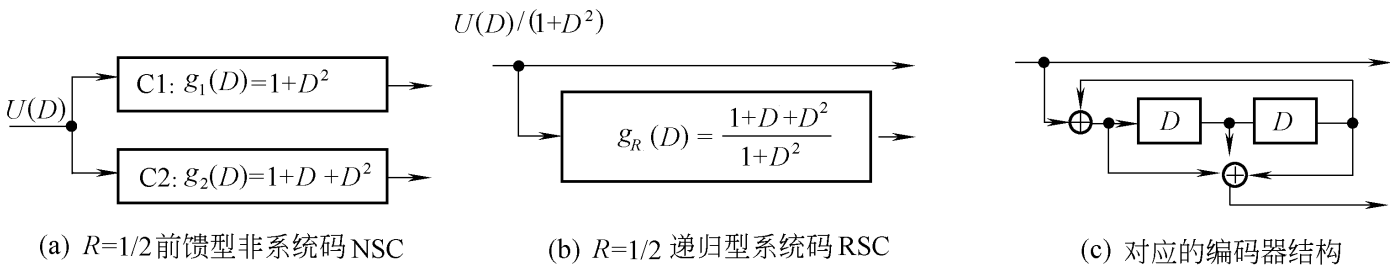


图 5 6 13 将前馈型非系统码转化成递归型系统码

8. Turbo 码译码器

Turbo 码译码器采用反馈结构,以迭代方式译码。与图 5 6 12 的两个编码器相对应,译码器也有两个,分别写作 DEC₁ 和 DEC₂,它们的关系可以是并行级联(parallel concatenation code),也可以是串行级联(series concatenation code),两种情况下的译码结构如图 5 6 14 所示。

图(a)是串行级联译码。在译码之前,首先要进行数据的串/并转换。另外,由于编码器的收缩功能,部分校验位没有传送过来,为此,串/并转换之后必须对接收序列进行内插,在被删除的数据位上补以中间量(如 0),以保证序列的完整性。

DEC₁ 的输入有三部分:信息码 x_k 、校验码 y_{1k} 以及 DEC₂ 输出反馈的外信息 z_k 。DEC₁ 根据译码算法(MAP,SOVA 等)完成对编码器 1 的译码,输出译码软信息 $L_1(d_k)$ 。由于是系统码, $L_1(d_k)$ 直接表示信息序列的判决值。又由于编码器 1 未经交织而编码器 2 的输入

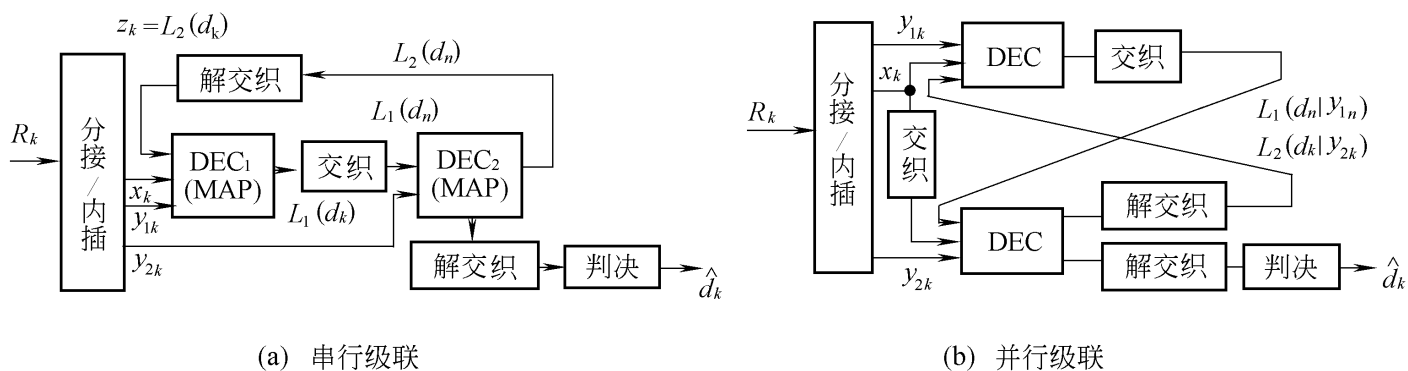


图 5 6 14 Turbo 码递归型译码器

是 $\{d_k\}$ 的交织序列 $\{d_n\}$, 因此 $L_1(d_k)$ 在进入 DEC_2 之前须交织成序列 $L_1(d_n)$ 以便与编码器 2 匹配。 DEC_2 利用 $L_1(d_n)$ 和 y_{2k} 完成对编码器 2 的译码, 得到软输出 $L_2(d_n)$ 。对 $L_2(d_n)$ 解交织即得 $L_2(d_k)$, 该值可反馈到 DEC_1 , 也可判决后作为译码输出估值 \hat{d}_k 。 DEC_1 提供给 DEC_2 的译码信息 $L_1(d_k)$ 与另一支输入 y_{2k} 虽然代表同一种信息, 但它们是相互独立传送的, $L_1(d_k)$ 对 y_{2k} 而言是一种附加信息, 使输入到 DEC_2 的信息量增加, 不确定度(信息熵)减小, 从而提高了译码正确性。既然 y_{2k} 独立于 y_{1k} , 在 DEC_2 译码时用过的信息 y_{2k} 并不曾被 DEC_1 使用过。因此, 若将 DEC_2 的译码信息反馈到 DEC_1 , 必然有助于提高 DEC_1 的译码性能。如此反复反馈, 整个译码器的性能可逐步提高。当然, 这种提高是有限度的, 随着反复次数增加, DEC_2 与 DEC_1 译码信息中相互独立的成分(即附加信息)越来越少, 最终降至零。此时, 信息量已被用尽, 迭代不再影响性能, 译码结束。上述迭代译码过程类似于涡轮机中蒸汽的反复循环使用, Turbo 码名字的原意即源于此。

迭代译码的引入, 使并行结构的译码器成为可能, 如图 5 6 14(b) 所示。此时两个译码器的功能是一样的, 都是利用另一译码器提供的附加信息进行译码, 然后将自己译码信息作为附加信息提供给对方。整个译码过程可以看作是两个子译码器信息互相交换的过程。当两个子译码器中的信息得到充分交换, 并达到平衡状态时, 译码达到最佳效果。

Turbo 码两个子译码器的译码、交织、解交织等运算必然造成时延, 使外信息 z_k 不可能立即反馈到 DEC_1 。两次迭代的时差表现为差分变量, 使得不可能有真正意义上的反馈, 而是流水线式的迭代结构。译码器视作由若干完全相同的软入软出的基本单元构成, 每一节的结构完全相同。采用这种形式将使译码器的结构非常简单, 且利于集成。

9. Turbo 码的性能分析

目前对 Turbo code 的性能研究还远远不够, 迭代译码算法的误码性能只能通过仿真得到, 而码本身的性能受交织器特性及码率调整器特性的影响。从模拟结果看, Turbo 码的性能与 shannon 限已相差甚微, 但是理论分析尚不完善。

1995 年, R. Podemski 等给出了计算 Hamming 距离谱(HDS)的算法——修正的 Fanno 算法, 并利用最小 Hamming 距离(MHD)与 HDS 对 Turbo 码的性能进行了分析, 分析结果与模拟结果相当接近。

Perez 等从距离谱的观点分析了 Turbo 码在低信噪比时的优异性能, 在 Turbo 码的编码器中, 交织器起着“谱窄化”的作用, 使得 Turbo 码中重量小的码字数目减少, 而这正是影响 Turbo 码性能的主要因素之一, Perez 等还通过距离谱解释了 Turbo 码性能曲线尾部平坦的现象。

10. Turbo 码优异性能的物理解释

Turbo 码的工作机理到目前为止还远没有被彻底弄清,但粗略的物理解释还是很明显的。大家知道,一种编码的误码性能取决于其码距, A, B 两个码字距离越远,把 B 错译成 A 的概率越小。Turbo 码采用并行结构的级联系统码,两个码分别对交织前后的信息序列进行编码,得到相应的校验序列。显然,影响误码性能主要是低重量的信息序列编码后的校验重量,对于不同的低重量信息序列经过一次分量编码(卷积码)后的校验重量是不同的,而我们知道单靠卷积码的码重是不足以提供接近极限的译码性能的,但若大部分具有低校验重量的信息序列经交织后再次编码可获得较高的校验重量,则从总体来看,大部分的码字都有较大的码重,从而提高误码性能。也就是说尽管从某个分量码看,信息序列 A 和 B 的编码距离较近,但只要它们在另一个分量码中有较大的距离,我们还是能很容易地区分它们。而软输出迭代译码算法正好符合这种情况,即当处理距离较近的分量码时,软输出算法对 A 和 B 求同存异,对 A 和 B 中不同的位给出一个模糊输出,留待另一个分量码译码算法处理。

从上述物理解释可直接得到一个重要结论:用递归码做分量码要优于非递归码,在非递归码的低重信息序列编码中,单错事件(即错误路径从离开到返回正确路径只有一个信息位错)的概率较大而第一层码中的单错事件经交织后也会在第二层码中以很大的概率产生单错事件。而递归码不会发生单错事件,其双错事件的两个错码经交织后会离得很远,从而产生很大的校验位错误,因而从总的码重分布来看更集中于平均码重附近。

11. Turbo 码的应用

从上面的介绍,可以看出, Turbo 码有着接近信道极限的性能。因而特别适用于对功率要求严格的情形,如卫星通信中能源极端受限,移动通信中的电池寿命指标要求较高,军事通信中要求发射信号功率尽可能低以降低被发现的概率,因此 Turbo 码在这些方面特别有吸引力。不过, Turbo 码的固有缺点:有较大的延时,在很大程度上限制了它的更广泛应用。

另外,由于 Turbo 码接近于随机码,有很好的距离特性,因而有很强的抗衰落和抗干扰能力, Turbo 码理想交织后在瑞利衰落信道中的性能,比 AWGN 下的未编码性能还有数 dB 的增益。另外,在军事通信中常见的部分带干扰环境下,只要接收机能监测到哪些频点受干扰,对信号进行带删除纠错译码,就能得到其他码难以达到的性能,因而特别适合于各种恶劣环境下的通信。

习 题

5 1 设二进制对称信道的概率转移矩阵为
$$\begin{bmatrix} 2/3 & 1/3 \\ 1/3 & 2/3 \end{bmatrix},$$

(1) 若 $p(x_0) = 3/4$, $p(x_1) = 1/4$, 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。

5 2 某信源发送端有 2 个符号, x_i , $i = 1, 2$, $p(x_1) = a$, 每秒发出一个符号。接收端有 3 种符号 y_j , $j = 1, 2, 3$, 转移概率矩阵

$$P = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 1/4 & 1/4 \end{bmatrix}。$$

(1) 计算接收端的平均不确定度;

(2) 计算由于噪声产生的不确定度 $H(Y/X)$;

(3) 计算信道容量。

5 3 在有扰离散信道上传输符号 1 和 0, 在传输过程中每 100 个符号发生一个错传的符号。已知 $p(0) = 1/2$, $p(1) = 1/2$, 信道每秒钟内允许传输 1000 个符号。求此信道的信道容量。

5 4 具有 6.5 MHz 带宽的某高斯信道, 若信道中信号功率与噪声功率谱密度之比为 45.5 MHz, 试求其信道容量。

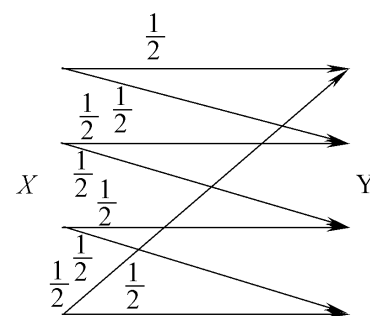
5 5 设有扰离散信道的传输情况分别如图题所示。

(1) 求出该信道的信道容量。

(2) 找一个循环码长为 2 的重复码, 其信息传输率为 $1/2 \log_2 5$ 。当输入码字为等概分布时, 如果按最大似然译码规则设计译码器, 求译码器输出端的平均错误概率。

5 6 发送端有 3 种等概符号 (x_1, x_2, x_3), $p(x_i) = 1/3$, 接收端收到 3 种符号 (y_1, y_2, y_3), 信道转移概率矩阵

$$P = \begin{bmatrix} 0.5 & 0.3 & 0.2 \\ 0.4 & 0.3 & 0.3 \\ 0.1 & 0.9 & 0 \end{bmatrix}$$



图题 5 5

(1) 求接收端收到一个符号后得到的信息量 $H(Y)$;

(2) 计算噪声熵 $H(Y/X)$;

(3) 计算当接收端收到一个符号 y_2 的错误概率;

(4) 计算从接收端看的平均错误概率;

(5) 计算从发送端看的平均错误概率;

(6) 从转移矩阵中你能看出该信道的好坏吗?

(7) 计算发送端的 $H(X)$ 和 $H(X/Y)$ 。

5 7 电视图像由 30 万个像素组成, 对于适当的对比度, 一个像素可取 10 个可辨别的亮度电平, 假设各个像素的 10 个亮度电平都以等概率出现, 实时传送电视图像每秒发送 30 帧图像。为了获得满意的图像质量, 要求信号与噪声的平均功率比值为 30 dB, 试计算在这些条件下传送电视的视频信号所需的带宽。

5 8 一个平均功率受限制的连续信道, 其通频带为 1 MHz, 信道上存在白色高斯噪声。

(1) 已知信道上的信号与噪声的平均功率比值为 10, 求该信道的信道容量;

(2) 若信道上的信号与噪声的平均功率比值降至 5, 要达到相同的信道容量, 信道通频带应多大?

(3) 若信道通频带减小为 0.5 MHz 时, 要保持相同的信道容量, 信道上的信号与噪声的平均功率比值应等于多大?

5 9 写出构成 GF(2) 上的所有 4 重的矢量空间, 并找出其中一个二维子空间及其相应的对偶子空间。

5 10 若 S_1 和 S_2 是矢量空间 V 的两个子空间, 证明 S_1 和 S_2 的交也是 V 的子空间。

5 11 某系统 (8, 4) 码, 其后 4 位校验位 v_i , $i = 0, \dots, 3$ 与信息位 u_i , $i = 0, \dots, 3$ 的关系是

$$v_0 = u_1 + u_2 + u_3$$

$$v_1 = u_0 + u_1 + u_2$$

$$v_2 = u_0 + u_1 + u_3$$

$$v_3 = u_0 + u_2 + u_3$$

求:该码的生成矩阵和校验矩阵,该码的最小距离,并画出该编码器硬件逻辑连接图。

5 12 将例 5 4 1 的(7,4)码缩短为(5,2)码,写出缩短码的生成矩阵和校验矩阵,并且列出缩短前、后的所有码字加以比较。

5 13 列出本章例 5 4 1 (7,4)汉明码的标准阵列译码表。若收码 $R = (0010100, 0111000, 1110010)$,由标准阵列译码表判断发码是什么。

5 14 设计一个由 $g(x) = 1 + x + x^4$ 生成的(15,11)循环汉明码的编码器。

5 15 考虑一个由 $g(x)$ 生成的 (n, k) 二元循环码 C 。若 $g^*(x) = x^{n-k}g(x^{-1})$ 定义为是 $g(x)$ 的反多项式,证明:用 $g^*(x)$ 也能生成 (n, k) 循环码。

5 16 根据例 5 4 4,设计一个(7,3)循环码,

(1) 列出所有码字证明其循环性。

(2) 写出系统形式的生成矩阵。

5 17 计算(7,4)系统循环汉明码最小重量的可纠差错图案和对应的伴随式。

5 18 某帧所含信息是(0000110101100010101100),循环冗余校验码的生成多项式是例 5 4 6 中提供的 CRC ITU T。问附加在信息位后的 CRC 校验码是什么?(用多项式长除法)

5 19 生成某(2,1,3)卷积码的转移函数矩阵是

$$G(D) = (1 + D^2, 1 + D + D^2 + D^3),$$

(1) 画出编码器结构图。

(2) 画出编码器的状态图。

(3) 求该码的自由距离 d_f

5 20 某卷积码 $G_0 = (1 \ 0 \ 0)$, $G_1 = (1 \ 0 \ 1)$, $G_2 = (1 \ 1 \ 1)$,

(1) 画出该码的编码器。

(2) 画出该码的状态图和网格图。

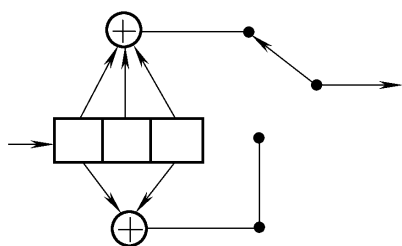
(3) 求出该码的转移函数和自由距离。

5 21 某码率为 1/2、约束长度 $K=3$ 的二进制卷积码,其编码器如图题 5 21 所示。

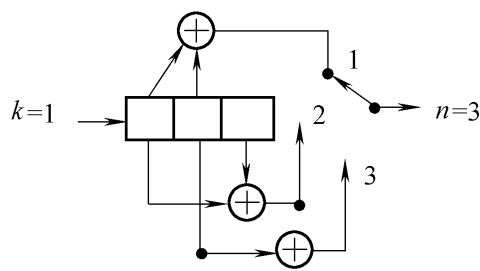
(1) 画出树图、格栅图和状态图。

(2) 求转移函数 $T(D, N, J)$, 据此指出自由距离。

5 22 某(3,1)卷积码的框图如图题 5 22 所示,



图题 5 21



图题 5 22

- (1) 画出该码的状态图。
 - (2) 求转移函数 $T(D)$ 。
 - (3) 求该码的自由距离 d_{free} , 在格栅图上画出相应路径(与全 0 码字相距 d_{free} 的路径)。
 - (4) 对 4 位信息比特 (x_1, x_2, x_3, x_4) 和紧接的 2 位 0 比特编码, 以 0.1 的差错概率通过 BSC 信道传送。已知接收序列是 (111 111 111 111 111 111), 试用维特比算法找出最大似然的发送数据序列。
- 5 23 证明: 由 CRC ITU T 生成多项式 $g(x) = x^{16} + x^{12} + x^5 + 1$ 生成的码字的重量一定是偶数。
- 5 24 例 5 4 7 中, 若 8 进制符号是 $6 \ 2 \ 3$, 生成多项式不变, 求: 编出的 8 进制 RS 码及其二进衍生码各是什么?

