信息论与编码理论

马啸

2024 年 10 月 9 日

目录

第一章

预备知识

§1.1 信息论的基本问题

所谓通信，是从“此处”向“彼处”，或者从“此时”向“彼时”传递消息。Shannon [1] 指出数字通信系统分为信源、编码器、信道、译码器、信宿五个模块，其中的编译码模块又可以细分为信源编译码与信道编译码模块等，见图 1.1。这些模块分述如下。

- “信源”产生语音、文字、图片、视频等消息，可以视作一个随机过程。特别地，“信宿”在接收前是不知道这些消息的。
- “编码器”的任务就是将消息变换成适合信道的传输信号，而“译码器”是从接收信号恢复消息。可以进一步分为两个部分：
  - 信源编码：纷繁复杂的信源总可以用二进制数位来精确或近似表征。
  - 信道编码：在发送信号中加入某种约束，使得接收端可以检错或者纠错。信道编码是物理层的关键技术之一。
- 发送信号经过“信道”后（按照某概率法则）随机转换为输出信号。

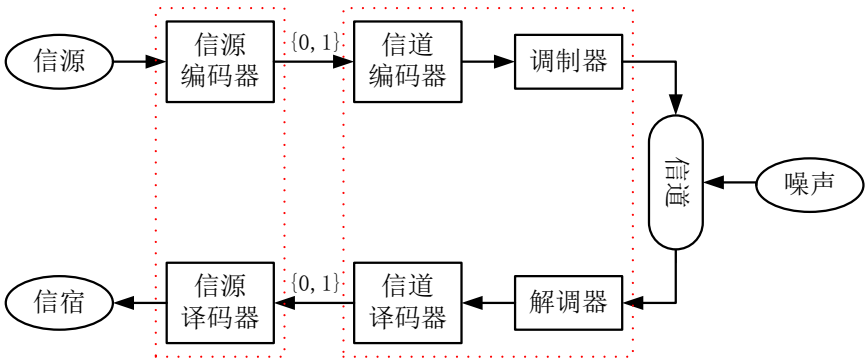


图 1.1: 数字通信系统框架

在理想情况下，信源编码与信道编码的分离并不会带来极限性能的损失，反而有很多优点。信息论的基本问题是研究通信系统的性能极限。信源编码解决的主要问题是如何用二进制序列高效地、精确或近似地表征消

息，而信道编码解决的主要问题是如何高效地、可靠地传输这些二进制序列。总的来说，信息论研究，在长时间平均意义下，可靠传输一个信源符号需要使用信道的最少次数是多少 [2]。

§1.2

概率论回顾

1.2.1 随机模型三要素

概率论是研究随机现象并揭示其规律的学科。为此，我们需要建立一个称之为随机试验的概率模型。一个随机试验（或称随机模型）由下面三个要素组成：

1. 样本空间（sample space）是一个集合，由所有可能的试验结果构成，通常记作 Ω。样本空间可以是有限的，也可以是无限的。样本点可以是非数值类的，也可以是数值类的，此时样本点可以是标量，也可以是向量，甚至无穷序列。

2. 事件（event）是 Ω 的子集，而所有事件构成的类 $\mathcal{F}$ 须满足

2.1 $\Omega \in \mathcal{F}$（必然事件）；

2.2 $A \in \mathcal{F} \Rightarrow \bar{A} \in \mathcal{F}$（补运算封闭），这里“$\bar{A}$”表示 A 的补集；

2.3 $A_i \in \mathcal{F} \Rightarrow \bigcup_i A_i \in \mathcal{F}$（可列并运算封闭）。

由此可推出不可能事件 $\varnothing \in \mathcal{F}$，可列交运算封闭，即事件的“加（并）、减（补）、乘（交）”运算封闭。换句话讲，事件类是根据研究问题而定义的子集类。与集合论里的术语稍有不同的是，我们用和事件与积事件分别表示集合的并与交。当两个事件对应的子集不相交时候，我们说两个事件是不相容的。

3. 概率是一个映射 $P : \mathcal{F} \to (-\infty, +\infty)$，满足

3.1 $P(\Omega) = 1$（规范性）；

3.2 $P(A) \geqslant 0$（非负性）；

3.3 $P(\biguplus_i A_i) = \sum_i P(A_i)$（可列可加性，$\biguplus_i$ 表示两两不相容的 A_i 的并）。

由 3.2、3.3 可以推出 $P(\varnothing) = 0$。此处，P 类似归一化的“计数、长度、面积、体积”等测度。

【例题 1.1】 抛一枚硬币，根据实际情况，或可选择如下模型。

模型 1

（1）样本空间：$\Omega = \{$正，反$\}$。

（2）事件类：$\mathcal{F} = \{\varnothing, \{$正$\}, \{$反$\}, \{$正，反$\}\}$。

（3）概率：$P(\{$正$\}) = \frac{1}{2}$，$P(\{$反$\}) = \frac{1}{2}$。

模型 2

（1）样本空间、事件类同模型 1。

（2）概率：$P(\{$正$\}) = \frac{1}{4}$，$P(\{$反$\}) = \frac{3}{4}$。

模型 3

（1）样本空间：$\Omega = \{$正，反，立$\}$。

（2）事件类、概率从略。

【例题 1.2】 掷一均匀骰子

（1）样本空间 $\Omega = \{1, 2, 3, 4, 5, 6\}$。

（2）取 $A = \{1, 3, 5\}$。$\{\Omega, A, \varnothing\}$ 是事件类吗？

（3）$\{\Omega, A, \bar{A}, \varnothing\}$ 是事件类吗？

（4）$\mathcal{F} = 2^{\Omega}$ 是事件类吗？

（5）对于有限样本空间，我们通常取所有子集构成的类为事件类，即任意一个子集都视作事件。若有限样本空间大小为 $|\Omega|$，问共有多少子集？

（6）若样本空间为有限样本空间，则概率的定义通常以定义单点集的概率为基本方式。若骰子是均匀的，则定义

$$P(\{1\}) = P(\{2\}) = \cdots = P(\{6\}) = \frac{1}{6}。$$

注：我们可以用 $P(1)$ 简单记概率 $P(\{1\})$，不过要注意概率 P 是一个广义的函数，其定义域是事件类，而自变量是事件。我们也可以用 $\Pr\{A\}$ 来表示事件 A 的概率。

1.2.2 概率的基本运算律

在概率中，我们有以下运算法则：

1. **加法**：对于事件 A，B，

- $P(A\cup B)=P(A)+P(B)-P(AB)$；$P(A\cup B)\le P(A)+P(B)$。
- 若 A 与 B 不相容，则 $P(A\cup B)=P(A)+P(B)$（分类）。
- **容斥定理**：若事件集 A_1，$\cdots$，A_n 为有限集，则有

$$P\{\bigcup_{i=1}^{n}A_i\}=\sum_{i=1}^{n}P\{A_i\}-\sum_{1\le i<j\le n}P\{A_i\cap A_j\}+\sum_{1\le i<j<k\le n}P\{A_i\cap A_j\cap A_k\}-\cdots+(-1)^{n-1}P\{A_1\cap\cdots\cap A_n\}。$$

2. **乘法**：

- 条件概率定义：设 A 与 B 为样本空间 Ω 中的两个事件，其中 $P(A)>0$。那么在事件 A 发生的条件下，事件 B 发生的**条件概率**为：

$$P(B|A)=\frac{P(AB)}{P(A)};$$

- $P(AB)=P(B|A)P(A)$；若 A 与 B **相互独立**，则 $P(AB)=P(A)P(B)$；
- 链式法则：$P(A_1A_2\dots A_n)=P(A_1)P(A_2|A_1)\cdots P(A_n|A_1A_2...A_{n-1})$；
- **多事件独立的定义**：设有 A_1，A_2，$\dots$，$A_n(n\ge 2)$ 个事件，如果其中任意 k $(2\le k\le n)$ 个事件的积事件的概率都等于各事件概率之积，即

$$P(A_{i_1}A_{i_2}\cdots A_{i_k})=\prod_{j=1}^{k}P(A_{i_j}),$$

对于所有 $1\le i_1<i_2<\cdots<i_k\le n$ 均成立，则称它们相互独立。

3. **减法**：

- $P(\bar{A})=1-P(A)$；
- $P(A-B)=P(A)-P(AB)$。

4. **全概率公式**：若 B_i 是样本空间 Ω 的划分，则有 $P(A)=\sum\limits_i P(B_i)P(A|B_i)$（分类、分步）。

5. **贝叶斯公式**：$P(B_i|A)=\frac{P(B_i)P(A|B_i)}{P(A)}=\frac{P(B_i)P(A|B_i)}{\sum\limits_j P(B_j)P(A|B_j)}$。

贝叶斯分析在工程中有广泛的应用背景。我们可以把 B_i 视作引起 A 发生的原因，$P(B_i)$ 称之为先验概率，而 $P(B_i|A)$ 是后验概率，反映事件 A 对于 B_i 概率的影响。条件概率 $P(A|B_i)$ 通常称为似然概率，是“由因及果”，而贝叶斯公式计算的后验概率是“由果及因”，是概率推断的重要工具。在实际应用中，我们通常可以借助图模型来分析复杂的概率问题，概率树可以用来描述“分类”与“分步”：“分类”用“分支”，“分步”用“分节”。从根节点（全集，或者说“必然事件”）开始做划分，随着树的生长，划分得越来越细。每个子节点是父节点的子事件，所有子节点划分（不重不漏）它们的父节点。根节点的概率是 1。每条分支有个权重，对应给定父节点条件下相应子节点发生的概率。因此，从一个节点出发的各个分支的权重之和为 1。某个节点的概率是从根节点到达该节点的路径的概率，而该概率是边概率的连乘积。一个子树的概率是该子树所有叶子对应的概率的和，这个子树的概率是等于该子树的根节点的概率的。在概率树中，一个节点的概率既可以由其父亲节点通过一次乘法算得（由上至下），也可以由其儿子节点通过加法算得（由下至上）。有些问题在题目中描述比较直接，有些需要自己提炼，整理出“概率树”，不同的“概率树”对应不同的概率模型。

【例题 1.3】 老师准备从五个学生 A、B、C、D、E 中选三个代表班级参加随机过程课程竞赛。

（1）A 被选中的概率是多少?

（2）老师公布了其中一个参赛同学:E。此时 A 被选中的概率是多少?

（3）在老师未公布之前，A 向助教打听自己有无入选。助教按照纪律要求不能告知 A 是否入选，但告知 E 入选了。此时，A 入选的概率是多少?

1.2.3 随机变量及其分布函数

【定义 1.4】 随机变量 X 不是（自）变量，而是样本空间 Ω 到实数集 $\mathbb{R}$ 的映射，即 $X:\Omega\to\mathbb{R}$，满足：对于任意实数，$\{\omega\in\Omega|X(\omega)\leq x\}$ 是一个事件。为简便起见，我们把该事件记为 $\{X\leq x\}$。

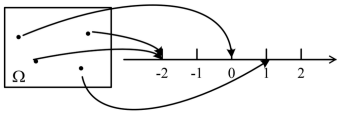


图 1.2: 随机变量

例如，抛一枚硬币，概率模型如例 1.1 模型 1。我们可以定义随机变量 $X(\text{正})=1$，$X(\text{反})=0$。引入随机变量后，样本点就有了数值映像。这有助于我们利用数学分析的工具揭示一些随机现象的统计规律。但需要注意的是，不同的样本点可以映射到相同的数值。

设 X 是一个随机变量。由于 $\{X\leq x\}$ 是事件，我们可以定义 $F_X(x)=P(X\leq x)$，$x\in\mathbb{R}$，称之为随机变量 X 的累积分布函数（cumulative distribution function，CDF）。当 x 在 $(-\infty,\infty)$ 变化时，累积分布函数反映了随机变量取值在实数轴上的分布情况。如果 X 取值有限或者可列，则称为离散型随机变量，我们可以定义概率质量函数（probability mass function，PMF） $P_X(x)=P(X=x)$，$x\in\mathcal{X}$（$\mathbb{R}$ 的有限子集或可列子集）满足 $P_X(x_i)\geqslant 0$ 且 $\sum_i P_X(x_i)=1$。如果 $F_X(x)$ 连续且几乎处处可微，我们称 X 为连续型随机变量，并定义概率密度函数（probability density function，PDF）$f_X(x)$，$x\in\mathcal{X}$（$\mathbb{R}$ 上的区间 $\mathcal{X}$）满足 $f_X(x)\geqslant 0$ 和 $\int_{\mathbb{R}} f_X(x)\mathrm{d}x=1$。图1.3、图1.4与图1.5分别给出了离散型随机变量的 PMF、连续型随机变量的 PDF 与一般随机变量的 CDF 的示意图。

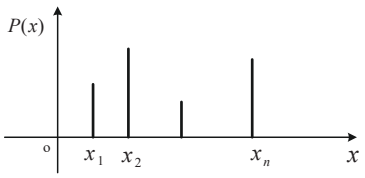


图 1.3: 离散型随机变量的 PMF (又称分布律)，“谱线”

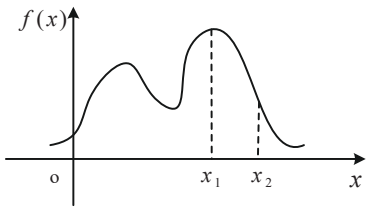


图 1.4: 连续型随机变量的 PDF，“谱密度”

一个随机变量 X 的累积分布函数 $F_X(x)\triangleq P(X\leqslant x)$，$x\in\mathbb{R}$ 满足如下性质：

1. 对离散型随机变量 X，$F_X(x)=\sum\limits_{x_i\leqslant x} P_X(x_i)$ 是阶梯函数，其中 $P_X(x)$，$x\in\mathcal{X}$ 是 X 的概率质量函数。
2. 对连续型随机变量 X，$F_X(x)=\int_{y\leqslant x} f_X(y)\mathrm{d}y$ 是连续函数，其中 $f_X(x)$，$x\in\mathcal{X}$ 是 X 的概率密度函数。

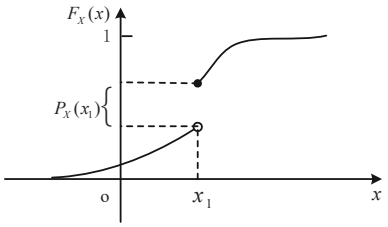


图 1.5: 随机变量的 CDF

3. $F_X(x)$ 是单调递增（未必严格单调）函数，且

$$\lim_{x\to-\infty} F_X(x)=0,\qquad \lim_{x\to+\infty} F_X(x)=1。$$

4. $F_X(x)$ 的间断点至多可列个；若在 x_1 处有跳跃，则跳跃的高度恰好是概率 $P_X(x_1)$。

【例题 1.5】 常见的离散随机变量的概率质量函数分述如下，其中 $0<p<1$，$\lambda>0$ 是参数。

- 两点分布：随机变量 X 取 0 与 1 两个值，其分布律是 $P_X(1)=p$，$P_X(0)=1-p$。
- 二项分布：设 n 是正整数，记 $B(n;p)$ 为二项分布，其分布律是 $P_X(m)=\binom{n}{m}p^m(1-p)^{n-m}$，$0\le m\le n$。
- 几何分布：随机变量 X 取正整数，其分布律是 $P_X(m)=p(1-p)^{m-1}$，$m\ge 1$。
- 泊松分布：随机变量 X 取非负整数，其分布律是 $P_X(n)=\frac{\lambda^n\exp(-\lambda)}{n!}$，$n\ge 0$。

常见的连续随机变量的概率密度函数分述如下，其中 $\lambda>0$，$\sigma>0$，μ 是参数。

- 指数分布：随机变量 X 的密度函数是 $f_X(x)=\lambda\exp(-\lambda x)$，$x\ge 0$。
- 高斯分布：随机变量 X 的密度函数是 $f_X(x)=\frac{1}{\sqrt{2\pi}\sigma}\exp(\frac{-(x-\mu)^2}{2\sigma^2})$，$x\in\mathbb{R}$。
- 均匀分布：给定 $a>0$。随机变量 X 的密度函数是 $f_X(x)=\frac{1}{a}$，$0\le x\le a$。

注：密度函数的定义域与累积分布函数的定义域一致，都是整个实数轴。在给出密度函数的时候，通常只给出函数的支撑集（support set），即函数值不为 0 的自变量所构成的子集合。在支撑集之外，函数值默认为 0。

以上讨论的只限于一个随机变量的情况，如果随机试验的结果需要同时考虑两个或两个以上的随机变量的情况，则可以定义复随机变量，随机向量，随机序列，随机过程等。

我们称 $F_{X^n}(x^n)=P(X_1\le x_1，X_2\le x_2，\cdots，X_n\le x_n)$ 为 n 维随机向量的联合分布函数（joint distribution function）。它表示事件 $\{X_1\le x_1\}$，$\{X_2\le x_2\}$，$\cdots$，$\{X_n\le x_n\}$ 同时出现的概率。联合分布函数亦称多维分布函数。一般地，多维随机变量（包括随机序列、随机过程）可借助联合分布 $F_{X^n}(x^n)=P\left(X_1\leqslant x_1，X_2\leqslant x_2，\cdots，X_n\leqslant x_n\right)$ 来刻画，其中 x^n 表示 $(x_1，x_2，\cdots，x_n)$。图1.6给出了分布函数的示意图，即试验结果对应的随机变量取值落在阴影部分的概率，而这个阴影部分随着 $(x，y)$ 的变化可以“扫描”整个平面。

对于二维离散随机变量 $(X，Y)$，我们有以下分布律：

- 联合分布律：$P_{X，Y}(x，y)$，$x\in\mathcal{X}$，$y\in\mathcal{Y}$。
- 条件分布律：$P_{X|Y}(x|y)=\frac{P_{X，Y}(x，y)}{P_Y(y)}$，$P_{Y|X}(y|x)=\frac{P_{X，Y}(x，y)}{P_X(x)}$。
- 边缘分布律：$P_X(x)=\sum\limits_y P_{X，Y}(x，y)$，$P_Y(y)=\sum\limits_x P_{X，Y}(x，y)$。

【定义 1.6】 设 X 是随机变量，而 $\mathbf{g}(x)$ 是一个普通函数，则 $Y=\mathbf{g}(X)$ 也是一个随机变量，即

$$Y:\Omega\xrightarrow{X}\mathbb{R}\xrightarrow{g}\mathbb{R}$$

【例题 1.7】 设 X 是随机变量，则 $Y=X^2$，$Z=P_X(X)$，$L=-\log P_X(X)$ 均是随机变量。他们的分布律如下所示。注：随机变量“被函数”之后，“谱线”的数目可能减少，高度可能增加。

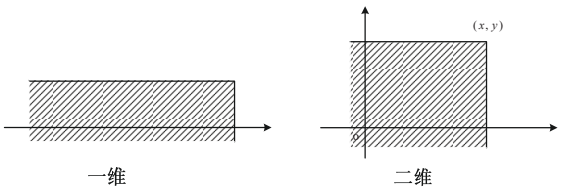


图 1.6: 分布函数

X	-1	0	1	2
$P_X(x)$	0.2	0.3	0.2	0.3

$Y=X^2$	0	1	4
$P_Y(y)$	0.3	0.4	0.3

$Z=P_X(X)$	0.2	0.3
$P_Z(z)$	0.4	0.6

$L=-\log P_X(X)$	2.32	1.74
$P_L(l)$	0.4	0.6

1.2.4 数字特征

【定义 1.8】 设离散型随机变量 X 的概率质量函数为 $P_X(x_k) \triangleq p_k$，$k=1$，2，$\cdots$。若级数 $\sum\limits_{k=1}^{\infty} x_k p_k$ 绝对收敛，则 X 的数学期望定义为：

$$\mathbf{E}(X)=\sum_{k=1}^{\infty} x_k p_k。$$

设连续型随机变量的概率密度函数为 $f(x)$，若积分 $\int_{-\infty}^{\infty} xf(x)\mathrm{d}x$ 绝对收敛，则 X 的数学期望定义为：

$$\mathbf{E}(X)=\int_{-\infty}^{\infty} xf(x)\mathrm{d}x。$$

而一般随机变量 X 的数学期望若存在的话，可以表示为以下的 Stieltjes（斯蒂尔杰斯）积分：

$$\mathbf{E}(X)=\int_{-\infty}^{\infty} x\mathrm{d}F_X(x)。$$

其中，$F_X(x)$ 为 X 的累积分布函数。

随机变量的期望 $\overline{X}=\mathbf{E}(X)$，有时也记作 $\mathbf{E}[X]$，通常被认为是一个随机变量的一个“典型值”。我们有如下命题。

【命题 1.9】 对于任意非负的随机变量 X，$\mathbf{E}[X]=\int F_X^c(y)dy$，其中，$F_X^c(x)=1-F_X(x)=P(X>x)$。

证明： 证明从略。（见拓展阅读材料） □

【定义 1.10】 设 X 是一个随机变量，如果 $D(X) \triangleq \mathbf{E}\left([X-\mathbf{E}(X)]^2\right)$ 存在，则称其为 X 的**方差**。

我们有如下命题。

【命题 1.11】 设 X，Y 是随机变量，$z=g(x)$ 是一个普通函数。则当所涉及的数学期望存在时，我们有

（1）$\mathbf{E}(X+Y)=\mathbf{E}(X)+\mathbf{E}(Y)$（不管 X 与 Y 是否独立）；

（2）若 X 与 Y 独立，则 $\mathbf{E}(XY)=\mathbf{E}(X)\mathbf{E}(Y)$；

（3）若 X 与 Y 独立，则 $D(X+Y)=D(X)+D(Y)$。

（4）$\mathbf{E}(Y)=\mathbf{E}(\mathbf{E}(Y|X))$；

（5）$\mathbf{E}(Z)=\mathbf{E}(g(X))$。

证明： 证明从略。 □

注：给定 $X=x$，$\mathbf{E}(Y|X=x)$ 是一个数，而 $\mathbf{E}(Y|X)$ 可以看作是随机变量 X 的函数。随机变量的函数的期望公式不只是简单地等量替换，要注意区分期望所对应的变量。

【例题 1.12】 概率模型中的事件 A 可以定义一个二元随机变量 $\mathbb{I}_A$，称之为 A 的示性变量，满足 $\mathbb{I}_A(\omega)=1$，$\omega \in A$，而 $\mathbb{I}_A(\omega)=0$，$\omega \notin A$。反之，一个二元随机变量必定是某个事件 A 的示性变量。我们有

$$P_{\mathbb{I}_A}(0)=1-P(A)；\ P_{\mathbb{I}_A}(1)=P(A)。$$

$$\mathbf{E}[\mathbb{I}_A] = P(A)；\ \sigma_{\mathbb{I}_A} = \sqrt{P(A)(1-P(A))}。$$

注：概率是期望，期望是积分；积分是期望，期望是概率。

【例题 1.13】 期末某校评优，甲、乙、丙、丁、戊 5 名同学分别获得“德、智、体、美、劳”单项奖状。不巧的是，辅导员发放奖状时完全随机放乱了。请问，5 名同学中平均意义上有几人拿到了与自己匹配的奖状？

1.2.5 尾部概率

在实际应用中，一方面，事件的概率未必有简易的计算方法；另一方面，很多情况下非平凡的概率界也可以用来指导系统的设计。下面介绍一些重要的不等式。

1. Markov（马尔科夫）不等式（若均值有限，则尾部概率线性递减趋于 0）

设非负随机变量 X 的期望 $\mathbf{E}(X)$ 的值有限，则对于 $y>0$， $P(X \geq y) \leq \frac{\mathbf{E}(X)}{y}$。

证明:

$$\mathbf{E}(X) = \int_0^{+\infty} x \mathrm{d}F_X \geqslant \int_y^{+\infty} x \mathrm{d}F_X \geqslant y \int_y^{+\infty} \mathrm{d}F_X = yP(X \geqslant y)。$$

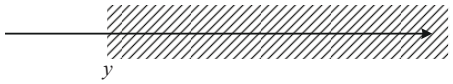


图 1.7: Markov（马尔科夫）不等式

□

2. Chebyshev（切比雪夫）不等式（若方差有限，则尾部概率平方递减趋于 0）

对于任意 $\delta > 0$，$P(|X - \mathbf{E}(X)| \geqslant \delta) \leqslant \frac{\sigma_X^2}{\delta^2}$，其中 σ_X^2 表示 X 的方差。

证明: 对 $(X - \mathbf{E}(X))^2$ 应用 Markov 不等式，

$$\begin{aligned} & P(|X - \mathbf{E}(X)| \geqslant \delta) \\ = \ & P(|X - \mathbf{E}(X)|^2 \geqslant \delta^2) \\ \leqslant \ & \frac{\mathbf{E}(X - \mathbf{E}(X))^2}{\delta^2} = \frac{\sigma_X^2}{\delta^2}。 \end{aligned}$$

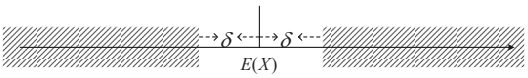


图 1.8: Chebyshev（切比雪夫）不等式

□

3. Chernoff（切诺夫）界（若矩生成函数有限，则尾部概率指数递减趋于 0）

$$P(X \geqslant y) \leqslant \mathbf{E}(e^{sX})e^{-sy}， \quad 对所有， \ s \geqslant 0。$$

证明: 对 e^{sX} 应用 Markov 不等式，

$$P(X \geqslant y) = P\left(e^{sX} \geqslant e^{sy}\right) \leqslant \frac{\mathbf{E}(e^{sX})}{e^{sy}}， \ s \geqslant 0。$$

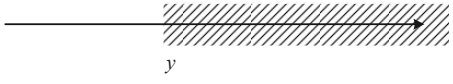


图 1.9: Chernoff（切诺夫）界

□

类似地，$P(X \leqslant y) \leqslant \mathbf{E}(e^{sX})e^{-sy}$， 对所有 $s \leqslant 0$。

上述几个不等式也可以借助积分的性质（非负函数的积分非负）来证明，如图 1.10 所示。为此，对于 $y > 0$，定义示性函数（indicator function）

$$Z(\omega) = \begin{cases} 1, & X(\omega) \geqslant y \\ 0, & \text{其他} \end{cases}$$

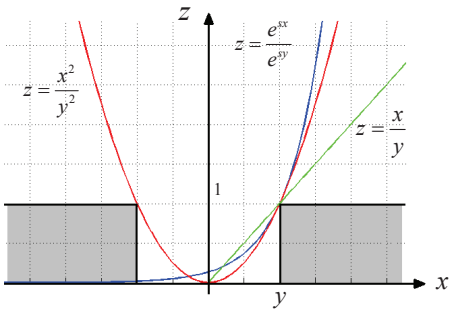


图 1.10: 基于积分性质的尾部概率不等式的证明示意图

又 $\Pr\{X \geqslant y\} = \Pr\{Z = 1\} = \mathbf{E}(Z)$，可得:

$$Z \leqslant \frac{X}{y} \Rightarrow \Pr\{X \geqslant y\} = \mathbf{E}(Z) \leqslant \frac{\mathbf{E}(X)}{y}$$
$$Z \leqslant \frac{X^2}{y^2} \Rightarrow \Pr\{|X| \geqslant y\} = \mathbf{E}(Z) \leqslant \frac{\mathbf{E}(X^2)}{y^2}$$
$$Z \leqslant \frac{e^{sX}}{e^{sy}} \Rightarrow \Pr\{X \geqslant y\} = \mathbf{E}(Z) \leqslant \frac{\mathbf{E}(e^{sX})}{e^{sy}}.$$