第 6 章 密码学

信息(如语言、文字、数据、图像等)需要利用通信网络(如电话、电报、传真、微波、卫星、光纤等)传送和交换,需要利用计算机处理和存储。显然,一部分信息由于其重要性,在一定时间内必须严加保密,严格限制其被利用的范围。利用密码对各类电子信息进行加密,以保证在其处理、存储、传送和交换过程中不会泄露,是迄今为止对电子信息实施保护,保证信息安全的唯一有效措施。

电话将为每个人提供方便的通信;高速的“电子邮件”会取代传统的“书面邮件”,商业上可能用“电子邮件”来签署、交换各类合同;银行和金融界中电子资金传递系统和信用卡将被广泛应用……显然,在这类商业或个人通信中,人们常常希望能对他们的通信内容实施加密保护和有效地证实鉴别。

计算机系统要求只有合法用户才能接入系统;广大用户希望自己输入、处理和存储的信息能不被他人利用;软件工作者希望他们辛勤劳动创造出来的系统软件和应用软件不会被其他人无偿占有;……凡此种种,人们都要求利用密码对重要信息实施保护。

作为从事通信专业的工程技术人员,应对密码学有所了解。本章在介绍密码学基本概念的基础上,着重叙述密码学发展史上两个具有里程碑作用的加密算法——数据加密标准(简称 DES)和公开密钥密码——的原理及其实现,最后再对信息安全和数字签名作常识性介绍。

6.1 密码学的基础知识

6.1.1 密码学的基本概念

人们希望把重要信息通过某种变换转换成秘密形式的信息。转换方法可以分为两大类:一类是隐写术,隐蔽信息载体(信号)的存在,古代常用。另一种是编码术,将载荷信息的信号进行各种变换使它们不为非授权者所理解。在利用现代通讯工具的条件下,隐写术受到很大限制,但编码术却以计算机为工具取得了很大的发展。我们把对真实数据施加变化的过程称为加密 E_K ,把加密前的真实数据称为明文 M ,加密后输出的数据称为密文 C 。从密文恢复出明文的过程称为解密 D_K 。加密实际上是明文到密文的函数变换,变换过程中使用的参数叫密钥 K 。完成加密和解密的算法称为密码体制。

人们一方面要把自己的信号隐蔽起来,另一方面则想把别人的隐蔽信息挖掘出来,于是,就产生了密码设计的逆科学——密码分析。密码分析研究的问题是如何把密文转换成明文,把密文转换成明文的过程称为破译。破译也是进行函数变换,变换过程中使用的参数也叫密钥。对于某一个明文以及由它产生的密文,加密时使用的密钥与解密时使用的密钥可以相同(单密钥),也可以不同(双密钥)。后面将会详细介绍。

一般地,如果求解一个问题需要一定量的计算,但环境所能提供的实际资源却无法实现

它,则称这种问题是计算上不可能的。如果一个密码体制的破译是计算上不可能的,则称该密码体制是计算上安全的。

密码体制必须满足三个基本要求:

- 对所有的密钥,加密和解密都必须迅速有效;
- 体制必须容易使用;

体制的安全性必须只依赖于密钥的保密性,而不依赖算法 E 或 D 的保密性。

第一个要求对于计算机系统是十分重要的,在进行数据传输时通常需要进行加密和解密。如果它们的运算速度过于缓慢,就会成为整个计算机网络的薄弱环节。还有存储量(程序的长度、数据分组长度、高速缓存大小)、实现平台(硬件、软件、芯片)、运行模式等因素均需折衷考虑。第二个要求意味着编码员应能方便地找到具有逆变换的密钥加以解密。第三个要求意味着加密算法和解密算法都应该很强,能使破译者仅知道加密算法还不足以破译密码。这项要求是完全必要的,因为算法要交给公众使用,破译者也会知道它。因而,无论什么人,只要根据特定的密钥 K ,就可以用 E_K 进行加密变换,用 D_K 进行解密变换。但与此同时,却不容许相反的情况成立,也就是知道 E_K 和 D_K 后不应该能导出密钥 K 。只有这样,才能阻止密码分析员破译密码。

密码体制要实现的功能可分为保密性和真实性两种。保密性要求密码分析员无法从截获的密文中求出明文。包括两项要求:

即使截获了一段密文 C ,甚至知道了与它对应的明文 M ,密码分析员要从中系统地求出解密变换仍然是计算上不可能的。

密码分析员要由截获的密文 C 系统地求出明文 M 是计算上不可能的。

第一个要求保证了不能系统地求出解密变换。第二个要求保证了在不知道解密变换的情况下无法从密文解出明文。无论被截获的密文消息的数量和长度是多少,为了实现保密性,这两个要求都必须成立。

保密性只要求对变换 D_K (解密密钥)加以保密,只要不影响 D_K 的保密,变换 E_K 可以公布于众。图 6 1 1(a)表示了这种情况。



图 6 1 1 加解密变换

数据的真实性要求密码分析员无法用虚假的密文 C 代替真实密文 C 而不被觉察。包括两个要求:

对于给定的 C ,即使密码分析员知道对应于它的明文 M ,要系统地求出加密变换 E_K 仍然是计算上不可能的。

密码分析员要系统地找到密文 C ,使 $D_K(C)$ 是明文空间上有意义的明文,这在计算上是不可能的。

第一个要求保证了不能系统地求出加密变换 E_K 。第二个要求保证了在不知道加密变换 E_K 的情况下不能找到虚假密文 C ,使它在解密后变为有意义的明文。与保密性类似,为

了实现真实性,无论被截获的密文数量多大,这两个要求都必须成立。真实性只要求变换 E_K (加密密钥)保密,变换 D_K 可公布于众。图 6 1 1(b)表示了这种情况。

密码体制可分为对称(单密钥)体制和非对称(双密钥)体制。在对称体制中,加密密钥和解密密钥相同或者很容易相互推导出。由于我们假定加密方法是众所周知的,所以这就意味着变换 E_K 和 D_K 很容易互相推导。因此,如果对 E_K 和 D_K 都保密,则保密性和真实性就都有了保障。但这种体制中 E_K 和 D_K 只要暴露其中一个,另一个也就暴露了。所以,对称密码体制必须同时满足保密性和真实性的全部要求。

对称体制用于加密私人文件十分方便。每个用户 A 都用自己的秘密变换 E_K 和 D_K 加密解密文件,如果其他用户无法得到 E_K 和 D_K ,就能保障 A 的数据的保密性和真实性。在用于保护计算机网络中的信息传输时,发送者和接收者公用秘密的通讯密钥,它通过保密信道分配给发、收双方。大量数据在加密后以密文形式由非保密信道传输。如果密码分析员无法根据截获的密文破译出明文,那么只要通讯双方诚实可靠、互相信赖,他们就能在通讯中既保障保密性又保障真实性。

直到 20 世纪 70 年代中期,所有密码体制都是对称密码体制。因此,对称(单密钥)体制通常也叫传统(或经典)体制。最有代表性的传统密码体制是美国政府颁布的数据加密标准(DES:Data Encryption Standard),将在 6.2 节中详细介绍。

非对称(双密钥)密码体制的加密密钥和解密密钥中至少有一个在计算上不可能被另一个导出。因此,在变换 E_K 或 D_K 中有一个可公开而不影响另一个的保密。

在非对称密码体制中,通过保护两个不同的变换分别获得保密性和真实性。保护 D_K 获得保密性,保护 E_K 获得真实性。公开密钥体制即是这种,如图 6 1 2(a)所示。用户 B 通过保密自己的解密密钥来保障他接收信息的保密性,但不能保证真实性,因为任何知道 B 的加密密钥的人都可以将虚假消息发给他。而图 6 1 2(b)中用户 A 通过保密自己的解密密钥来保障他发送信息的真实性。但任何知道 A 的加密密钥的人都可以破译消息,保密性不能保证。

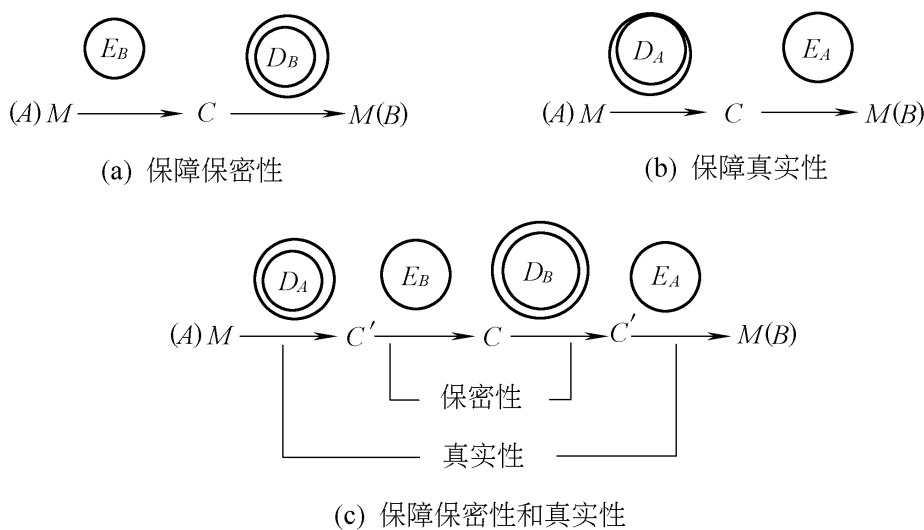


图 6 1 2 保密性和真实性

为了既实现保密性又实现真实性,发送者和接收者都必须各自运用两组变换。如图 6 1 2(c)所示,设 A 把消息 M 发送给 B,他首先使用他的秘密解密变换 D_A ,然后再用 B 的公开加密变换 E_B 将消息加密,密文发送至 B。B 用自己的秘密解密变换 D_B 和 A 的公开加密变换

E_A 两次解密后得到明文 M 。由公开变换不能简单地推导秘密变换是公开密钥体制和传统体制的主要区别。最有代表性的公开密钥密码体制是 6.3 节中将要介绍的 RSA 算法。

公开密钥真实性系统可用来识别企图进入绝密地区(如计算机房、核反应堆房等地)的人的身份。具体做法是,中央控制当局用其秘密变换为所有允许进入该地区的人建立密码标识卡(无法伪造)。卡上的密文含有姓名、声调、指纹、允许进入的地区和可以进入的时间等信息,中央控制当局的公开变换则分发到进行出入控制的所有关口。任何想出入受控地区的人都必须通过一个专用设施。在那里,他的识别信息如声调、指纹等被取样,而存储在个人标识卡上的加密信息则被解密,然后两者进行核实检查,辩明真伪。

还可以利用公开密钥的真实性来实现数字签名,在电子邮政和电子资金传送领域内得到应用。

根据加密明文数据时的加密单位的不同,可以把密码分为分组密码和序列密码两大类。设 M 为密码消息,将 M 分成等长的连续区组 M_1, M_2, \dots ,并且用同一密钥 K 为各区组加密,即

$$C = E_K(M) = E_K(M_1) E_K(M_2) \dots$$

则称这种密码为分组密码。分组的长度一般是几个字符。

若将 M 分成连续的字符或位 m_1, m_2, \dots ,并用密钥序列 $K = k_1, k_2 \dots$ 的第 i 个元素给 m_i 加密,即

$$C = E_K(M) = E_{K_1}(m_1) E_{K_2}(m_2) \dots$$

则称该密码为序列密码。以下要介绍的 DES 和 RSA 密码体制都是采用分组密码。

6.1.2 密码学中的熵概念

密码学和信息论一样,都是把信源看成是符号(文字、语言等)的集合,并且它按一定的概率产生离散符号序列。第 2 章中介绍的多余度的概念也可用在密码学中,用来衡量破译某一种密码体制的难易程度。香农对密码学的重大贡献之一在于他指出:多余度越小,破译的难度就越大。可见对明文先压缩其多余度,然后再加密,可提高密文的保密度。

香农在理论上提出了衡量密码体制保密性的尺度:在截获密文后,明文在多大程度上仍然无法确定。即如果无论截获了多长的密文都得不到任何有关明文的信息,那么就说这种密码体制是绝对安全的。

所有实际密码体制的密文总是会暴露某些有关明文的信息。在一般情况下,被截获的密文越长,明文的不确定性就越小,最后会变为零。这时,就有了足够的信息唯一地决定明文,于是这种密码体制也就在理论上可破译了。

但是理论上可破译,并不能说明这些密码体制不安全,因为把明文计算出来的时空需求也许会超过实际上可供使用的资源。因此,重要的不是密码体制的绝对安全性,而是它在计算上的安全性。

可将密码系统的安全问题与噪声信道问题进行类比。噪声相当于加密变换,接收的失真消息相当于密文,密码分析员则可类比于噪声信道中的计算者。

下面应用熵的概念。熵表示了消息的不确定性,它告诉我们,如果消息被噪声通道改变或隐藏在密文中,我们必须知道多少位才能算出正确消息。例如,如果密码分析员知道密文

块“ZSJP7K”所对应的明文要么是“MALE”,要么是“FEMALE”,那么其不确定性仅为一位。为了确定明文,密码分析员只要区分明文的两种可能值的一个位就行了。但是若上述密文块对应一个工资值,则其不确定性就不止一位了。如果知道一共只有 N 种不同的工资额,那么它不会超过 $\log_2 N$ 位。

随机变量的不确定性可以通过给予附加信息而减少。正如前面介绍过条件熵一定小于无条件熵。例如,令 X 是 32 位二进制整数并且所有值的出现概率都相等,则 X 的熵 $H(X) = 32$ 比特。假设已经知道 X 是偶数,那么熵就减少了一位,因为 X 的最低位肯定是零。

对于给定的 Y , X 的条件熵 $H(X/Y)$

$$H(X/Y) = - \sum_{i,j} p(x_i y_j) \log_2 p(x_i/y_j)$$

被称为疑义度。在密码学中,将用到两种疑义度:

(1) 对于给定密文,密钥的疑义度可表示为

$$H(K/C) = - \sum_j p(c_j) \sum_i p(k_i/c_j) \log_2 p(k_i/c_j) \tag{6.1.1}$$

(2) 对于给定密文,明文的疑义度可表示为

$$H(M/C) = - \sum_j p(c_j) \sum_i p(m_i/c_j) \log_2 p(m_i/c_j) \tag{6.1.2}$$

设明文熵为 $H(M)$, 密钥熵为 $H(K)$, 从密文破译来看,密码分析员的任务是从截获的密文中提取有关明文的信息

$$I(M; C) = H(M) - H(M/C) \tag{6.1.3}$$

或从密文中提取有关密钥的信息

$$I(K; C) = H(K) - H(K/C) \tag{6.1.4}$$

对于合法的接收者,在已知密钥和密文条件下提取明文信息,由加密变换的可逆性知

$$H(M/CK) = 0 \tag{6.1.5}$$

因而此时有

$$I(M; CK) = H(M) - H(M/CK) = H(M) \tag{6.1.6}$$

从(6.1.3)式和(6.1.4)式可知, $H(M/C)$ 和 $H(K/C)$ 越大,窃听者从密文能够提取出有关明文和密钥的信息就越小。

因为

$$H(K/C) + H(M/KC) = H(M/C) + H(K/MC) \quad (M \text{ 和 } K \text{ 交换})$$

$$H(M/C) \quad (\text{熵值 } H(K/MC) \text{ 总是大于或等于零})$$

根据(6.1.5)式,上式得

$$H(K/C) \geq H(M/C) \tag{6.1.7}$$

即已知密文后,密钥的疑义度总是大于等于明文的疑义度。可以这样来理解:由于可能存在多种密钥把一个明文消息 M 加密成相同的密文消息 C ,即满足

$$C = E_K(M)$$

的 K 值不止一个。但用同一个密钥对不同明文加密而得到相同的密文则较困难。

又因为 $H(K) \geq H(K/C)$, 由(6.1.7)式得 $H(K) \geq H(M/C)$, 则

$$I(M; C) = H(M) - H(M/C) \geq H(M) - H(K) \tag{6.1.8}$$

(6.1.8)式说明,保密系统的密钥量越少,密钥熵 $H(K)$ 就越小,其密文中含有的关于明文的

信息量 $I(M; C)$ 就越大。至于密码分析者能否有效地提取出来,则是另外的问题了。作为系统设计者,自然要选择有足够多的密钥量才行。

6.2 数据加密标准 DES

1977 年 7 月美国国家标准局公布了采纳 IBM 公司设计的方案作为非机密数据的正式数据加密标准(DES:Data Encryption Standard)。DES 密码是一种采用传统加密方法的区组密码,它的算法是对称的,既可用于加密又可用于解密。

6.2.1 换位和替代密码

根据加密时对明文数据的处理方式的不同,可以把密码分为换位密码和替代密码两类。换位密码是对数据中的字符或更小的单位(如位)重新组织,但并不改变它们本身。替代密码与此相反,它改变数据中的字符,但不改变它们之间的相对位置。

现代编码术所使用的基本方法仍然是换位和替代,但是其侧重点却不同。传统方法中都使用简单的算法,依靠增加密钥长度提高安全性。现在则是把加密算法搞得尽可能复杂,使密码分析员即使获得大量密文,也无法破译出有意义的明文。

换位和替代密码可使用简单的硬件来实现。如图 6 2 1 的硬件可实现换位(简称 P 盒)加密,其输出信息序列即为输入信息序列的一个重排列。 n 位 P 盒的输入与输出有 $n!$ 种不同的连接方法,要判明 P 盒输入的第 i 位对应于输出的第几位是不困难的。只要将第 i 位置 1,其余各位都置 0 送入 P 盒的输入端,看输出端的哪一位为 1 就行了。

图 6 2 2 表示替代(简称 S 盒)加密,其输出信息序列是输入信息序列的替代。S 盒由三级构成,第一级将输入的二进制数转换成十进制数(n 位的二进制数可以转换成 2^n 个十进制数)。第二级是一个换位盒(P 盒),用来进行十进制数的换位,形成一个排列(有 $2^n!$ 种可能的排列)。第三级再将排列的结果转换成二进制数输出。

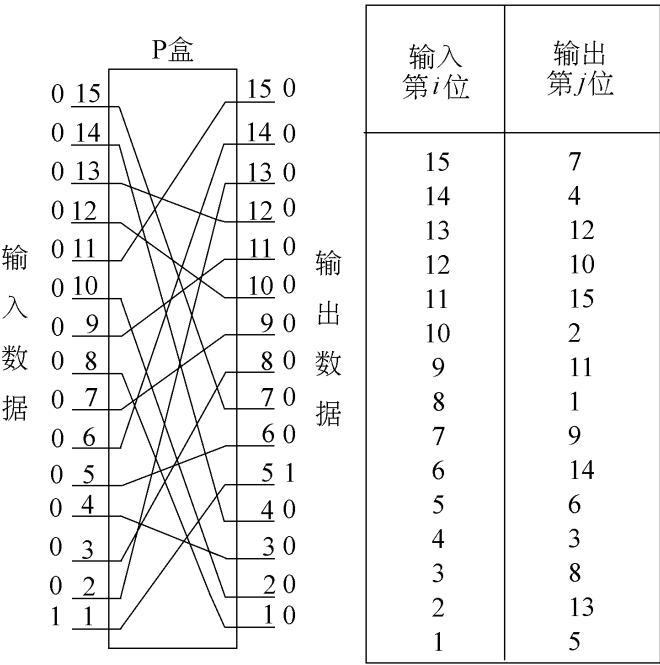


图 6 2 1 换位盒(P 盒)