1.2.6 大数定律及中心极限定理

在试验不变的条件下，重复试验多次，随机事件的频率接近于它的概率，这就是大数定律。定理叙述如下。

【定理 1.14】 设 X_1，X_2，...，X_n，... 是独立同分布的，且具有数学期望 $\mathbf{E}(X)$ 和方差 σ^2，则对于任意 $\varepsilon > 0$，序列 $S_n = \sum_{k=1}^{n} X_k$ 满足（Chebyshev 不等式）：

$$P(|S_n/n - \mathbf{E}(X)| \geqslant \varepsilon) \leqslant \sigma^2/n\varepsilon^2。$$

这个定理表明：

1. 统计平均在概率意义上趋于集合平均，即 $S_n/n \xrightarrow{P} \mathbf{E}(X)$；
2. 随着 n 的增大，S_n/n 的分布函数趋于一个阶跃函数，阶跃点在 $\mathbf{E}(X)$。

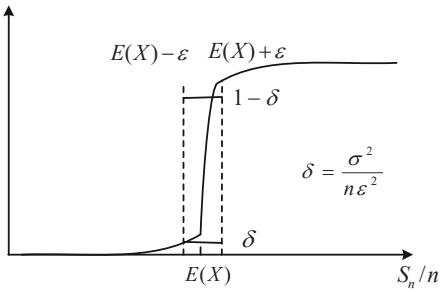


图 1.11

【例题 1.15】 设 X 是指数分布，$P\{X > x\} = \exp(-x)$。取一样本 x，放入信封 A；再以 $1/2$ 的概率计算 $y = 2x$ 或 $y = x/2$，把 y 放入信封 B。请问，选 A 还是选 B 能得到更大的数。A 与 B 信封若混淆了。请问，选定一个之后，要不要换？

在客观实际中，一些现象受到许多相互独立的随机因素的综合影响，如果每个因素所产生的作用都很微小时，总的影响可以看作是服从正态分布的。这时，我们可以应用中心极限定理。

【定理 1.16】 设 X_1，X_2，...，X_n 是独立同分布的随机变量，具有数学期望 $\mathbf{E}(X)$ 和方差 σ^2，则序列 $S_n = \sum_{k=1}^{n} X_k$ 满足：

$$\lim_{n\to\infty} P\left(\frac{S_n - n\mathbf{E}(X)}{\sqrt{n}\sigma} \leqslant y\right) = \int_{-\infty}^{y} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx。$$

用语言描述即，$\frac{S_n - n\mathbf{E}(X)}{\sqrt{n}\sigma}$ 的分布函数趋于正态分布函数（注意，前者可能没有密度函数）。

1.2.7 简单事实

总结以上内容，我们可以得到以下事实：

1. 概率界
   - $P(A \cup B) \geqslant \max\{P(A)，P(B)\}$
   - $P(AB) \leqslant \min\{P(A)，P(B)\}$
   - $P(A \cup B) \leqslant P(A) + P(B)$（并集上限）
2. 事件序列概率极限：
   - $P\left\{\bigcup_{n=1}^{\infty} A_n\right\} = \lim_{n\to\infty} P\{A_n\}$，for $A_1 \subseteq A_2 \subseteq ...$
   - $P\left\{\bigcap_{n=1}^{\infty} A_n\right\} = \lim_{n\to\infty} P\{A_n\}$，for $A_1 \supseteq A_2 \supseteq ...$
3. 必然事件 $\neq$ 概率等于 1 的事件，不可能事件 $\neq$ 概率等于 0 的事件；若样本空间是离散的，可以不加区分这些细微差别。
4. 若随机变量 X 的数学期望是 $\mathbf{E}(X)$，则一定存在 $x \leqslant \mathbf{E}(X)$，使 $P(X \leq x) > 0$（“抽屉”原理）。
5. 独立同分布的 X_1，X_2，$\cdots$，X_n，$\cdots$，具有数学期望 $\mathbf{E}(X)$ 和方差 σ^2，设 $S_n = \sum_{k=1}^{n} X_k$。
   - 不能说 $S_n \to n\mathbf{E}(X)$；
   - 也不能说 $S_n \sim \mathcal{N}(n\mathbf{E}(X)，n\sigma^2)$。
   - 以 $P_{X_k}(0) = 3/4$，$P_{X_k}(1) = 1/4$ 为例，画图说明。

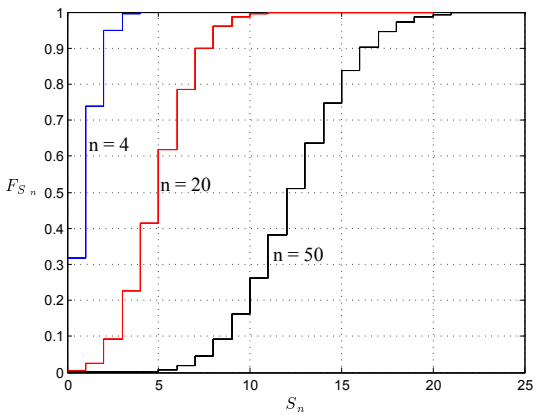


图 1.12

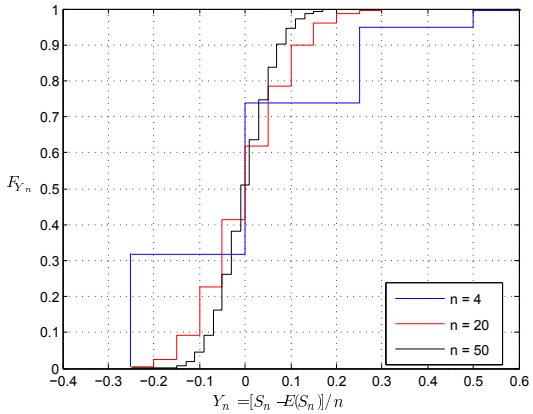


图 1.13

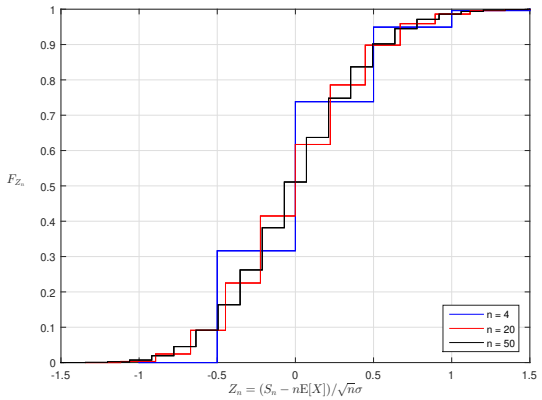


图 1.14

1.2.8 Monte Carlo 仿真

给定一个随机试验及一个事件 A，我们可以用如下方法计算概率 $P(A)$。这类方法称为蒙特卡洛（Monte Carlo）方法。

重复做试验 n 次，且假定 n 次试验之间相互独立，得到试验结果 $\omega_1, \omega_2, \cdots, \omega_n$，定义随机变量序列

$$Z_i = \begin{cases} 1, & \omega_i \in A \\ 0, & \omega_i \notin A \end{cases}$$

则 $Z_1, Z_2, \cdots, Z_n$ 是独立同分布的随机变量序列，且 $\mathbf{E}(Z_i) = P(A)$。所以，根据大数定律，我们可以估计 $P(A) \approx \frac{\sum_i Z_i}{n}$。这里“$\approx$”是在概率意义下的近似，其近似的可信度随着 n 增大而增大。通常，我们可选取 $\sum_i Z_i$ 是否超过某个门限值作为试验的终止条件。比如，事件 A 是否已经发生了 1000 次左右。这个选取要靠经验，也要兼顾计算耗时量。

§1.3 习题

1. 给出两个事件不相容的定义；给出两个事件独立的定义；讨论事件不相容与事件独立的关系。

2. 举例说明三个事件两两独立并不蕴含三个事件独立。

3. 举例说明增加条件，概率并无确定的变化方向。即举例说明 $P(A|B_1) = P(A)$，$P(A|B_2) < P(A)$。

4. 两个随机变量 X，Y 独立，当且仅当 $F_{X,\,Y}(x，\,y) = F_X(x) \cdot F_Y(y)$ 对于所有的 x，y 成立。试给出两个离散型随机变量独立的定义，两个连续型随机变量独立的定义。

5. 三个离散型随机变量 X，Y，Z 若满足，对于任意 $x \in \mathcal{X}$，$y \in \mathcal{Y}$，$z \in \mathcal{Z}$，$P_{X,\,Y,\,Z}(x，\,y，\,z) = P_X(x) \cdot P(y|x) \cdot P(z|y)$，我们记这个变量构成马尔科夫链，记作 $X - Y - Z$。证明，若 $X - Y - Z$ 是马尔科夫链，则 $Z - Y - X$ 也是马尔科夫链。

6. 独立掷两个均匀骰子，点数记为 X，Y，记 $M_+ = \max(X，\,Y)$，$M_- = \min(X，\,Y)$，求 M_+，M_- 的分布律；求 $\mathbf{E}M_+$，$\mathbf{E}M_-$；说明 $\mathbf{E}M_+ + \mathbf{E}M_- = \mathbf{E}X + \mathbf{E}Y$。

7. Alice 向 Bob 发送字符 0 或 1，发送 0 的概率为 p_0，传输过程中字符有概率 p_e 被翻转，求当 Bob 收到字符为 1 时，Alice 发送字符为 0 的概率。

8.（研究型问题）在《万里归途》电影里有一个情景，穆夫塔刁难宗大伟，发起“轮盘赌”。现设枪中子弹数服从概率为 $\frac{1}{2}$ 的 0-1 两点分布。请细化概率模型，分析随着“轮盘赌”进行，枪中无子弹的概率是如何变化的？具体地，无子弹的初始概率为 $\frac{1}{2}$。选择至少两个概率模型（必要时可以改变两轮之间的规则），分析第 $i \geq 1$ 枪后无子弹的概率。

§1.4

拓展阅读

积分、期望、蒙特卡洛仿真

我们讨论非负函数或者非负随机变量。在此前提下，分析一些概念的几何意义，有助于直观理解。

一个函数 $y=f(x)$ 在一个给定区间 $[a,b]$ 上的定积分，即是这个曲线下方的面积。

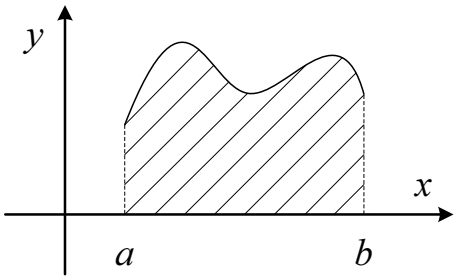


图 1.15

从定义上讲，可以分成四步来描述这个积分过程。

第一步，把区间 $[a,b]$ 分成很多小区间，每个区间的长度记为 Δx；

第二步，在每个区间内取一个点，计算每一个区间对应“梯形”的近似面积 $f(x)\Delta x$；

第三步，对这些面积求和；

第四步，当区间越分越细时，上述和的极限是积分 $\int_a^b f(x)\mathrm{d}x$。

一个非负随机变量 X 的数学期望 $\mathbf{E}(X)$ 是概率加权和。对于离散型随机变量而言，$\mathbf{E}(X)=\sum xP(x)$；对于连续型随机变量而言，$\mathbf{E}(X)=\int xf(x)\mathrm{d}x$。这里的 $P(x)$ 是概率质量函数，$f(x)$ 是概率密度函数。对于一般的随机变量 X，其累积分布函数记作 $F_X(x)$，则数学期望定义为

$$\mathbf{E}(X)=\int x\mathrm{d}F_X(x).$$

这个形式是黎曼-斯蒂尔杰斯（Riemann–Stieltjes）积分，与常见的黎曼积分相比，其定义也可以描述为四个步骤，但梯形的面积近似用 $f(x)\cdot\Delta F_X(x)$ 来取代，也就是说，横坐标的“长度”不是线性的，而是由 $F_X(x)$ 来定义。这样，我们可以认为 $\mathbf{E}(X)$ 是如下阴影部分的“面积”，其中 a,b 分别是 X 的最小值与最大值。

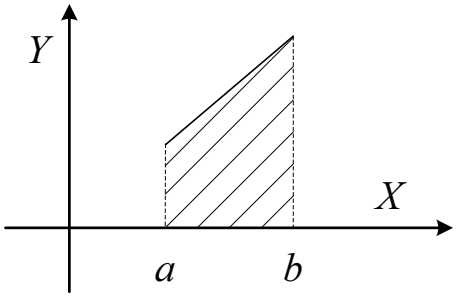


图 1.16

再次强调，这部分面积的计算是按照 $F_X(x)$ 的尺度计算的。

设 $Y=X^2$，则 $\mathbf{E}(Y)$ 对应如下阴影部分的面积。

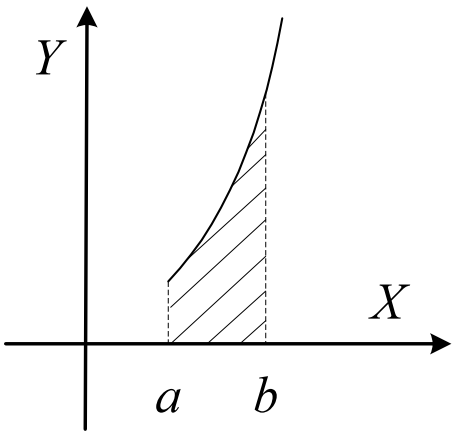


图 1.17

一个事件 A 的概率记作 $P(A)$，其可以表示成一个数学期望 $\mathbf{E}(I_A)$，其中 I_A 表示 A 的示性函数。当一个样本点 $\omega \in A$ 时，$I_A(\omega) = 1$；否则 $I_A(\omega) = 0$。由于 $I_A(\omega)$ 是定义在样本空间 Ω 上的函数，所以，当样本空间是多维时，$I_A(\omega)$ 的图示就不方便了，这也表明 $I_A(\omega)$ 更具一般性。

蒙特卡洛仿真可以很方便地估算 $P(A)$ 的值。基本方法是，产生大量的样本点 ω，统计 $\omega \in A$ 发生的频率即可。蒙特卡洛仿真可以估算 $P(A)$，而 $P(A)$ 是示性函数的积分，因而，蒙特卡洛仿真也可以用来计算积分。

第一种情况，常规积分 $\int_a^b f(x)\mathrm{d}x$ 的仿真方法。

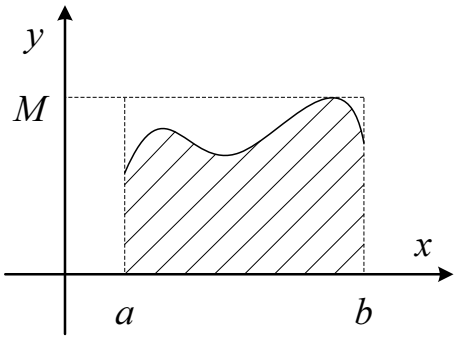


图 1.18

产生 $[a, b] \times [0, M]$ 区域中的均匀分布，统计落在阴影部分的频率，用该频率乘以矩形部分面积 $M(a-b)$ 来估计积分值。

第二种情况，$\mathbf{E}(X) = \int_a^b x\mathrm{d}F_X$。

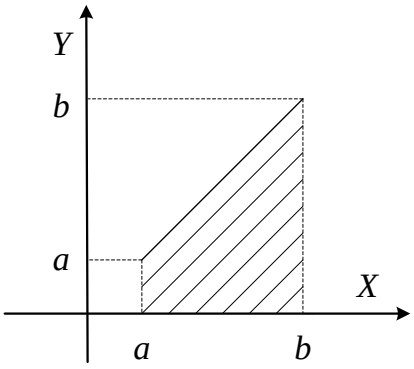


图 1.19

产生 X 的样本，同时产生 $[a,b]$ 之间的均匀分布 Y，统计 (X,Y) 落在阴影部分的频率。此时，矩形“面积”为 b，因为矩形在 X 方向上的“长度”是用 F_X 来计算的，长度为 1。再用统计出的频率乘以矩形部分面积 b 来估计积分值。

再看一个例子。设 X 是非负随机变量，说明 $\mathbf{E}(\min(1,X))=\Pr\{X\geq U\}$，其中 U 与 X 相互独立，且服从 $[0,1]$ 区间上的均匀分布。我们可以看出，$Y=\min(1,X)$ 的图形如下，

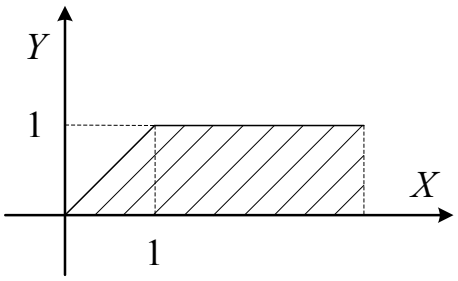
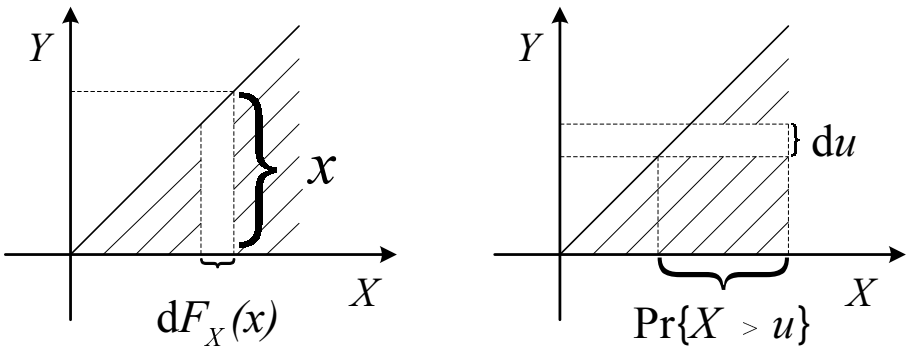


图 1.20

因此，$\mathbf{E}(Y)$ 即是上图阴影部分的面积，这个面积的计算可以归为产生 (X,U)，统计频率即可，其意义就是 $\Pr\{X\geq U\}$。

最后，我们再来解释一下非负随机变量的一个重要公式 $\mathbf{E}(X)=\int_0^{+\infty}F_X^{\mathrm{c}}(x)\mathrm{d}x$，即 X 的均值等于其累积分布补函数的积分。我们已经知道，$\mathbf{E}(X)$ 是如下阴影部分的概率。



方法一　　方法二

图 1.21

阴影部分的概率有两种计算方法。

一是“先纵后横”，给定 x，其对应横坐标“长度”是 $\mathrm{d}F_X(x)$，高度是 x。因而纵向小矩形面积是 $x\mathrm{d}F_X(x)$，横向累积即期望 $\mathbf{E}(X)=\int x\mathrm{d}F_X$。

二是“先横后纵”，给定 u，这个 u 对应的高度是 $\mathrm{d}u$，宽度是 $\Pr\{X>u\}$，所以 $\mathbf{E}(X)=\int\Pr\{X>u\}\mathrm{d}u$。

第二章

离散信源编码定理

这一章，我们考虑离散信源：时间与取值都是离散的。一般地，一个时间离散的信源可以表示为一个随机变量序列：$X_1, X_2, \cdots, X_n, \cdots$，其中，$X_t$ 取值在 $\mathcal{X}$ 上，其统计规律可以用一族联合分布律 $\{P_{X^n}(x^n), n \geq 1\}$ 来表征。设 $\mathcal{D} = \{0, 1, \cdots, D-1\}$ 是字符集，我们用 $\mathcal{D}^*$ 表示由 $\mathcal{D}$ 构成的字符串的全体，包括空字符串，即 $\mathcal{D}^* = \bigcup_{\ell \geq 0} \mathcal{D}^\ell$。信源编码的一般框架可以描述为 [3]：

$$\begin{aligned} \textbf{编码：} \quad & \phi_n : \mathcal{X}^n \longmapsto \mathcal{D}^* \\ \textbf{译码：} \quad & \psi_n : \mathcal{D}^* \longmapsto \mathcal{X}^n \\ \textbf{码率：} \quad & R_n = \tfrac{1}{n} \textstyle\sum_{x^n} P_{X^n}(x^n) \ell(\phi_n(x^n)) \\ \textbf{译码错误：} \quad & \varepsilon_n = \Pr\{\psi_n(\phi_n(X^n)) \neq X^n\} \end{aligned}$$

在上述码率 R_n 的表达式中，$\ell(\phi_n(x^n))$ 表示码字 $\phi_n(x^n)$ 的长度。由此，我们知道码率表示在统计意义下每个信源符号所用的码字的平均长度。离散信源编码的问题就是通过证明 ϕ_n 与 ψ_n 的存在性，寻找满足 $\lim\limits_{n\to\infty} \varepsilon_n = 0$ 的码率 R_n 的下极限。无论是理论还是实践，$\mathcal{D} = \{0, 1\}$ 都是最重要的情形，即我们主要考虑信源输出序列的二元序列表征问题。

§2.1 离散无记忆信源编码定理

我们以下讨论一种简单但非常重要的信源：**离散无记忆信源**。
设随机变量序列 $\mathbf{X} = (X_1, X_2, \cdots, X_n, \cdots)$，满足：

$$\begin{aligned} \textbf{无记忆的：} \quad & P_{X^n}(x^n) = \textstyle\prod_{1 \leq t \leq n} P_{X_t}(x_t) \quad \text{for} \quad n > 1 \\ \textbf{平稳的：} \quad & P_{X_t}(x) \equiv P_{X_1}(x) \triangleq P_X(x) \quad \text{for} \quad t > 1 \end{aligned}$$

这样一个随机过程 $\mathbf{X}$ 也称为“独立同分布”的，其统计规律可由一维边缘分布刻画。

【定义 2.1】 设 X 为离散随机变量，概率质量函数为 $P_X(x)$，$x \in \mathcal{X}$。X 的熵，记作 $H(X)$，定义为：

$$H(X) = \sum_{x \in \mathcal{X}} P_X(x) \log \frac{1}{P_X(x)}.$$

关于熵的定义，我们有如下注释：

(1) 根据熵的定义，若 log 以 2 为底，则熵的单位是 比特/符号（bits/symbol）；若 log 以 e 为底，则熵的单位是 奈特/符号（nats/symbol）。如果我们已知信源每秒发出的符号数，则熵的单位可以是 比特/秒（bits/second）或 奈特/秒（nats/second）。

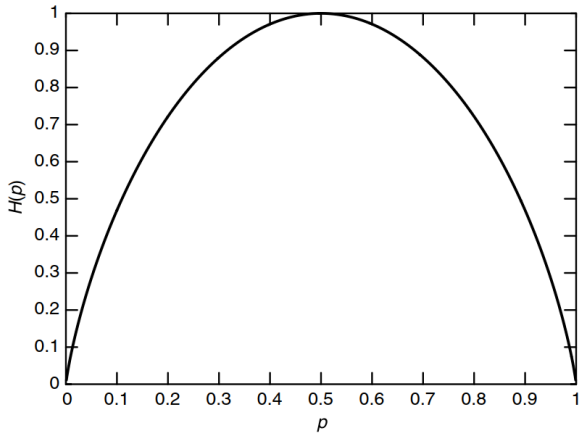


图 2.1: $H(p)$ 与 p 的关系曲线

(2) $I(x) \triangleq \log(1/P_X(x))$ 也被称为 x 的**自信息**。所以，我们也可以说熵是自信息的数学期望。概率小的样本点的自信息大，但是在熵中权重较小；而概率大的样本点的自信息小，但在熵中权重较大。

(3) 我们约定 $0 \log 0 \triangleq 0$，这样约定是考虑到 $\lim\limits_{x \to 0^+} x \log x = 0$。

(4) $H(X) \geq 0$，这是因为自信息是非负的，所以熵作为期望值，也具有非负性。特别地，除了确定性变量外，$H(X) > 0$。

(5) $H(X) \leq \log |\mathcal{X}|$，这个证明并不显然。需要说明的是，熵 $H(X)$ 不依赖于 X 的取值，而仅仅依赖于其概率分布律 P_X。所以，熵可以看作一个概率向量 P_X 的函数，并且可以证明该函数是凹函数（concave function）。由此，可以证明均匀分布律对应最大的熵。

(6) 对于二元随机变量，$P_X(1) = p$，$P_X(0) = 1 - p$，我们定义熵函数 $H(p) = -p \log p - (1-p) \log(1-p)$。该函数如图 2.1所示。可以看出，$H(0) = H(1) = 0$，而对于 $p > 0$， $H(p) > 0$，且在 $p = \frac{1}{2}$ 时达到最大熵 $H(\frac{1}{2}) = 1$ 比特/符号（bits/symbol）。

下面的信源编码定理表明只要编码速率大于信源的熵，就可以找到错误概率任意小的编译码方案。

【定理 2.2】 （离散无记忆信源编码定理） 设码率 $R > H(X)$，存在固定码长编码 (ϕ_n, ψ_n)，使得 $R_n \leq R$，$\varepsilon_n \to 0$。当允许可变码长时，$\varepsilon_n = 0$。

□

注：信源编码根据编码器的输入与输出的长度大致可以分为三类：（1）定长-定长：输入长度固定，输出长度也固定；（2）定长-变长：输入长度固定，输出长度不定，即不同的固定长度的信源符号序列对应的码字序列可能具有不同长度。大家熟知的例子是摩尔斯电码（Morse code）；（3）变长-变长：输入长度不定，输出长度也不定。我们在本章主要讨论的是输入长度固定的，其中情形（1）称之为固定码长编码。

我们可以使用三种方法来证明这个定理：典型序列、随机装箱和型方法。

§2.2 典型序列方法

信源编码的基本目的是：用二元序列表征“纷繁复杂”的信源输出。一个信源输出的序列，如果编码映射到一个“独享的”码字序列，则该信源序列在译码时就可以恢复，不会产生错误。而如果一个信源序列没有与之对应的码字序列，则该信源序列在译码时就会发生错误。设编码速率为 R，则大约有 2^{nR} 个二元序列可以用作码字。那么，我们应该给哪些序列指派码字序列呢？由于我们的目标是尽可能降低译码错误概率，所以一个简单的办法是，把所有信源序列按照其发生的概率大小排序，然后给排在前 2^{nR} 个序列进行编码，而其余序列忽略。如此，在译码时前 2^{nR} 个序列不会发生错误，但其余的序列就会发生错误。困难的是，当 n 很大时，排序遇到了挑战。而事实证明，在 n 充分大时，无需对信源序列进行排序，因为几乎所有的信源序列都具有相近的概率。这一点是下面弱大数定律所要表达的内容。

【例题 2.3】 掷一个骰子，结果为 1，2，3，4，5，6 的概率分别为 $\frac{1}{2}$，$\frac{1}{4}$，$\frac{1}{8}$，$\frac{1}{16}$，$\frac{1}{32}$，$\frac{1}{32}$，只用两比特表示掷骰子的结果，如何编码使得译码错误率最小？

【引理 2.4】 **弱大数定律** 若 $X_1, X_2, \ldots, X_n, \ldots$ 是独立同分布的随机变量序列，一维分布是 $P_X(x)$，则 $X_1, X_2, \ldots, X_n, \ldots$ 的样本熵 $-\frac{1}{n}\log P_{X^n}(X^n)$ 依概率收敛于 $H(X)$，记作 $\lim\limits_{n\to\infty} -\frac{1}{n}\log P_{X^n}(X^n) \overset{P}{=} H(X)$，即，$\forall \varepsilon > 0$，我们有

$$\lim_{n\to\infty} \Pr\left\{\left|-\frac{1}{n}\log P_{X^n}(X^n) - H(X)\right| \geq \varepsilon\right\} = 0.$$

由此引理，我们可以引入典型集的概念，即所有样本熵在集合熵附近的序列的集合。

【定义 2.5】 **典型集** [3] 设 $X_1, X_2, \ldots, X_n, \ldots$ 独立同分布，服从 $P_X(x)$，$x \in \mathcal{X}$，给定 $\varepsilon > 0$，一个序列 $(x_1, x_2, \ldots, x_n) \in \mathcal{X}^n$ 是 $\varepsilon-$ 典型的，如果

$$2^{-n(H(X)+\varepsilon)} \leq P_{X^n}(x^n) \leq 2^{-n(H(X)-\varepsilon)},$$

所有 $\varepsilon-$ 典型序列的全体记作 $A_\varepsilon^{(n)}$，称为**典型集**。

【例题 2.6】 投掷一枚硬币，正面朝上的概率为 $\frac{3}{4}$，反面朝上的概率为 $\frac{1}{4}$。则有随机变量 $\mathbf{X} \in \{0, 1\}$，其概率质量函数为 $P(1) = \frac{3}{4}$ 和 $P(0) = \frac{1}{4}$，熵 $H(X) = -\frac{3}{4}\log\frac{3}{4} - \frac{1}{4}\log\frac{1}{4} \approx 0.8113$。投掷硬币 5 次，则有 32 (2^5) 个序列：$(0, 0, 0, 0, 0)$、$(0, 0, 0, 0, 1)$、$\ldots$。从典型序列的定义可以看出，给定 n，判定一个序列是不是典型的，与设定的 ε 有关。在本例中，$n = 5$，当 $\varepsilon = 0.1$ 时，则一个有 4 次正面朝上的序列是典型的。例如，序列 $(0, 1, 1, 1, 1)$，其样本熵为：$-\frac{1}{5}\log P(0, 1, 1, 1, 1) \approx 0.7320$，落在区间 $[0.7113, 0.9113]$ 内，是一个典型序列。而当 $\varepsilon = 0.01$ 时，则没有序列是典型的。由此可以想象，固定 n，若 ε 取值很大，则所有序列都是典型的；若 ε 取值很小，则典型序列的数量（典型集 $A_\varepsilon^{(n)}$ 的元素个数）会减少。信息论关心的是另一个问题，任意给定一个 $\varepsilon > 0$（无论多小），n 充分大时，哪些序列是典型的？如果硬币是均匀的（正面朝上的概率 $P(1)$ 与反面朝上的概率 $P(0)$ 均等于 $\frac{1}{2}$），按照我们目前关于典型序列的定义，则所有序列都是典型的。

由引理 2.4，我们可以证明如下渐近均分性（**AEP**）[3]。

【定理 2.7】 **渐近均分性**

(1) 若 $x^n \in A_\varepsilon^{(n)}$，则 $H(X) - \varepsilon \leq -\frac{1}{n}\log P_{X^n}(x^n) \leq H(X) + \varepsilon$。

(2) 当 n 充分大时，$\Pr\{A_\varepsilon^{(n)}\} \geq 1 - \varepsilon$。

(3) $|A_\varepsilon^{(n)}| \leq 2^{n(H(X)+\varepsilon)}$，其中 $|A|$ 表示集合 A 中元素的个数。

(4) 当 n 充分大时，$|A_\varepsilon^{(n)}| \geq (1-\varepsilon)2^{n(H(X)-\varepsilon)}$。

□

注：由定义 2.5 我们可以看出，典型序列是其概率限定在一个范围内的序列，即，典型序列的概率有非平凡的上下界。定理 2.7 第 (1) 点指出典型序列的样本熵也具有上下界。第 (2) 点说明典型集的概率可以接近于 1，这是

大数定律的推论。而第 (3) 点与第 (4) 点给出了典型序列的个数，也即典型集的大小的上下界。这两点可以这样直观想象，因为每个典型序列的概率有上下界，所以，典型序列的个数不能太多，否则典型集的概率大于 1 而产生矛盾；类似地，典型序列的个数也不能太少，否则典型集的概率可能远小于 1。

证明:

(1) 由定义 2.5 可得，若 $x^n \in A_\varepsilon^{(n)}$, 则满足
$2^{-n(H(X)+\varepsilon)} \le P_{X^n}(x^n) \le 2^{-n(H(X)-\varepsilon)}$,
则有 $H(X)-\varepsilon \le -\frac{1}{n}\log P_{X^n}(x^n) \le H(X)+\varepsilon$。

(2) 当 n 充分大时，由引理 2.4，$\forall \varepsilon > 0$，我们有
$\lim\limits_{n\to\infty} \Pr\left\{\left|-\frac{1}{n}\log P_{X^n}(X^n) - H(X)\right| < \varepsilon\right\} = 1$,
可得 $\Pr\{A_\varepsilon^{(n)}\} \ge 1-\varepsilon$。

(3) $\sum_{x\in A_\varepsilon^{(n)}} P_{X^n}(x^n) \le \sum_{x\in\mathcal{X}} P_{X^n}(x^n) = 1$，由定义 2.5,
可得 $\sum_{x\in A_\varepsilon^{(n)}} 2^{-n(H(X)+\varepsilon)} \le \sum_{x\in A_\varepsilon^{(n)}} P_{X^n}(x^n) \le 1$,
则 $|A_\varepsilon^{(n)}|\, 2^{-n(H(X)+\varepsilon)} \le 1$,
$|A_\varepsilon^{(n)}| \le 2^{n(H(X)+\varepsilon)}$。

(4) 当 n 充分大时，$\Pr\{A_\varepsilon^{(n)}\} \ge 1-\varepsilon$。由定义 2.5,
可得 $1-\varepsilon \le \Pr\{A_\varepsilon^{(n)}\} = \sum_{x\in A_\varepsilon^{(n)}} P_{X^n}(x^n) \le \sum_{x\in A_\varepsilon^{(n)}} 2^{-n(H(X)-\varepsilon)} = |A_\varepsilon^{(n)}|\, 2^{-n(H(X)-\varepsilon)}$,
可得 $|A_\varepsilon^{(n)}| \ge (1-\varepsilon)2^{n(H(X)-\varepsilon)}$。

□

离散无记忆信源编码定理证明（典型序列方法）:

(1) **编码:**

第一步，将 $\mathcal{X}^n$ 中的所有序列划分成两个集合：典型集 $A_\varepsilon^{(n)}$ 及其补集 $A_\varepsilon^{(n)^c} = \mathcal{X}^n - A_\varepsilon^{(n)}$。

第二步，将所有的典型序列按照某种顺序（比如字典序）编号，并把这些编号转化为二进制，用作对应序列的编码。由于典型序列的个数 $|A_\varepsilon^{(n)}| \le 2^{n(H(X)+\varepsilon)}$，所以典型序列可以用不超过 $\lceil n(H(X)+\varepsilon)\rceil$ 比特表示，其中 $\lceil x\rceil$ 表示“向上取整”，即，不小于 x 的最小整数。类似地，由于非典型序列的个数显然不超过 $|\mathcal{X}|^n$，非典型序列可以用不超过 $\lceil n\log|\mathcal{X}|\rceil$ 比特表示。

第三步，分别给典型集 $A_\varepsilon^{(n)}$ 中的所有序列前面加上 0，非典型集 $A_\varepsilon^{(n)^c}$ 的序列前面加上 1。这样就得到了 $\mathcal{X}^n$ 的所有序列的一个编码方案，其中，描述典型序列需要的总长度不超过 $\lceil n(H(X)+\varepsilon)+1\rceil$ 比特，非典型序列的编码总长度不超过 $\lceil n\log|\mathcal{X}|+1\rceil$ 比特。

(2) **译码:**

起始位 0 或 1，作为标识位，识别是典型序列或非典型序列，标明紧随码字的长度。由于编码是一对一映射，那么根据序列的整数表示可以找到对应序列的编号，从而实现译码。

(3) **码率:**

$$
\begin{aligned}
R_n &= \frac{1}{n}\sum_{x^n} P_{X^n}(x^n)\ell(x^n) \\
&= \frac{1}{n}\left[\sum_{x^n\in A_\varepsilon^{(n)}} P_{X^n}(x^n)\ell(x^n) + \sum_{x^n\in A_\varepsilon^{(n)^c}} P_{X^n}(x^n)\ell(x^n)\right] \\
&\le \frac{1}{n}\left[\sum_{x^n\in A_\varepsilon^{(n)}} P_{X^n}(x^n)(n(H(X)+\varepsilon)+2) + \sum_{x^n\in A_\varepsilon^{(n)^c}} P_{X^n}(x^n)(n\log|\mathcal{X}|+2)\right] \\
&= \frac{1}{n}\left[\Pr\{A_\varepsilon^{(n)}\}(n(H(X)+\varepsilon)+2) + \Pr\{A_\varepsilon^{(n)^c}\}(n\log|\mathcal{X}|+2)\right] \\
&\le \frac{1}{n}\left[n(H(X)+\varepsilon) + \varepsilon n\log|\mathcal{X}| + 2\right] = H(X)+\varepsilon',
\end{aligned}
$$

由定理条件知，$R > H(X)$，所以我们可以取充分小的正数 ε，使得对所有充分大的 n，$\varepsilon' \triangleq \varepsilon + \varepsilon \log|\mathcal{X}| + \frac{2}{n} < \frac{R-H(X)}{2}$。这样，我们得到上述编码的码率 $R_n \leq H(X) + \varepsilon' \leq R$。

(4) **译码错误概率:**

在这个编码方案中，我们将 $\mathcal{X}^n$ 上的所有序列的编号的二进制作为编码，进行一对一映射，因而在译码时可以对应恢复，不会发生错误。即，译码错误概率 ε_n 为 0。

□

【例题 2.8】 我们以例 2.6 为例对上述证明进行说明。在此例子中，$n = 5$，熵 $H(X) \approx 0.8113$，共有 32 个序列。当 $\varepsilon = 0.1$ 时，这些序列可以分为两类：典型的与非典型的，分别按照字典序排序及编号如下表：

表 2.1

序列类型	序列	十进制序号	二进制序号	码字
典型序列	(0，1，1，1，1)	0	000	0000
	(1，0，1，1，1)	1	001	0001
	(1，1，0，1，1)	2	010	0010
	(1，1，1，0，1)	3	011	0011
	(1，1，1，1，0)	4	100	0100
非典型序列	(0，0，0，0，0)	0	00000	100000
	(0，0，0，0，1)	1	00001	100001
	(0，0，0，1，0)	2	00010	100010
	(0，0，0，1，1)	3	00011	100011
	(0，0，1，0，0)	4	00100	100100
	(0，0，1，0，1)	5	00101	100101
	(0，0，1，1，0)	6	00110	100110
	(0，0，1，1，1)	7	00111	100111
	(0，1，0，0，0)	8	01000	101000
	(0，1，0，0，1)	9	01001	101001
	(0，1，0，1，0)	10	01010	101010
	(0，1，0，1，1)	11	01011	101011
	(0，1，1，0，0)	12	01100	101100
	(0，1，1，0，1)	13	01101	101101
	(0，1，1，1，0)	14	01110	101110
	(1，0，0，0，0)	15	01111	101111
	(1，0，0，0，1)	16	10000	110000
	(1，0，0，1，0)	17	10001	110001
	(1，0，0，1，1)	18	10010	110010
	(1，0，1，0，0)	19	10011	110011
	(1，0，1，0，1)	20	10100	110100
	(1，0，1，1，0)	21	10101	110101
	(1，1，0，0，0)	22	10110	110110
	(1，1，0，0，1)	23	10111	110111
	(1，1，0，1，0)	24	11000	111000
	(1，1，1，0，0)	25	11001	111001
	(1，1，1，1，1)	26	11010	111010

(1) **编码:** 分别把典型序列和非典型序列的编号转化为二进制，在典型序列的编号前加上 0，非典型序列加上 1，作为对应序列的码字。典型序列的码长 $\ell(x^n) = 4 < \lceil n(H(X)+\varepsilon)+1 \rceil = 6$，非典型序列的码长 $\ell(x^n) = 6 \leq \lceil n \log|\mathcal{X}| + 1 \rceil = 6$。

(2) **译码:** 根据编码输出的起始位识别该序列是典型序列还是非典型序列，若起始位为 0，则该序列为典型序列，码长为 4；若起始位为 1，则该序列为非典型序列，码长为 6。根据码字可以找到对应序列实现译码。更具体地，假设信源输出（即编码器的输入）是 (1，1，1，1，0) (1，0，1，1，1) (0，1，0，0，1) (1，1，1，1，1)，则编码输出为 01000001101001111010，而这个码字序列可以唯一译码得到信源输出。方法是，根据第一位是 0，可以识别第一个序列为典型序列，码长为 4，则第一个序列的码字为 0100，对应译码可以知道此序列是典型序列编号为 4 的序列。以此类推，可以将编码输出 01000001101001111010 译码为典型序列 4 号、典型序列 1 号、非典型序列 9 号、非典型序列 26 号，对应找到信源输出序列。

注：上述例子说明了码表构造与译码的过程，但可以验证，这并不是一个有效的压缩编码，因为码字序列可能比直接表示观察结果还要长。实际上，要达到压缩效果，ε 要充分小，码长要充分大。此时，用查表方法是不切实际的。

§2.3 随机装箱方法

随机装箱方法，是信息论中很有用的一个工具，也是概率论中一个重要的模型，“把 m 个球随机投放到 n 个箱子里”。在这个有名的模型中，我们通常会问几个问题，如：有多少箱子是空的？箱子最多有多少球？球在箱中的分布是怎样的？我们这里主要利用这样一个事实：如果箱子个数相比于球数目足够多，则每个箱子里有两个球的机会就非常小。

离散无记忆信源编码定理证明（随机装箱方法）[3]:

设有 2^{nR} 个箱子（若 nR 不是整数，则取 $2^{\lceil nR\rceil}$），这些箱子可以用 nR 比特进行编号，记为 $\{0,1,2,\ldots,2^{nR}-1\}$。所有的序列 $\mathbf{x}\in\mathcal{X}^n$ 被随机投掷到这些箱子里。然后这些箱子向编译码器公开，即：$\phi_n:\mathcal{X}^n\to\{0,1,2,\ldots,2^{nR}-1\}$。

若信源输出序列 $\mathbf{x}\in\mathcal{X}^n$，则信源编码器输出 $\phi_n(\mathbf{x})$。可以看出，编码是一个“多对一”的映射，即，不同的信源序列可能具有相同的码字序列。

对于译码 $\phi_n(\mathbf{x})$，我们在编号 $\phi_n(\mathbf{x})$ 的箱子中寻找一个典型序列。若有且仅有一个典型序列 $\hat{\mathbf{x}}$ 在箱子中，则译码器输出序列 $\hat{\mathbf{x}}$，即：$\psi_n:\{0,1,2,\ldots,2^{nR}-1\}\to\mathcal{X}^n$。若箱子中没有典型序列，或者有超过一个典型序列，则译码报错。

可以看出，这样的码是定长到定长的编码，其编码速率 $R_n=R$。

下面我们来分析译码错误概率。对于任意 $R>H(X)$，我们可以取 $\varepsilon>0$，使得 $R>H(X)+\varepsilon$。可以看出，有两种情况会导致错误发生。一种情况是，信源输出的序列不是典型序列；另一种情况是，信源输出的序列是典型序列，但它所在的箱子里还有另外的典型序列。

$$\begin{aligned}
\varepsilon_n &= \Pr\{\psi_n(\phi_n(\mathbf{X}))\neq\mathbf{X}\}\\
&\le \Pr\{\mathbf{X}\notin A_\varepsilon^{(n)}\}+\sum_{\substack{\mathbf{x},\mathbf{x}'\in A_\varepsilon^{(n)}\\ \mathbf{x}'\neq\mathbf{x}}} P_{\mathbf{X}}(\mathbf{x})\Pr\{\phi_n(\mathbf{x}')=\phi_n(\mathbf{x})\}\\
&\le \varepsilon+\sum_{\mathbf{x}'\in A_\varepsilon^{(n)}}\sum_{\mathbf{x}\in A_\varepsilon^{(n)}} P_{\mathbf{X}}(\mathbf{x})2^{-nR}\\
&= \varepsilon+\sum_{\mathbf{x}'\in A_\varepsilon^{(n)}} 2^{-nR}\sum_{\mathbf{x}\in A_\varepsilon^{(n)}} P_{\mathbf{X}}(\mathbf{x})\\
&\le \varepsilon+\sum_{\mathbf{x}'\in A_\varepsilon^{(n)}} 2^{-nR}\\
&\le \varepsilon+2^{-n[R-(H(X)+\varepsilon)]}\le 2\varepsilon
\end{aligned}$$

当 n 足够大时，有 $\varepsilon_n\to 0$。

□

随机线性装箱：上面是对一般信源的编码方法，当信源为二元信源时，我们可以设计更简单高效的编码方法。假设信源服从二元 Bernoulli 分布，则信源输出的符号集是 $\mathcal{X}=\mathbb{F}_2=\{0,1\}$，我们可以用随机线性装箱方法来实现编码。这里，我们假设信源输出的序列是 $\mathbf{x}=(x_1,x_2,\ldots,x_n)$，其中 $x_i\in\mathbb{F}_2$。我们可以用一个**均匀随机**矩阵 $G\in\mathbb{F}_2^{n\times nR}$ 来实现编码，即：$\phi_n:\mathbf{x}\mapsto\mathbf{x}G$。对于译码，我们采用与随机装箱方法相同的译码方法。可以证明对于任意两个不同的信源输出序列 $\mathbf{x}$ 和 $\mathbf{x}'$，有 $\mathbf{x}G=\mathbf{x}'G$ 的概率也是 2^{-nR}，即

$$\Pr\{\phi_n(\mathbf{x}')=\phi_n(\mathbf{x})\}=2^{-nR}.$$

因此，随机装箱的证明方法对于随机线性装箱也是适用的。

§2.4 型方法

型方法 [2] 的基本概念如下。

【定义 2.9】 型 (type) 设序列 $\mathbf{x} \in \mathcal{X}^n$，记 $N(a|\mathbf{x})$ 表示字符 $a \in \mathcal{X}$ 在序列 $\mathbf{x}$ 中出现的次数，则称

$$P_{\mathbf{x}}(a) = \frac{N(a|\mathbf{x})}{n},\ a \in \mathcal{X}$$

是序列 $\mathbf{x}$ 的型。也称之为组合（composition），在概率统计中称之为经验分布。

从定义可以看出，序列的型是一个统计概念，与这个序列来自哪一个概率模型是没有关系的。

【例题 2.10】 序列 (01011) 的型是 $(\frac{2}{5}, \frac{3}{5})$，而序列 (11000) 的型是 $(\frac{3}{5}, \frac{2}{5})$。

我们可以看出，不同的序列可能有相同的型，一个型是一个概率质量函数，而一个概率质量函数（除非各个分量是有理数）未必对应一个型。更准确地说，如果我们用 $\mathscr{P}_n$ 表示所有 $\mathbf{x} \in \mathcal{X}^n$ 对应的型的集合，用 $\mathscr{P}$ 表示 $\mathcal{X}$ 上所有概率向量的集合，即 $\mathscr{P} = \{P(a), a \in \mathcal{X} | P(a) \geqslant 0, \sum_{a\in\mathcal{X}} P(a) = 1\}$，则我们有 $\mathscr{P}_n$ 是 $\mathscr{P}$ 的真子集合。实际上，$\mathscr{P}_n$ 的大小是有限的，而 $\mathscr{P}$ 的大小是无限的。当然，我们可以用组合方式求出 $\mathscr{P}_n$ 的大小来，但在这里，一个界就够了。我们有如下引理：

【引理 2.11】 对于有限集 $\mathcal{X}^n$，

$$|\mathscr{P}_n| \leqslant (n+1)^{|\mathcal{X}|}.$$

证明: $\mathscr{P}_n$ 为所有 $\mathbf{x} \in \mathcal{X}^n$ 对应的型（概率向量）的集合，那么型向量具有 $|\mathcal{X}|$ 个分量。每个分量可取 $n+1$ 个不同的值，那么，型至多有 $(n+1)^{|\mathcal{X}|}$ 种选择。 □

在一个给定的有限集 $\mathcal{X}$ 上，我们可以引入不同的概率质量函数 P 与 Q。那么，如何衡量二者的差异呢？一种可以想到的衡量办法是欧氏距离，或者其他距离。在信息论中，更常用的是相对熵，也称为“K-L 距离（Kullback Leibler distance）”，定义如下：

【定义 2.12】 相对熵（Relative Entropy） 有限集 $\mathcal{X}$ 的两个概率密度函数 P 和 Q 之间的相对熵定义为

$$D(P\|Q) = \sum_{x\in\mathcal{X}} p(x) \log \frac{p(x)}{q(x)}.$$

一般地，相对熵有如下性质：

(1) $D(P\|Q) \neq D(Q\|P)$。

(2) $D(P\|Q) \geqslant 0$，相对熵总是非负的。当且仅当对于所有的 $x \in \mathcal{X}$ 都有 $p(x) = q(x)$ 时，$D(P\|Q) = 0$。相当于说，若存在字符 $x \in \mathcal{X}$，使得 $p(x) \neq q(x)$，则 $D(P\|Q) > 0$。

【例题 2.13】 随机变量 $\mathcal{X} = \{0, 1\}$，设 P 与 Q 为 $\mathcal{X}$ 的两个概率质量函数，分别为：

$$p(x) = \begin{cases} p, & x = 0; \\ 1-p, & x = 1 \end{cases}$$

$$q(x) = \begin{cases} q, & x = 0; \\ 1-q, & x = 1 \end{cases}$$

则有：

$$D(P\|Q) = \sum_{x\in\mathcal{X}} p(x) \log \frac{p(x)}{q(x)} = p \log \frac{p}{q} + (1-p) \log \frac{1-p}{1-q}$$
$$D(Q\|P) = \sum_{x\in\mathcal{X}} q(x) \log \frac{q(x)}{p(x)} = q \log \frac{q}{p} + (1-q) \log \frac{1-q}{1-p}$$

当且仅当 $p = q$ 时，$D(P\|Q) = D(Q\|P) = 0$。假设 $p = \frac{1}{4}, q = \frac{1}{8}$，则 $D(P\|Q) \approx 0.0832$，$D(Q\|P) \approx 0.0696$，$D(P\|Q) \neq D(Q\|P)$。

【定义 2.14】 **型类（type class）**分布 $P \in \mathscr{P}_n$ 的型类 $T(P)$ 是指长度为 n 且型为 P 的序列的集合，即

$$T(P) = \{\mathbf{x} \in \mathcal{X}^n : P_{\mathbf{x}} = P\}.$$

【引理 2.15】 对于 $\mathcal{X}$ 上的一个概率分布 Q，序列 $\mathbf{x}$ 的概率仅依赖于它的型 $P_{\mathbf{x}}$，即，

$$Q(\mathbf{x}) = 2^{-n[H(P_{\mathbf{x}})+D(P_{\mathbf{x}}||Q)]}.$$

证明:

$$\begin{aligned}
Q(\mathbf{x}) &= \prod_{i=1}^{n} Q(x_i) \\
&= \prod_{a\in\mathcal{X}} Q(a)^{N(a|\mathbf{x})} \\
&= \prod_{a\in\mathcal{X}} Q(a)^{nP_{\mathbf{x}}(a)} \\
&= \prod_{a\in\mathcal{X}} 2^{nP_{\mathbf{x}}(a)\log Q(a)} \\
&= \prod_{a\in\mathcal{X}} 2^{n(P_{\mathbf{x}}(a)\log Q(a) - P_{\mathbf{x}}(a)\log P_x(a) + P_{\mathbf{x}}(a)\log P_x(a))} \\
&= 2^{n\sum_{a\in\mathcal{X}}(-P_{\mathbf{x}}(a)\log\frac{P_{\mathbf{x}}(a)}{Q(a)} + P_{\mathbf{x}}(a)\log P_x(a))} \\
&= 2^{-n[H(P_{\mathbf{x}})+D(P_{\mathbf{x}}||Q)]}
\end{aligned}$$

若 $P_{\mathbf{x}} = Q$，则 $Q(\mathbf{x}) = 2^{-nH(P_{\mathbf{x}})} = 2^{-nH(Q)}$。 □

【引理 2.16】 对于任意一个型类 $T(P)$，它的大小满足

$$\frac{1}{(n+1)^{|\mathcal{X}|}} 2^{nH(P)} \leqslant |T(P)| \leqslant 2^{nH(P)}.$$

证明: 我们首先证明型类的上界。假设序列 $\mathbf{x}$ 的型为 P。对于概率分布 P，根据性质 (1)，我们可以知道序列 $\mathbf{x}$ 的概率为 $P(\mathbf{x}) = 2^{-nH(P)}$ 和型类 $T(P)$ 的概率为 $P(T(P)) = |T(P)|2^{-nH(P)}$。因为 $P(T(P)) \leq 1$，所以 $|T(P)| \leq 2^{nH(P)}$。

接下来，我们证明型类大小的下界。给定两个型 P 和 $\hat{P}$, 我们考虑两个型类 $T(P)$ 和 $T(\hat{P})$ 在概率分布律 P 控制下的概率比值，即，

$$\begin{aligned}
\frac{P(T(P))}{P(T(\hat{P}))} &= \frac{|T(P)|\prod_{a\in\mathcal{X}} P(a)^{nP(a)}}{|T(\hat{P})|\prod_{a\in\mathcal{X}} P(a)^{n\hat{P}(a)}} \\
&= \frac{\binom{n}{nP(a_1), nP(a_2), \ldots, nP(a_{|\mathcal{X}|})} \prod_{a\in\mathcal{X}} P(a)^{nP(a)}}{\binom{n}{n\hat{P}(a_1), n\hat{P}(a_2), \ldots, n\hat{P}(a_{|\mathcal{X}|})} \prod_{a\in\mathcal{X}} P(a)^{n\hat{P}(a)}} \\
&= \prod_{a\in\mathcal{X}} \frac{(n\hat{P}(a))!}{(nP(a))!} P(a)^{n(P(a)-\hat{P}(a))}.
\end{aligned}$$