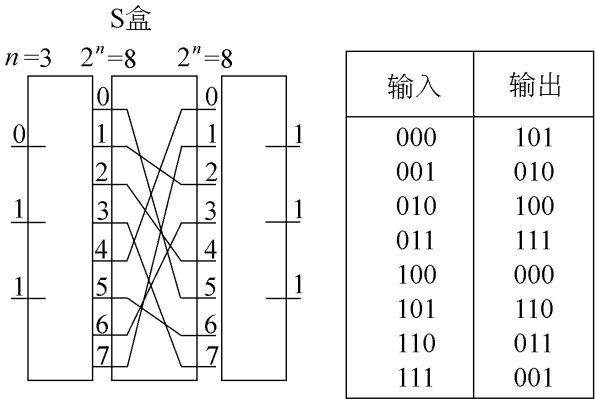


图 6 2 2 替代盒(S 盒)

S 盒比 P 盒复杂,因此位数较多的 S 盒很难实现。但位数相同时,S 盒的输入输出对应关系比 P 盒多,因而有较高的安全性。例如在 $n = 4$ 时,P 盒的输入输出对应关系只有 $4! = 24$ 种,S 盒却有 $2^4! = 16! = 2 \times 10^{13}$ 种。

单独使用 P 盒或位数较少的 S 盒,都不能达到较高的安全性,因为人们可以比较容易地检测出它们的输入输出对应关系。但若交替结合使用这两者,则可以大大提高安全性。图 6 2 3 表示由 15 位的 P 盒与 5 个并置的 3 位 S 盒所组成的 7 层硬件密码产生器。设 P 盒和 S 盒的输入输出对应关系分别如图 6 2 1 和 6 2 2 所示,并令输入信息的最低位为 1,其余各位为 0,则在该密码输出端将出现图示的信息。

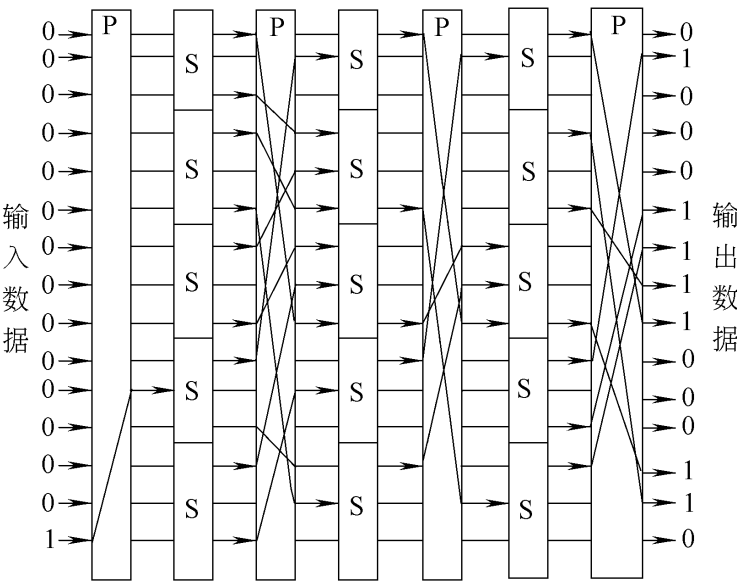


图 6 2 3 P 盒和 S 盒的结合使用

这里每层 S 盒由 5 个 3 位的 S 盒并联构成。在理论上它也可以由唯一一个 15 位的 S 盒形成,但是那会使设备的第二级需要 $2^{15} = 32\,768$ 根交叉线,这在工艺上是无法实现的。因此,在 P 盒与 S 盒结合使用时,S 盒层总是分成若干个位数较少的 S 盒,然后把它们并置在一起。

6.2.2 DES 密码算法

DES 密码就是在上述换位和替代密码的基础上发展的。图 6 2 4 为其算法框图,将输入明文序列分成区组,每组 64 比特。首先将 64 比特进行初始置换 IP。置换规则如表 6 2 1,即将输入的第 58 位置换到第 1 位输出,第 50 位换到第 2 位,……,依表类推,第 7 位换到最后一位等等。

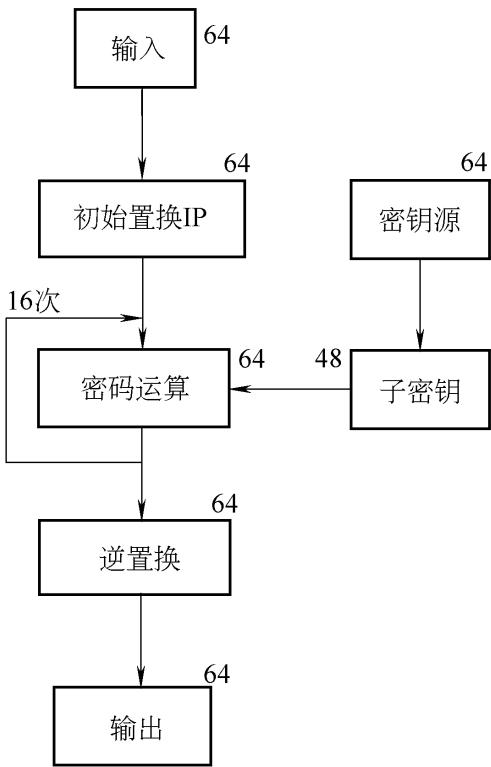


图 6 2 4 DES 算法

表 6 2 1 初始置换表 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

然后进行密码运算,它是在密钥控制下的 16 步非线性变换,如图 6 2 5 所示。先将 64 比特分成左右两组各 32 比特 L_0 和 R_0 ,迭代运算如下:

$$\begin{aligned} L_1 &= R_0, & R_1 &= L_1 \oplus f(R_0, K_1) \\ L_2 &= R_1, & R_2 &= L_1 \oplus f(R_1, K_2) \\ &\dots & &\dots \\ L_{16} &= R_{15}, & R_{16} &= L_{15} \oplus f(R_{15}, K_{16}) \end{aligned}$$

其中 $f(R_{i-1}, K_i)$ 是密码计算函数,如图 6 2 6 所示,将 32 比特 R_{i-1} 经过表 6 2 2 的扩充函数 E 变成 48 比特,与 48 比特的子密钥 K_i 按位模 2 加,再经 8 个 S 盒。这些 S 盒的功能是把 6 比特数变换成 4 比特数,替代函数如表 6 2 3 所示。具体做法是以 6 比特数中的第 1 和第 6 比特组成的二进制数为行号,以第 2,3,4,5 比特组成的二进制数为列号,查找 S_i ,行列交叉处即是要输出的 4 比特数。例如输入 S_1 的 6 比特数为 110010,则以“10”即 2 为行,以“1001”即 9 为列,输出为 12 即“1100”。8 个 S 盒的输出拼接为 32 比特数据区组,最后经 P 盒换位输出,换位函数如表 6 2 4。

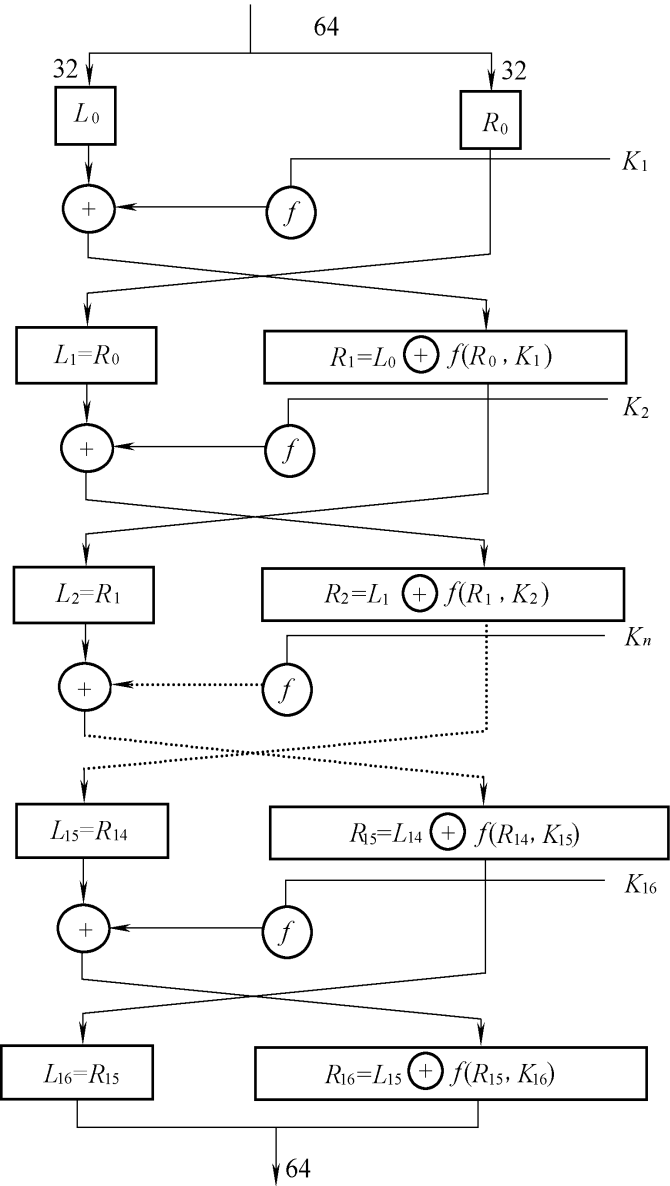


图 6 2 5 密码运算

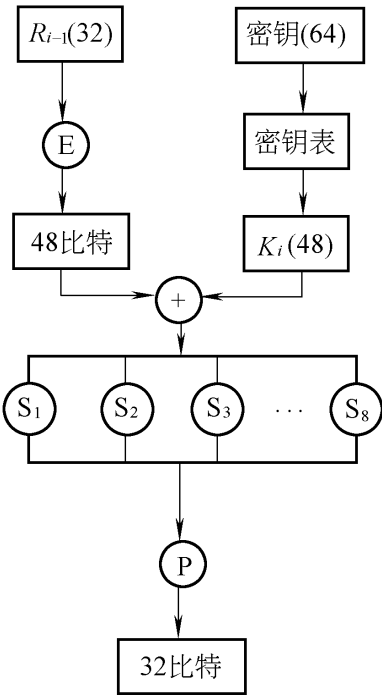


图 6 2 6 密码计算函数 $f(R, K)$

16 个子密钥是由同一个 64 比特的密钥源 $K = k_1 k_2 \dots k_{64}$ 循环移位产生。密钥源中 56 比特是随机的,所有 8 的倍数位 $k_8, k_{16}, \dots, k_{64}$ 是为奇偶校验而设。图 6 2 7 为计算子密钥的流程图,首先对 64 比特的密钥源进行第一次置换选择,变成 56 比特,置换选择规则如表6 2 5。

表 6 2 2 扩充函数 E

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

表 6 2 3 替代函数

替代函数	列																	行
(S_i)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
S_1	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7	0	
	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8	1	
	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0	2	
	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13	3	
S_2	15	1	8	14	6	11	3	4	9	7	5	13	12	0	5	10	0	
	3	13	4	7	15	2	8	15	12	0	1	10	6	9	11	5	1	
	0	14	8	11	10	4	13	1	5	8	12	6	9	3	2	15	2	
	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9	3	
S_3	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8	0	
	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1	1	
	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7	2	
	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12	3	
S_4	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15	0	
	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9	1	
	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4	2	
	3	15	0	6	10	1	13	8	9	4	5	11	12	4	2	14	3	
S_5	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9	0	
	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6	1	
	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14	2	
	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3	3	
S_6	12	1	10	15	9	2	6	8	0	13	3	4	4	7	5	11	0	
	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8	1	
	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6	2	
	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13	3	
S_7	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1	0	
	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6	1	
	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2	2	
	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12	3	
S_8	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7	0	
	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2	1	
	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8	2	
	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11	3	

表 6 2 4 换位函数 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

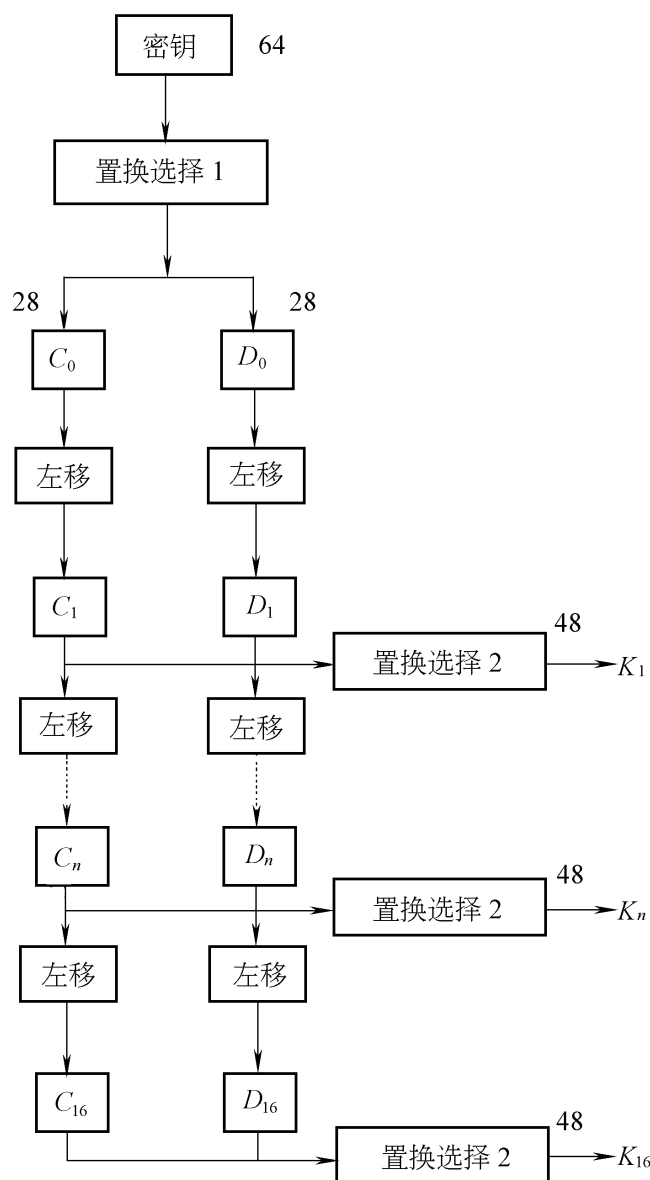


图 6 2 7 密钥表计算

表 6 2 5 置换选择 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

然后将 56 比特分存到两个 28 比特的寄存器 C_0 和 D_0 中。除了寄存器 (C_0, D_0) 外, 还有 16 对寄存器, 即 $(C_1, D_1), \dots, (C_{16}, D_{16})$ 。每个寄存器都是 28 比特。加密时, 寄存器 (C_{i+1}, D_{i+1}) 中的内容是将 C_i 和 D_i 中的内容分别向左移 1 至 2 位得到的。而且这种移位方式是按循环移位寄存器方式进行, 也即从寄存器左边移出的比特, 又从右边补入到寄存器的头一位。移位多少与寄存器的位置 (即序号) 有关, 如表 6 2 6 所示。即寄存器 (C_0, D_0) 的内容向左循环移 1 位, 分别装入 C_1 和 D_1 。而 C_1 和 D_1 的内容向左循环移 1 位分别装入 C_2 和 D_2 , 依此类推。在经过 16 次的循环移位后, 一共移了 28 位, 保证了 $C_{16} = C_0, D_{16} = D_0$ 。从 C_i 和 D_i 的输出拼接成的 56 比特再经第二次置换选择就得到了 48 比特的子密钥 k_i , 置换选择 2 如表 6 2 7 所示。

表 6 2 6 寄存器的移位数

寄存器序号 (i)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
左移位数	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

表 6 2 7 置换选择 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

经过 16 次密码运算后, 必须再进行逆初始置换, 它是初始置换的逆变换。这样就保证了加密和解密是可逆的, 可以共用同一个程序或硬件, 只是所用于子密钥的顺序相反而已。如加密时采用 K_1, K_2, \dots, K_{16} , 则解密时就用 $K_{16}, K_{15}, \dots, K_1$ 。逆置换的规则如表 6 2 8。

表 6 2 8 逆初始置换

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

6.2.3 DES 密码的安全性

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。

如果 DES 算法中每次迭代所用的子密钥都相同, 即

$$K_1 = K_2 = \dots = K_{16}$$

就称给定的密钥 K 为弱密钥。此时有

$$\text{DES}_K[\text{DES}_K(x)] = x, \quad \text{DES}_K^{-1}[\text{DES}_K^{-1}(x)] = x$$

即以 K 对 x 加密两次或解密两次都恢复出明文。其加密运算和解密运算没有区别。而对一般密钥只满足

$$\text{DES}_K^{-1}[\text{DES}_K(x)] = \text{DES}_K[\text{DES}_K^{-1}(x)] = x$$

弱密钥下使 DES 在选择明文攻击下的搜索量减半。

弱密钥的构造是由子密钥产生器中寄存器 C 和 D 中的存数在循环移位下出现的重复图样决定的, 参看图 6 2 7。若 C 和 D 中存数为 0 或 1 重复 28 次的图样, 即 $(0, 0, \dots, 0)$, 或 $(1, 1, \dots, 1)$, 则在循环左移位下保持不变, 因而相应的 16 个子密钥都相同。可能产生弱密钥的 C 和 D 的存数有四种组合, 其十六进制表示为

$$\begin{array}{ll}
 (0, 0) & \backslash \quad 00 \ 00 \ 00 \ 00 \ 00 \ 00 \ 00 \ 00 \\
 (0, 15) & \backslash \quad 00 \ 00 \ 00 \ 0F \ FF \ FF \ FF \\
 (15, 0) & \backslash \quad FF \ FF \ FF \ F0 \ 00 \ 00 \ 00 \\
 (15, 15) & \backslash \quad FF \ FF \ FF \ FF \ FF \ FF \ FF
 \end{array}$$