进一步，根据不等式 $\frac{m!}{n!} \geqslant n^{m-n} (\forall m, n)$，我们可以得到

$$\begin{aligned}
\frac{P(T(P))}{P(T(\hat{P}))} &\geqslant \prod_{a\in\mathcal{X}} (nP(a))^{n(\hat{P}(a)-P(a))} P(a)^{n(P(a)-\hat{P}(a))} \\
&= \prod_{a\in\mathcal{X}} n^{n(\hat{P}(a)-P(a))} \\
&= n^{n(\sum_{a\in\mathcal{X}}\hat{P}(a) - \sum_{a\in\mathcal{X}}P(a))} \\
&= n^{n(1-1)} \\
&= 1.
\end{aligned}$$

因此，对于概率分布 P，在所有的型类中概率最大的型类为 $T(P)$，即，$P(T(P)) = \max_{\hat{P}\in\mathscr{P}_n} P(T(\hat{P}))$。

最后，根据

$$1 = \sum_{\hat{P}\in\mathscr{P}_n} P(T(\hat{P})) \leqslant (n+1)^{|\mathcal{X}|} \max_{\hat{P}\in\mathscr{P}_n} P(T(\hat{P})) = (n+1)^{|\mathcal{X}|} P(T(P)) = (n+1)^{|\mathcal{X}|}|T(P)|2^{-nH(P)},$$

我们可以推出 $|T(P)| \geq (n+1)^{-|\mathcal{X}|}2^{nH(P)}$。 □

【引理 2.17】 对于 $\mathcal{X}$ 上的一个概率分布 Q，任意一个型类 $T(P)$ 的概率满足

$$\frac{1}{(n+1)^{|\mathcal{X}|}} 2^{-nD(P||Q)} \leqslant Q(T(P)) \leqslant 2^{-nD(P||Q)}.$$

证明: 根据引理 2.16，型类 $T(P)$ 的大小有上下界：

$$\frac{1}{(n+1)^{|\mathcal{X}|}} 2^{nH(P)} \leqslant |T(P)| \leqslant 2^{nH(P)}.$$

根据引理 2.15，对于概率分布 Q，型类 $T(P)$ 中的任一序列 $\mathbf{x}$ 的概率为

$$Q(\mathbf{x}) = 2^{-n[H(P)+D(P||Q)]}.$$

从而，我们可以得到型类 $T(P)$ 的概率上下界：

$$\frac{1}{(n+1)^{|\mathcal{X}|}} 2^{-nD(P||Q)} \leqslant Q(T(P)) \leqslant 2^{-nD(P||Q)}.$$

□

注：不同的序列可能具有相同的型，具有相同的型的序列的集合称为型类，引理 2.15 说明具有相同型的序列的概率相等，仅依赖于它的型。引理 2.16 给出了型类的大小的上下界，引理 2.17 给出了型类概率的上下界。其中，$H(P_{\mathbf{x}})$ 定义为型的熵，$D(P_{\mathbf{x}}||Q)$ 定义为型的相对熵。

离散无记忆信源编码定理证明（型方法）：

设 $R_n = R - |\mathcal{X}|\frac{\log(n+1)}{n}$，则 $\{R_n\}$ 是以极限为 R 的递增序列，表示为 $R_n \uparrow R$。设序列集 $A_n = \{\mathbf{x} \in \mathcal{X}^n : H(P_{\mathbf{x}}) \leqslant R_n\}$，则 A_n 的大小为

$$\begin{aligned}
|A_n| = \sum_{P\in\mathcal{P}_n: H(P)\leqslant R_n} |T(P)| &\leqslant \sum_{P\in\mathcal{P}_n: H(P)\leqslant R_n} 2^{nH(P)} \\
&\leqslant \sum_{P\in\mathcal{P}_n: H(P)\leqslant R_n} 2^{nR_n} \\
&\leqslant (n+1)^{|\mathcal{X}|} 2^{nR_n} \\
&= 2^{n\left(R_n + |\mathcal{X}|\frac{\log(n+1)}{n}\right)} = 2^{nR}
\end{aligned}$$

由于集合 A_n 的大小不超过 2^{nR}，所以集合中的每个序列可以有对应的二进制编号，长度不超过 nR。编译码即查表对照即可。

译码错误事件为 $\bar{A}_n$。则对于任意满足 $H(Q) < R$ 的信源分布 Q，译码错误的概率为

$$\varepsilon_n = Q(\bar{A}_n) = 1 - Q(A_n) = \sum_{P\in\mathcal{P}_n: H(P)>R_n} Q(T(P))$$

因为 $R_n \uparrow R$，且 $H(Q) < R$，则存在 n_0 使得 $n \geqslant n_0$，所有 $R_n \geqslant R_{n_0} > H(Q)$。因为 $H(P) > R_{n_0} > H(Q)$，且 $D(P\|Q) > 0$，所以 $\min_{P:H(P)\geqslant R_{n_0}} D(P\|Q) = \delta > 0$。因此，对于 $n \geqslant n_0$，译码错误的概率为

$$\begin{aligned}
\varepsilon_n &\leqslant \sum_{P\in\mathcal{P}_n: H(P)\geqslant R_{n_0}} Q(T(P)) \\
&\leqslant (n+1)^{|\mathcal{X}|} 2^{-n\left(\min_{P:H(P)\geqslant R_{n_0}} D(P\|Q)\right)} \\
&\leqslant 2^{-n\left(\delta - |\mathcal{X}|\frac{\log(n+1)}{n}\right)}
\end{aligned}$$

而且，$|\mathcal{X}|\frac{\log(n+1)}{n} \to 0$ 意味着存在 n_1 使得 $n \geqslant n_1$ 时，$|\mathcal{X}|\frac{\log(n+1)}{n} < \delta/2$。因此，当 $n \to \infty$ 时，ε_n 以指数增长收敛于 0。

□

型的方法是组合方法，其关键是所有 n 长序列 $\mathcal{X}^n$ 可以分成至多多项式个型类，而每一个型类中序列具有相等的仅依赖于型的概率。因此，计算任何 $A \subseteq \mathcal{X}^n$ 的概率问题，可以转化为组合计数问题，即计算 A 与各个型类交集的大小。

给定 $n > 0, \varepsilon > 0$，我们可以定义型典型序列，相对于概率典型更强。一个序列 $\mathbf{x}$ 是型典型的是指：对于所有 $a \in \mathcal{X}$，当 $P(a) > 0$，有 $|\frac{1}{n}N(a|\mathbf{x}) - P(a)| < \frac{\varepsilon}{|\mathcal{X}|}$；当 $P(a) = 0$，有 $N(a|\mathbf{x}) = 0$，其中 $P(a), a \in \mathcal{X}^n$ 是真实的概率分布律。

因此，典型集由这样的序列组成，任意元素出现的频率与真实概率之间的差异不超过 $\frac{\varepsilon}{|\mathcal{X}|}$。根据强大数定律，型典型集的概率随着 $n \to +\infty$ 趋于 1。型典型集的优点在于可以用其来证明一些更强的结论，特别是在通用编码理论、率失真理论和大偏差理论中。

§2.5 通用信源编码定理

至此，我们介绍了三种证明方法。以下，我们深入分析一下这三种方法的内在区别。

典型序列方法是确定性编码，编码函数的构造需要知道信源的概率分布。只有这样，才能确定典型集与非典型集，从而给典型集中的序列进行编号。典型序列方法也给非典型序列编号，只是长度可以比典型序列编号长度长很多。典型序列方法提供了一种定长—变长编码方法，其错误概率是 0。由于码表建好之后，编译码可以通过查表实现，因而译码端无需知道信源的分布律。但典型序列方法构造的码表依赖于信源分布律，所以，若信源分布律有所变化，或用错了分布律，编码性能会下降。

随机装箱方法是一种证明技巧，其得到的序列与箱号的对应关系是随机的。极端情形下，所有的序列是有机会全部投入到第 0 号箱子的，尽管这种机会非常小。这说明，有些编码方案是很差的。随机装箱方法中证明的错误概率可以趋近于 0，是指平均错误概率。这里的“平均”，是指对于所有可能的装箱方法进行概率加权平均。这个平均错误概率不超过给定的 ε，表明一定有一种装箱方法的错误概率不超过 ε。实际上，一种随机产生的装箱方法，以很大概率是一个好的编码方法。另外，值得注意的是，随机装箱时，不需要知道信源的概率分布律。但是，译码端必须得知道概率分布律，否则，译码器无法确认箱中的典型序列。从这个角度看，随机装箱有一定的通用性，不管信源分布律 Q，只要 $H(Q) < R$，且序列被随机扔到 2^{nR} 个箱子里，而译码端知道 Q，就可以使得译码错误概率不超过既定的 ε。这种性质在分布式信源编码中非常重要。最后，可以看出随机装箱方法提供一种定长—定长的编码方案，其错误概率可以趋于 0，但不等于 0。

型方法给出了一种确定性的编码方案,并且这种编码方案有个显著特点,可应用到所有分布律 Q 满足 $H(Q) < R$ 的信源。我们前面给出的证明提供了一种定长—定长的编码方案，其错误概率趋于 0。我们也可以基于型给出定长—变长的编码方案，并证明其速率趋近于熵，而错误概率等于 0。以此作为思考题，暂略去。

【定义 2.18】 通用分组码 *对于某类信源 X，分组码 $(2^{nR}, n)$ 称为通用的，若函数 ϕ_n 和 ψ_n 不依赖于 X 的分布 Q，且若 $R > H(Q)$，则当 $n \to \infty$ 时，$\varepsilon_n \to 0$。*

【定理 2.19】 通用编码定理 *对于任意满足 $H(Q) < R$ 的分布为 Q 的信源，存在一列通用分组码 $(2^{nR}, n)$，使得 $\lim\limits_{n\to\infty} \varepsilon_n = 0$。*

□

§2.6

通用信源编码定理的逆定理

【定理 2.20】 **信源编码定理的逆定理** 若 $R < H(X)$, 设定长编码的长度为 L_n, 给定 L_n, 有 $R_n \leqslant R < H(X)$, 当 $n \to \infty$ 时，译码错误概率 $\varepsilon_n \to 1$。

□

证明:

定长编码的速率 $R_n = L_n/n \leqslant R = \log(2^{nR})/n < H(X)$

译码正确的概率为：

$$\begin{aligned}\Pr\{\psi(\phi(X^n)) = X^n\} &= \sum_{x^n:\psi(\phi(x^n))=x^n} P_{X^n}(x^n) \\ &= \sum_{x^n\in A_\varepsilon^{(n)}} P_{X^n}(x^n) + \sum_{x^n\notin A_\varepsilon^{(n)}} P_{X^n}(x^n)\end{aligned}$$

码本大小/码字个数至多为 2^{nR}，至多保证 2^{nR} 个典型序列编码正确。根据 AEP，典型序列中每个 x^n 的概率至多为 $2^{-n(H-\varepsilon)}$，所以，当 $n \to \infty$ 时，$\Pr\{\psi(\phi(X^n)) = X^n\} \leqslant 2^{nR}2^{-n(H-\varepsilon)} + \varepsilon \to 0$

所以，译码错误概率 $\varepsilon_n = 1 - \Pr\{\psi(\phi(X^n)) = X^n\} \to 1$。

□

对于定长—变长的情况，我们有如下逆定理：

若 $\varepsilon_n \to 0$，则 $\lim inf R_n \geq H(X)$。

证明梗概如下：

一个序列 X^n 若要译码不出错，则需要给予一个码字。设 $\delta > 0$，则

$$\begin{aligned}R_n &= \frac{1}{n}E\ell(X^n) \\ &\geq \frac{1}{n}\sum_{X^n:\ell(X^n)\leq n(H(X)-\delta)} P(X^n)\ell(X^n) + \frac{1}{n}\sum_{X^n:\ell(X^n)>n(H(X)-\delta)} P(X^n)\ell(X^n) \\ &> \frac{1}{n}n(H(X)-\delta)\Pr\{\ell(X^n) > n(H(X)-\delta)\} \\ &= (H(X)-\delta)(1-\varepsilon_n - \Pr\{\ell(X^n) \leq n(H(X)-\delta)\}) \\ &\to H(X)-\delta\end{aligned}$$

由于 δ 是任意的，我们得证。

在上面的证明中, 第二行用“$\geq$”是由于我们仅考虑译码正确的序列。我们把正确译码的序列分成两类，一类是码字长度不超过 $n(H(X)-\delta)$ 的，一类是码字长度大于 $n(H(X)-\delta)$ 的。这两类的概率和是 $1-\varepsilon_n$，其中 ε_n 是错误概率且 $\varepsilon_n \to 0$。我们也用到了一个事实，第一类的概率

$$\Pr\{\ell(X^n) \leq n(H(X)-\delta)\} \to 0$$

这是因为，长度不超过 $n(H(X)-\delta)$ 的码字至多有 $2^{n(H(X)-\delta)}$ 个，而这些码字若分给典型序列，其覆盖概率不超过 $2^{n(H(X)-\delta)} \cdot 2^{-n(H(X)-\frac{\delta}{2})} \to 0$，若分给非典型序列，其覆盖概率也可以任意小。

§2.7

熵的直观解释

我们已经证明了 $H(X)$ 是描述随机变量 X 所需的比特数的下确界，也就是说，任何算法所得到的比特数平均意义上不小于 $H(X)$。由此，我们可以得到 $H(X)$ 的一些性质，这些性质可以直接证明，也可以直观理解。

(1) $H(X) \geq 0$。描述一个随机变量所需比特数是不可能小于 0 的。

(2) $H(X) \leq \log|\mathcal{X}|$。为表示 X，最简单也是“最无效”的方法是用二进制展开表示 $|\mathcal{X}|$ 中所有的符号。例如，若 $|\mathcal{X}| = 16$，则，每个符号可以用 4 比特表示，所以，$H(X) \leq 4$。若 $|\mathcal{X}|$ 不是 2 的幂次，我们从 $H(X) \leq \lceil \log|\mathcal{X}| \rceil$ 可以得到 $H(X^n) \leq \lceil \log|\mathcal{X}|^n \rceil \leq \log|\mathcal{X}|^n + 1$，两边除以 n，得到 $H(X) = \frac{1}{n}H(X^n) \leq \log|\mathcal{X}| + \frac{1}{n}$，令 n 趋于无穷即可。

注：由 $H(X) \leq \log|\mathcal{X}|$，我们知道均匀分布的熵最大，这个在之前是由 $H(p)$ 的凸性证明的。

(3) $H(X,Y) \leq H(X) + H(Y)$。这个性质是说，描述二维随机变量 (X,Y) 所需的比特数不会超过分别描述 X,Y 所需的比特数之和。换言之，一种描述 (X,Y) 的方法是，不考虑 X,Y 之间的关联性，分别给出 X,Y 的二进制表示。当然，这种方法只有当 (X,Y) 独立时才是最优的。

(4) $H(X|Y) \leq H(X)$。这个性质是讲，条件熵不增。当我们观察到 Y 时，我们如何描述 X 呢？一种办法（一般来讲不是最优的）是忽略 Y 的观测值，所需比特数是 $H(X)$，所以 $H(X|Y) \leq H(X)$。

(5) Fano 不等式。设 $X \to Y \to \hat{X}$ 构成一个马尔科夫链，即，在已知 Y 的条件下，X 与 $\hat{X}$ 独立。或者，可以想像这样一个场景，X 是一个系统的输入，而 Y 是输出，$\hat{X}$ 是由 Y 按照某种算法对 X 的估计值。那么，我们有 $H(X|Y) \leq P_e \log(|\mathcal{X}| - 1) + H(P_e)$，其中 $P_e = \Pr\{X \neq \hat{X}\}$，即估计的错误概率。这个如何理解呢？我们来看不等式左边 $H(X|Y)$ 表示观测到 Y 的条件下描述 X 所需的比特数，而不等式右边给出了一种表示方法。信源产生的符号是 $X_1, X_2, ..., X_n$，我们观察到 $Y_1, Y_2, ..., Y_n$，则可以估算 $\hat{X_1}, \hat{X_2}, ..., \hat{X_n}$。由于 Y 在译码时是已知的，所以，我们可以由 Y 计算 $\hat{X}$。这样，若 $X = \hat{X}$，则我们不需要表示 X，我们只需表示 $X \neq \hat{X}$ 的情形。所以，给定 Y，我们分两步来表示 X，一是确定哪些 $\hat{X}$ 是错的，这个描述需要 $H(P_e)$ 比特；二是在错的地方，要描述 X，仅需 $\log(|\mathcal{X}| - 1)$ 比特即可，这里减去 1 是因为 $X \neq \hat{X}$ 时，X 只能取其他符号。因此，平均需要 $H(P_e) + P_e \log(|\mathcal{X}| - 1)$ 比特。所以，Fano 不等式成立。

§2.8

拓展阅读

随机装箱与随机线性装箱问题

随机装箱问题

假设有 2^{nR} 个箱子，记为 $\{0，1，2，\ldots，2^{nR-1}\}$。有 2^{nH} 个球，记为 $\{0，1，2，\ldots，2^{nH-1}\}$。将 2^{nH} 个球（$H < R$）独立均匀随机（i.u.d）地投掷到这些箱子中。问：

1. 0 号球与 1 号球同在 0 号箱子的概率是多少？
2. 0 号球与 1 号球在同一个箱子内的概率是多少？
3. 利用并集限推导存在某个箱子内有 2 个或 2 个以上球的概率的一个上界。

随机线性装箱问题

设有 2^{nR} 个箱子，这些箱子可以用比特串 $\boldsymbol{v} \in \mathbb{F}_2^{nR}$ 编号。设有 2^{nH} 个球，这些球可以用比特串 $\boldsymbol{u} \in \mathbb{F}_2^{nH}$ 编号。假设 $\mathbf{G}$ 是一个 nH 行、nR 列的均匀随机矩阵（每个元素独立均匀地取值为 0 或 1）。对于每个球 $\boldsymbol{u}$，由 $\boldsymbol{u}\mathbf{G}$ 计算箱号 $\boldsymbol{v}$，即编码 $\phi_n(\boldsymbol{u}) = \boldsymbol{u}\mathbf{G}$

1.（思考题）随机装箱的每个箱子内球的数目是随机变量。随机线性装箱每个箱子内的球的数目也是随机的，不过，固定 $\mathbf{G}$，箱子中球的数目最多取两个值，0 或某个 $N_g > 0$。
2.（证明题）随机线性装箱中，（$\boldsymbol{u}'$，$\boldsymbol{u}$）落在同一箱子的概率为 2^{-nR}（成对独立）。
3.（思考题）随机装箱中，（$\boldsymbol{u}$，$\boldsymbol{u}'$，$\boldsymbol{u}''$）落在 0 号箱子的概率为 $2^{-nR} \cdot 2^{-nR} \cdot 2^{-nR}$。这个对于随机线性装箱成立吗？

关于条件熵不增的说明

对于随机变量 X，在给定另一随机变量 $Y = y$ 的条件下，$H(X|Y = y)$ 其实会出现三种情况：$H(X|Y = y) < H(X)$，$H(X|Y = y) = H(X)$，$H(X|Y = y) > H(X)$。但是，$H(X|Y = y)$ 的统计平均 $H(X|Y) = \sum P(y)H(X|Y = y)$ 一定满足 $H(X|Y) \leq H(X)$，也就是我们说的“条件熵不增”。

【例题 2.21】 设 U 和 Y 是两个服从两点分布 $(P(0), P(1)) = (\frac{1}{2}, \frac{1}{2})$ 的随机变量。令随机变量 $X = U$ or Y，则 X 服从两点分布 $(P(0), P(1)) = (\frac{1}{4}, \frac{3}{4})$。此时，容易计算得到 $H(X|Y = 0) = 1$，$H(X|Y = 1) = 0$ 和 $0 < H(X) < 1$。因此，在给定 $Y = y$ 的条件下，X 的熵 $H(X|Y = y)$ 可能增加也可能减少。不过，根据 $H(X|Y) = P(Y = 0)H(X|Y = 0) + P(Y = 1)H(X|Y = 1) = \frac{1}{2} < H(X)$，我们可以看出 X 的条件熵 $H(X|Y)$ 并没有增加。

关于 Fano 不等式的一个注

Fano 不等式（Fano's inequality）也称为 Fano 引理（Fano lemma）是信息论中的一个定理，是 Robert Fano 是 1950 年代于麻省理工学院教授博士讨论班的时候推导的，后来放在 1961 年编写的教科书“The Theory of Information and Coding”中。Fano 不等式在信息论中，提供了译码器错误概率的下界。下面是摘自教科书的原文。

Fano's inequality Let X and Y be random variables, each taking values in the (finite) set $\{x_1, x_2, \ldots, x_r\}$. Let $P_e = \Pr\{X \neq Y\}$. Then

$$H(X|Y) \leq H_2(P_e) + P_e \log(r - 1),$$

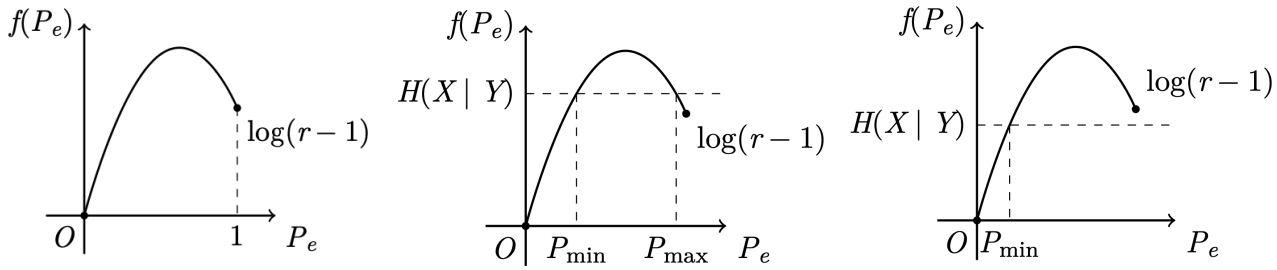
where $H_2(P_e) = -P_e \log P_e - (1-P_e)\log(1-P_e)$。

Problem 1.12 Show that Fano's inequality implies both an upper bound and a lower bound on P_e in terms of $H(X \mid Y)$. Interpret the upper bound heuristically.

从 **Problem 1.12** 来看，McEliece 的表述设置与其他教材稍有不同。我们可以根据 Fano 不等式按照下述方式得到 P_e 的上下界。记 Fano 不等式的右式为 $f(P_e)$，即

$$f(P_e) \triangleq H_2(P_e) + P_e \log(r-1).$$

由于 $f(P_e)$ 是上凸（concave）的，如图 2.2a，因此我们对 $H(X|Y)$ 的取值可以分三种情况讨论。



(a) Fano 不等式的右式 $f(P_e)$

(b) $H(X \mid Y) > \log(r-1)$

(c) $0 < H(X \mid Y) \le \log(r-1)$

图 2.2: Fano 不等式的图示

1. $H(X \mid Y) > \log(r-1)$。此时，$P_{\min} \le P_e \le P_{\max}$，参考图 2.2b
2. $0 < H(X \mid Y) \le \log(r-1)$。此时，有正下界 $P_{\min}$ 且 $P_e \le 1$（平凡）, 参考图 2.2c。
3. $H(X \mid Y) = 0$，则有平凡上下界 $0 \le P_e \le 1$。

我们可以根据下面的例子知道非平凡的界并不总是能推出来的。

【例题 2.22】 设 $X, Y \in \mathcal{X} = \mathbb{Z}_5 = \{0,1,2,3,4\}$，$Y = X + 1$. 显然

$$P_e = \Pr\{X \neq Y\} = 1.$$

此时，$H(X \mid Y) = 0 \le H_2(P_e) + P_e \log 4$。就是说存在 $H(X \mid Y) = 0$，推不出 P_e 严格小于 1.

考虑 $r = 2$，即二元情形。此时若熵 $H(X \mid Y) > 0$，则有非平凡上下界 $P_{\min} \le P_e \le P_{\max}$。直观解释是，若 $H(X \mid Y) > 0$，有模糊度，则 P_e 不可能等于 0，否则就没有模糊度了。同理，P_e 也不可能等于 1，否则意味着 Y 与 X 总相反，这样也会有 $H(X \mid Y) = 0$。

§2.9 习题

1. 设随机变量 X 服从几何分布，即，$P_X(m) = p(1-p)^{m-1}$，$0 < p < 1, m \in \mathbb{Z}_+$。计算 X 的熵 $H(X)$。
2. 随机变量 X 的熵 $H(X)$ 可能等于无穷吗？
3. 设 f 是一个函数，X 是一个随机变量。分析 $H(X)$ 和 $H(f(X))$ 的关系。
4. 设 X，Y 是两个随机变量，请证明 $H(X,Y) \leq H(X) + H(Y)$。
5. 掷一枚均匀硬币 n 次，给出概率典型序列与型典型序列的定义。分析哪些序列是“典型”的，理解哪一个定义更合理。
6. 把 m 个球均匀随机地扔进 n 个箱子。证明第一个箱子球数不少于 M 的概率至多是 $\binom{m}{M}(\frac{1}{n})^M$。
7. 把 m 个球均匀随机地扔进 n 个箱子。找一个 m 的下界，使得有个箱子里有至少两个球的概率超过 $\frac{1}{2}$。
8. 参见第2.3节，回答以下问题并加以详细讨论：
   (a) 对于随机装箱，各个箱子里的序列数目相同吗？
   (b) 对于随机线性装箱，各个箱子里的序列数目相同吗？

第三章

离散信源编码算法

§3.1 信源编码与性能评估

我们已经从理论上证明了离散无记忆信源的熵是信源压缩的极限，即对充分长的信源序列进行压缩，存在编码算法使得描述每个信源符号平均所需的比特数逼近熵，但证明中引入的编码方法并不实用。这一章我们主要介绍几种易于理解且可以实现的信源编码算法。

考虑离散无记忆信源，其产生的信源符号序列 $X_1, X_2, \cdots, X_n, \cdots$ 是独立同分布的随机变量序列，概率质量函数记为 $p(x) \triangleq P_X(x), x \in \mathcal{X}$。我们仅考虑如何有效地把 X 序列表示为二元序列。

一种直接的方法是建立一个映射 $\phi : \mathcal{X} \to \mathcal{D} = \{0,1\}^*$，然后将其拓展到序列上，即 $\phi(X_1, X_2, \cdots, X_n) = \phi(X_1)\phi(X_2)\cdots\phi(X_n)$。一个映射 ϕ 若作为信源编码方法，则应该满足如下性质：

1. 非奇异性，即 ϕ 应为单射，若 $x \in \mathcal{X}, y \in \mathcal{X}$ 是两个不同的符号，则 $\phi(x) \neq \phi(y)$。这是一个最基本的要求，否则无法正确译码。
2. 可拓展性，即 ϕ 拓展为序列上的映射后仍然是单射：若对于任意 (m,n)，$x^n \in \mathcal{X}^n, y^m \in \mathcal{X}^m$ 是两个不同的序列，则 $\phi(x^n) \neq \phi(y^m)$。这个要求并不是多余的，其不能从非奇异性推导出。

【例题 3.1】 设 $\mathcal{X} = \{a,b,c\}$，定义映射 ϕ 如下，

$$\phi(a) = 0, \phi(b) = 010, \phi(c) = 10.$$

则 ϕ 是非奇异的，但不可拓展，因为 $\phi(ac) = \phi(b)$。

给定一个信源编码映射 ϕ，我们用 $\ell(\phi(x))$ 表示 $\phi(x)$ 的长度，则平均长度定义为

$$\mathbf{E}[\ell(\phi(X))] = \sum_x p(x)\ell(\phi(x)).$$

一个编码方法的平均码长越小，则压缩效果越显著。

非奇异性与可拓展性对应着唯一可译性，即若二进制序列是编码器的输出，则可以唯一地译码出一个信源符号序列。简单的唯一可译码是前缀码，其定义如下。

【定义 3.2】 设 ϕ 是 $\mathcal{X}$ 上的编码映射，对于任意两个不同的符号 $x \in \mathcal{X}, y \in \mathcal{X}$，有 $\phi(x)$ 不是 $\phi(y)$ 的前缀，则 ϕ 称为前缀码。

在例 3.1 中给出的 ϕ 不是前缀码，因为 $\phi(a) = 0$ 是 $\phi(b) = 010$ 的前缀。给定一个前缀码 ϕ，其码长序列 $\ell(\phi(x)), x \in \mathcal{X}$ 应该满足如下 Kraft 不等式。

【定理 3.3】 **（Kraft 不等式）**

设 ϕ 是前缀码，则

$$\sum_{x\in\mathcal{X}} 2^{-\ell(\phi(x))} \leq 1.$$

反之，若 $L(x)$ 是定义在 $\mathcal{X}$ 上的满足上述 Kraft 不等式的正整数函数，则存在一个前缀码 ϕ，使得 $\ell(\phi(x)) = L(x), x \in \mathcal{X}$。

证明:

所有二元序列可以用一个有根二元概率树表示，每条边度量 $\frac{1}{2}$，对应给定父节点时候的条件概率，一个码字 $\phi(x)$ 对应从根节点出发的一条长为 $\ell(\phi(x))$ 的路径，以 $\phi(x)$ 为前缀的其他序列全部剪枝。这样处理之后，一棵树中的叶子对应一个码字，所有这些叶子的概率之和不超过 1，即 $\sum\limits_{x\in\mathcal{X}} 2^{-\ell(\phi(x))} \leq 1$。

反之，若一组正整数 $L(x)$ 满足上述不等式，则我们可以构造一棵概率树，其叶子对应一个前缀码。

□

前缀码作为一类唯一可译码，其码长满足 Kraft 不等式。一个有趣的问题是：唯一可译码是否满足 Kraft 不等式呢？我们有如下定理。

【定理 3.4】 **（McMillian）**

对于唯一可译码 $\phi : \mathcal{X} \to \{0, 1\}$，我们有

$$\sum_{x\in\mathcal{X}} 2^{-\ell(\phi(x))} \leq 1.$$

证明:

考虑 ϕ 的 k 次拓展，即 $\phi(x_1, x_2, \cdots, x_k) = (\phi(x_1), \phi(x_2), \cdots, \phi(x_k))$，其码长是 $\ell(\phi(x_1, x_2, \cdots, x_k)) = \sum\limits_{i=1}^{k} \ell(\phi(x_i))$。我们有

$$\left[\sum_{x\in\mathcal{X}} 2^{-\ell(\phi(x))}\right]^k = \sum_{x^k\in\mathcal{X}^k} 2^{-\ell(\phi(x^k))}.$$

记 $\ell_{\max} = \max\limits_{x\in\mathcal{X}} \ell(\phi(x))$，我们可以对上述等式右端进行同类项合并，

$$\sum_{x^k\in\mathcal{X}^k} 2^{-\ell(\phi(x^k))} = \sum_{m=1}^{k\ell_{\max}} N_m \cdot 2^{-m},$$

其中 N_m 是长为 m 的码字序列的个数。由于唯一可译，我们知道 $N_m \leq 2^m$，由此可得

$$\left[\sum_{x\in\mathcal{X}}^{k} 2^{-\ell(\phi(x))}\right]^k \leq k\ell_{\max},$$

两边开 k 次根号，并令 k 趋于无穷，得到 Kraft 不等式，

$$\sum_{x\in\mathcal{X}} 2^{-\ell(\phi(x))} \leq 1.$$

□

由定理 3.3 与定理 3.4 知，我们只需考虑如何设计前缀码即可。换句话讲，采用前缀码并不会增加码长。下面我们来证明最优的前缀码的平均码长不小于信源的熵。

【定理 3.5】 设 X 是定义在有限字符集 $\mathcal{X}$ 上的随机变量，则 X 的前缀码的平均码长不小于 $H(X)$，即 $L \geq H(X)$。

证明:

设 ϕ 是一个前缀码，其码长集是 $\ell(\phi(x)), x \in \mathcal{X}$。由 Kraft 不等式

$$\sum_{x\in\mathcal{X}} 2^{-\ell(\phi(x))} \leq 1.$$

满足上述约束条件的平均码长

$$L = \sum_x p(x)\ell(\phi(x))$$

的下界可由拉格朗日乘子法得到，其“最优”的码长（不考虑整数约束）是 $\ell^*(\phi(x)) = \log \frac{1}{p(x)}, x \in \mathcal{X}$.

我们可以直接证明如下，

$$\begin{aligned} L - H(X) &= \sum_x p(x) \left[\ell(\phi(x)) - \log \frac{1}{p(x)} \right] \\ &= \sum_x p(x) \log \frac{p(x)}{2^{-\ell(\phi(x))}} \\ &\geq 0. \end{aligned}$$

这里用到了相对熵的性质以及 Kraft 不等式。

□

§3.2

Huffman 码

设 X 是 $\mathcal{X}$ 上的随机变量。为方便起见，记 $\mathcal{X}=\{1,2,\cdots,m\}, P_i=P_X(X=i)$。令 $\ell_i=\lceil\log\frac{1}{P_i}\rceil$，可以验证

$$\sum_{i=1}^{m}2^{-\ell_i}\leq 1.$$

因此，存在以 $\{\ell_1,\ell_2,\cdots,\ell_m\}$ 为码长的前缀码，其平均码长 $L=\sum\limits_i P_i\ell_i=\sum\limits_i P_i\lceil\log\frac{1}{P_i}\rceil$ 满足 $H(X)\leq L\leq H(X)+1$。当我们考虑 $\mathcal{X}^n$ 上的独立同分布随机向量 $(X_1,X_2,\cdots,X_m)$ 的编码时，可以推出其平均码长 $nH(X)\leq L_n\leq nH(X)+1$。所以，$\frac{L_n}{n}$ 在 n 很大时，接近 $H(X)$，这再次证明存在编码方案，使得每符号所用的比特数平均意义下逼近信源熵 $H(X)$。

不过，$\ell_i=\lceil\log\frac{1}{P_i}\rceil, 1\leq i\leq m$ 未必是一组最优码长，最优的是如下 Huffman 编码，其平均码长最短。

Huffman 编码算法

输入：$\mathcal{X}$ 上的概率分布律，

$$P_1\geq P_2\geq\cdots\geq P_m.$$

输出：m 个二元码字，分别对应 $\mathcal{X}$ 的 m 个符号。

初始化：建一个森林，有 m 棵树，每棵树表示一个符号，赋予度量 $P_i, 1\leq i\leq m$。

递推循环：若森林中只有一棵树，则退出循环。否则，找两棵度量最小的树，将它们合并到一棵树，其度量是两棵树度量之和。

编码：循环退出后，得到一棵二元树，从根节点开始，左分支标记为 0，右分支标记为 1，每个叶子节点对应一个符号。从根节点到叶子节点的路径即对应的码字。

注：上述算法每循环一次，树的数目减少 1。由于初始只有 m 棵树，所以，上述算法经过 $m-1$ 次循环必定结束。在循环执行过程中，若有子树度量相同且都排在最轻两个之列，则可以随便选取一棵，或者选取“矮的”（层数少的）树参与合并。

【例题 3.6】 设 X 是 $\mathcal{X}$ 上的随机变量，记 $\mathcal{X}=\{1,2,3,4,5\}$，其对应的概率分布为 $0.25, 0.25, 0.2, 0.15, 0.15$。图示如下：

1. 初始化 5 棵树。

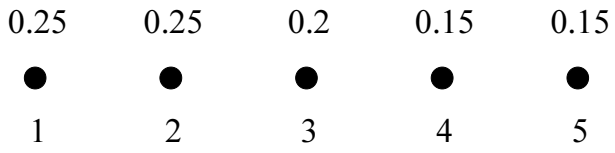


图 3.1

2. 将度量最小的两棵树合并后，得到 4 棵树。

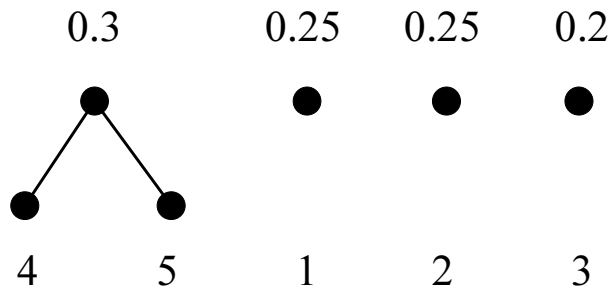


图 3.2

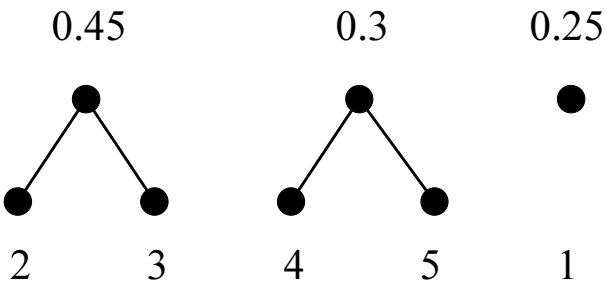


图 3.3

3. 继续上述过程，得到 3 棵树。

4. 继续重复，得到 2 棵树。

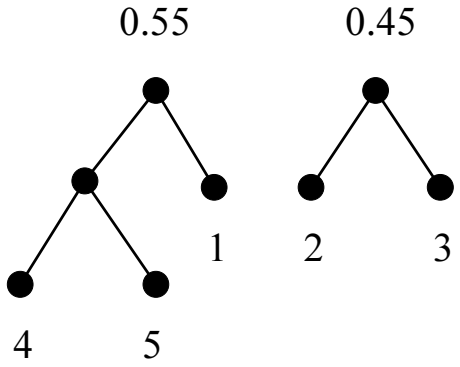


图 3.4

5. 当只剩 1 棵树时，退出循环，得到一棵二元树。从根节点开始，左分支标记为 0，右分支标记为 1。

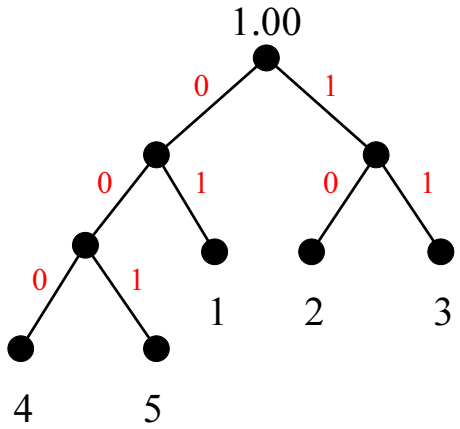


图 3.5

得到码字如下：

4：000

5：001

1：01

2：10

3：11

【定理 3.7】 Huffman 码是平均码长最短的一种码。

证明:

任何一个前缀码均可用一棵二叉树表示，其中一个叶子节点对应一个码字。一个最优码对应的二叉树记为 T，应该满足如下性质。

1. T 应是满树，即任何内部节点必有两个子节点。否则，若一个内部节点仅有一个子节点，则可把这个子节点删掉得到一个更短的前缀码。

2. 若 $P_i > P_j$，则 $\ell_i \leq \ell_j$。也就是说，概率大的符号的码长不能超过概率小的符号的码长。否则，可以将 i 与 j 对应的码字对换，得到的码有更短的平均码长。

3. 概率最小的码字在最深层，且可以通过调整顺序，使得 m 与 $m-1$ 互为兄弟，即二者仅差一个比特。

4. 若 $\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_{m-1}, \mathbf{c}_m$ 是对应 $P_1 \geq P_2 \geq \cdots \geq P_{m-1} \geq P_m$ 的最优码，则 $\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_{m-2}, \tilde{\mathbf{c}}_{m-1}$ 是对应 $P_1, P_2, \cdots, P_{m-2}, P_{m-1}+P_m$ 的最优码，其中，$\tilde{\mathbf{c}}_{m-1}$ 是 $\mathbf{c}_{m-1}$ 与 $\mathbf{c}_m$ 的共同前缀。事实上，若 $\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_{m-2}, \tilde{\mathbf{c}}_{m-1}$ 不是最短的码，则可以找一个更短的码，并把 $\tilde{\mathbf{c}}_{m-1}$ 扩展一位，得到 $\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_{m-1}, \mathbf{c}_m$ 更短。

□

§3.3 Shannon-Fano-Elias 码

设 X 是 $\mathcal{X}$ 上的随机变量。为方便起见，记 $\mathcal{X}=\{1,2,\cdots,m\}$，$P_x=P_X(X=x)$。对于 $x\geq 0$，我们定义 $F(x)=\sum\limits_{a\leq x}P_a$，$\bar{F}(x)=\sum\limits_{a<x}P_a+\frac{1}{2}P_x$。其中，当 $a\in\mathcal{X}$，$P_a>0$；当 $a\notin\mathcal{X}$，$P_a=0$。如图 3.6所示。

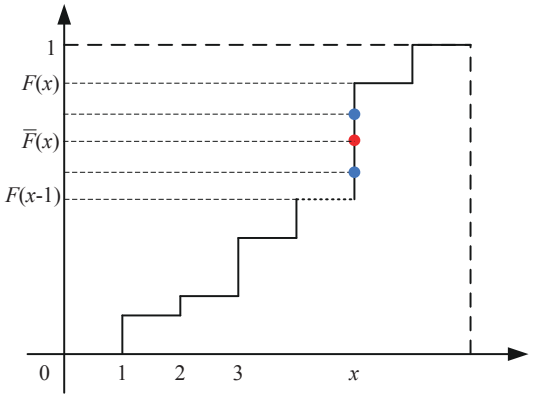


图 3.6

给定符号 $x\in\mathcal{X}$，我们可以利用 $\bar{F}(x)$ 得到它的码字，具体如下。$\bar{F}(x)$ 是一个可以用无限比特表示的实数。我们保留 $\bar{F}(x)$ 的小数点后 $\ell(x)$ 位并将它记作 $\lfloor\bar{F}(x)\rfloor_{\ell(x)}$。当 $\ell(x)=\lceil\log\frac{1}{P_x}\rceil+1$ 时，我们有

$$\bar{F}(x)-\lfloor\bar{F}(x)\rfloor_{\ell(x)}<\frac{1}{2^{\ell(x)}}<\frac{P_x}{2}=\bar{F}(x)-F(x-1).$$

因此，$\lfloor\bar{F}(x)\rfloor_{\ell(x)}\in(F(x-1),\bar{F}(x))$，意味着 $\lfloor\bar{F}(x)\rfloor_{\ell(x)}$ 就可以表示 x。我们取 $\lfloor\bar{F}(x)\rfloor_{\ell(x)}$ 小数点后 $\ell(x)$ 比特作为 x 的码字 $z_1z_2\ldots z_{\ell(x)}$。一个码字 $z_1z_2\ldots z_{\ell(x)}$ 对应一个区间 $[0.z_1z_2\ldots z_{\ell(x)},0.z_1z_2\ldots z_{\ell(x)}+\frac{1}{2^{\ell(x)}})$。根据

$$\lfloor\bar{F}(x)\rfloor_{\ell(x)}>F(x-1)$$

和

$$\lfloor\bar{F}(x)\rfloor_{\ell(x)}+2^{-\ell(x)}<\lfloor\bar{F}(x)\rfloor_{\ell(x)}+\frac{P_X(x)}{2}=\lfloor\bar{F}(x)\rfloor_{\ell(x)}+F(x)-\bar{F}(x)<F(x)$$

可以推出所有码字对应的区间不相交。因此，Shannon-Fano-Elias 码是前缀码，其平均码长

$$L=\sum_{x\in\mathcal{X}}P_x(\lceil\log\frac{1}{P_x}\rceil+1)<\sum_{x\in\mathcal{X}}P_x(\log\frac{1}{P_x}+2)=H(X)+2.$$

§3.4 算术码

对于一个信源符号序列，我们可以对每个信源符号进行独立地编码，如：Huffman 码编码，Shannon-Fano-Elias 码编码，从而产生最终的码字序列。此外，我们也可以考虑符号的前后关系，采用算术码编码。算术码编码的主要思想如下：假设已知信源符号的概率分布。给定一个编码区间，每个符号对应其中的一个子区间，子区间的长度占比等于对应符号的概率。具体地，从 $[0,1)$ 开始，根据信源符号序列，不断地更新编码区间，直至编码结束。下面举一个实例来说明 [4]。

【例题 3.8】 设信源符号是 $\{a,e,i,o,u,!\}$ 上的随机变量，其对应的概率分布如下：

表 3.1

符号	概率	区间
a	0.2	[0,0.2)
e	0.3	[0.2,0.5)
i	0.1	[0.5,0.6)
o	0.2	[0.6,0.8)
u	0.1	[0.8,0.9)
!	0.1	[0.9,1.0)

我们对符号序列 $eaii!$ 进行编码。首先，根据信源符号的概率分布将初始区间 $[0,1)$ 分割为五个子区间，如表 3.1所示。第一个编码符号是 e，对应的子区间是 $[0.2,0.5)$，因此将编码区间更新为 $[0.2,0.5)$。接下来，根据信源符号的概率分布对区间 $[0.2,0.5)$ 进行分割。下一个编码符号是 a，其所对应的子区间是 $[0.2,0.26)$，因此将编码区间更新为 $[0.2,0.26)$。依此类推，最终得到区间 $[0.23354,0.2336)$。在这个区间中任意选一个数，如 0.23358 作为序列 $eaii!$ 经过算术码编码后的结果。上述过程可参考图 3.7。

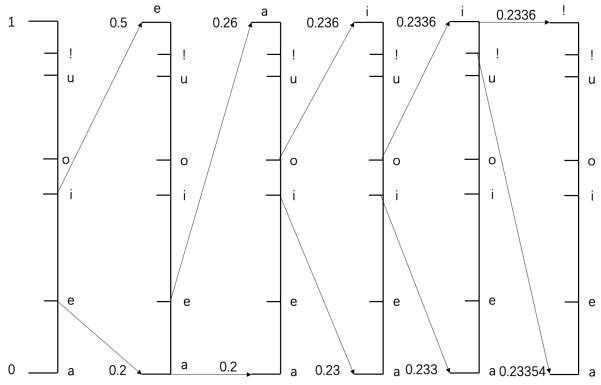


图 3.7

算术码解码的过程与编码的过程相反，相当于编码的逆运算，其主要思想如下：给定一个解码区间，按照信源符号的概率分割成若干个小区间。对于要解码的小数，确定它位于哪个子区间并输出该子区间对应的符号。进一步地，该子区间将作为下一轮的解码区间，并在下一轮解码中将它进行分割。同编码一样，从 $[0,1)$ 开始，不断地更新解码区间，直到所有的符号都被解码出来。以上面的例子为例说明。

【例题 3.9】 设信源符号是 $\{a,e,i,o,u,!\}$ 上的随机变量，其对应的概率分布如表 3.1。我们对 0.23358 进行解码。首先，按照信源符号的概率分布将初始区间 $[0,1)$ 分割为五个子区间，如表 3.1所示。观察到 0.23358 落在了区间 $[0.2,0.5)$ 中，该区间所对应的符号是 e，于是可以解码出符号 e。选择子区间 $[0.2,0.5)$ 作为下一次解码的区间，并根据信源符号的概率分布对区间 $[0.2,0.5)$ 进行分割。由于 0.23358 落在子区间 $[0.2,0.26)$ 中，该区间所对应的符号是 a，于是可以解码出符号 a。依此类推，直到解码出所有符号 $eaii!$。

§3.5

Lempel-Ziv 78 编码

设信源符号集 $\mathcal{X}=\{x_1,x_2,\cdots,x_k\}$，共 k 个符号。设输入信源符号序列为 $\mathbf{u}=(u_1,u_2,\cdots,u_L)$。编码时将此序列分成不同的段。分段的规则为：尽可能取最少个相连的信源符号，并保证各段都不相同。开始时，先取一个符号作为第一段，然后维续分段。若出现与前面相同的符号时，就再取紧跟后面的一个符号一起组成一个段，使之与前面的段不同。这些分段构成字典。当字典达到一定大小后，再分段时就应查看有否与字典中的短语相同，若有重复就添加符号，以便与字典中短语不同，直至信源符号序列结束。这样，不同的段内的信源符号可看成一短语，可得不同段所对应的短语字典表。每个短语对应的码字为：其最长前缀字段所在的段号 + 其末尾符号对应的序号。设 $\mathbf{u}$ 构成的字典中的短语共有 $M(\mathbf{u})$ 个。若编为二元码，段号所需码长 $n=\lceil\log M(\mathbf{u})\rceil$，每个符号需要的码长为 $\lceil\log k\rceil$。单符号的码字的段号为 0。

Lempel-Ziv 78 编码的编码方法很便捷，译码也很简单，可以一边译码一边建立字典，只需要传输字典的大小，无需传输字典本身。当编码的信源序列较短时，该算法性能似乎会变坏，但是当序列增长时，编码效率会提高，平均码长会逼近信源熵。说明如下：将符号集大小为 k，长度为 L 的信源序列 $\mathbf{u}$ 分为 $M(\mathbf{u})$ 个码段。设最长的段的长度为 ℓ_{max}，可以证明，每个源符号的平均码长有

$$H(U)+\frac{\log k}{\ell_{max}}<\bar{n}<H(U)+\frac{\log k+2}{\ell_{max}}$$

将 k 趋于无穷时，ℓ_{max} 也趋于无穷，平均码长趋近于信源熵。

【例题 3.10】 设 $\mathcal{X}=\{x_1,x_2,x_3,x_4\}$，信源序列为 $x_1x_2x_1x_3x_2x_4x_2x_4x_3x_1x_1x_4\cdots$，按照分段规则，可以分为 $x_1,x_2,x_1x_3,x_2x_4,x_2x_4x_3,x_1x_1,x_4$，共 7 段。每个符号编码如表 3.2所示，字典表如表 3.3所示。

表 3.2

x_1	x_2	x_3	x_4
00	01	10	11

表 3.3

段号	短语	编码
1	x_1	000 00
2	x_2	000 01
3	x_1x_3	001 10
4	x_2x_4	010 11
5	$x_2x_4x_3$	100 10
6	x_1x_1	001 00
7	x_4	000 11

该字典表中，共有 4 个信源符号和 7 个短语。因此，我们用 2 比特表示信源符号，3 比特表示段号。那么，一个短语总共使用 5 比特。

§3.6

信息论的基本量与性质

这一节，我们介绍一些信息论的基本信息量及其性质[1]。

【定义 3.11】 （熵）
一个概率分布为 $p(x)$ 的离散型随机变量 X 的熵 $H(X)$ 定义为

$$\begin{aligned} H(X) &= -\sum_{x\in\mathcal{X}} p(x)\log p(x) \\ &= -\mathbf{E}_{p(x)}\log p(X). \end{aligned}$$

有时也将上面的量记为 $H(p)$，表示概率分布 $p(x)$ 的熵。其中，对数 log 所用的底是 2，熵的单位用比特表示。此外，我们约定 $0\log 0 = 0$。

【定义 3.12】 （联合熵）
对于一对服从联合分布 $p(x,y)$ 的离散随机变量 (X,Y)，其联合熵 $H(X,Y)$ 定义为

$$\begin{aligned} H(X,Y) &= -\sum_{x\in\mathcal{X}}\sum_{x\in\mathcal{Y}} p(x,y)\log p(x,y) \\ &= -\mathbf{E}_{p(x,y)}\log p(X,Y). \end{aligned}$$

【定义 3.13】 （条件熵）
对于一对服从联合分布 $p(x,y)$ 的离散随机变量 (X,Y)，其条件熵 $H(Y|X)$ 定义为

$$\begin{aligned} H(Y|X) &= \sum_{x\in\mathcal{X}} p(x)H(Y|X=x) \\ &= -\sum_{x\in\mathcal{X}} p(x)\sum_{y\in\mathcal{Y}} p(y|x)\log p(y|x) \\ &= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log p(y|x) \\ &= -\mathbf{E}_{p(x,y)}\log p(Y|X). \end{aligned}$$

【定理 3.14】 （熵的性质）
1. $H(X)\geqslant 0$。
2. $H_b(X) = (\log_b a)\, H_a(X)$ 。
3. (条件不增加熵) 对两个随机变量 X 和 Y，有

$$H(X|Y)\leqslant H(X),$$

当且仅当 X 与 Y 相互独立，等号成立。
4. $H(X_1,X_2,\cdots,X_n)\leqslant\sum_{i=1}^{n} H(X_i)$，当且仅当随机变量 X_i 相正独立，等号成立。
5. $H(X)\leqslant\log|\mathcal{X}|$，当且仅当 X 服从 $\mathcal{X}$ 上的均匀分布，等号成立。

【定义 3.15】 （相对熵）
两个概率分布为 $p(x)$ 和 $q(x)$ 之间的相对熵定义为

$$\begin{aligned} D(p||q) &= \sum_{x\in\mathcal{X}} p(x)\log\frac{p(x)}{q(x)} \\ &= \mathbf{E}_{p(x)}\log\frac{p(X)}{q(X)}. \end{aligned}$$

在上述定义中，我们约定 $0\log\frac{0}{0}=0$，$0\log\frac{0}{q}=0$，$p\log\frac{p}{0}=\infty$ (基于连续性)。因此，若存在字符 $x\in\mathcal{X}$ 使得 $p(x)>0$，$q(x)=0$，则有 $D(p\|q)=\infty$ 。

[1]为了内容的完整性，我们会在这节中对已在前文中出现过的部分内容进行重述。

【定义 3.16】 （互信息）

考虑两个随机变量 X 和 Y，它们的联合概率密度函数为 $p(x,y)$，边际概率密度函数分别是 $p(x)$ 和 $p(y)$。互信息 $I(X;Y)$ 为联合分布 $p(x,y)$ 和乘积分布 $p(x)p(y)$ 之间的相对熵，即:

$$\begin{aligned} I(X;Y) &= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \\ &= D(p(x,y)\|p(x)p(y)) \\ &= \mathbf{E}_{p(x,y)}\log\frac{p(X,Y)}{p(X)p(Y)}. \end{aligned}$$