相应的输入的秘密密钥 K 的十六进制表示为

(0 , 0)	\	01	01	01	01	01	01	01	01
(0 , 15)	\	1F	1F	1F	1F	0E	0E	0E	0E
(15 , 0)	\	E0	E0	E0	E0	1F	1F	1F	1F
(15 , 15)	\	FE	FE	FE	FE	FE	FE	FE	FE

若给定密钥 K , 相应的 16 个子密钥只有两种图样, 且每种都出现 8 次, 就称它为半弱密钥。半弱密钥的特点是成对地出现, 且具有下述性质: 若 K_1 和 K_2 为一对互逆的半弱密钥, x 为明文组, 则有

$$\text{DES}_{K_1}(\text{DES}_{K_2}(x)) = \text{DES}_{K_2}(\text{DES}_{K_1}(x)) = x$$

称 K_1 和 K_2 是互为对合的。若寄存器 C 和 D 的存数图样是 2 的重复数字, 如 (0101...01) 或 (1010...10), 则这种图样对于偶次循环移位具有自封闭性, 对于奇数次循环具有互封闭性。而 (00...0) 和 (11...1) 图样显然也具有上述性质。若 C 和 D 的初值选自这四种图样, 则所产生的子密钥就会只有两种, 且每种都出现 8 次。可能的组合有 $4 \times 4 = 16$ 个, 其中弱密钥有 4 个, 半弱密钥有 12 个, 组成 6 对。

如果随机地选择密钥, 则在总数 2^{56} 个密钥中, 弱密钥所占比例极小, 而且稍加注意就不难避开。因此, 弱密钥的存在不会危及 DES 的安全性。

对 DES 安全性批评意见中, 较为一致的看法是 DES 的密钥短了些, IBM 最初向 NSA (美国的国家安全局) 提交的建议方案采用 112 比特密钥, 但公布的 DES 标准采用 64 比特密钥。有人认为 NSA 故意限制 DES 的密钥长度, 以保证他自己能够破译, 但其他预算经费较少的单位则无法破译。DES 的密钥量为 $2^{56} = 7.2 \times 10^{16}$ 个。有人则认为 56 比特已足够了, 选择长的密钥会使成本提高、运行速度降低。若要对 DES 进行密钥搜索破译, 分析者在得到一组明文 密文对条件下, 可对明文用不同的密钥加密, 直到得到的密文与已知的明文 密文对中的相符, 就可确定所用的密钥了。密钥搜索所需的时间取决于密钥空间的大小和执行一次加密所需的时间。若假设 DES 加密操作需时为 $100 \mu\text{s}$ (一般微处理器能实现), 则搜索整个密钥空间需时为 7.2×10^{15} 秒, 近似为 2.28×10^8 年。若以最快的 LSI 器件, DES 加密操作时间可降到 $5 \mu\text{s}$, 也要 1.1×10^4 年才能穷尽密钥。

但是由于最新的两个破译法——差分 and 线性密码分析法——的出现以及计算机技术的发展, 在 1993 年破译 DES 的费用为 100 万美元, 需时 3.5 小时。RSA 数据安全公司为破译 DES 提供 10 000 美元奖金。现已被 DESCHALL 小组经过近 4 个月的努力, 通过 Internet 搜索了 3×10^{16} 个密钥, 找出了 DES 的密钥, 恢复出明文。1998 年 5 月美国 EFF (Electronic Frontier Foundation) 宣布, 他们以一台价值 20 万美元的计算机改装成的专用解密机, 用了 56 小时破译可采用 56 比特密钥的 DES。因此在现有的条件下破译 56 比特密钥的 DES 已经是完全可能的了。据报道, 美国国家标准和技术协会正在征集新的称之为 AES (Advanced Encryption Standard) 加密标准, 新算法很可能要采用 128 比特密钥。

自 DES 正式成为美国标准以来, 已有许多公司设计并推出了实现 DES 算法的产品。有的设计专用 LSI 器件或芯片, 有的用现成的微处理器实现。有的只限于实现 DES 算法, 有的则可运行各种工作模式。对于器件所提供的物理保护也各不相同, 从没有保护的单片到可防篡改的装置。美国 NSA 至少已认可了 31 种硬件和固件实现产品, 每年平均批准 3 件。硬件实现的价格为 1 000 美元左右, 而完整的加密机为 3 000 美元左右。在这方面, 其他任何算法都无法和 DES 竞争。

6.2.4 DES 密码的改进

尽管 DES 算法十分复杂,但它基本上还是采用 64 比特字符的单字母表替换密码。当同样的 64 比特明文块进入编码器后,得到的是同样的 64 比特的密文块。破译者可利用这个性质来破译 DES。

要了解这种单字母替换码的缺点,可用下面的一个例子说明:一张工资表中共有 3 列:姓名、职称、工资。采用二进制编码,姓名 16 个字节(16×8 比特),职称和工资各 8 字节(8×8 比特)。用 DES 算法对这张工资表进行加密,以便传给银行,将正确的工资数打入每个员工的储蓄卡中。但是若有人想篡改(调换或替代),则是非常容易的事。由于每位员工有 4 个 64 比特块,只需将两人的第 3、4 个 64 比特密文块调换一下即可,而银行在解密时,不会发现问题。

为了改进 DES 算法,可采用密码块链接的方法,该方法适用于所有分组密码。如图 6 2 8 所示,每个明文块在加密之前都与前一个密文块进行异或操作,如第一个明文块 M_1 先与一个随机选择的初始矢量 V 异或,再进行加密得到密文块 C_1 ;将 C_1 与 M_2 异或,再进行加密得到密文块 C_2依此类推,这样同一个明文块在不同位置就会生成不同的密文块,加密不再是一个单字母替换密码,如果密文块移位后,就会导致解密时明文毫无意义。另外由于同一个明文块不会生成同一个密文块,这也给密码破译者带来了困难。

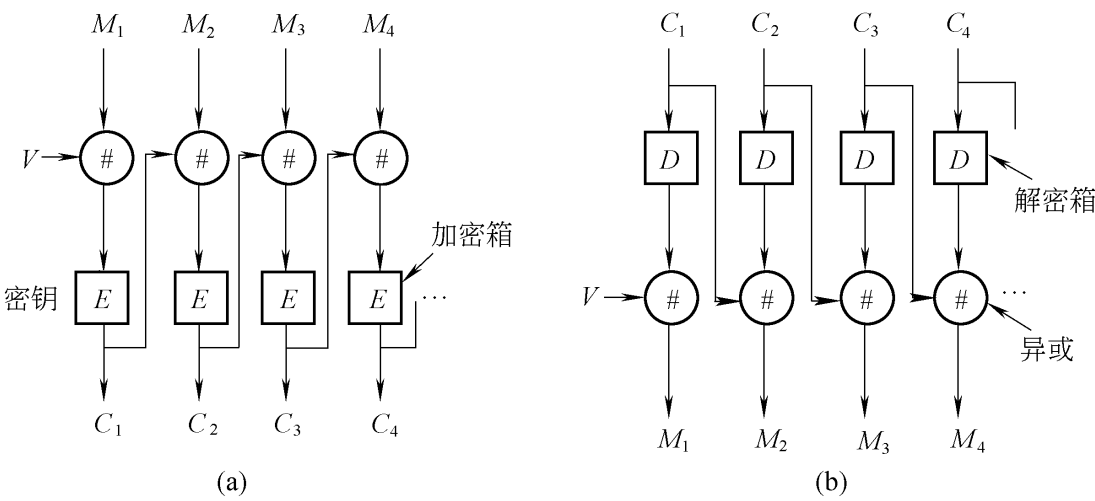


图 6 2 8 密码块链接

但是密码块链接也有缺点,只有当所有 64 比特块到达后才能开始解码。如果使用交互式终端,即用户可以键入少于 8 字符的数据行,然后停下来等待响应,那么这种方式就不适用。此时可采用按字节加密的方式——密码反馈方式,如图 6 2 9 所示,(a)图中显示了当字节 0~9 被加密及发送后加密机的状态。当明文的第 10 个字节 M_{10} 到达时,DES 算法对 64 比特的移位寄存器内容进行加密,生成 64 比特的密文,读取密文最左边的字节与 M_{10} 异或,生成密文 C_{10} 后被输出。同时移位寄存器左移 8 位, C_2 从最左边移出, C_{10} 填入到 C_9 右边的空位中。由于移位寄存器的内容与所有前面的明文有关,所以内容相同的多次明文将产生不同的密文。显然在这种密码块链接中,也需要一个初始矢量。

图(b)是这种加密方式的解密过程,要恢复原来的明文 M_{10} ,则对接收到的 C_{10} 进行异或时,需用原来的加密字节。也就是说,解密时的移位寄存器必须与加密时的移位寄存器保持一致,对它产生的 64 个比特也进行加密操作,就可以生成原来加密时的字节,从而正确解

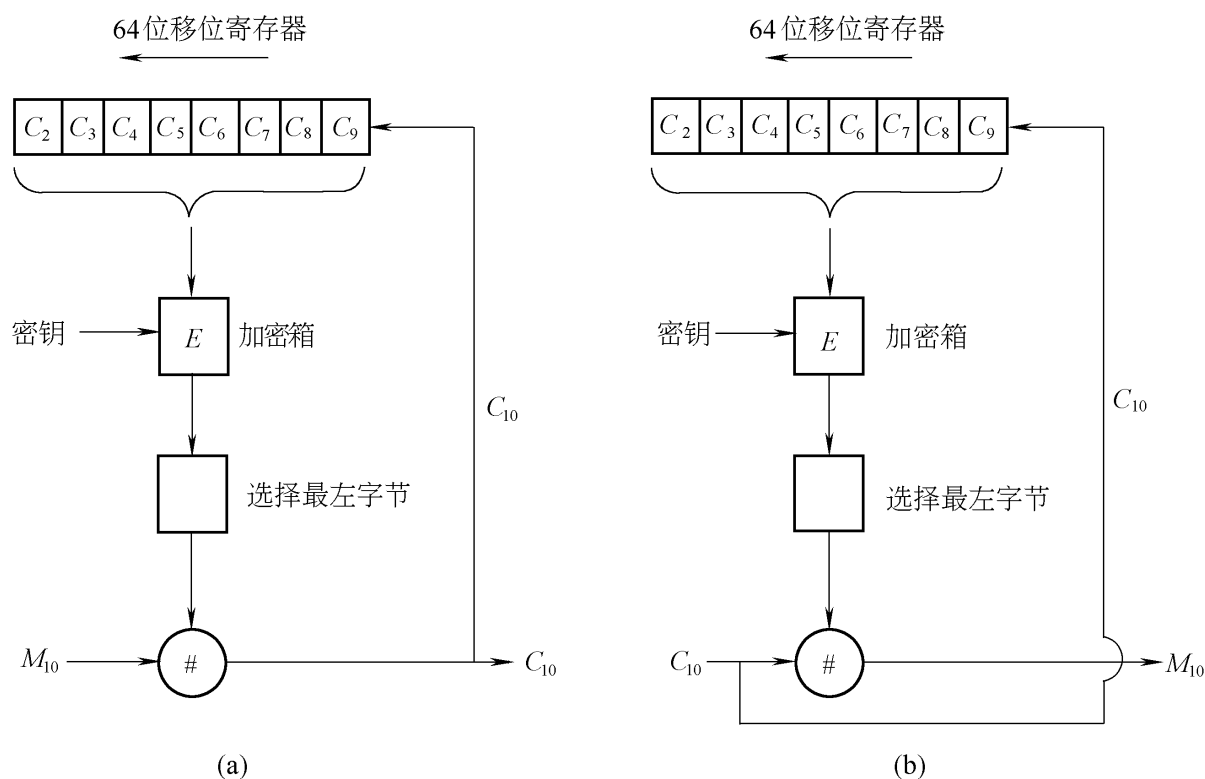


图 6 2 9 密码反馈方式

密。但是如果在传输过程中,有某一比特密文发生错误,则当这一字节在移位寄存器中时,解密的 8 字节都会出错,直到该错误字节移出寄存器为止,以后的字节才可能正确解密。

6.3 国际数据加密算法

尽管一次加密的 DES 仍然广泛应用于保密中,如银行的自动取款机(ATM)。但专家们对 DES 不安全的原因作了大量的分析,认为这种方法在十年或更久以前(当它刚被发明时)是很适用的,而现在已不再能满足需要。人们开始寻求更安全的块密码,曾提出了许多算法,其中最令人感兴趣最重要的就是 IDEA(International Data Encryption Algorithm),即国际数据加密算法。

IDEA 由瑞士的两名科学家于 1990 年提出,最早称作 PES(Proposed Encryption Standard),后改称为 IDEA,1992 年进行了改进,强化了抗差分攻击法的能力。

6.3.1 算法原理

输入和输出字长为 64 比特,密钥长 128 比特,8 轮迭代体制。采用下述几种基本运算:

逐位 $\text{mod } 2$ 和,记作 \oplus ;

$\text{mod } 2^{16}$ (即 65536)整数加,记作 \oplus ;

$\text{mod } (2^{16} + 1)$ (即 65537)整数乘,记作 \odot ;

三个运算中任意两个运算不满足分配律。例如:

$$a \odot (b \oplus c) \neq (a \odot b) \oplus (a \odot c)$$

三个运算中任意两个运算间不满足结合律。例如:

$$a \odot (b \odot c) \neq (a \odot b) \odot c$$

前三种运算之间不具兼容性。这些运算使输入之间实现了较复杂的组合运算,8 次迭

代后经过一个输出变换给出密文。IDEA 可用于各种标准工作模式。实现时考虑了下述三个方面：

(1) 基本构件——乘/加单元。实现 16 比特为字长的非线性 S 盒,如图 6 3 1 所示。它是 IDEA 实现中的关键非线性构件。通过 8 轮迭代,能够完成更好的扩散和混淆。研究表明,为实现完善混淆至少需要 4 轮迭代。

(2) 硬件。加密、解密运算相似,差别是密钥时间表,类似于 DES,具有对合性,可用同一器件实现。由于采用规则的模块结构,易于设计 ASIC 实现。

(3) 软件。采用子段结构:以 16 比特为字长进行处理。采用简单运算,三种运算易于编程实现加、移位等。

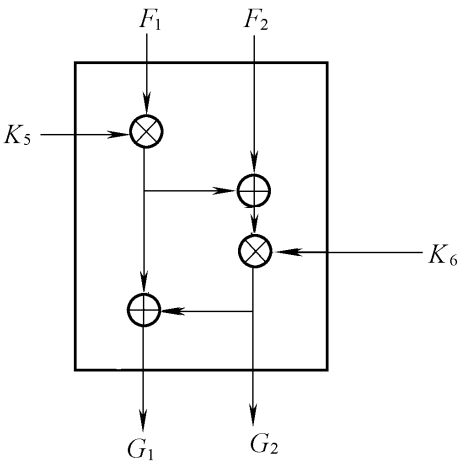


图 6 3 1 乘加单元

6.3.2 加密解密过程

加密过程的框图如图 6 3 2 和 6 3 3 所示。它由两部分组成：一个是对输入 64 比特明文组的 8 轮迭代产生 64 比特密文输出；另一个是由输入的 128 比特会话密钥,产生 8 轮迭代所需的 52 个子密钥,共 52×16 比特。运算过程组成均采用 16 比特。

每轮的迭代过程如图 6 3 4 所示,每次迭代所用密钥不同,结构相同。输出变换如图6 3 5

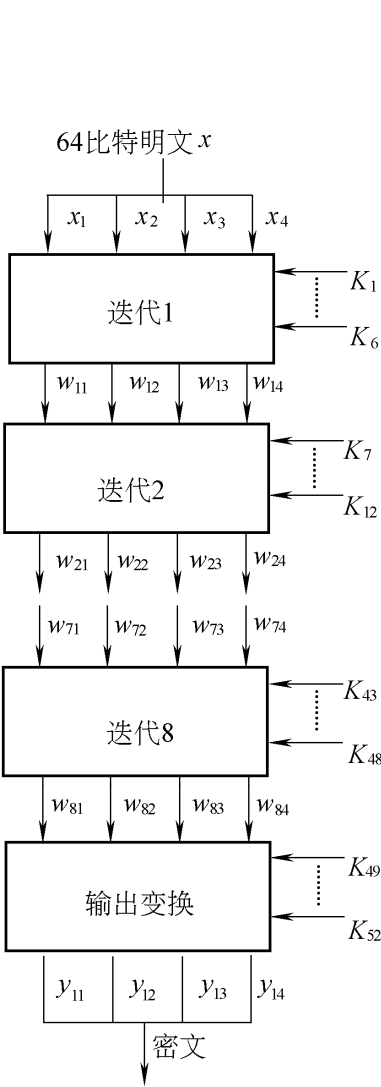


图 6 3 2 IDEA 算法框图

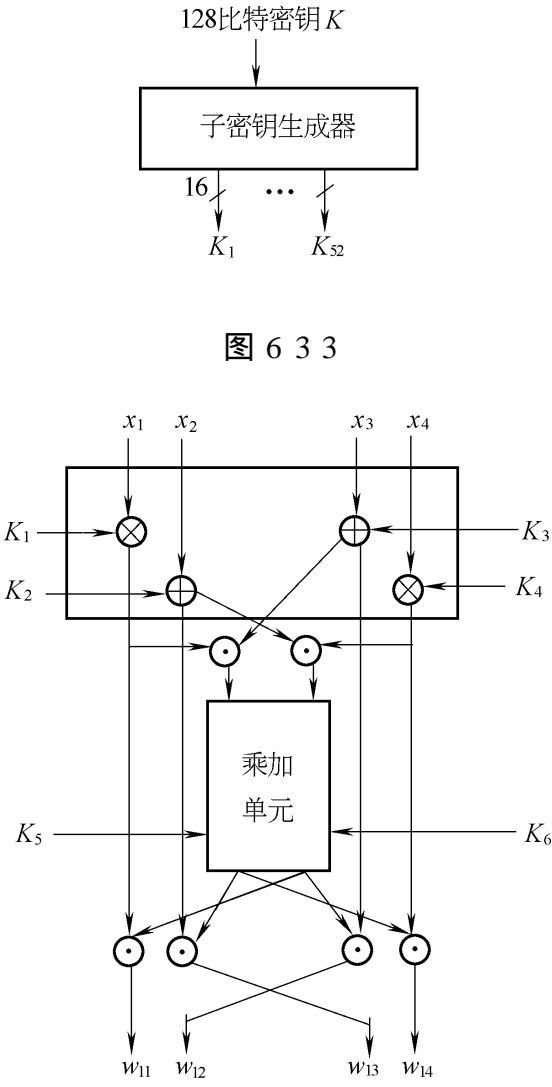


图 6 3 3

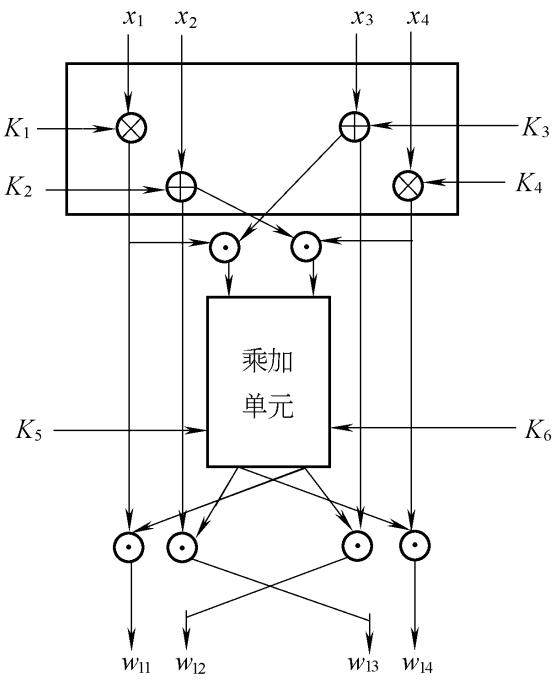


图 6 3 4 IDEA 的一次迭代过程

所示,主要功能是保证 IDEA 整个加密、解密具有对合性质。

子密钥产生器是以输入的 8×16 比特会话密钥作为前 8 个子密钥 K_1, K_2, \dots, K_8 , 然后将 128 比特移位寄存器循环左移 25 位, 形成子密钥 $K_9, K_{10}, \dots, K_{15}$, 如图 6 3 6 所示。重复移位过程, 直到给出子密钥 $K_{49}, K_{50}, K_{51}, K_{52}$ 。这种迭代每轮需要 6 个子密钥, 而密钥产生器每轮移位后给出 8 个子密钥, 所以 IDEA 算法中每轮所用子密钥将从 128 比特会话密钥移位寄存器中的不同位置取出。