【定理 3.17】 （相对熵和互信息的性质）

1. $I(X;Y)=H(X)-H(X|Y)=H(Y)-H(Y|X)=H(X)+H(Y)-H(X,Y)$。
2. $D(p||q)\geqslant 0$，当且仅当 $\forall x\in\mathcal{X}$，$p(x)=q(x)$，等号成立。
3. $I(X;Y)=D(p(x,y)||p(x)p(y))\geqslant 0$，当且仅当 $p(x,y)=p(x)p(y)$（即 X 与 Y 相互独立），等号成立。
4. 若 $|\mathcal{X}|=m$，u 是 $\mathcal{X}$ 上的均匀分布，则 $D(p||u)=\log m-H(p)$ 。

【定理 3.18】 （链式法则）

1.（熵）

$$H\left(X_1,X_2,\cdots,X_n\right)=\sum_{i=1}^{n}H\left(X_i|X_{i-1},\cdots,X_1\right).$$

2.（互信息）

$$I\left(X_1,X_2,\cdots,X_n;Y\right)=\sum_{i=1}^{n}I\left(X_i;Y|X_1,X_2,\cdots,X_{i-1}\right).$$

3. （相对熵）

$$D\left(p(x,y)||q(x,y)\right)=D\left(p(x)||q(x)\right)+D\left(p(y|x)||q(y|x)\right).$$

【定理 3.19】 （Fano 不等式）

对于两个随机变量 X 和 Y，记 $P_e=\Pr\{X(Y)\neq X\}$，则 $H\left(P_e\right)+P_e\log|\mathcal{X}|\geqslant H(X|Y)$。

【定理 3.20】 （数据处理不等式）

若 $X\to Y\to Z$ 构成马尔科夫链，则 $I(X;Y)\geqslant I(X;Z)$。

【定理 3.21】 （信息量的凹凸性）

1. 熵 $H(p)$ 关于概率分布 p 是凹的。
2. 相对熵 $D(p||q)$ 关于一对概率分布 (p,q) 是凸的。
3. 设一对随机变量 $(X,Y)\sim p(x,y)=p(x)p(y|x)$。如果条件概率分布 $p(y|x)$ 固定，则互信息 $I(X;Y)$ 是关于概率分布 $p(x)$ 的凹函数；而 $p(x)$ 的分布固定，则互信息 $I(X;Y)$ 是关于 $p(y|x)$ 的凸函数。

§3.7 习题

1. Shannon 码编码需要知道概率质量函数 $p_i, 0 \le p_i \le 1, \sum_{i=1}^{M} p_i = 1$，其码长 $\ell_i = \lceil \log \frac{1}{p_i} \rceil$。若用了一个不匹配的概率质量函数 $q_i, 0 \le q_i \le 1$，$\sum_{i=1}^{M} q_i = 1$，则其码长是 $\tilde{\ell_i} = \lceil \log \frac{1}{q_i} \rceil$。
   (1) 求其平均码长，并计算不匹配概率对应的平均码长与匹配概率对应的平均码长之差。
   (2) 请设计一个程序例子，用 Matlab 程序仿真，体现上述计算的合理性。

2. 记 $\mathcal{X} = \{0, 1\}$ 为二元集，$\mathcal{X}^n$ 为所有 n 重二元数组。$x^n \in \mathcal{X}^n$ 的汉明重量是 x^n 中 1 的个数。
   (1) $\mathcal{X}^n$ 中有多少个二元数组？
   (2) 重量是 w 的 n 重二元数组有几个？
   (3) 以 $n = 4$ 为例，重为 2 的数组有 6 个：$(0,0,1,1), (0,1,0,1), (1,0,0,1), (0,1,1,0), (1,0,1,0), (1,1,0,0)$，因此我们称其可以承载 2 比特信息。即可以建立一个单射：$00 \to (0,0,1,1)$，$01 \to (0,1,0,1)$，$10 \to (1,0,0,1)$，$11 \to (0,1,1,0)$。记 $\mathcal{X}^n$ 中重量为 w 的全体二元数组为 T_w，求最大的 k，使得存在 $\mathcal{X}^k \to T_w$ 的单射。
   (4) **思考题**：任意给定 n，w，如何建立 $\mathcal{X}^k \to T_w$ 之间的单射？此单射可以称为等重编码，那么应该如何译码？

第四章

信道编码定理

§4.1 一般码的定义

【定义 4.1】设 $\mathcal{A}$ 是一个非空集合，$\mathcal{A}$ 上所有 n-重组的全体，记作 $\mathcal{A}^n = \{(a_0, a_1, \cdots, a_{n-1}) | a_i \in \mathcal{A}, 0 \leq i < n\}$，也称为 $\mathcal{A}$ 的 n-重笛卡儿积。一个码（或码表），记作 $\mathscr{C}(n, M)$，可以表示成一个 $M \times n$ 的阵列

$$\mathscr{C} = \begin{bmatrix} \mathbf{c}^{(0)} \\ \mathbf{c}^{(1)} \\ \vdots \\ \mathbf{c}^{(M-1)} \end{bmatrix},$$

其中 $\mathbf{c}^{(i)} \in \mathcal{A}^n, 0 \leq i < M$，称为码字。我们称 $\mathscr{C}$ 是定义在 $\mathcal{A}$ 上的码。

【例题 4.2】 设 $\mathcal{A} = \{0, 1\}$，则 $\mathcal{A}$ 上的码通常称为二元码。例如：

$$\mathscr{C} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

若 $\mathcal{A}$ 是有限集合，且 $\mathcal{A}$ 的元素个数 $|\mathcal{A}| > 2$，我们称 $\mathcal{A}$ 上的码为多元码。

【例题 4.3】 实数集 $\mathbb{R}$ 上的码有时候也称为星座（constellation），如：

$$\mathscr{C} = \begin{bmatrix} +1 & +1 \\ -1 & +1 \\ -1 & -1 \\ +1 & -1 \end{bmatrix}$$

称为正交相移键控（quadrature phase shift keying，QPSK）信号星座，可以用等间隔分布在平面单位圆上的四个点表示。

一般码的编译码从概念上来讲是很简单的。编码是从消息集合 $\mathcal{M} = \{0, 1, \cdots, M-1\}$ 到 $\mathscr{C}$ 的一个映射，即 $\phi : \mathcal{M} \longmapsto \mathscr{C}$。若要传送消息 $i \in \mathcal{M}$，编码器输出第 i 个码字 $\mathbf{c}^{(i)}$，即 $\phi(i) = \mathbf{c}^{(i)}$。

设 $\mathbf{c}^{(i)}$ 经过一个信道之后，接收端收到一个 n-重组 $\mathbf{y} \in \mathcal{B}^n$，其中 $\mathcal{B}$ 是信道的输出字符集合。通常情况下，$\mathbf{c}^{(i)}$ 与 $\mathbf{y}$ 之间的对应关系不是确定性的，而是一个“一对多”的对应关系。确切地说，给定 $\mathbf{c}^{(i)}$，$\mathbf{y}$ 是按照某

种条件概率分布律分布在接收空间 $\mathcal{B}^n$ 上。一个译码准则就是如何从 $\mathbf{y}$ 推测 $\mathbf{c}^{(i)}$。从概念上讲，一个译码准则就是把 $\mathcal{B}^n$ 划分成 M 个互不相交的区域，即 $\mathcal{B}^n = \bigcup_{i=0}^{M-1} \mathcal{D}^{(i)}$，且 $\mathcal{D}^{(i)} \bigcap \mathcal{D}^{(j)} = \varnothing, i \neq j$。译码描述为一个映射 $\psi : \mathcal{B}^n \longmapsto \mathcal{M}$。具体地，若 $\mathbf{y} \in \mathcal{D}^{(i)}$，则 $\psi(\mathbf{y}) = i$。有时候，我们也把整个接收空间划分成 $M+1$ 个互不相交的区域 $\mathcal{B}^n = \bigcup_{i=0}^{M} \mathcal{D}^{(i)}$。若接收 $\mathbf{y} \in \mathcal{D}^{(M)}$，则译码器输出“译码失败”。这种情况下，我们称为不完全译码，而之前的称为完全译码。

上面描述的编译码“算法”从原理上看很简单，但是显然不适合很大的 M。一方面，我们需要足够的空间存储码表；另一方面，我们译码时也要搜索整个码表。

§4.2

一般码的性能参数

一个码 $\mathscr{C}(n,M)$ 的重要参数是码率与误码性能。码率的定义比较简单：$R=\frac{\log M}{n}$。一般我们取以 2 为底的对数。此时，码率的单位是 比特/符号，或者 比特/信道使用，意指平均每使用一次信道，可以传输的信息量。误码性能讨论如下。

由于接收码字与发送码字之间的依赖关系不是确定性关系，因而译码结果有可能与发送消息不一致。设信道转移概率已知，记为 $P(\mathbf{y}|\mathbf{x}),\forall\mathbf{x}\in\mathcal{A}^n,\forall\mathbf{y}\in\mathcal{B}^n$。注意，若 y 是连续变量，则 $P(y|x)$ 表示条件概率密度，对 y “求和”可以理解为“积分”。记 E 是译码错误事件，则给定发送消息是 i 的条件下的译码错误概率是

$$P(E|\text{发送 }i)=\sum_{\mathbf{y}\notin\mathcal{D}^{(i)}}P(\mathbf{y}|\mathbf{c}^{(i)}).$$

若每个消息发送的概率已知，则译码错误概率

$$P(E)=\sum_{i=0}^{M-1}P(\text{发送 }i)\cdot P(E|\text{发送 }i),$$

此概率称为误帧率（frame error rate, FER）。

前面已经看到，一个确定的译码算法对应一个划分。常见的准则之一是最小错误概率译码准则，即寻找一个划分，使得 $P(E)$ 达到最小。而要确定一个划分，就是要对每个 $\mathbf{y}\in\mathcal{B}^n$ 找一个“归宿” $\mathcal{D}^{(i)}$，根据贝叶斯公式，

$$P(\text{发送 }i|\mathbf{y})=\frac{P(\text{发送 }i)\cdot P(\mathbf{y}|\mathbf{c}^{(i)})}{P(\mathbf{y})}.$$

我们可以计算给定 $\mathbf{y}$ 的条件下的所有后验概率 $P(\text{发送 }i|\mathbf{y}),0\le i<M$。这组概率可以这样直观理解，若我们独立重复地观察所讨论的编码传输系统，则在充分长的观察样本序列中，$\mathbf{y}$ 可能发生了很多次，比如发生了 N 次；在这 N 次中，大致有 $N\cdot P(\text{发送 }i|\mathbf{y})$ 次是对应发送 i 的。因此，为了最小化误码率（最大化正确译码概率），我们应该把 $\mathbf{y}$ 判决为 $\hat{i}$，使得 $P(\text{发送 }\hat{i}|\mathbf{y})=\max\limits_{i}P(\text{发送 }i|\mathbf{y})$。当 $P(\text{发送 }i)=\frac{1}{M},0\le i<M$ 时，最大后验概率译码可以简化为求 $\hat{i}$，使得 $P(\mathbf{y}|\text{发送 }\hat{i})=\max\limits_{i}P(\mathbf{y}|\text{发送 }i)$。这个准则称为最大似然序列译码（maximum likelihood decoding, MLD），其在发送消息等概率时，等价于最大后验概率译码，可以使得误码率最小。在先验概率 $P(\text{发送 }i)$ 未知或者定义不明确时，我们通常采用最大似然序列译码。

当然，若有多个 i，使得 $P(\text{发送 }i|\mathbf{y})$ 达到最大，我们可以按照某个既定规则选择其中一个，或者宣告译码失败。这个译码准则使得误码率达到最小，又称为最大后验概率序列译码（sequence maximum a posteriori decoding, Sequence MAP）。

我们考虑离散无记忆信道，其输入集为 $\mathcal{X}$，输出集记为 $\mathcal{Y}$，转移概率记为 $P(y|x),x\in\mathcal{X},y\in\mathcal{Y}$。这里，对于给定 $P(y|x),x\in\mathcal{X},y\in\mathcal{Y}$ 是条件概率质量函数或者是条件概率密度。我们的讨论以 x,y 是有限集为主，若是连续集，则可以通过量化转化为有限集。

【定义 4.4】 图 4.1 展示了一个二元对称信道。这个信道的输入集为 $\mathcal{X}=\{0,1\}$，输出集为 $\mathcal{Y}=\{0,1\}$，转移概率为 $P(y|x)=\begin{cases}p, & x\neq y\\ 1-p, & x=y\end{cases}$。也就是说，在这个信道上，发送字符会以 p 的概率发生翻转。

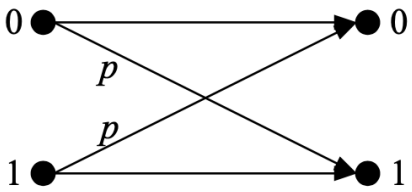


图 4.1: 二元对称信道

【例题 4.5】 考虑例 4.2 中的二元码 $\mathscr{C}$，发送码字 $\boldsymbol{c}\in\mathscr{C}$，接收端收到接收向量 $\boldsymbol{y}$。当 $\boldsymbol{y}\notin\mathscr{C}$ 时，发生的错误图样 $\boldsymbol{e}\in\{001,\ 010,\ 100,\ 111\}$ 能被检查出来，其中 $\boldsymbol{y}=\boldsymbol{c}+\boldsymbol{e}$。

【定义 4.6】 图 4.2 展示了一个二进制相移键控调制下的加性高斯白噪声信道。符号 X，Y 和 Z 分别表示信道输入，信道输出和信道噪声。这个信道的输入集为 $\mathcal{X}=\{+1,-1\}$，输出集为 $\mathcal{Y}=\mathbb{R}$，转移概率为 $P(y|x)=\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(y-x)^2}{2\sigma^2}}$。也就是说，噪声服从均值为 0，方差为 σ^2 的高斯分布。

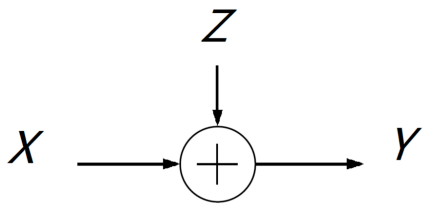


图 4.2: 二进制相移键控调制下的加性高斯白噪声信道

【例题 4.7】 考虑加性高斯白噪声信道，信道输入集为 $\mathcal{X}=\{-1,1\}$，信道噪声 $Z\sim\mathcal{N}(0,\sigma^2)$。当接收端收到 $y=0.7$，根据后验概率译码准则，我们可以计算

$$P(x=1|y=0.7)=\frac{P(x=1)P(y=0.7|x=1)}{P(y=0.7)},$$

$$P(x=-1|y=0.7)=\frac{P(x=-1)P(y=0.7|x=-1)}{P(y=0.7)}.$$

当先验概率均匀时，MAP 等价于 MLD。其中，

$$P(y=0.7|x=1)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(0.7-1)^2}{2\sigma^2}\right),$$

$$P(y=0.7|x=-1)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(0.7+1)^2}{2\sigma^2}\right).$$

【例题 4.8】 考虑加性高斯白噪声信道和例 4.3 中的码 $\mathscr{C}$，接收端收到接收向量 $\boldsymbol{y}=(y_0,\ y_1)\in\mathbb{R}^2$。给定任何一个码字记 $\boldsymbol{c}=(x_0,\ x_1)\in\mathscr{C}$，我们可以计算

$$P(\boldsymbol{y}|\boldsymbol{c})=\prod_{i=0}^{1}\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y_i-x_i)^2}{2\sigma^2}\right).$$

最大似然译码输出 $\arg\max\limits_{\boldsymbol{c}\in\mathscr{C}}P(\boldsymbol{y}|\boldsymbol{c})$。可以看到，在加性高斯白噪声信道下，最大似然译码等价于最小欧氏距离译码，即 $\arg\max\limits_{\boldsymbol{c}\in\mathscr{C}}P(\boldsymbol{y}|\boldsymbol{c})\iff\arg\min\limits_{\boldsymbol{c}\in\mathscr{C}}\left((y_0-x_0)^2+(y_1-x_1)^2\right)$。

【定义 4.9】 两个随机变量 X,Y 的互信息为

$$I(X;Y)=\sum_{x,y}P(x,y)\log\frac{P(y|x)}{P(y)}.$$

由定义直接验证，我们有如下性质：

1.

$$\begin{aligned}I(X;Y)&=H(X)-H(X|Y)\\&=H(Y)-H(Y|X).\end{aligned}$$

2. $I(X;Y)\geq 0$，当且仅当 X 与 Y 相互独立时，等号成立，即对于任意的 $x\in\mathcal{X},y\in\mathcal{Y}$，我们有 $P(x,y)=P(x)P(y)$。

3. $I(X_1,X_2;Y)=I(X_1;Y)+I(X_2;Y|X_1)$，这里条件互信息的定义类似于互信息的定义，

$$I(X_2;Y|X_1)=\mathbf{E}\log\frac{P(Y|X_1,X_2)}{P(Y|X_1)},$$

其中期望 $\mathbf{E}$ 是对 X_1,X_2,Y 的联合分布而言。

【定义 4.10】 设 $(X_1,Y_1),(X_2,Y_2),\cdots,(X_n,Y_n)$ 是独立同分布的随机向量序列，对 $\forall\varepsilon>0$，我们称序列 $(x_1,x_2,\cdots,x_n)$ 与 $(y_1,y_2,\cdots,y_n)$ 是联合典型的，是指

$$|\frac{1}{n}\log P(x_1,x_2,\cdots,x_n)-H(X)|\le\varepsilon$$

$$|\frac{1}{n}\log P(y_1,y_2,\cdots,y_n)-H(Y)|\le\varepsilon$$

$$|\frac{1}{n}\log P((x_1,y_1),(x_2,y_2),\cdots,(x_n,y_n))-H(X,Y)|\le\varepsilon.$$

我们把联合典型序列全体记作 $A_\varepsilon^{(n)}$，有如下引理，

【引理 4.11】 1. 当 $n\to\infty$ 时，$P((X^n,Y^n)\in A_\varepsilon^{(n)})\to 1$，即联合典型序列集的概率收敛到 1。

2. $|A_\varepsilon^{(n)}|\le 2^{n(H(X,Y)+\varepsilon)}$，即联合典型序列集的大小受联合熵限制。

3. 若 $(\widetilde{X^n},\widetilde{Y^n})\sim P(x^n)P(y^n)$，即 $\widetilde{X^n}$ 与 $\widetilde{Y^n}$ 独立，但与 (X^n,Y^n) 具有相同的边缘分布，则

$$P((\widetilde{X^n},\widetilde{Y^n})\in A_\varepsilon^{(n)})\le 2^{-n(I(X;Y)-3\varepsilon)}.$$

当 n 充分大，有

$$P((\widetilde{X^n},\widetilde{Y^n})\in A_\varepsilon^{(n)})\ge(1-\varepsilon)2^{-n(I(X;Y)-3\varepsilon)},$$

即独立抽取的 $\widetilde{X^n}$ 与 $\widetilde{Y^n}$ 使联合典型序列的概率很小，但也不会太小。

【定理 4.12】 设 $x\in\mathcal{X}$ 与 $y\in\mathcal{Y}$ 是离散无记忆信道对应的输入与输出符号，转移概率分布律是 $P_{Y|X}(y|x),x\in\mathcal{X},y\in\mathcal{Y}$。对于任意给定的输入分布 $P_X(x),x\in\mathcal{X}$，设 $R<I(X;Y)$。则存在一个编译码方案 $\mathscr{C}(n,M)$ 满足

$$R_n=\frac{\log M}{n}\ge R,$$

且当 $n\to\infty$ 时，$\mathrm{FER}\to 0$。

这个定理至少有两种证明方法。一种是基于 Shannon 引入的典型序列 [1]，一种是基于 Gallager 引入的错误指数 [5]。前者从概念上讲较为简单，下面简述之。

证明:

码表产生：我们用随机方式构造一个码表 $\mathscr{C}(n,M)$，其可以看作一个 $M*n$ 的阵列，其中每个元素独立且服从相同的分布 $P(x),x\in\mathcal{X}$。

编码发送：若要传输消息 $i,1\le i\le M$，我们发送 $\mathscr{C}(n,M)$ 的第 i 个码字，即阵列的第 i 行。

译码：设收到 $\mathbf{y}$，译码器找与 $\mathbf{y}$ 联合典型的码字序列。若有且仅有一个码字序列与接收序列 $\mathbf{y}$ 联合典型，则输出该序列作为发送码字的估计，否则报错。

误码性能分析：从随机码表的构造看，各行具有对称性。我们不妨假设发送的是第一个码字，即码表的第一行，记作 $\mathbf{x}_1$。记 E_i 表示事件码字 i 与 $\mathbf{y}$ 联合典型，则有两类错误，一是 $\bar{\mathrm{E}}_1$，即发送序列 $\mathbf{x}_1$ 与 $\mathbf{y}$ 不联合典型，二是有 $\mathrm{E}_i,i\ne 1$ 发生。总之，利用并集限，以及 $P(\mathrm{E}_i)$ 对所有 $i\ge 2$ 相等，我们有

$$\begin{aligned}P(\mathrm{E}) &= P(\bar{\mathrm{E}}_1)+P(\bigcup_{i\ge 2}\mathrm{E}_i)\\ &\le P(\bar{\mathrm{E}}_1)+MP(\mathrm{E}_2).\end{aligned}$$

上式第一项 $P(\bar{\mathrm{E}}_1)\to 0$，因为发送序列与接收序列联合典型的概率是趋于 1 的（引理 4.11），而

$$\begin{aligned}MP(\mathrm{E}_2) &\le M2^{-n(I(X;Y)-3\varepsilon)}\\ &= 2^{-n(I(X;Y)-R_n-3\varepsilon)}.\end{aligned}$$

所以，当 $R < I(X;Y)$，我们可以取 $R_n = (I(X;Y)+R)/2$，其满足 $R_n \geq R$，且

$$R_n \leq I(X;Y) - \frac{I(X;Y)-R}{2} \triangleq I(X;Y) - \delta,$$

因此可以选择 ε 使得 $P(\mathrm{E}) \to 0$。

□

从上述证明可以看出，定理对于任意给定的输入分布均成立，由此我们引入如下定义。

【定义 4.13】 离散无记忆信道的信道容量定义为

$$C = \max_{P(x)} I(X;Y),$$

即输入输出之间的最大平均互信息量。

可以证明，对一般信道而言，存在一个分布律 $P^*(x), x \in \mathcal{X}$，使得 $I(X;Y)$ 达到最大。在这种情况下，如果我们在上述定理的证明中取 $P^*(x)$ 来产生随机码表，我们就可以得到如下的信道编码定理。

【定理 4.14】 记离散无记忆信道的容量是 C，对于任意 $R < C$，存在编译码方案，其码率不小于 R，而错误概率趋于 0。反之，若 $R > C$，这样的编译码方案不存在。

证明:

定理的第一部分已经证明，只需用 $P^*(x), x \in \mathcal{X}$ 产生码表即可。

定理的第二部分，需要用到 Fano 不等式。具体地，设消息 $W, 0 \leq W < 2^{nR}$ 是均匀分布的随机变量，则有一个 Markov 链 $W \to X^n \to Y^n$，由此得到

$$\begin{aligned} H(W) &= I(W;Y^n) + H(W|Y^n) \\ &\leq I(W, X^n; Y^n) + H(W|Y^n) \\ &= I(X^n;Y^n) + I(W;Y^n|X^n) + H(W|Y^n) \\ &= I(X^n;Y^n) + H(W|Y^n). \end{aligned}$$

其中 $I(W;Y^n|X^n) = 0$，因为已知 X^n 的条件下，W 与 Y^n 是相互独立的。

而 $I(X^n;Y^n) = H(Y^n) - H(Y^n|X^n), H(Y^n) \leq \sum_{i=1}^{n} H(Y_i)$ 与 $H(Y^n|X^n) = \sum_{i=1}^{n} H(Y_i|X_i)$，我们有

$$I(X^n;Y^n) \leq \sum_{i=1}^{n} I(X_i;Y_i) \leq nC.$$

由 $H(W) = nR$ 及 Fano 不等式 $H(W|Y^n) \leq 1 + P_e \cdot nR$，我们有

$$nR \leq nC + 1 + P_e \cdot nR.$$

若编码方案使得 $\lim_{n\to\infty} P_e = 0$，我们必有 $R \leq C$，这就证明了信道编码的逆定理部分。 □

影响一个码的误码性能的因素很多，其中一个重要的因素是距离特性。

【定义 4.15】 设 $\mathcal{A}$ 是一个非空集合，我们称 $\mathcal{A}$ 为距离空间，是指在 $\mathcal{A}$ 上引入了一个二元实函数 $\rho(x,y)$，满足下列三个条件：对于任意的 $x, y, z \in \mathcal{A}$，

(1) $\rho(x,y) \geqslant 0$，且 $\rho(x,y) = 0$ 当且仅当 $x = y$（非负性）；

(2) $\rho(x,y) = \rho(y,x)$（对称性）；

(3) $\rho(x,z) \leqslant \rho(x,y) + \rho(y,z)$（三角不等式：三角形两边之和不小于第三边）。

我们称 ρ 是 $\mathcal{A}$ 上的一个距离。同一个集合上可以根据研究需要，定义不同的距离。若为避免混淆，以 ρ 为距离的距离空间 $\mathcal{A}$ 记作 $(\mathcal{A}, \rho)$。

【例题 4.16】 对于任意给定的非空集合 $\mathcal{A}$，我们总可以引入汉明（Hamming）距离：

$$\rho(x,y) = \begin{cases} 0, & x = y \\ 1, & x \neq y \end{cases},$$

其中 $x, y \in \mathcal{A}$。此时我们称 $(\mathcal{A}, \rho)$ 为汉明空间，有时记 ρ 为 d_H。

设 d_H 是 $\mathcal{A}$ 上的汉明距离，则可以定义 $\mathcal{A}^n$ 上的距离：$d_H(\mathbf{x},\mathbf{y})=\sum\limits_{t=0}^{n-1} d(x_t,y_t)$。此距离也称为 $\mathcal{A}^n$ 上的汉明距离。码 $\mathscr{C}$ 的最小汉明距离定义为：$d_{\min}=\min\{d(\mathbf{c}_i,\mathbf{c}_j)|\mathbf{c}_i,\mathbf{c}_j\in\mathscr{C},i\neq j\}$。

由于任意两个码字至少有 $d_{\min}$ 个位置不同，所以任意两个码字中不可能存在完全相同的 $n-d_{\min}+1$ 位，并且，存在两个码字，其中的 $n-d_{\min}$ 位相同。

我们有如下的命题，称为辛格尔顿（Singleton）界。

【命题 4.17】 设有限字符集 $\mathcal{A}$ 上的码 $\mathscr{C}(n,M)$ 具有最小汉明距离 $d_{\min}$，则 $M\leq|\mathcal{A}|^{n-d_{\min}+1}$。

证明：从 $\mathscr{C}$ 中任意删除掉 $d_{\min}-1$ 列，则剩余的子阵列仍然具有不同的行。由不同行的个数最多 $|\mathcal{A}|^{n-d_{\min}+1}$ 得证。