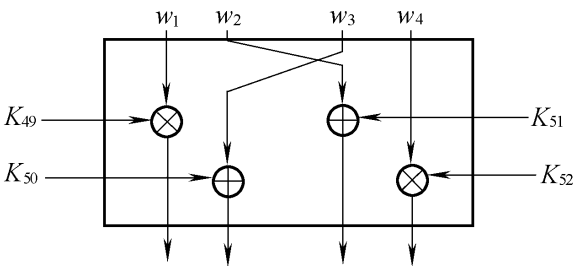


图 6 3 5 IDEA 的输出变换

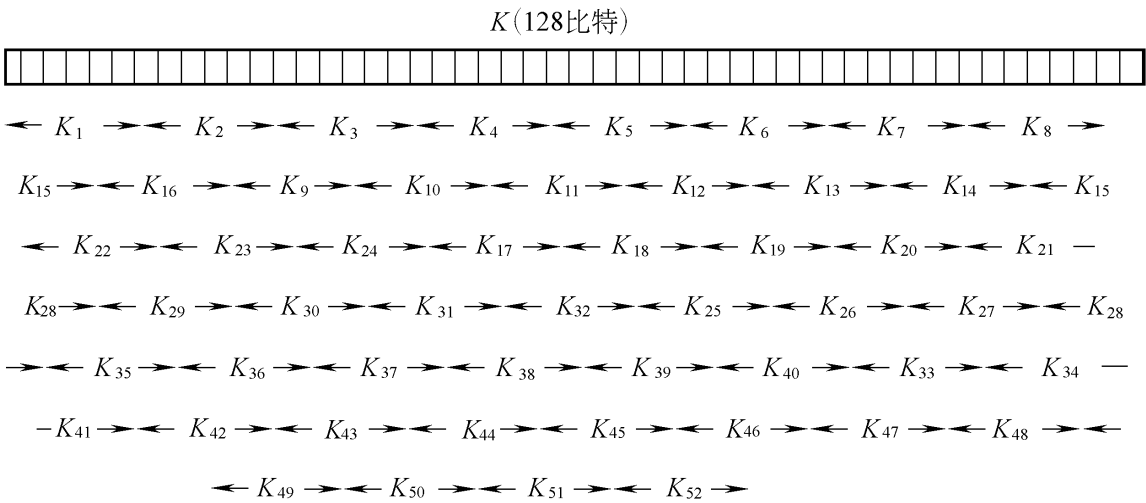


图 6 3 6 IDEA 的子密钥

IDEA 算法的解密过程和加密相同, 只是解密所用的子密钥与加密子密钥之间具有如表 6 3 1 给出的关系。表中的密钥满足下述关系:

$$K_i^{-1} \cdot K_j^{-1} = 1 \pmod{2^{16} + 1}$$
$$- K_i \cdot K_j = 0 \pmod{2^{16}}$$

表 6 3 1 IDEA 加密、解密子密钥

	加 密 钥	解 密 钥	
第 1 轮	$K_1, K_2, K_3, K_4, K_5, K_6$	$Z_1, Z_2, Z_3, Z_4, Z_5, Z_6$	$K_{49}^{-1}, -K_{50}, -K_{51}, K_{52}^{-1}, K_{47}, K_{48}$
第 2 轮	$K_7, K_8, K_9, K_{10}, K_{11}, K_{12}$	$Z_7, Z_8, Z_9, Z_{10}, Z_{11}, Z_{12}$	$K_{43}^{-1}, -K_{45}, -K_{44}, K_{46}^{-1}, K_{41}, K_{42}$
第 3 轮	$K_{13}, K_{14}, K_{15}, K_{16}, K_{17}, K_{18}$	$Z_{13}, Z_{14}, Z_{15}, Z_{16}, Z_{17}, Z_{18}$	$K_{37}^{-1}, -K_{39}, -K_{38}, K_{40}^{-1}, K_{35}, K_{36}$
第 4 轮	$K_{19}, K_{20}, K_{21}, K_{22}, K_{23}, K_{24}$	$Z_{19}, Z_{20}, Z_{21}, Z_{22}, Z_{23}, Z_{24}$	$K_{31}^{-1}, -K_{33}, -K_{32}, K_{34}^{-1}, K_{29}, K_{30}$
第 5 轮	$K_{25}, K_{26}, K_{27}, K_{28}, K_{29}, K_{30}$	$Z_{25}, Z_{26}, Z_{27}, Z_{28}, Z_{29}, Z_{30}$	$K_{25}^{-1}, -K_{27}, -K_{26}, K_{28}^{-1}, K_{23}, K_{24}$
第 6 轮	$K_{31}, K_{32}, K_{33}, K_{34}, K_{35}, K_{36}$	$Z_{31}, Z_{32}, Z_{33}, Z_{34}, Z_{35}, Z_{36}$	$K_{19}^{-1}, -K_{21}, -K_{20}, K_{22}^{-1}, K_{17}, K_{18}$
第 7 轮	$K_{37}, K_{38}, K_{39}, K_{40}, K_{41}, K_{42}$	$Z_{37}, Z_{38}, Z_{39}, Z_{40}, Z_{41}, Z_{42}$	$K_{13}^{-1}, -K_{15}, -K_{14}, K_{16}^{-1}, K_{11}, K_{12}$
第 8 轮	$K_{43}, K_{44}, K_{45}, K_{46}, K_{47}, K_{48}$	$Z_{43}, Z_{44}, Z_{45}, Z_{46}, Z_{47}, Z_{48}$	$K_7^{-1}, -K_9, -K_8, K_{10}^{-1}, K_5, K_6$
输出置换	$K_{49}, K_{50}, K_{51}, K_{52}$	$Z_{49}, Z_{50}, Z_{51}, Z_{52}$	$K_1^{-1}, -K_2, -K_3, K_4^{-1}$

6.3.3 算法的安全性

如果采用穷搜索破译,要求进行 2^{128} 10^{38} 次试探。若每秒可完成 100 万次加密,需 10^{13} 年;若用 10^{24} 个 ASIC 芯片阵需要一天。有关专家研究表明,IDEA 算法没有似 DES 意义下的弱密钥,8 轮迭代使得没有任何捷径破译,在差分和线性攻击下是安全的。当然,若将字长由 16 比特增加到 32 比特,密钥相应长 256 比特,采用 2^{32} 模加, $2^{32} + 1$ 模乘,则可进一步强化 IDEA。

6.4 公开密钥加密法

如果将国际数据加密算法用于电子邮件和电子资金传送时,因为必须把密钥分配给许多通信者,就显示出不足。密钥分配增加了暴露报文或截获者获得报文的危险性。提出公开密钥加密法(PKC),使用两个不同密钥来减小上述危险性。一个公开作为加密密钥,另一个为用户专用,作为解密密钥。通信双方无需事先交换密钥就可进行保密通信。而要从公开的公钥或密文分析先后明文或秘密密钥,在计算上是不可能的。若以公开密钥作为加密密钥,以用户专用密钥作为解密密钥,则可实现多个用户加密的消息只能由一个用户解读;反之,以用户专用密钥作为加密密钥,而以公开密钥作为解密密钥,则可实现由一个用户加密的消息而使多个用户解读。前者可用于保密通信,后者可用于数字签名。

PKC 算法的成功在于加密函数的单向性,即求逆函数的困难性。即使知道加密函数也不可能导出解密函数,也就是加密函数的逆函数。

PKC 使用特殊的数学函数,称为单向窍门函数 $y = F(x)$ 。它满足这些特性: 对自变量 x 的任意给定值,容易计算 $y = F(x)$ 的值。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“窍门”(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} ,这是计算上不可能的。而预定的接收者则可以用窍门信息(解密密钥 K_d)简单地求解 $x = F_{K_d}^{-1}(y)$ 。

6.4.1 公开密钥密码体制

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)。如图 6 4 1 所示即为这种加密体制。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 6 4 2 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

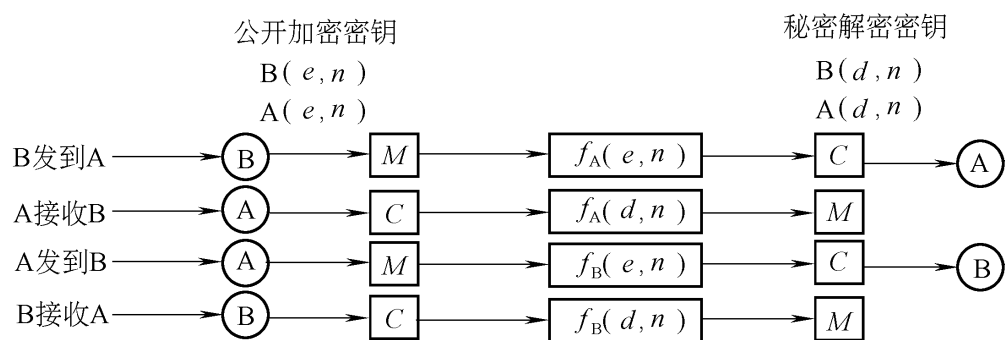


图 6 4 1 公开密钥密码体制

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。

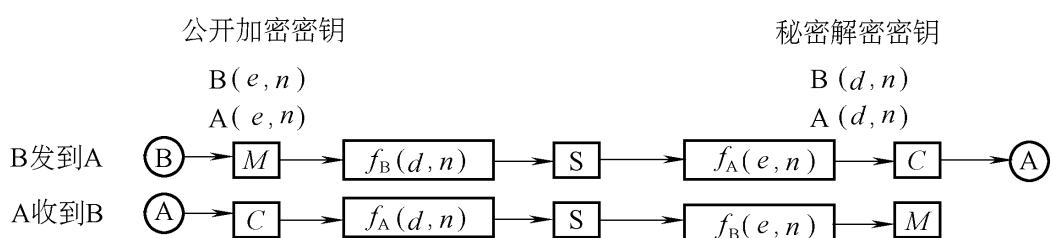


图 6 4 2 公开密钥密码体制

6.4.2 RSA 密码体制

RSA 体制是根据 PKC 算法由美国麻省理工学院(MIT)的研究小组提出的,该体制的名称是用了三位作者(Rivest, Shamir 和 Adleman)英文名字的第一个字母拼合而成。该体制的理论基础是数论中的下述论断:要求得到两个大素数(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 体制的加密过程通过三个数 e, d, n 来实现。

加密时: $y = x^e \pmod n$

解密时: $x = y^d \pmod n$

上面同余方程(即方程两边余数相等)的意思是:加密时将明文 x 自乘 e 次,然后除以模数 n ,余数便是密文 y ;解密运算是将密文 y 自乘 d 次,再除以 n ,余数便是明文 x 。

在设计过程中,需要两个密钥:一个公开密钥(e, n),一个秘密密钥(d, n)。具体做法如下:

- (1) 选取两个很大的素数 p 和 q ,令模数 $n = p \times q$;
- (2) 求 n 的欧拉函数 $\phi(n) = (p - 1) \times (q - 1)$,并从 2 至 $[\phi(n) - 1]$ 中任选一个数作为加密指数 e ;
- (3) 解同余方程 $(e \times d) \pmod{\phi(n)} = 1$,求得解密指数 d ;
- (4) (e, n) 即为公开密钥, (d, n) 即为秘密密钥。

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 6 4 1 在 RSA 方法中,令 $p = 3, q = 17$,取 $e = 5$,试计算解密密钥 d 并加密 $M = 2$ 。
解

$$n = p \times q = 51$$

$$(n) = (p - 1) \times (q - 1) = 32$$

$$(5 \times d) \bmod 32 = 1, \text{可解得 } d = 13$$

于是

$$y = x^e \bmod n = 2^5 \bmod 51 = 32$$

验算

$$y^d \bmod n = 32^{13} \bmod 51 = 2 = x$$

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 6 4 3 所示:

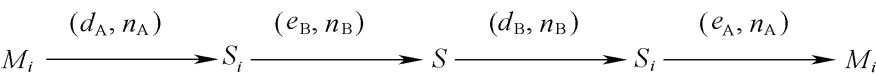


图 6 4 3 RSA 公开密钥体制签名略图

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。

例 6 4 2 用户 A 发送给用户 B 一份密文,用户 A 用首字母 B = 02 来签署密文。用户 A 知道三个密钥: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)	

A 计算他的签名: $S_i = M_i^{d_A} \bmod n_A = (02)^{23} \bmod 123 = 8388608 \bmod 123 = 8$

再次加密签名: $S = S_i^{e_B} \bmod n_B = 8^{13} \bmod 51 = 549755813888 \bmod 51 = 26$

接收者 B 必须恢复出 02 = B 认证他接收的密文。接收者也知道三个密钥: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)

用户 B 用自己的秘密解密密钥一次解密:

$$S_i = S^{d_B} \bmod n_B = 26^5 \bmod 51 = 11881376 \bmod 51 = 8$$

再用 A 用户的公开密钥加密:

$$M_i = S_i^{e_A} \bmod n_A = 8^7 \bmod 123 = 2097153 \bmod 123 = 2$$

结果 $M_i = 2$,就是 B = 02 值,可以确认密文的发送者是 A,B 用户能够确信这点,是因为只有用户 A 具有秘密的解密密钥(d_A, n_A),只有 A 自己能生成用他的公开密钥(e_A, n_A)能够解

密的密文。

RSA 方法中,由于不能由模 n 简单地求得 (n) ,也无法简单地由 e 推算 d ,因而 e 和 n 可以公开,而不会泄露 (n) 和 d 。机密核心在于秘密密钥 d ,一旦 d 失窃,别人也就窃得了相应的被加密信息。因此,必须对秘密密钥 d 采取防窃措施。

一个现代密码体制必须能经得住训练有素的密码分析家借助计算机寻找秘密密钥的攻击或用某些其它方法试破密文的攻击。在 RSA 体制中,如果密码分析家(知道公开密钥 e 和 n)能把 n 分解成 p 和 q ,那么他就可以计算出 (n) ,接着找出秘密密钥分量 d 。

例如,公开密钥为 $(5,51)$, $n = 51$ 的因数只有 3 和 17。这样小的 n 很容易被分解并找出秘密密钥 (d, n) 。当前的技术进展使分解算法和计算能力在不断提高,计算所需的硬件费用在不断下降,110 位十进制数字早已能分解。表 6 4 1 给出以 NSF 算法破译 RSA 体制与穷搜索密钥法破译单密钥体制的等价密钥长度。因此今天要用 RSA,需要采用足够大的整数 n 。

表 6 4 1 破译单密钥体制和 RSA 体制的等价密钥长度

单密钥体制(比特)	RSA 体制(比特)	单密钥体制(比特)	RSA 体制(比特)
56	384	112	1792
64	512	128	2304
80	768		

512 比特(154 位)、664 比特(200 位)已有实用产品,也有人想用 1024 比特的模,若以每秒可进行 100 万步的计算资源分解 664 比特大整数,需要完成 10^{23} 步,即要用 1000 年。据研究 1024 比特模在今后 10 年内足够安全,而 150 位数将在本世纪被分解。目前 512 比特模在短期内仍十分安全,但大素数分解工作在 WWW 上大协作已构成对 512bit 模 RSA 的严重威胁,很快可能要采用 768bit 甚至 1024bit 的模。

RSA 算法的硬件实现速度很慢,最快也只有 DES 的 1/ 1000,512 比特模下的 VLSI 硬件实现只达 64 kbit/ s。目前计划开发 512 比特 RSA 达 1Mb/ s 的芯片。软件实现的 RSA 的速度只有 DES 的软件实现的 1/ 100,在速度上 RSA 无法与对称密钥体制相比,因而 RSA 体制多用于密钥交换和认证。512 比特 RSA 的软件实现的速率可达 11 kbit/ s。

6.4.3 报文摘要

这种方案基于单向散列(Hash)函数的思想,该函数从一段很长的报文中计算出一个固定长度的比特串,作为该报文的摘要(message digest)。它具有下列重要性质:

- 给出报文 P 就易于计算出报文摘要 MD(P);
- 只给出 MD(P),几乎无法找出 P ;
- 无法生成两条具有同样报文摘要的报文;

从一段明文中计算一段报文摘要比用公开密钥算法加密明文要快得多,因此采用报文摘要节省了加密时间,同时也节省了报文传输和存储的开销。首先用户 A 计算明文信息的报文摘要,然后在报文摘要上签名,并将签名的摘要和明文一起发送给用户 B。如果第三者替换了 P ,当 B 用户计算 MD(P)时就会发现这一点。

1. MD5 算法

目前已提出多种报文摘要,应用最广的一种是 MD 5。输入报文的长度是任意的,而输

出的报文摘要长度固定为 128 比特。它以一种充分复杂的方式将各比特打乱,每个输出比特都受每一个输入比特的影响。下面介绍 MD 5 的算法,采用 MD 5 算法产生报文摘要的全过程如图 6 4 4 所示,其过程如下所述:

- (1) 在原报文长度 K 后添加若干比特,使其长度比 512 的整数倍少 64。即使有些报文的长度已经达到要求,但还是必须添加。例如原报文长度为 448,则需要添加 $512 \times 2 - 64 = 960$ bit。因此添加长度的范围在 1~512 之间,添加的内容是第 1 比特为 1,其余为 0。
- (2) 用 64 bit 表示原报文的长度 K ,再添加在后面。如果原报文的长度大于 2^{64} ,则仅表示 $K \bmod 2^{64}$ 的余数。

这样经过上述两步的添加,其长度就为 512 的整数倍了。图 6 4 4 中用 Y_0, Y_1, \dots, Y_{L-1} 分别表示 512 的比特块。为了便于在 32 位的机器上运算,每块可用 16 个 32 位字表示,共 $N=16 \times L$ 个字。

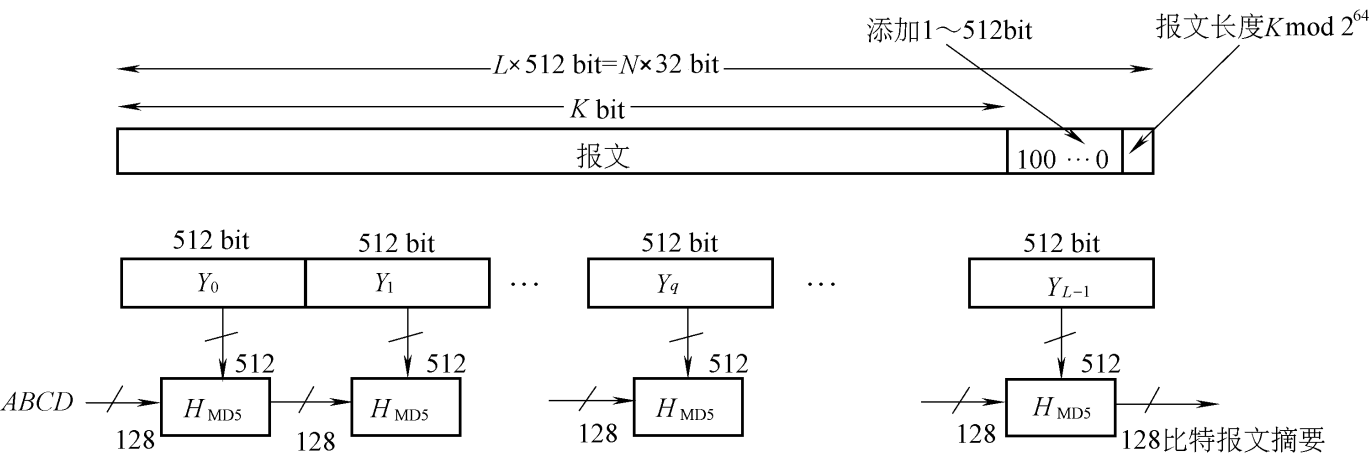


图 6 4 4 采用 MD5 算法产生报文摘要

- (3) 依次对 L 组 512 比特块进行处理,算法的核心是 H_{MD5} 模块。用 128 bit 的存储器来存放散列(hash)函数的中间结果和最终结果,由 4 个 32 位寄存器 A, B, C, D 组成,它们的初始存放数用十六进制表示为 $A = 01234567, B = 89AABCDEF, C = FEDCBA98, D = 76543210$ 。

- (4) H_{MD5} 模块的处理过程如图 6 4 5 所示,有 4 轮运算。 Y_q 表示输入的第 q 组 512 比特, $q = 0, 1, \dots, L - 1$,每轮使用一次。 $T[1, \dots, 64]$ 为 64 个元素表,分成 4 组参与不同轮的计算, $T[i]$ 为 $2^{32} \times \text{abs}[\sin(i)]$ 的 32 位二进制整数部分, i 是弧度,其数值如表 6 4 2 所示。该 32 比特是将输入数据打乱,随机化。 MD_q 为寄存器 $ABCD$ 的中间结果, MD_0 是初始化值, MD_L 是最终的报文摘要结果。

- (5) 四轮运算的结构类似,如图 6 4 6 所示,运算关系如下式:

$$A \leftarrow B + CLS_s\{A + g(BCD) + X[k] + T[i]\} \tag{6 4 1}$$

式中 $ABCD$ 为寄存器的内容, g 为基本逻辑函数 F, G, H, J 中之一,每轮用一种。CLS 将 32 比特数循环左移 s 位。 $X[k]$ 是第 q 组 512 比特中第 k 个 32 位字, $k = 1, 2, \dots, 16$,因此 (6 4 1) 式的迭代运算需要 16 次。函数 F, G, H, J 的逻辑关系不同,函数定义式如表 6 4 3 所示,表 6 4 4 为逻辑真值表。

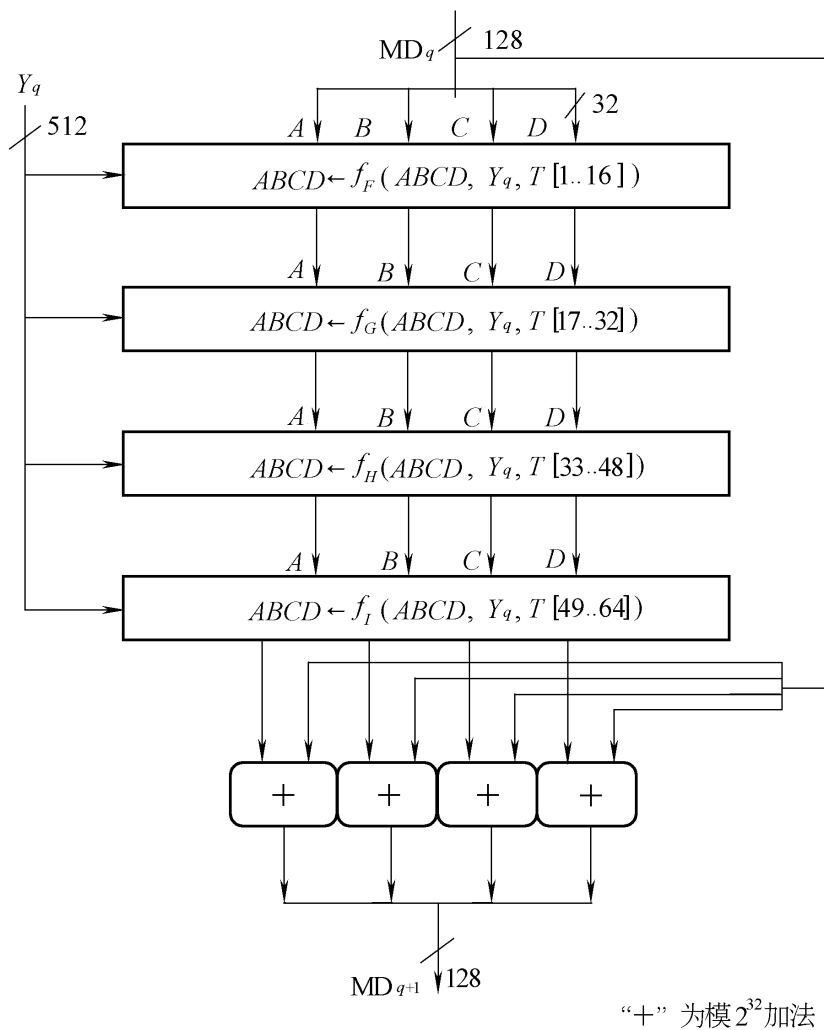


图 6 4 5 处理 512 比特块的算法 H_{MD5}

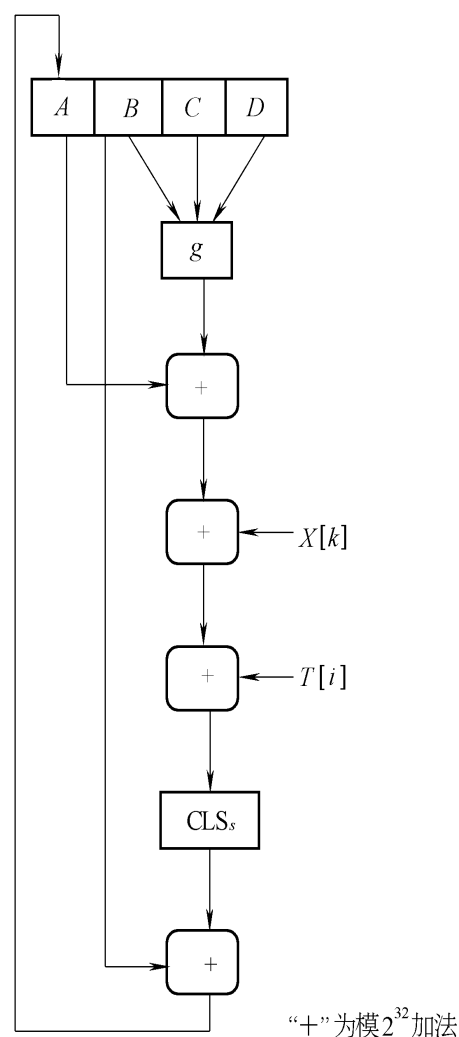


图 6 4 6 (6 4 1) 式运算图

表 6 4 2 从 $\sin e$ 函数构造的 T 表

$T[1] = D76AA478$	$T[17] = F61E2562$	$T[33] = FFFA3942$	$T[49] = F4292244$
$T[2] = E8C7B756$	$T[18] = C0408340$	$T[34] = 8771F681$	$T[50] = C32AFF97$
$T[3] = 242070DB$	$T[19] = 265E5A51$	$T[35] = 69D96122$	$T[51] = AB9423A7$
$T[4] = C1BDCEEE$	$T[20] = E9B6C7AA$	$T[36] = FDE5380C$	$T[52] = FC93A039$
$T[5] = F57C0FAF$	$T[21] = D62F105D$	$T[37] = A4BEEA44$	$T[53] = 655B59C3$
$T[6] = 4787C62A$	$T[22] = 02441453$	$T[38] = 4BDECFA9$	$T[54] = 8F0CCC92$
$T[7] = A8304613$	$T[23] = D8A1E681$	$T[39] = F6BB4B60$	$T[55] = FFEFF47D$
$T[8] = FD469501$	$T[24] = E7D3FBC8$	$T[40] = BEBFBC70$	$T[56] = 85845DD1$
$T[9] = 698098D8$	$T[25] = 21E1CDE6$	$T[41] = 289B7EC6$	$T[57] = 6FA87E4F$
$T[10] = 8B44F7AF$	$T[26] = C33707D6$	$T[42] = EAA127FA$	$T[58] = FE2CE6E0$
$T[11] = FFFF5BB1$	$T[27] = F4D50D87$	$T[43] = D4EF3085$	$T[59] = A3014314$
$T[12] = 895CD7BE$	$T[28] = 455A14ED$	$T[44] = 04881D05$	$T[60] = 4E0811A1$
$T[13] = 6B901122$	$T[29] = 49E3E905$	$T[45] = D9D4D039$	$T[61] = F7537E82$
$T[14] = FD987193$	$T[30] = FCEFA3F8$	$T[46] = E6DB99E5$	$T[62] = BD3AF235$
$T[15] = A679438E$	$T[31] = 676F02D9$	$T[47] = 1FA27CF8$	$T[63] = 2AD7D2BB$
$T[16] = 49B40821$	$T[32] = 8D2A4C8A$	$T[48] = C4AC5665$	$T[64] = EB86D391$

表 6 4 3 基本函数的逻辑运算关系

轮	基本函数 g	$g(B, C, D)$	轮	基本函数 g	$g(B, C, D)$
f_F	$F(B, C, D)$	$(B \cdot C) \quad (\overline{B} \cdot D)$	f_H	$H(B, C, D)$	$B \quad C \quad D$
f_G	$G(B, C, D)$	$(B \cdot C) \quad (C \cdot \overline{D})$	f_I	$I(B, C, D)$	$C \quad (B \cdot \overline{D})$

表 6 4 4 基本函数的真值表

B	C	D	F	G	H	I
0	0	0	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	1	1	0
0	1	1	1	0	0	1
1	0	0	0	0	1	1
1	0	1	0	1	0	1
1	1	0	1	1	0	0
1	1	1	1	1	1	0

2. MD5 的安全性

求具有相同 Hash 值的两个消息在计算上是不可行的。MD5 的输出为 128 bit,若采用纯强力攻击寻找一个消息具有给定 Hash 值的计算困难性为 2^{128} ,用每秒可试验 1 000 000 000个消息的计算机需时 1.07×10^{22} 年。若采用生日攻击法,寻找有相同 Hash 值的两个消息需要试验 2^{64} 个消息,用每秒可试验 1 000 000 000 个消息的计算机需时 585 年。差分攻击对 MD5 的安全性不构成威胁。

3. 公开密码体制的优、缺点

在传统的密码体制中,由于加密密钥和解密密钥可以简单地互导,因此密钥必须首先经由安全通道分发给通讯双方,随后才能利用公开通道建立起安全通讯,因而密钥分配问题是传统密码体制的薄弱环节。公开密钥密码体制不存在这个问题,所以特别适合于在计算机网络中建立分散于各地的用户之间的秘密通讯联系。

与传统密码体制相比,公开密码体制的优点是:

- (1) 减少了密钥数量。这对于多用户的商用密码通信系统和计算机通信网络具有十分重要的意义。如前所述,在 n 个用户的密码系统中,采用传统密码体制,需要 $n(n - 1)/ 2$ 个密钥。采用公钥密码体制,只需要 n 对密钥,而真正需要严加保管的只有用户自己的秘密密钥。
- (2) 彻底消除了经特殊保密的密钥信道分送密钥的困难,消除了密钥在分送过程中被窃的可能性,大大提高了密码体制的安全性。
- (3) 便于实现数字签名,完满地解决了对发方和收方的证实问题,彻底解决了发、收双方就传送内容可能发生的争端,为在商业上广泛应用创造了条件。

在目前,公钥密码体制的缺点也是显然的,它的工作基础是利用了单向函数的单向性,一般说来加密和解密要经过较复杂的计算过程,而传统密码体制算法较简单,可采用大规模集成电路实现。因此,公钥密码体制对信息加密和解密的工作速率还远低于传统密码体制。但由于公钥密码体制彻底克服了传统体制在密钥分送和保存上的巨大困难,且能实现加密信息的电子签名,显示了美好的发展前景,可以预料随着密码学的进一步发展,公钥密码体

制一定会获得广泛应用。

6.5 模拟信号加密

上面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。电话是目前最重要的、业务最繁忙的通信,因此这里就以话音加密为例,总的来说,话音加密可分为数字加密和模拟加密两大类。

模拟话音的特性可用它的时间、频率和幅度的三维语音图来完整地表示,因而对模拟话音进行加密,可以按照一定规律改变话音的幅度、频率和时间的特征实现加密。其中单独处理幅度、频率和时间的分别称为幅度置乱、频率置乱和时间置乱,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。

数字话音加密就是将模拟话音数字化,再对数字序列进行加密变换。在接收端,先解码,再数模转换成模拟的话音。数字加密有两种方法,一种是比特置乱,即将数字话音分为相同长度的不同时段,然后对时段内的数字脉冲的位置按所选择的密钥规则进行扰乱实现加密。第二种方法是比特掩盖加密,就是在数字话音序列上叠加伪随机序列。

模拟加密主要是采用话音的时域或频域置乱,实现起来比较容易,加密后的话音频带较窄,故可在原来的模拟信道中传输。但一般地说,模拟加密话音的保密度较差,较容易被窃听者破译。而数字加密因为是在话音数字化后采取置乱措施,为了获得高保密度和良好的话音质量,需要用较高的取样速率,因此需要通频带较宽的信道进行传输,此外还有一种模拟加密方式,它是在话音数字化后进行时域或频域置乱,再通过数模变换成为模拟加密话音。因此这种加密方式兼有模拟和数字加密的优点,即既可获得较高的保密度,又可在模拟信道中传输。

6.6 通信网络中的加密

根据网络的构形和通信的特点,在通信网络中可根据不同的要求采用三种加密方式:链路加密、节点加密和端到端加密。对这几种加密方式,数据加密设备(DEE)都须和数据线路终结设备(DCE)、数据终端设备(DTE)之间保持一致,并使用户感到透明。

链路加密是在相邻的网络节点间对数据进行保护,也即在邻近两个节点之间的链路上传送的数据是加密的,而在节点中的信息是以明文形式出现的,因而对用户程序员与系统程序员都是透明的。其加密可在密码设备中实现,或用加密软件来完成。当用密码设备时,在节点和有关的调制解调器之间安置两个装配了相同密钥的密码设备,如图 6 6 1 所示,它既

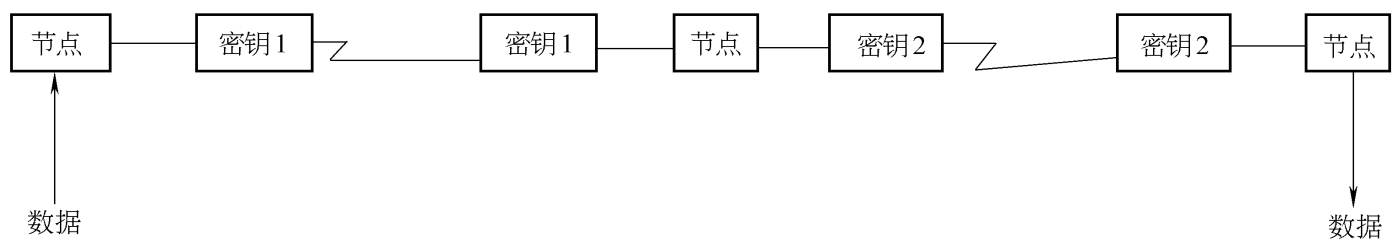


图 6 6 1 链路加密

有加密能力又有解密能力。

节点加密类似于链路加密,即每对节点共用一个密钥来保护两个节点间的通信数据。不同的是节点加密时,数据在发送节点和接收节点是以明文形式出现,而在中间节点,数据并不象链路那样使用明码,是在一个安全的模块(设备)内部,从一个密钥控制下的密文转换成另外一个密钥控制下的密文。其过程如图 6 6 2 所示。故不同节点之间密码的密钥可以是不同的。