给定一个码，我们可以固定一个码字，不妨设为 $\mathbf{c}^{(0)}$，然后考察所有码字与 $\mathbf{c}^{(0)}$ 的汉明距离。这些距离可以用一个母函数的形式记录：

$$A(X)=\sum_{i=0}^{M-1} X^{\rho(\mathbf{c}^{(i)},\mathbf{c}^{(0)})}.$$

这个多项式（合并同类项）的系数也称为 $\mathscr{C}$ 的条件距离谱。形象地说，是站在 $\mathbf{c}^{(0)}$ 的位置四周巡望，看看其他码字离多远。我们也可以考察任何两个码字之间的距离。不同的距离至多有 $\binom{M}{2}=\frac{M(M-1)}{2}$ 种。

【例题 4.18】 考虑二元码

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

若从 $\mathbf{c}^{(0)}=(0,0,0)$ 的角度看，其汉明距离谱 $A(X)=1+3X^2$，其中，有一个码字距其 0，三个码字距其 2。

【例题 4.19】 若一个码是 $n-$ 维实空间 $\mathbb{R}^n$ 的一个非空子集，我们可以考虑欧几里德（Euclid）距离：$\rho(\mathbf{x},\mathbf{y})=\sqrt{\sum\limits_{t=0}^{n-1}(x_t-y_t)^2}$。对于欧氏距离空间的码，用平方距离谱取代距离谱更方便。

【例题 4.20】 考虑 QPSK 星座，

$$\mathscr{C}=\begin{bmatrix}+1&+1\\-1&+1\\-1&-1\\+1&-1\end{bmatrix},$$

若从 $\mathbf{c}^{(0)}=(+1,+1)$ 的角度看，其欧氏平方距离谱 $A(X)=1+2X^4+X^8$。这个多项式表示有一个星座点（其实就是自身）离 (+1, +1) 的距离为 0，有两个星座点离 (+1, +1) 的平方距离为 4，有一个星座点离 (+1, +1) 的平方距离为 8。

在实际工程中，M 通常可以写成 2^k 的形式。此时，消息集可以看作是 $\{0,1\}^k$。一个 k-重二元组，经过编码之后映射成一个 n-重码元组。在这种场景下，我们可以定义误比特率（bit error rate, BER）。若发送的消息是 $U\in\{0,1\}^k$，而译码消息是 $\hat{U}$，则 BER 定义为

$$\text{BER}=\frac{\mathbf{E}d_H(U,\hat{U})}{k},$$

其中 $\mathbf{E}$ 表示数学期望。

§4.3 一般码的仿真

仿真的主要目的是估计 FER 或 BER。仿真模块按照类可以分为三个模块：一个模块是信源---信宿；一个模块是一般码，包括编译码；一个模块是信道。我们建议初学者仅考虑时间离散无记忆信道，其数学描述如下。

若输入一个 $n-$ 重组 $\mathbf{x} \in \mathcal{A}^n$，则在概率 $P(\mathbf{y}|\mathbf{x})$ 的控制下输出 $\mathbf{y} \in \mathcal{B}^n$，且

$$P(\mathbf{y}|\mathbf{x}) = \prod_{i=0}^{n-1} P(y_i|x_i).$$

在仿真中，我们假定 $P(y_i|x_i)$ 是已知的。更多情况下，我们约定 y_i 与 x_i 之间通过运算式子建立联系，而这个运算过程中有随机数参与。

信源---信宿的角色是产生待发送的数据，比较译码的结果，并统计错误事件。一般码模块要定义码长，定义码表的大小等，也就是要定义阵列 $\mathscr{C}$。此外，一般码模块还要包括编译码函数等。

这些模块定义好之后，可以按照下述流程完成仿真。

0. 初始化：num_BE = 0，num_FE = 0，tot_Fr = 0。

1. 循环：

while（tot_Fr $\leq$ max_Fr 或 num_FE $\leq$ max_FE）

按照均匀分布产生一个消息 $u \in \mathcal{M}$；

编码得到码字 $\mathbf{c}^{(u)}$；

经过信道得到 $\mathbf{y}$；

译码得到 $\hat{u}$；

tot_Fr $\longleftarrow$ tot_Fr+1；

如果 $\hat{u} \neq u$，则 num_FE $\longleftarrow$ num_FE+1, num_BE $\longleftarrow$ num_BE+$d_H(\hat{u}, u)$。

2. 计算：

$$\text{FER} = \frac{\text{num\_FE}}{\text{tot\_Fr}},$$

$$\text{BER} = \frac{\text{num\_BE}}{\text{tot\_Fr} * k}, \text{ 其中 } k = \log_2 |\mathcal{M}|.$$

在上述流程中，max_Fr 与 max_FE 是预先设定的，用以控制仿真时间与估计精度。通常情况下，max_FE 设置在 100~1000 之间，而 max_Fr 要设置到 $\frac{\text{max\_FE}}{\text{FER}}$ 的级别。虽然 FER 是未知的，不过可以有个预期的数量级。

【例题 4.21】 考虑二进制相移键控调制下的加性高斯白噪声信道，[7，4，3] 汉明码硬判决译码、软判决译码和不编码的仿真性能曲线对比如下图所示。

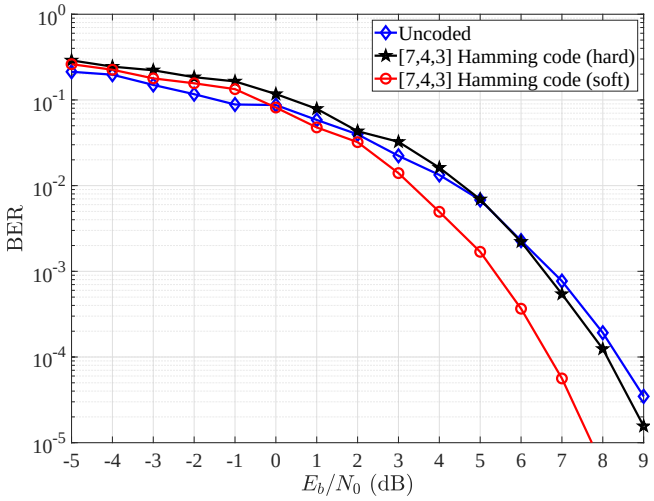


图 4.3: [7，4，3] 汉明码与不编码的 BER 仿真性能对比曲线

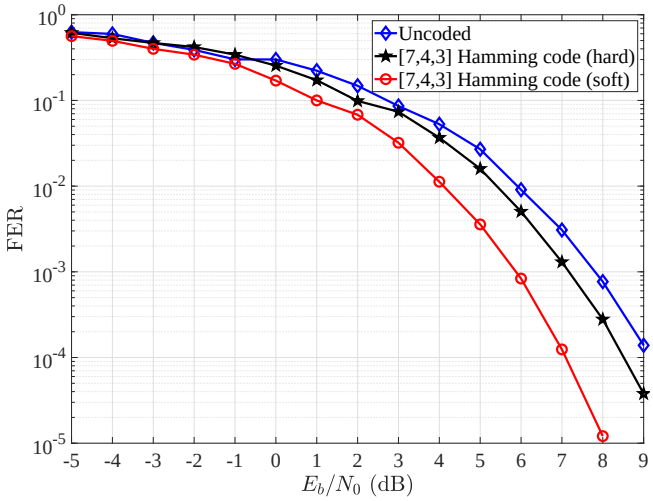


图 4.4: [7，4，3] 汉明码与不编码的 FER 仿真性能对比曲线

§4.4

随机一般码集合

由定义，我们知道一个一般码也就是一个 $M \times n$ 的阵列。如果编码符号集 $\mathcal{A}$ 是有限集，则这样的阵列有 $|\mathcal{A}|^{nM}$ 个，因而也就有 $|\mathcal{A}|^{nM}$ 个不同的码表。特别需要指出的是，按照定义，一个码表中可以含有相同的行。从编码的角度，把不同的消息映射到相同的码字的做法显然是不可取的。但这样做，给研究带来诸多方便。

我们在证明信道编码定理时候，已经接触到随机码的概念。这里给出两个具体的例子，帮助读者进一步理解。一个随机码集合可以描述为一个码表的集合，且其中每个码表均被赋予一个概率（或概率密度）。

【例题 4.22】 考虑 $\mathcal{A} = \{0, 1\}$ 上的 $(2, 2)$ 码集，这样的码表共有 16 个，

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}, \cdots, \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

若我们赋予每个阵列概率 $\frac{1}{16}$，可以得到一个随机码集。当然，我们也可以赋予不同的概率。从概念上讲，任何概率向量 $(p_0, p_1, \cdots, p_{15}), p_i \geq 0, \sum p_i = 1$, 均可以定义随机码集。

【例题 4.23】 考虑 $\mathbb{R}$ 上的 $(2, 2)$ 码集，这样的码表具有 $\begin{pmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{pmatrix}$ 的形式，因而有无穷多个码表。如果我们约定 c_{ij} 是按照正态分布 $\mathcal{N}(0, 1)$ 独立产生的，我们同样也得到了一个随机码集。

随机码集是 Shannon（香农）研究信道编码定理时引入的重要工具。Shannon 的基本想法回顾如下。

考虑一个随机码集 (n, M)，给定其中一个特定的码表 $\mathscr{C}$，我们可以定义 FER，这个 FER 可以看作 $\mathscr{C}$ 的函数。进而，我们可以定义码集的平均 FER，即

$$\text{FER} = \sum_{\mathscr{C}} P(\mathscr{C}) \cdot \text{FER}(\mathscr{C}).$$

当 $\mathscr{C}$ 是定义在连续集上时，上面的求和可以换成积分。

一个简单的事实是，对于某个 $\epsilon > 0$，如果可以证明 FER $\leq \epsilon$，则一定有某个特定的码表使得 FER$(\mathscr{C}) \leq \epsilon$。在很多情况下，分析一个具体码的性能比较困难，而给出一个平均 FER 的上界却相对“容易”（当然，对于初学者也不那么容易）。

§4.5 错误指数

信道编码定理也可以利用错误指数来证明。为此，我们仍然考虑离散无记忆信道。设 $P_X(x), x \in \mathcal{X}$ 是输入分布，而 $W_{X|Y}(y|x), x \in \mathcal{X}, y \in \mathcal{Y}$ 是信道转移概率，其对应的互信息是 $I(X;Y)$。为了简便，后文省略 P_X 和 $W_{X|Y}$ 的下标。

考虑码长为 N，码率为 R，总共可以传输 $M \triangleq \lfloor 2^{NR} \rfloor$ 个消息。假设在每次传输中，消息是随机均匀产生的。在发送端，采用随机编码的方式产生码本，记作 $\mathcal{C} = \{\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(M-1)}\}$，其中，$\boldsymbol{x}^{(i)} = (x_0^{(i)}, x_1^{(i)}, \ldots, x_{N-1}^{(i)}) \in \mathcal{X}^N$。这里，每个码字比特 $x_j^{(i)}$ 以 $P(x)$ 的概率生成。对于消息 $m \in \{0, 1, \ldots, M-1\}$，我们使用 N 次信道 W 发送其对应的码字 $\boldsymbol{x}^{(m)}$。此时，码字 $\boldsymbol{x}^{(m)}$ 会以 $W^N(\boldsymbol{y}|\boldsymbol{x}^{(m)}) = \prod_{i=0}^{N-1} W(y_i|x_i^{(m)})$ 的概率变成序列 $\boldsymbol{y}$。在接收端，采用最大似然译码准则将接受序列 $\boldsymbol{y}$ 判决为消息 $\hat{m}$，即，$\hat{m}(\boldsymbol{y}) = \arg \max\limits_{0 \le m' \le M-1} W^N(\boldsymbol{y}|\boldsymbol{x}^{(m')})$。因此，给定一个码本 $\mathcal{C} = \{\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(M-1)}\}$，可以推出其在最大似然译码下的错误率为

$$P_e(\mathcal{C}) \triangleq \sum_{i=0}^{M-1} \frac{1}{M} \sum_{\boldsymbol{y}: \hat{m}(\boldsymbol{y}) \neq m} W^N(\boldsymbol{y}|\boldsymbol{x}^{(i)}). \tag{4.1}$$

然而，想要计算一个特定的码本的 $P_e(\mathcal{C})$ 不是件容易的事。不过，我们可以计算一个码簇 (N, R, P_X)[1]在最大似然译码下的平均错误率，记作 $\overline{P}_e(N, R, P_X)$。下面，我们对 $\overline{P}_e(N, R, P_X)$ 进行推导。首先，在随机编码的情况下，一个消息 m 对应的码字是随机的，概率分布为 $P^N(\boldsymbol{x}^{(m)}), \boldsymbol{x}^{(m)} \in \mathcal{X}^N$。给定消息 m，发送码字 $\boldsymbol{x}^{(m)}$ 和接受序列 $\boldsymbol{y}$，定义事件

$$A_{m'} \triangleq \{W^N(\boldsymbol{y}|\boldsymbol{x}^{(m')}) \ge W^N(\boldsymbol{y}|\boldsymbol{x}^{(m)})\},\ m' \neq m, \tag{4.2}$$

表示基于接收序列 $\boldsymbol{y}$，消息 m' 比消息 m 更似然。由于消息 m' 的码字也是随机的，我们可以推出，对于任意的 $s > 0$，有

$$\begin{aligned} \Pr\{A_{m'}\} &= \sum_{\boldsymbol{x}^{(m')}: W^N(\boldsymbol{y}|\boldsymbol{x}^{(m')}) \ge W^N(\boldsymbol{y}|\boldsymbol{x}^{(m)})} P^N(\boldsymbol{x}^{(m')}) && (4.3) \\ &\le \sum_{\boldsymbol{x}^{(m')}} P^N(\boldsymbol{x}^{(m')}) \frac{(W^N(\boldsymbol{y}|\boldsymbol{x}^{(m')}))^s}{(W^N(\boldsymbol{y}|\boldsymbol{x}^{(m)}))^s}. && (4.4) \end{aligned}$$

进一步地，在消息为 m，发送码字为 $\boldsymbol{x}^{(m)}$ 和接受序列为 $\boldsymbol{y}$ 的条件下，最大似然译码错误的概率为

$$P_e\{\text{error}|m, \boldsymbol{x}^{(m)}, \boldsymbol{y}\} \le \Pr\{\bigcup_{m' \neq m} A_{m'}\} \le (\sum_{m' \neq m} \Pr\{A_{m'}\})^\rho = ((M-1)\Pr\{A_{m' \neq m}\})^\rho. \tag{4.5}$$

这里，$0 < \rho \le 1$。而码簇 (N, R, P_X) 的最大似然错误概率则可由下式给出:

$$\begin{aligned} &\overline{P}_e(N, R, P_X) && (4.6) \\ &\triangleq \sum_{i=0}^{M-1} \frac{1}{M} \sum_{\boldsymbol{x}^{(i)} \in \mathcal{X}^N} \sum_{\boldsymbol{y} \in \mathcal{Y}^N} P^N(\boldsymbol{x}^{(i)}) W^N(\boldsymbol{y}|\boldsymbol{x}^{(i)}) P_e\{\text{error}|i, \boldsymbol{x}^{(i)}, \boldsymbol{y}\} && (4.7) \\ &= \sum_{\boldsymbol{x}^{(m)} \in \mathcal{X}^N} \sum_{\boldsymbol{y} \in \mathcal{Y}^N} P^N(\boldsymbol{x}^{(m)}) W^N(\boldsymbol{y}|\boldsymbol{x}^{(m)}) P_e\{\text{error}|m, \boldsymbol{x}^{(m)}, \boldsymbol{y}\} && (4.8) \\ &\le \sum_{\boldsymbol{x}^{(m)} \in \mathcal{X}^N} \sum_{\boldsymbol{y} \in \mathcal{Y}^N} P^N(\boldsymbol{x}^{(m)}) W^N(\boldsymbol{y}|\boldsymbol{x}^{(m)}) \left((M-1) \sum_{\boldsymbol{x}^{(m')}} P^N(\boldsymbol{x}^{(m')}) \frac{(W^N(\boldsymbol{y}|\boldsymbol{x}^{(m')}))^s}{(W^N(\boldsymbol{y}|\boldsymbol{x}^{(m)}))^s} \} \right)^\rho && (4.9) \\ &= (M-1)^\rho \sum_{\boldsymbol{y} \in \mathcal{Y}^N} \left[\sum_{\boldsymbol{x}^{(m)} \in \mathcal{X}^N} P^N(\boldsymbol{x}^{(m)}) (W^N(\boldsymbol{y}|\boldsymbol{x}^{(m)}))^{1-s\rho} \right] \left[\sum_{\boldsymbol{x}^{(m')} \in \mathcal{X}^N} P^N(\boldsymbol{x}^{(m')}) (W^N(\boldsymbol{y}|\boldsymbol{x}^{(m')}))^s \right]^\rho. && (4.10) \end{aligned}$$

[1]这里的码簇是指一个码本集合，集合中的一个码本 $\mathcal{C} = \{\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(M-1)}\}$ 可以被看作是一个概率为 $\Pr\{\mathcal{C}\} = \prod_{i=0}^{M-1} P^N(\boldsymbol{x}^{(i)})$ 的样本。

取 $s=1/(1+\rho)$，$0\le\rho\le 1$。由于码字比特之间是独立生成的而信道是无记忆的，我们有

$$\overline{P}_e(N,R,P_X)\le (M-1)^{\rho}\sum_{\boldsymbol{y}\in\mathcal{Y}^N}\left[\sum_{\boldsymbol{x}\in\mathcal{X}^N}P^N(\boldsymbol{x})(W^N(\boldsymbol{y}|\boldsymbol{x}))^{1/(1+\rho)}\right]^{1+\rho} \tag{4.11}$$

$$=(M-1)^{\rho}\left(\sum_{y\in\mathcal{Y}}\left[\sum_{x\in\mathcal{X}}P(x)(W(y|x))^{1/(1+\rho)}\right]^{1+\rho}\right)^{N} \tag{4.12}$$

$$\le e^{NR\rho}\left(\sum_{y\in\mathcal{Y}}\left[\sum_{x\in\mathcal{X}}P(x)(W(y|x))^{1/(1+\rho)}\right]^{1+\rho}\right)^{N} \tag{4.13}$$

$$=\exp\{-N[E_0(\rho,P)-\rho R]\}, \tag{4.14}$$

其中，

$$E_0(\rho,P)=-\ln\sum_{y\in\mathcal{Y}}\left[\sum_{x\in\mathcal{X}}P(x)(W(y|x))^{1/(1+\rho)}\right]^{1+\rho}. \tag{4.15}$$

定义随机编码的错误指数为

$$E_r(R)=\max_{0\le\rho\le 1}\max_{P}[E_0(\rho,P)-\rho R]. \tag{4.16}$$

因此，

$$\overline{P}_e(N,R,P_X)\le\exp\{-NE_r(R)\}. \tag{4.17}$$

最后，我们指出只有当 $R<I(X;Y)$ 时，$E_r(R)>0$。从而，可以得到结论：存在一个码本 $\mathcal{C}$，在最大似然译码下有 $P_e(\mathcal{C})\to 0\ (N\to\infty)$。

§4.6 习题

1. 考虑例 4.2 中的二元码，若我们作一个映射 $0 \longmapsto +1,\ 1 \longmapsto -1$，则可以得到实数域上的码。请问，若两个二元码字之间的汉明距离是 d，这两个二元码字在实数空间的欧氏距离是多少？

2. 在例 4.3 中，我们给出了 QPSK 星座，类似地，我们可以定义 8-PSK 星座，即等间隔分布在单位圆周上的 8 个点。

（1）请画出 8-PSK 的星座图，并给出其距离谱函数。

（2）若某个时刻从中选一点 x 发送，接收到 $y = x + w$，其中，w 是二维高斯白噪声。试问，最大似然译码准则是什么？在此译码准则下，接收空间是如何划分的？

3. 对于如下二元对称信道：

输入 $x \in \{0, 1\}$，

输出 $y \in \{0, 1\}$，

转移概率 $P(y|x) = \begin{cases} p, & x \neq y \\ 1-p, & x = y \end{cases}$，$x, y \in \{0, 1\}$。

现给定码表 $\{000, 111\}$。

编码：$0 \longmapsto 000,\ 1 \longmapsto 111$。

一个码字在信道上传输之后，以 p 的概率发生反转，因此，可能有 8 种接收序列。试问，最大似然译码准则是什么？在此译码准则下，接收空间是如何划分的？

4. 给定一矩阵 $\mathbf{H} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$，考虑码（或码表）$\mathscr{C}$ 满足 $\mathscr{C} = \{\boldsymbol{x} \in \mathbb{F}_2^7 \mid \mathbf{H}\boldsymbol{x}^T = \mathbf{0}\}$。

（1）该码表有多少个码字？

（2）任取 $\boldsymbol{c} \in \mathscr{C}$ 在二元对称信道上传输，得到接收向量 $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$，其中，$\boldsymbol{e} \in \mathbb{F}_2^7$ 为错误图样。分析在二元对称信道下该码的译码错误概率 （注：$\mathscr{C}$ 为（7，4）汉明码）。

第五章

线性分组码

上一章我们引入了 $\mathcal{A}$ 上的码的概念，其中 $\mathcal{A}$ 是一个一般集合，没有特定的结构。从这一章开始，我们将讨论几种常见的有结构的码。

§5.1 有限域

设 $\mathbb{F}$ 是一个非空集合，至少包含两个不同的元素。在 $\mathbb{F}$ 上定义两种二元运算，分别记作“$+$”与“$\times$”，即

$+$：$\forall \alpha, \beta \in \mathbb{F}, \exists! \gamma \in \mathbb{F}$（“$\exists!$”表示“存在且唯一”），使得 $\gamma = \alpha + \beta$，称为和。

$\times$：$\forall \alpha, \beta \in \mathbb{F}, \exists! \gamma \in \mathbb{F}$，使得 $\gamma = \alpha \times \beta$，称为积。

为简单起见，我们通常记 α 与 β 的积为 $\alpha \cdot \beta$ 或 $\alpha\beta$。需要指出的是，所谓二元运算是指两个操作数对应一个确定的结果。上述和与积的运算符号“$+$”与“$\times$”只是表示符号，与实数的和运算、积运算或“大相径庭”。

我们称 $\mathbb{F}$ 为域，是指所定义的二元运算满足以下九条规律：

1. $(\mathbb{F}, +)$ 是一个交换群，即

1.1 交换律：$\forall \alpha, \beta \in \mathbb{F}$，有 $\alpha + \beta = \beta + \alpha$。

1.2 结合律：$\forall \alpha, \beta, \gamma \in \mathbb{F}$，有 $(\alpha + \beta) + \gamma = \alpha + (\beta + \gamma)$。

1.3 零元：$\forall \alpha \in \mathbb{F}, \exists \theta \in \mathbb{F}$，使得 $\alpha + \theta = \theta + \alpha = \alpha$。

1.4 负元：$\forall \alpha \in \mathbb{F}, \exists \beta$，使得 $\alpha + \beta = \theta$。

可以证明，“零元”是唯一的，通常简记为 0。也可以证明，给定 $\alpha \in \mathbb{F}$，负元 β 也是唯一的，简记为 $-\alpha$（相当于实数中的相反数）。

2. $(\mathbb{F} \setminus \{0\}, \times)$ 是一个交换群，即

2.1 交换律：$\forall \alpha, \beta \in \mathbb{F} \setminus \{0\}$，有 $\alpha \cdot \beta = \beta \cdot \alpha$。

2.2 结合律：$\forall \alpha, \beta, \gamma \in \mathbb{F} \setminus \{0\}$，有 $(\alpha \cdot \beta) \cdot \gamma = \alpha \cdot (\beta \cdot \gamma)$。

2.3 幺元：$\forall \alpha \in \mathbb{F} \setminus \{0\}, \exists e \in \mathbb{F}$，使得 $\alpha \cdot e = e \cdot \alpha = \alpha$。

2.4 逆元：$\forall \alpha \in \mathbb{F} \setminus \{0\}, \exists \beta$，使得 $\alpha \cdot \beta = e$。

同样可以证明，“幺元”是唯一的，通常简记为 1。也可以证明，给定 $\alpha \in \mathbb{F} \setminus \{0\}$，逆元 β 也是唯一的，简记为 α^{-1}（相当于实数中的倒数）。

3. 分配律：$\forall \alpha, \beta, \gamma \in \mathbb{F}$，有 $\alpha \cdot (\beta + \gamma) = \alpha \cdot \beta + \alpha \cdot \gamma$。

【例题 5.1】 所有实数构成的集合在通常意义下构成一个域，记作 $\mathbb{R}$。我们常见的例子还有有理数域（记作 $\mathbb{Q}$）、复数域（记作 $\mathbb{C}$）。但是全体整数构成的集合在通常的运算下不构成一个域。考虑如下例子：2 是一个整数，但我们找不到整数 x，使得 $2x = 1$。

+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

×	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

+	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

×	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1
3	0	3	1	2

通俗地讲，一个域就是可以做“加、减、乘、除”四项基本运算的集合。我们之前遇到的域多是无限集合，而在编码领域还经常用到有限域，即元素个数 $|\mathbb{F}| < \infty$ 的域。

【例题 5.2】 最简单的有限域 $\mathbb{F} = \{0, 1\}$，其运算规定如下：

$0 + 0 = 0$，$0 + 1 = 1$，$1 + 1 = 0$；

$0 \cdot 0 = 0$，$0 \cdot 1 = 0$，$1 \cdot 1 = 1$。

通常，对于素数 p，集合 $\mathbb{F} = \{0, 1, \cdots, p-1\}$ 在模 p 意义下做加法与乘法，构成一个域，记作 $\mathbb{F}_p$，也称为素数域。

【例题 5.3】 有限域 $\mathbb{F}_3 = \{0, 1, 2\}$ 的运算可以用下述两个表格定义。

同样地，我们可以定义 $\mathbb{F}_5$，$\mathbb{F}_7$，等等。

【例题 5.4】 $\mathbb{Z}_4 = \{0, 1, 2, 3\}$ 在模 4 运算下，不是域。因为 $2 \cdot 2 = 4 = 0 \pmod 4$，这不符合我们之前的认识：即非零元素之积不应该为零。但是按照域的定义，$\mathbb{Z}_4$ 违反了哪一条呢？比如，我们可以断言，找不到 x，使得 $2x = 1$。详细证明略去。

但是，如果我们规定如下运算，

我们可以验证 $\mathbb{F}_4 = \{0, 1, 2, 3\}$ 构成一个域。

从上面的例子来看，域不仅与集合有关，还与集合上定义的运算有关。对于有限域，有如下优美的结论。

【命题 5.5】 当且仅当 $q = p^m$（p 是素数，m 是正整数）时，存在（在同构意义下是唯一存在的）有限域 $\mathbb{F}_q$。

注意，当 $m > 1$ 时，$\mathbb{F}_{p^m}$ 的运算法则不是简单的模 q 运算。有限域由于其元素个数有限，有许多组合计数的问题，具有简单而美的结构。

§5.2 线性空间

设 $\mathbb{F}$ 是一个域，考虑其 $n-$ 重笛卡儿积 $\mathbb{F}^n$。我们在 $\mathbb{F}^n$ 上定义如下运算，称为向量加法。