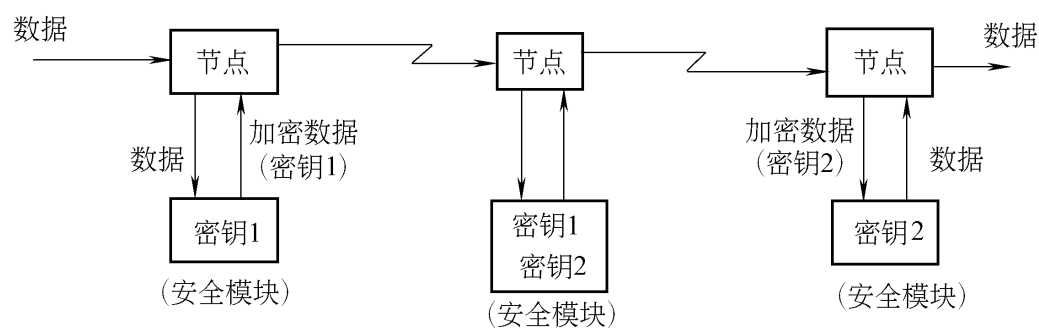


图 6 6 2 节点加密

端到端加密是当数据在用户间传输时一直受到保护。只是在终端才进行解密,在整个传输过程中是以一个确定的密钥和算法进行加密的,如图 6 6 3 所示,在中间节点或在与它们有关的安全模块内永远不会以明码的形式出现。

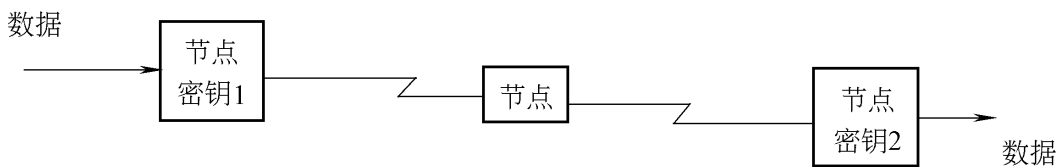


图 6 6 3 端对端加密

使用链路和节点加密,用户一般并不知道信息正在接受密码保护(即密码功能由网络提供,对用户是透明的)。使用链路加密时,在加密数据通过的所选择的路径中的每一个节点必须有自己单独的密码设备。这些设备被连结到它的输入和输出端口。使用节点加密时,在所选择的密码数据所通过的路径中每一个节点必须有它自己的安全模块。使用终端到终端加密时,只要求发出或者接收加密信息的那些节点具有加密能力。这有效地减少了网络中必须使用密码学或密码设备的地点。

6.7 信息安全和确认技术

随着信息技术的发展,大规模的计算机通信网已成为一个很普遍的传送、存储和处理信息的系统。通信网的服务范围已扩展到了包括电子资金传送、有价值的合作数据传送和医学记录信息存储等重要功能,将网络作为个人和敏感的通信使用已越来越普遍。在这样大规模的信息系统中,信息资源的共享是很方便的,但并不是任何信息资源都可供每个人自由享用。对不同范围的信息就其使用目的、价值和后果而言,共享的范围应有严格的限制;但另一方面,信息资源也应得到充分的保护以防人为的篡改、破坏。因此信息系统的安全问题是极为重要的急待解决的问题,同时也是一个复杂的问题。

6.7.1 信息安全的基本概念

在一个大规模的计算机通信网中,它所包含的信息安全问题是多方面的。在网络的不同层次上,有不同的安全要求,信息安全措施有技术的,也有管理的。当前密码学研究人员最为关心的是网络信息系统中信息传输时窃听泄密问题、数据库存储等系统的资源接入控制问题和对信息进行完整性保护以防篡改、破坏、病毒侵入等问题。现代密码学为信息系统的安全性提供了有效的技术保证,信息保密系统为传输和存储中的信息提供了加密手段,基于密码技术发展起来的数字签名、身份验证、消息确证系统为抵抗攻击者对信息系统进行主动攻击提供了强有力的手段。

在网络通信中,主要的安全防护措施被称作安全业务。有五种通用的安全业务:

(1) 认证业务

认证业务提供了关于某个人或某个事情身份的保证。这意味着当某人(或某事)声称具有一个特别的身份(如某个特定的用户名称)时,认证业务将提供某种方法来证实这一声明是正确的。口令是一种提供认证的熟知方法。

(2) 访问控制业务

访问控制的目标是防止对任何资源(如计算资源、通信资源或信息资源)进行非授权的访问。所谓非授权访问包括未经授权的使用、泄露、修改、销毁以及颁发指令等。访问控制直接支持保密性、完整性、可用性以及合法使用的安全目标。可采用防火墙技术。

(3) 保密业务

保密业务就是保护信息不泄露或不暴露给那些未授权掌握这一信息的实体(例如人或组织)。一般采用数据加密的方法。

(4) 数据完整性业务

数据完整性业务(或简称为完整性业务)是对安全威胁所采取的一类防护措施,这种威胁就是以某种违反安全策略的方式,改变数据的价值和存在。改变数据的价值是指对数据进行修改和重新排序;而改变数据的存在则意味着新增或删除它。依赖于应用环境,以上任何一种威胁都有可能产生严重的后果。

(5) 不可否认业务

不可否认业务与其他安全业务有着最基本的区别。它的主要目的是保护通信用户免遭来自于系统其他合法用户的威胁,而不是来自于未知攻击者的威胁。“否认”最早被定义成一种威胁,它是指参与某次通信交换的一方事后虚伪地否认曾经发生过本次交换。不可否认业务是用来对付此种威胁的。事实上这种业务不能消除业务否认。也就是说,它并不能防止一方否认另一方对某件已发生的事情所作出的声明。它能够做的只是提供无可辩驳的证据,以支持快速解决这种纠纷。通常采用数字签名技术。

6.7.2 数字签名

数字签名在信息安全,包括身份认证、数据完整性、不可否认性以及匿名性等方面有重要应用,特别是在大型网络安全通信中的密钥分配、认证以及电子商务系统中具有重要作用。

信息安全系统除了信息保密外,还需要抵抗对手的主动攻击,即在一个网络中,信息发

送方和接收方之间产生以下几方面的问题：

伪造：接收方伪造一份来自某一发送方的文件；

篡改：接收方篡改接收到的文件或其中的数据；

冒充：网络中任一用户冒充另一用户作为接收方或发送方；

否认：发送/接收方不承认曾发送/接收过某一文件。

这些属于接收方和发送方双方之间的问题，仅用数据加密的方法而不让第三方获得数据，是无法解决的。在不使用计算机网络交换文件的场合，常使用手写签名来防止上述问题的发生。但在计算机网络中，由于用户地理位置不同，而且传输的文件是数据形式，所以无法使用手写签名。为此，必须设计一个手迹签名的代替方案，有一个这样的系统能用以下的方式将一个“签名的”文件发送到另一方：接收者可以确认发送者的身份；发送者以后不能否认文件是他发的；接收者自己不能伪造该文件。

第一个条件是必须的，比如在一个经济系统中，当一位顾客通过计算机发订货单，向一家银行订购一吨黄金，银行计算机需要证实发出订购要求的计算机确实属于付款的公司。第二个条件用于保护银行不受欺骗。假设银行为该顾客买入了这吨黄金，但金价随后立即暴跌，狡猾的顾客可能会控告这家银行，声称自己从未发出过任何订购黄金的订单。第三个条件用来在下述情况下保护顾客，如金价暴涨，银行伪造一个文件，说顾客只要买一条黄金而不是一吨黄金。

满足上述要求的数字签名将在以下方面优于手写签名。例如数字签名可以通过计算机网络使地理位置不同的用户实现签名；数字签名既可有手写签名那样的可见性，又可将签名存储于计算机系统之中；数字签名与整个文件的每一组成部分都有关，从而保证了不变性，而手写签名的文件则可以改换某一页内容；数字签名可以对一份文件的一部分进行签署，这是手写签名所不能做到的；手写签名一般要经过专家的鉴定才能确认，而在一个具有良好数字签名方案的网络内，接收方可以立即识别接收的文件中的签名的真伪。

数字签名技术就是利用数据加密技术、数据变换技术，根据某种协议来产生一个反映被签署文件的特征以及反映签署人的特性的数字化签名，以保证文件的真实性和有效性。

数字签名技术是建立在其他一些技术基础之上的。这些基础影响到数字签名的安全性、实用性以及实现的方法。首先，数字签名是在网络环境下应用的，因此与网络的组成有很大关系。如果是局域网，因各用户所处的地理位置接近，对数字签名的功能要求就不高；如果是远程网，则要求有很强的数字签名方案。在公用数据网中，由于入网的用户类型和数目繁多，对数字签名的要求较高；而在本系统内部专用的网中，要求相对较低。如果网络中有网络管理中心或安全控制中心，则在实现数字签名时可充分利用这一条件；相反地，在没有这类控制中心的简单网中，则要设计其他类型的数字签名方案。

其次，数字签名的实现是在网络内已具有数据加密功能的前提下进行的，即假定第三者至多能得到签名参与者双方交换的密码数据，而不能获得其明文数据。除此之外，签名双方在签名过程中自始至终利用了数据加密来达到签名有效性的目的。所以，数据加密是数字签名的重要基础。目前有两大类加密算法：一类是秘密密钥加密方法，它的代表是 DES 算法；另一类是公开密钥加密算法，它的代表是 RSA 算法。

1. 秘密密钥的数字签名

这种签名方法需要一个众人信任的中心权力者(BB)，他知道每件事情。每个用户选择

一个秘密密钥,将其亲手交给 BB。这样只有用户 A 和 BB 知道 A 的秘密密钥 K 。如果用户 A 要将一文件传送给 B 用户,则须经下述过程:

- (1) 用户 A 用自己的秘密密钥加密报文 P 得到 $K_A(P)$, 并发送给 BB;
- (2) BB 解密 $K_A(P)$ 得到 P , 然后建立一个由 A 的名字和地址、日期、初始报文组成的一个新报文 $(A + D + P)$ 。再用一个对任何人都保密的密钥 X 加密产生 $X(A + D + P)$, 并回送给 A。这样 BB 可以确认请求确实来自于 A, 因为只有 A 和 BB 知道 K_A 。如果一个冒充者发送给 BB 一个报文, 那么用 K_A 解密出的报文将无任何意义;
- (3) 用户 A 发送 $X(A + D + P)$ 给 B 用户;
- (4) 用户 B 发送 $X(A + D + P)$ 给 BB, 请求得到 $K_B(A + D + P)$ 作为结果;
- (5) 用户 B 将 $K_B(A + D + P)$ 解密得到明文信息 A, D 和 P 。

如果用户 A 否认发送过 P 给用户 B, 则 B 可以将 $X(A + D + P)$ 提供给法官。法官命令 BB 将其解密, 当法官看到 A, D 和 P 时知道用户 A 在撒谎。因为用户 B 不知道 X , 所以不能伪造 $X(A + D + P)$ 。图 6 7 1 所示就是在两个陌生人之间采用秘密进行的密钥报文签名传送, 可以看到每传送这样的一条报文必须请求 BB 两次。

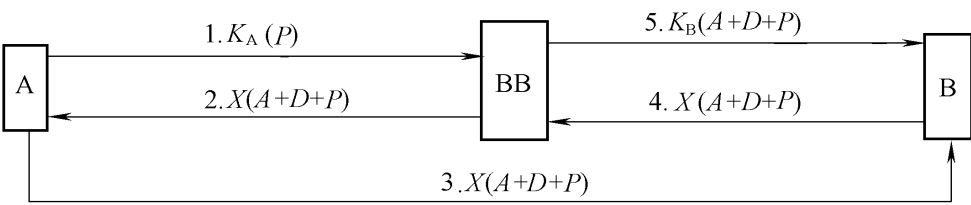


图 6 7 1 在两个陌生人 A 和 B 之间采用秘密密钥报文签名传送

2. 公开密钥的数字签名

使用秘密密钥加密法进行数字签名的关键问题是每个人都信任中心权力者, 而该中心权力者要读取所有签名过的信息。最能担当此重任的代表是政府、银行和律师。但是并不是所有的公民都非常信任这些部门的。因此如果文件签名不需要任何可信赖机构将会更好。公开密钥加密法可以满足这一要求。

如图 6 7 2 所示, 用户 A 用自己的私有密钥将明文 P 加密得到 $D_A(P)$, 再用用户 B 的公开密钥加密得到 $E_B(D_A(P))$, 将此密文传送给用户 B。用户 B 用自己的私有密钥 D_B 将此解密得到 $D_A(P)$, 并把这条信息存放在一个安全的地方, 然后用用户 A 的公开密钥 E_A 解密得到初始明文 P 。如果用户 A 后来否认曾经发送过报文 P 给 B 用户, 用户 B 只需出示 $D_A(P)$ 给法官, 法官用 E_A 来解密就能证明该条消息确实是 A 用户发送的。因为 B 用户不知道 A 的私有密钥, 只有用户 A 才能产生出该密文。

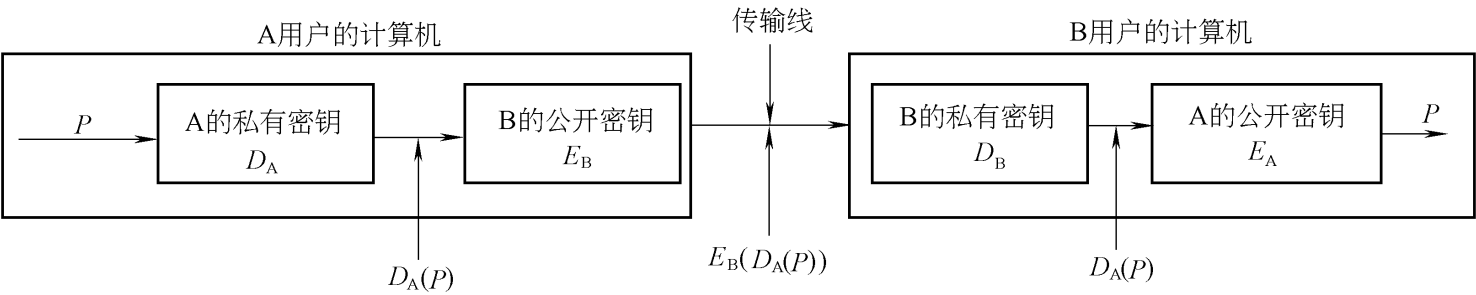


图 6 7 2 使用公开密钥加密法的数字签名

6.7.3 防火墙

随着 Internet 的飞速发展,计算机网络的资源共享进一步加强,随之而来的信息安全问题也日益突出。对网络的主要威胁有非法入侵和病毒传播,影响 Email, IP, Web 乃至整个系统的安全。现在采用的有效措施有设置防火墙、采用口令、用户标识号 ID 等。

所谓防火墙就是一个或一组系统,用来在两个或多个网络间加强访问控制。它是一个网络与其他网络之间的可控网关,通常它置于一个私有的、有确认的网络和公开的 Internet 之间。它的功能类似于大厅的警卫,目的在于把那些不受欢迎的人隔离在特定的网络之外,但又丝毫不影响正常工作。其原理可以想象成一对开关,其中一个开关用来阻止传输,另一个开关用来允许传输。比如在企业网和 Internet 网设立防火墙软件,使企业信息系统对于来自 Internet 的访问采取有选择的接收方式。它可以允许或禁止某一类具体的 IP 地址访问,也可以接收或拒绝 TCP/ IP 上的某一类具体的应用。如果在某一台 IP 主机上有高度机密的信息或危险的用户,则可以使用防火墙过滤掉从该主机发出的包。如果一个企业只是使用 Internet 的电子邮件和 WWW 服务器向外部提供信息,那么就在防火墙上设置使得只有这两类应用的数据包可以通过。虽然防火墙有很多种类型,但其主要技术有下列三种:

(1) 包过滤(Packet Filter)

该技术是在网络层中对数据包实施有选择的通过。依据系统内事先设定的过滤逻辑,检查数据流中每个数据包后,根据数据包的源地址、目的地址、所用的 TCP 端口与 TCP 链路状态等因素来确定是否允许数据包通过;

(2) 应用网关(Application Gateway)

这是建立在网络应用层上的协议过滤技术。它针对特别的网络应用服务协议,即数据过滤协议,并且能够对数据包进行分析并形成相关的报告。在实际工作中,应用网关一般由专用工作站系统来完成;

(3) 代理服务(Proxy Service)

这是设置在 Internet 防火墙网关的专用应用级代码。这种代理服务是准许网管员允许或拒绝特定的应用程序或应用的一种特定功能。包过滤技术和应用网关是通过特定的逻辑判断来决定是否允许特定的数据包通过,一旦判断条件满足,防火墙内部网络的结构和运行状态便暴露在外来用户面前,这就引入了代理服务的概念,即防火墙内外计算机系统应用层的“链接”,由两个终止于代理服务的“链接”来实现,就成功地实现了防火墙内外计算机系统的隔离。同时代理服务还可用于实施较强的数据流监控、过滤、记录和报告等功能。代理服务技术主要通过专用计算机硬件(如工作站)来承担。

防火墙的具体实现有很多形式,其产品的侧重点各有不同,在实现上都有细小的差别,但原理和目的相似。

防火墙是保障网络和信息系安全的一道重要防线,没有防火墙保护的的企业网是难以想像的。然而防火墙只是保障网络和信息系安全的一个必要条件,而不是充分条件。防火墙的主要弱点在于:防火墙无法抵抗绕过防火墙的攻击;防火墙的主要功能是防止来自外部的黑客攻击,对于来自内部的攻击则无能为力。而人们通常认为内部攻击的危害性要远远大于外部攻击的危害性。因此必须将防火墙与其它安全设施有机地组合起来,才能构成有效的网络安全防卫体系。比如当用户通过防火墙访问网络资源时必须通过一个强有力

的认证过程。目前的认证手段也在不断发展,除了传统的口令技术外,还可采用一次性许可证、智能卡等技术来实现,好的认证方式还可利用指纹、音色以及视网膜纹等生物特征来实现。

6.7.4 密码学在电子支付系统中的应用

当前金融机构所使用的最方便的识别或者确认方法是顾客拥有的东西——银行卡片,以及顾客知道的东西——个体标识号 PIN。利用卡片上所记有的帐户号和由顾客记住的 PIN 之间的相对一致性来识别顾客。对一个骗子而言,占有卡片但不知道 PIN,或者知道 PIN 而没有相应的卡片,都不足以取得进入系统的权利。

曾使用过的卡片有两种,比较早的是采用磁卡,但是伪造或者复制这样的磁卡比较容易,复制时并不需要确知卡片上记录数据的内容、格式以及是否加密,只需将卡片上的数据从一个卡转移到另一个卡即可。为了增加安全性,可在卡片上构造一些随卡片而改变的随机特性,如在卡片的内磁芯上印制两组磁线道,使得没有两张卡片是同样的。但是这就增加了读卡机的复杂度和读取的费用。目前使用的则是智能安全卡,这种卡上装有一个微处理器,使得识别和确认运算可以直接在卡片上进行,而不必在系统入口点设备的逻辑电路中进行。此外还可将少量的重要顾客帐户信息储存在卡上,提供具有相当于储蓄存折所提供的自动化的记录,是一种智能化的安全的卡片。

使用保密的 PIN 是电子支付(EFT)系统中确认顾主的最好方法。PIN 本质上是卡片持有者的一种电子签名,它在 EFT 交易中的作用与传统的金融事务中书面签名的作用相同。PIN 是由卡主记忆的,而不得用可能被他人查出的方式记录下来。当卡主着手进行一项 EFT 交易时,他用专用键盘将其 PIN 输入进 EFT 的终端。除非 EFT 系统识别出他所输入的 PIN 与这一特定的帐号(由 EFT 终端从卡片中读出)相符,否则 EFT 系统就拒绝这笔交易。这样做的目的是:如果卡片丢失或被盗,其拾者或窃者无法去使用该卡片,因为他们不知道与此相关的 PIN。同样也可以防止那些能够伪造银行卡片的人。即使他能够伪造这样一种假银行卡片,也不能使用它,因为他不知道 PIN。

为了使 PIN 起到其应有的作用,它只能为卡片所有者所知,而不应让他人知道。PIN 的保密是极其重要的,只有采取严格的安全措施才能达到。一般推荐采用的 PIN 的长度是 4 位、5 位或 6 位的十进制数,这样既考虑了使用的方便,又保证了在有限时间内不能用试凑法测出。

PIN 可以由金融机构确定,也可以由卡主选定。每种方法都各有优缺点,但现有技术能以安全的方式来进行每种方法的实施,从而使得没有人能确定卡主的 PIN。如果卡主遗忘了 PIN 的值,也有安全技术来提醒卡主回忆他自己的 PIN。

前面介绍的数据加密标准 DES 算法就可用来加密 PIN,用严格 6 位十进制数或更长的数值来串连 PIN 明文,这个值是:一个随机数或伪随机数;一个在每次交易中增加的计数;帐号中的最无意义的数字。其结果在长度上必定不超过 64 位二进制比特。用 DES 和一个密钥对其进行块加密,则产生完整的 64 位密码就作为该项交易加密的 PIN。当然密钥也需受到保护。

6.7.5 密码学在电子数据交换中的应用

电子数据交换(EDI)的兴起,带来了交易方式的革命,对惯于白纸黑字立据为证的人们来说,纸张文件的消失和电子文电的出现在提高办事效率、加强商业竞争地位等方面无疑有着巨大的吸引力,但也可能产生认证上的问题。当前如何确保交易的准确、安全和可靠,已成为开放性 EDI 系统的关键问题。对 EDI 安全保密的主要威胁是:假冒、报文排序、报文丢失、修改信息、服务的否认和抵赖、遗漏信息等等,威胁可能是偶然的,也可能是预谋的;可能是主动的,也可能是被动的。在国外商业领域,DES 算法是应用于广泛的对称密码体制;而 RSA 则是日益为商界所接受的非对称密码体制。对 EDI 的加密而言,非对称密码体制更适合于鉴别业务和密钥管理;对称密码体制具有在文电内容的完整性和满足 EDI 系统处理文电能力上较为突出。故对 EDI 系统的加密保护需要两种密码体制的优化组合。

EDI 业务的源点鉴别和电文内容的完整性都可由数字签名来实现。在数字签名中,一种密码运算产生一个密码校验和,该校验和可以验证电文内容和完整性。而源点鉴别则可以由合法源点给出的含其秘密密钥的信息来实现。

习 题

6 1 用置换盒

3	5	1	2	8	7	4
---	---	---	---	---	---	---

 把字 SECURITY 进行移位。

6 2 若已知 DES 体制中 8 个 S 盒之一的 S 盒选择压缩函数如下:

<div>列 号</div> <div>行 号</div>	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 盒的输入矢量为 $X = (x_0\ x_1\ \dots\ x_5) = (010011)$ 。试求通过选择压缩函数 S 变换后的输出矢量。

6 3 对下面的每种情况求 d , 并给出 $(e \times d) \pmod{1}$:

- (1) $p = 5, q = 11, e = 3$
- (2) $p = 3, q = 41, e = 23$
- (3) $p = 5, q = 23, e = 59$
- (4) $p = 47, q = 59, e = 17$

6 4 用公开密钥 $(e, n) = (5, 51)$ 将报文 ABE, DEAD 用 $A = 01, B = 02, \dots$, 进行加密。

6 5 用秘密密钥 $(d, n) = (13, 51)$ 将报文 4, 1, 5, 1 解密。

6 6 用公开密钥 $(e, n) = (3, 55)$ 将报文 BID HIGH 用 $01 = A, 02 = B, \dots$, 进行加密。

6 7 用秘密密钥 $(d, n) = (5, 51)$ 将报文 4, 20, 1, 4, 20, 5, 4 解密。

6 8 用 $(d_B, n_B) = (7, 39)$ 和 $(e_A, n_A) = (5, 21)$ 签署报文 ED。

6 9 用 $(d_A, n_A) = (5, 21)$ 和 $(e_B, n_B) = (5, 51)$ 验证签名的数值 17, 1 是发送者(N, A)的字首。

附录：符号及含义

$A = \{ a_1, a_2, \dots, a_n \}$	包含 n 个元素的符号集
$B = \{ b_1, b_2, \dots, b_m \}$	包含 m 个元素的符号集
$X = \{ x_1, x_2, \dots, x_q \}$	X 为输入随机变量, 或信源随机变量;
$Y = \{ y_1, y_2, \dots, y_Q \}$	Y 为输出随机变量, 或信宿随机变量;
$S = \{ s_1, s_2, \dots, s_Q \}$	包含 Q 个状态的状态集
$X = (X_1 X_2 \dots X_L \dots X_L)$	L 长输入随机序列矢量,
$p(X = x_i)$	$A_i = A, X = A^L$
$p(X = x_i / Y = y_j) = p(x_i / y_j)$	输入符号概率, 变量 X 取 x_i 的先验概率;
$p(Y = y_j / X = x_i) = p(y_j / x_i) = p_{ij},$ $i = 1, \dots, q, \text{ 以及 } j = 1, \dots, Q$	条件概率, 或变量 X 的后验概率;
$p(s_j / s_i) = p_{ij}$	条件概率, 或离散无记忆信道转移概率;
$P = \begin{bmatrix} p_{11} & \dots & p_{1Q} \\ \dots & & \dots \\ p_{Q1} & \dots & p_{QQ} \end{bmatrix}$	从状态 i 转移到状态 j 的状态转移概率
$p(x_j / s_i)$	转移概率矩阵
$p(Y=0 / X=1) = p(Y=1 / X=0) = p$ $p(Y=1 / X=1) = p(Y=0 / X=0) = 1 - p$	在状态 i 时出现符号 x_j 的符号条件概率
$n(t)$	BSC 信道转移概率,
$H(X)$	零均值、方差为 σ^2 的高斯随机变量
$H(X) = H(X^L)$	加性噪声过程的一个样本函数
$H_L(X)$	输入符号的信息熵
$H(X / Y), H(Y / X)$	离散信源 L 长序列熵
$I(X; Y)$	离散信源 L 长序列的平均符号熵
$W^{(n)} = [W_1^{(n)} \quad W_2^{(n)} \quad \dots \quad W_r^{(n)}]$	条件熵
	输出 Y 对输入 X 提供的平均互信息
	n 时刻概率分布矢量, 其中 $W_j^{(n)} = p\{X_n = s_j\}$
	信息效率, 编码效率
	冗余度, 码的剩余度
$\overline{K_L}$	编码后码字的平均码长 (m 进制)
\overline{K}	编码后对应信源符号的平均码长 (比特),
	$\overline{K} = \frac{\overline{K_L}}{L} \log m$

R	码率, 每二进码元携带的信息量, 即信息传输率(效率)
$d(x, y)$	失真函数
D	平均失真
$R(D)$	信息率失真函数
C	$I(X; Y)$ 的最大值即信道容量
E_b/N_0	比特信噪比(能噪比)
X^N	N 维矢量空间
$m = (m_1, m_2, \dots, m_K)$	消息组
$c = (c_1, c_2, \dots, c_N) \quad X^N$	码字, 其中码元 $c_1, \dots, c_N \quad X = \{x_0, x_1, \dots, x_{q-1}\}$
$r = (r_1, r_2, \dots, r_N) \quad Y^N$	接收码
P_e	差错概率
$\overline{P_e}$	平均差错概率
d_{\min}	码的最小距离
t	纠错能力
$(c_5 \ c_4 \ c_3 \ c_2 \ c_1) \mathbf{G}$	矩阵运算: 左、上先
$c_5 x^5 + c_4 x^4 + c_3 x^3 + c_2 x^2 + c_1 x$	多项式运算, 高次先
硬件电路图	高位先

部分习题参考答案

- 2 1 (1) 4.17 比特; (2) 5.17 比特; (3) 4.337 比特/事件; (4) 3.274 比特/事件;
(5) 1.7105 比特
- 2 2 (1) 1 比特; (2) 0.08 比特; (3) 2 比特
- 2 3 1.42 比特
- 2 4 114.3 比特/消息; 1.905 比特/符号
- 2 5 4.17 比特; 1.24 比特
- 2 6 (1) 87.81 比特; (2) 1.95 比特/符号
- 2 7 1 比特, 2 比特, 3 比特.
- 2 8 (1) $I(\text{划}) = 2$ 比特, $I(\text{点}) = 0.42$ 比特; (2) 0.81 比特/符号
- 2 9 (1) 0.92 比特; (2) 0.86 比特; (3) 0.94 比特; (4) 0.91 比特
- 2 10 (1) $H(\text{colour}) = 1.24$ 比特
(2) $H(\text{colour, number}) = H(\text{number}) = \log_2 38 = 5.25$ 比特
(3) $H(\text{number/ colour}) = H(\text{colour, number}) - H(\text{colour}) = 4.01$ 比特
- 2 11 (1) $H(XY) = 2.3$ 比特/符号; (2) $H(Y) = 1.58$ 比特/符号;
(3) $H(X/Y) = 0.72$ 比特/符号
- 2 12 (1) $H(X) = 1$ 比特; $H(Y) = 1$ 比特; $H(Z) = 0.54$ 比特; $H(XZ) = 1.41$ 比特;
 $H(YZ) = 1.41$ 比特; $H(XYZ) = 1.81$ 比特
(2) $H(X/Y) = H(Y/X) = 0.81$ 比特; $H(X/Z) = 0.87$ 比特; $H(Z/X) = 0.41$ 比特;
 $H(Y/Z) = 0.87$ 比特; $H(Z/Y) = 0.41$ 比特; $H(X/YZ) = H(Y/XZ) = 0.4$ 比特; $H(Z/XY) = 0$
(3) $I(X; Y) = 0.19$ 比特; $I(X; Z) = I(Y; Z) = 0.13$ 比特; $I(X; Y/Z) = 0.47$ 比特;
 $I(Y; Z/X) = I(X; Z/Y) = 0.41$ 比特
- 2 13 3.415 比特/符号
- 2 16 (1) 0.8813 比特/符号; (2) 0.513 比特/符号
- 2 20 (1) 2.58 比特; (2) $H(X) = 3.32$ 比特
- 2 21 (1) 2.58 比特; (2) 4, 3
- 2 24 (1) 0.81 比特/符号; (2) $41 + 1.59m$; (3) 81 比特/序列
- 2 25 $p(0) = 0.4$, $p(1) = 0.6$

$$p_1 = 10/25$$

2 26 解得 $p_2 = 9/25$ 状态图如右所示:

$$p_3 = 6/25$$

2 27 解得 $p(00) = 5/14$
 $p(11) = 5/14$ 状态图如右所示:
 $p(01) = 2/14$
 $p(10) = 2/14$

2 28 0.69 比特/符号

2 31 (1) 联合熵 $H(X_1 X_2 X_3) = 3.968$ 比特
 平均符号熵 $H_L(X) = 1.323$ 比特/符号

(2) 极限熵 1.25 比特/符号

(3) $H_0 = \log n = \log 3 = 1.58$ 比特/符号, $\eta = 1 - \frac{H}{H_0} = 1 - (H/H_0) = 0.21$
 $H_1 = 1.4137$ 比特/符号, $\eta = 1 - 1.25/1.4137 = 0.115$
 $H_2 = H = 1.25$ 比特/符号, $\eta = 0$

2 32 (1) $p(0) = p(1) = p(2) = 1/3$

(2) $(1-p)\log(1/(1-p)) + p\log(2/p)$

(3) 1.58 比特/符号

(4) $p = 2/3$ 时 $\max H = 1.58$ 比特/符号; $p = 0$ 时 $H = 0$; $p = 1$ 时 $H = 1$

3 1 (1) C_1, C_2, C_3, C_6 ; (2) C_1, C_3, C_6 ;

(3) $H(X) = 2$ 比特/符号, 66.7%, 94.1%, 94.1%, 80%

3 2 (1) 200 bit/s; (2) 198.55 bit/s

3 3 (1) 200 bit/s; (2) 198.55 bit/s

3 5 (1) 1.98 比特/符号; (2) $p(0) = 0.8, p(1) = 0.2$; (3) $\eta = 0.66$;

(4) 0, 10, 110, 1110, 11110, 111110, 1111110, 1111111; (5) $\eta = 1$

3 8 0, 10, 11, 20, 21, 22; $\eta = 0.93$

3 13 (1) $a, ba, bb, ca, cba, cbb, cca, ccb, ccc$ 0.89

或 $b, c, ab, ac, aab, aac, aaaa, aaab, aaac$ 0.95

(2) $a, ba, bb, caa, cab, cba, cbb, cbca, cbcb$ 0.81

或 $a, ca, cb, baa, bab, bba, bbb, bca, bcb$ 0.84

或 $c, ba, bb, aa, aba, abb, aca, acba, acbb$ 0.87

4 1 0 1

1 0

4 2 $D_{\max} = 3/4, R(3/4) = 0, D_{\min} = 0, R(0) = 2$ 比特/符号

4 3 (1) $D = q(1-p)$

(2) $\max R(D) = H(U) = -p\log p - (1-p)\log(1-p)$; $q = 0$ 时 $D = 0$ 。

(3) $\min R(D) = 0$; $q = 1$ 时 $D = 1 - p$

4 7 7 比特

5 1 (1) $H(X) = 0.815$ 比特/符号, $H(X/Y) = 0.749$ 比特/符号, $H(Y/X) = 0.91$ 比特/符号,
 $I(X; Y) = 0.066$ 比特/符号;

(2) 0.082 比特/符号, $p(x) = 0.5$

5.2 (1) $H(Y) = \frac{3}{2} - \frac{1+a}{4} \log(1+a) - \frac{1-a}{4} \log(1-a) \text{ bit/s}$

(2) $H(Y/X) = \frac{3}{2} - \frac{a}{2} \text{ bit/s}$

(3) $C = 0.16 \text{ bit/s}$

5.3 $C = 920 \text{ bit/s}$

5.4 $C = 19.5 \text{ Mbit/s}$

5.5 (1) $C = 1$ 比特/信道符号; (2) $\overline{P_e} = 0.5$

5.6 (1) 1.46 比特/符号; (2) 1.18 比特/符号; (3) 0.8; (4) 0.73; (5) 0.73;
(6) 较差; (7) 1.58 比特/符号, 1.3 比特/符号

5.7 $W = 3 \text{ MHz}$

5.8 (1) $C = 3.46 \text{ Mbit/s}$; (2) $W = 1.34 \text{ MHz}$; (3) $\text{SNR} = 120$

5.9 所有 4 重向量空间: $\{1,0,0,0\}, \{0,1,0,0\}, \{0,0,1,0\}, \{0,0,0,1\}, \{0,0,0,0\},$
 $\{1,1,1,1\}, \{1,1,0,0\}, \{1,0,1,0\}, \{1,0,0,1\}, \{0,1,1,0\}, \{0,1,0,1\}, \{0,0,1,1\},$
 $\{1,1,1,0\}, \{1,1,0,1\}, \{0,1,1,1\}, \{1,0,1,1\}$

选一个 2 维子空间: $\{1,0,0,0\}, \{0,1,0,0\}$; 对偶子空间: $\{0,0,1,0\}, \{0,0,0,1\}$

5.11
$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}, H = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, d_{\min} = 4$$

5.12
$$G = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix}, H = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

5.13 发码为: 0010110, 0111010, 1100010

5.16 (1) 若选 $g(x) = x^4 + x^2 + x + 1$, 所有码字除 0000000 外具有循环性:
0010111, 0101110, 1011100, 0111001, 1110010, 1100101, 1001011
若选 $g(x) = x^4 + x^3 + x^2 + 1$, 所有码字除 0000000 外具有循环性:
0011101, 0111010, 1110100, 1101001, 1010011, 0100111, 1001110

(2)
$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

6.1 CRISEY TU

6.2 行 = 1, 列 = 9, Y = (0110)

6.3 (1) 27; (2) 7; (3) 3; (4) 157.

6.4 1, 32, 14, 4, 14, 1, 4

6.5 4, 1, 20, 1 或 DATA

6.6 8, 14, 9, 17, 14, 13, 17

6.7 4, 5, 1, 4, 5, 14, 4 或 DEAD END

6.8 8, 4; 8, 16

6.9 5, 1; 14, 1

参 考 文 献

- [1] 周炯磐 . 信息理论基础 . 北京:人民邮电出版社,1983
- [2] 周炯磐,丁晓明 . 信源编码原理 . 北京:人民邮电出版社,1996
- [3] 吴伟陵 . 信息处理与编码 . 北京:人民邮电出版社,1999
- [4] 吴伯修,祝宗泰,钱霖君 . 信息论与编码 . 南京:东南大学出版社,1991
- [5] 周荫清 . 信息理论基础 . 北京:北京航空航天大学出版社,1993
- [6] S .William .Cryptography and network security: principles and practice, Prentice YHall, Inc . 1999