对于 $\mathbf{a}=(a_1,a_2,\cdots,a_n)\in\mathbb{F}^n,\mathbf{b}=(b_1,b_2,\cdots,b_n)\in\mathbb{F}^n$，定义 $\mathbf{c}=\mathbf{a}+\mathbf{b}=(c_1,c_2,\cdots,c_n)$，其中 $c_t=a_t+b_t,1\leq t\leq n$。

我们也可以定义一个称为数乘的运算。

对于 $\lambda\in\mathbb{F},\mathbf{a}=(a_1,a_2,\cdots,a_n)\in\mathbb{F}^n$，定义数乘 $\lambda\cdot\mathbf{a}=(\lambda a_1,\lambda a_2,\cdots,\lambda a_n)$，也简记为 $\lambda\mathbf{a}$。

向量加法可以看作是“信号叠加”，而数乘可以看作是“信号缩放”，这两个运算都是线性运算。我们可以验证上述定义的两个运算满足如下八条规律。因此，$\mathbb{F}^n$ 是一般线性空间的一个特例。

1. 向量加法：
   1.1 交换律：$\mathbf{a}+\mathbf{b}=\mathbf{b}+\mathbf{a}$。
   1.2 结合律：$(\mathbf{a}+\mathbf{b})+\mathbf{c}=\mathbf{a}+(\mathbf{b}+\mathbf{c})$。
   1.3 零向量：$\mathbf{a}+\mathbf{0}=\mathbf{0}+\mathbf{a}=\mathbf{a}$。
   1.4 负向量：$\forall\mathbf{a},\exists\mathbf{b}$，使得 $\mathbf{a}+\mathbf{b}=\mathbf{0}$。
2. 数乘
   2.1 规范性：$1\cdot\mathbf{a}=\mathbf{a}$。
   2.2 累积性：$(\lambda_1\cdot\lambda_2)\mathbf{a}=\lambda_1\cdot(\lambda_2\mathbf{a})$。
3. 分配律
   3.1 $\lambda(\mathbf{a}+\mathbf{b})=\lambda\mathbf{a}+\lambda\mathbf{b}$。
   3.2 $(\lambda_1+\lambda_2)\mathbf{a}=\lambda_1\mathbf{a}+\lambda_2\mathbf{a}$。

上述八条规律是线性空间的共性，用于定义一般线性空间。在一般线性空间中，向量未必是 $n-$ 重数组的形式。下面我们着重考虑 $\mathbb{F}_q^n$，即有限域上的 $n-$ 重组全体构成的线性空间。在此约定下，我们不仅可以考虑一般线性空间中的维数、线性相关、线性无关、秩、极大线性无关组等概念，还可以计数。

【定义 5.6】 $\mathbb{F}_q^n$ 的一个非空子集 U，若在向量加法与数乘下封闭，则称 U 为一个线性子空间，即 $\forall\mathbf{a},\mathbf{b}\in U$，有 $(\mathbf{a}-\mathbf{b})\in U$；$\forall\mathbf{a}\in U,\lambda\in\mathbb{F}$，有 $\lambda\cdot\mathbf{a}\in U$。

【例题 5.7】 考虑有限域 $\mathbb{F}_2=\{0,1\}$ 及其上定义的线性空间 $\mathbb{F}_2^3$，这个线性空间包括 8 个向量 $\{\mathbf{v}=(v_1,v_2,v_3),v_i\in\mathbb{F}_2\}$。这个空间的自然基是 $\mathbf{e}_1=(0,0,1),\mathbf{e}_2=(0,1,0),\mathbf{e}_3=(1,0,0)$，就是说：

1. $\mathbf{e}_1,\mathbf{e}_2,\mathbf{e}_3$ 线性无关；
2. 任何 $\mathbb{F}_2^3$ 中的向量都可以由 $\mathbf{e}_1,\mathbf{e}_2,\mathbf{e}_3$ 线性表出。

我们可以验证，$\mathbf{a}_1=(0,0,1),\mathbf{a}_2=(0,1,1),\mathbf{a}_3=(1,1,1)$ 也满足上面两条性质，因而也是一组基。

§5.3

线性分组码

【定义 5.8】 设 $\mathbb{F}$ 是一个有限域。一个维数是 k，长度是 n 的线性分组码，记作 $\mathscr{C}[n,k]$，是 $\mathbb{F}^n$ 的一个 k 维线性子空间。

【例题 5.9】 我们可以验证，

$$\mathscr{C}=\{(0,0,0),(0,1,1),(1,0,1),(1,1,0)\}$$

中的向量在向量加法与数乘运算下封闭，因而构成一个线性子空间，也称为一个码，记作 $\mathscr{C}[3,2]$，因其是二维。

既然 $\mathscr{C}[n,k]$ 是线性子空间，则存在一组基。设 $\{\mathbf{g}_0,\mathbf{g}_1,\cdots,\mathbf{g}_{k-1}\}$ 是这样一组基，构造一个矩阵

$$\mathbf{G}=\begin{bmatrix}\mathbf{g}_0\\ \mathbf{g}_1\\ \vdots\\ \mathbf{g}_{k-1}\end{bmatrix},$$

这个矩阵称为生成矩阵，缘由如下：任何一个码字 $\mathbf{c}\in\mathscr{C}[n,k]$，总可以找到一组元素 $u_0,u_1,\cdots,u_{k-1}\in\mathbb{F}$，使得 $\mathbf{c}=u_0\mathbf{g}_0+u_1\mathbf{g}_1+\cdots+u_{k-1}\mathbf{g}_{k-1}$，即 $\mathbf{c}=\mathbf{uG}$。

对于给定的线性分组码，我们可以定义与之对偶的码。

【定义 5.10】 设 $\mathscr{C}[n,k]$ 是一个线性分组码，我们记 $\mathscr{C}^\perp$（“$\perp$”表示垂直）是与之对偶的码，其中 $\mathbf{x}\in\mathscr{C}^\perp$ 当且仅当 $\mathbf{x}$ 与所有 $\mathscr{C}$ 中的码字正交，即 $\forall\mathbf{c}\in\mathscr{C},\sum_{i=0}^{n-1}x_ic_i=0$。

需要指出的是，上述“点积”形式是域 $\mathbb{F}$ 上的运算。可以验证，对偶码自身也是线性分组码，维数是 $n-k$。设

$$\mathbf{H}=\begin{bmatrix}\mathbf{h}_0\\ \mathbf{h}_1\\ \vdots\\ \mathbf{h}_{(n-k-1)}\end{bmatrix},$$

是对偶码的生成矩阵，则称其为原码的校验矩阵。

综上所述，对于线性分组码 $\mathscr{C}[n,k]$，我们至少有两种描述，即

$$\begin{aligned}\mathscr{C}&=\{\mathbf{c}|\mathbf{c}=\mathbf{uG},\mathbf{u}\in\mathbb{F}^k\},\\&=\{\mathbf{c}|\mathbf{c}=\mathbf{Hc}^{\mathrm{T}}=\mathbf{0},\mathbf{c}\in\mathbb{F}^n\}.\end{aligned}$$

其中，“T”表示转置。注意，一个码可有不同形式的生成矩阵与校验矩阵。

§5.4

编译码算法

线性分组码的编码算法比较简单，设 $\mathscr{C}[n,k]$ 是一个线性分组码，$\mathbf{G}$ 是一个生成矩阵。编码可以表示成一个线性映射

$$\varphi : \mathbb{F}^k \to \mathbb{F}^n$$

使得，对于给定的 $\mathbf{u} \in \mathbb{F}^k$，有唯一码字 $\mathbf{c} = \mathbf{uG}$ 与之对应。这种一般的编码算法的复杂度与 $\mathbf{G}$ 的稀疏程度有关。在实际工程中，常用的是系统编码方法，此情形对应生成的矩阵 $\mathbf{G}$ 具有形式 $\mathbf{G} = [\mathbf{I}\,\mathbf{P}]$，其中 $\mathbf{I}$ 是 k 阶单位矩阵，而 $\mathbf{P}$ 是 $k \times (n-k)$ 的矩阵。系统码有个校验矩阵，形式为 $[-\mathbf{P}^{\mathrm{T}}\ \mathbf{I}]$。对于系统编码，码字具有形式 $\mathbf{c} = (\mathbf{u}, \mathbf{uP})$。就是说信息向量 $\mathbf{u}$ “原封不动”地出现在码字中。我们有如下命题。

【命题 5.11】 *任何一个线性分组码 $\mathscr{C}[n,k]$（必要时经过分量置换），均存在一个系统形式的生成矩阵。*

证明：设 $\mathbf{G}$ 是 $\mathscr{C}[n,k]$ 的一个生成矩阵，则经过行初等变换与列置换，则可以得到一个等价形式 $\mathbf{G} \sim [\mathbf{I}, \mathbf{P}]$。

注：在上述证明中，我们允许列置换，但不允许列向量的其他形式的变换。

线性分组码的译码算法需要结合信道模型来讨论。我们首先考虑二进制对称信道（binary symmetry channel, BSC）模型，如下图所示：

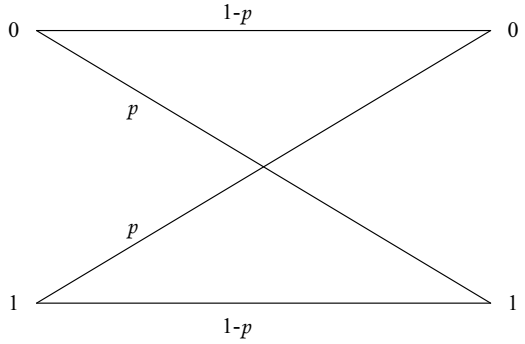


图 5.1: 二进制对称信道（BSC）模型

二进制对称信道是无记忆信道，输入是 $\{0,1\}$，以 $p(<\frac{1}{2})$ 的概率发生错误。设 $\mathbf{c} = (c_0, c_1, \cdots, c_{n-1})$ 是发送码字，则接收向量 $\mathbf{r} = (r_0, r_1, \cdots, r_{n-1})$ 可以表示成 $\mathbf{r} = \mathbf{c} + \mathbf{e}$，其中 $\mathbf{e}$ 称为错误图样向量，“+”是模 2 运算。我们说信道是 BSC 是指，$\mathbf{e}$ 是一个独立同分布的随机二进制向量的样本，其分量取 1 的概率是 p。在接收端，当收到 $\mathbf{r}$ 之后，我们可以计算 $\mathbf{s}^{\mathrm{T}} = \mathbf{Hr}^{\mathrm{T}}$，其中 $\mathbf{H}$ 是校验矩阵。由于 $\mathbf{Hc}^{\mathrm{T}} = \mathbf{0}$，我们可以得到 $\mathbf{s}^{\mathrm{T}} = \mathbf{He}^{\mathrm{T}}$，我们称 $\mathbf{s}$ 是伴随式。如果 $\mathbf{s} = \mathbf{0}$，我们认为没有错误发生。若 $\mathbf{s} \neq \mathbf{0}$，则一定有错误发生。利用这个性质，我们可以实现检错。而纠错是指，我们想从 $\mathbf{s}^{\mathrm{T}} = \mathbf{He}^{\mathrm{T}}$ 中解出 $\mathbf{e}$。这是一个线性方程组，包含有 $n-k$ 个方程，n 个未知数 $\{e_i, 0 \le i \le n-1\}$。这个方程组有 2^k 个解，译码就是从中选择一个解。

对于一般的线性方程组 $\mathbf{He}^{\mathrm{T}} = \mathbf{s}^{\mathrm{T}}$，通常的解法是求出齐次线性方程组 $\mathbf{He}^{\mathrm{T}} = \mathbf{0}^{\mathrm{T}}$ 的所有解，然后再求出一个特解（有可能不存在，但在译码这里一定是存在的）。那么，$\mathbf{He}^{\mathrm{T}} = \mathbf{s}^{\mathrm{T}}$ 的所有解就可以表示为特解 + 齐次线性方程组通解的形式。解方程组的过程也可以从几何的角度去描述。齐次线性方程组 $\mathbf{He}^{\mathrm{T}} = \mathbf{0}^{\mathrm{T}}$ 的解空间是一个线性子空间，在这里实际上就是线性分组码 $\mathscr{C}$ 本身，含有 2^k 个码字。而一般的线性方程组 $\mathbf{He}^{\mathrm{T}} = \mathbf{s}^{\mathrm{T}}$ 的解空间相当于把 $\mathscr{C}$ 进行了一个平移，如下图示意。

所以，若 $\mathbf{He}^{\mathrm{T}} = \mathbf{s}^{\mathrm{T}}$ 有解的话，与 $\mathbf{He}^{\mathrm{T}} = \mathbf{0}^{\mathrm{T}}$ 的解的个数是一样多的，同为 2^k 个。现在的问题是，如何从这 2^k 个解中挑选出一个作为译码器的输出。

这个挑选规则与信道特性紧密相关，在 BSC 信道条件下，一个错误图样 $\mathbf{e}$ 发生的概率是

$$P(\mathbf{e}) = p^{W_{\mathbf{H}}(\mathbf{e})}(1-p)^{n-W_{\mathbf{H}}(\mathbf{e})}$$

其中，$W_{\mathbf{H}}(\mathbf{e})$ 表示 $\mathbf{e}$ 的汉明重量。可以看出，$P(\mathbf{e})$ 是 $W_{\mathbf{H}}(\mathbf{e})$ 的减函数（注意 $p < \frac{1}{2}$）。由此，若 $\mathbf{e}_1$ 与 $\mathbf{e}_2$ 是两个解，但 $W_{\mathbf{H}}(\mathbf{e}_1) < W_{\mathbf{H}}(\mathbf{e}_2)$，我们应该选择哪一个呢？一个合理的选择是 $\mathbf{e}_1$，因为它较 $\mathbf{e}_2$ 发生的机会大，这就是最小汉明距离译码，在 BSC 信道条件下等价于最大似然译码。对于码参数比较小的码，我们通常列一个表，

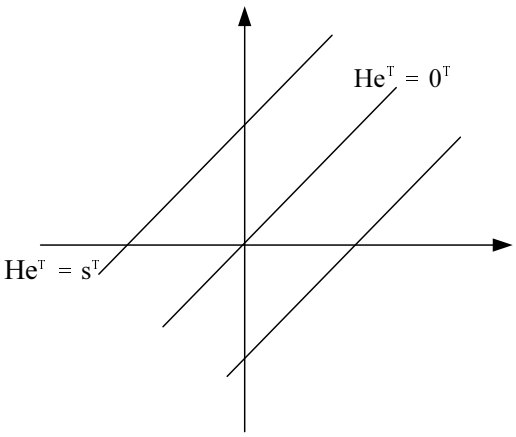


图 5.2

给出 $\mathbf{s}$ 与 $\mathbf{e}$ 之间的对应关系。当然，若某个 $\mathbf{s}$ 对应的解中有两个 $\mathbf{e}$，它们的重量相等，且同为最轻，则我们可以随意选择一个 $\mathbf{e}$，或者报告一个译码失败的信息。

§5.5

线性分组码的简单例子

这一节我们给出几个简单的例子以理解前几节的概念。

【例题 5.12】(重复码) 设 $u \in \mathbb{F}_2$ 是一个待传比特，一个简单的编码是 $u \longrightarrow \mathbf{c} = (u, u, \cdots, u)$，即把 u 重复 n 次。这个码的生成矩阵是 $\mathbf{G} = [1, 1, \cdots, 1]_{1\times n}$，对应的校验矩阵是 $\mathbf{H} = [\mathbf{1}^{\mathrm{T}}\ \mathbf{I}]$，即

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \cdots & 1 \end{bmatrix}_{(n-1)\times n}.$$

该码的码率是 $\frac{1}{n}$。若重复码在 BSC 信道上使用，则可用大数逻辑译码，即在接收端，根据一个码字中 1 的个数与 0 的个数的多少来判决。若这个数目相等（当 n 是偶数时才有可能相等），则宣告译码失败。设 BSC 的错误概率是 p，则大数逻辑译码的错误概率是

$$P_b = \sum_{t \geq \lceil \frac{n}{2} \rceil} \binom{n}{t} p^t (1-p)^{n-t}$$

可以证明，当 $p < \frac{1}{2}$ 时，P_b 随着 n 的增大趋于零，可以达到“可靠”通信。但是，要付出的代价是码率 $\frac{1}{n} \to 0$，这是通信系统要避免的。

【例题 5.13】(奇偶校验码) 设 $\mathbf{u} = [u_0, u_1, \cdots, u_{n-2}] \in \mathbb{F}_2^{n-1}$ 是一个待传信息，奇偶校验码的编码算法是计算 $c_{n-1} = \sum_{i=0}^{n-i} u_i$，对应码字是 $\mathbf{c} = [u_0, u_1, \cdots, u_{n-2}, c_{n-1}]$，其中 $\mathbf{u}$ 称为信息位，c_{n-1} 称为奇偶校验位。

奇偶校验码的码率是 $\frac{n-1}{n}$，生成矩阵是 $\mathbf{G} = [\mathbf{I}\ \mathbf{1}^{\mathrm{T}}]_{(n-1)\times n}$，校验矩阵是 $\mathbf{H} = [\mathbf{1}]_{1\times n}$。

可以验证，码长是 n 的重复码与奇偶校验码互为对偶码。前者的最小汉明距离是 n，而后者的最小汉明距离是 2。因而奇偶校验码在 BSC 上不具有纠错能力，但是可以检错。事实上，奇偶校验码可以发现任何奇数个错误。这两个码分别记为 $\mathscr{C}[n, 1, n]$ 和 $\mathscr{C}[n, n-1, 2]$。

【例题 5.14】(汉明码) 前两个例子是从编码的角度引出的，定义码的同时也定义了编码算法。汉明码比较方便的定义是从校验矩阵出发。设 $m > 1$ 是一个整数，考虑 $\mathbb{F}_2^m$ 中的非零向量，共有 $2^m - 1$ 个。以它们为列，构成一个 $m \times (2^m - 1)$ 的矩阵。

$$\mathbf{H} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 1 & 1 & \cdots & 1 \\ 1 & 0 & 1 & \cdots & 1 \end{bmatrix}.$$

该矩阵的秩是 m。因此，$\mathscr{C} = \{\mathbf{c} | \mathbf{H}\mathbf{c}^{\mathrm{T}} = 0\}$ 具有参数：码长 $2^m - 1$，维数 $2^m - 1 - m$。这样定义的码就是汉明码。汉明码的最小汉明距离是 3，可以纠正一个位错误，论证如下。

【命题 5.15】 一个线性分组码的最小汉明距离是 $d_{\min}$，当且仅当其校验矩阵 $\mathbf{H}$ 的任意 $d_{\min} - 1$ 列线性无关且存在某 $d_{\min}$ 列线性相关。

证明略。

从汉明码的定义可以看出，$\mathbf{H}$ 的任何两列不同，所以最小距离至少为 3。由于可以找到三列相加等于 0，我们由上述命题知道 $d_{\min} = 3$。

假定错误图样 $\mathbf{e}$ 的重量为 1，即只有一位错误，则 $\mathbf{s}^{\mathrm{T}} = \mathbf{H}\mathbf{e}^{\mathrm{T}}$ 刚好是错误位置对应的列。因此，可由 $\mathbf{s}$ 找出

错误的位置。为帮助理解，取 $m=3$ 为例，这就是常常用来作为例子的 [7，4，3] 汉明码，其校验矩阵是

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

设一个码字 $\mathbf{c}$，经过一个 BSC 信道，收到 $\mathbf{y}$，计算 $\mathbf{s}^{\mathrm{T}}=\mathbf{H}\mathbf{y}^{\mathrm{T}}$。若 $\mathbf{s}^{\mathrm{T}}=\mathbf{0}$，则我们认为无错；若 $\mathbf{s}^{\mathrm{T}}=\begin{bmatrix}0\\0\\1\end{bmatrix}$，我们认为第一位发生了错误。

汉明码的发明早于Shannon 1948 年的信息论奠基文章 [1]。但由于专利原因，公开时间较迟。Shannon 在 1948 年的论文里用到了 [7，4，3] 汉明码。

§5.6

一般线性分组码的仿真

一般线性分组码的仿真与一般码的仿真类似，不同之处在于，我们需要存储生成矩阵 **G** 以实现编码。而在接收端，如果利用标准阵列译码的话，需要给出每个伴随式对应的最轻错误图样。由此可以看出，尽管一般线性分组码的仿真较一般码简单，维数很大时也难以仿真。

§5.7 习题

1. 编写代码：仿真一个一般二元码在二元对称信道上的性能，并画出曲线（横轴为错误概率 p，纵轴为 FER，用对数坐标）。

2. 自学重复码、奇偶校验码、汉明码。

3. 阶不超过 32 的有限域有哪些？

4. 举出两个不同参数的汉明码例子，给出校验矩阵与生成矩阵。

5. 举一个与 [7,4,3] 汉明码不同的线性分组码 [7,4] 的例子。在 BSC 信道上仿真两个码的性能，并画曲线（横轴为错误概率 p，纵轴为 FER，用对数坐标），比较他们的性能曲线。

第六章

低密度一致校验码

上一章我们引入了线性分组码，其可以看作一个线性空间的线性子空间。一个线性分组码（线性子空间）可以用生成矩阵（线性子空间的一组基）来定义，也可以用校验矩阵（对偶空间的一组基）来定义。这一章，我们讨论一类特殊的线性分组码，其具有稀疏的校验矩阵。

§6.1 低密度一致校验码

一个矩阵 $\mathbf{H}_{m\times n}$ 称为稀疏的，如果其元素大部分为 0，只有少部分是非零。这个“定义”是直观描述，在文献中没有严格定义。通常认为 $\frac{\text{非零元素的数目}}{(mn)}$ 要远远小于 $\frac{1}{2}$。

【定义 6.1】 一个线性分组码 $\mathscr{C}$ 称之为低密度一致校验码 (low-density parity-check code, LDPC 码)，如果 $\mathscr{C}=\{\mathbf{c}|\mathbf{H}\mathbf{c}^{\mathrm{T}}=\mathbf{0},\mathbf{c}\in\mathbb{F}^n\}$，其中 $\mathbf{H}$ 是大小为 $m\times n$ 的稀疏矩阵。

由于 $\mathrm{rank}(\mathbf{H})\leq m$，所以该码的维数 $k\geq n-m$。如果 $\mathbf{H}$ 中，每行的非零元素个数（行重）相等，并且每列的非零元素个数（列重）也相等，我们称该码为规则的低密度一致校验码。否则，称之为非规则的低密度一致校验码。

【例题 6.2】 考虑 $\mathbb{F}_2$ 上的码，设 $\mathbf{I}$ 是 4×4 的单位矩阵，$\mathbf{P}$ 是 4×4 的循环移位矩阵，即

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

$$\mathbf{P}=\begin{bmatrix}0&0&0&1\\1&0&0&0\\0&1&0&0\\0&0&1&0\end{bmatrix}.$$

定义

$$\mathbf{H}=\begin{bmatrix}\mathbf{I}&\mathbf{I}&\mathbf{I}&\mathbf{I}\\\mathbf{I}&\mathbf{P}&\mathbf{P}^2&\mathbf{P}^3\end{bmatrix}.$$

则由 $\mathbf{H}$ 定义的 LDPC 码，具有码长 16，维数 ≥ 8。

【例题 6.3】 设 $\mathbf{1}_6 = [1,1,1,1,1,1]$，定义

$$\mathbf{H}_1 = \begin{bmatrix} \mathbf{1}_6 & 0 & 0 \\ 0 & \mathbf{1}_6 & 0 \\ 0 & 0 & \mathbf{1}_6 \end{bmatrix}.$$

定义

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_1\mathbf{\Pi}_1 \\ \mathbf{H}_1\mathbf{\Pi}_2 \end{bmatrix},$$

其中 $\mathbf{\Pi}_1$, $\mathbf{\Pi}_2$ 是两个随机置换矩阵，其作用是把 $\mathbf{H}_1$ 的列打乱。则由 $\mathbf{H}$ 定义的码是 $(3,6)$ 规则码。

§6.2 编码

LDPC 码作为一类特殊的线性分组码，可以按照一般线性分组码的思路去理解其编码过程。当然，如果考虑编码器的实现复杂度，则需要考虑码的结构及其性质。下面我们讨论编码的一般思路。

由于 LDPC 码一般是由校验矩阵定义的，我们从校验矩阵的角度讨论。设 LDPC 码的校验矩阵是 $\mathbf{H}_{m\times n}$，我们可以利用高斯消元法（行初等变换），将 $\mathbf{H}$ 化成如下形式，

$$\mathbf{H} \to \begin{bmatrix} \mathbf{I}_{r\times r} & \mathbf{P}_{r\times(n-r)} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \triangleq \tilde{\mathbf{H}},$$

其中 $r = \mathrm{rank}(\mathbf{H}), r \le m$。特别注意，在上述变换中，若需要的话，可以做列置换，但是不能做“列 + 列”的操作。

由一个码字满足 $\mathbf{H}\mathbf{c}^{\mathrm{T}} = \mathbf{0}$，我们有 $\tilde{\mathbf{H}}\mathbf{c}^{\mathrm{T}} = \mathbf{0}$。这样，我们可以把 $\mathbf{c}$ 分成两段，$\mathbf{c} = (\mathbf{v}, \mathbf{u})$，其中 $\mathbf{u}$ 是信息位，$\mathbf{v}$ 是校验位，满足 $\tilde{\mathbf{H}}\mathbf{c}^{\mathrm{T}} = \mathbf{v}^{\mathrm{T}} + \mathbf{P}\mathbf{u}^{\mathrm{T}} = \mathbf{0}$，即 $\mathbf{v}^{\mathrm{T}} = -\mathbf{P}\mathbf{u}^{\mathrm{T}}$。因此，编码算法简单描述为：

若 $\mathbf{H}$ 已经化成 $[\mathbf{I}, \mathbf{P}]$ 的系统形式，则对应任何一个信息向量 $\mathbf{u} \in \mathbb{F}^{n-r}$，校验向量可以计算为 $\mathbf{v} = -\mathbf{u}\mathbf{P}^{\mathrm{T}}$。

§6.3 译码

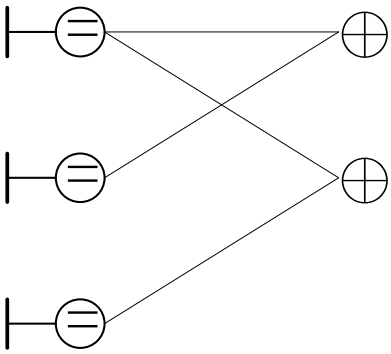
人们普遍接受的事实是，LDPC 码的译码算法应该结合图来描述，我们采用正规图的术语。相比于更常见的 Tanner 图表示，正规图把节点的功能与边的功能分离开来，更具广泛性。

设线性分组码 $\mathscr{C}[n,k]$ 的校验矩阵是 $\mathbf{H}_{m\times n}$，满足 $m\geq n-k$，而 $\mathrm{Rank}(\mathbf{H})=n-k$。我们可以用一个二分图来表示该码。在这个二分图里有两类节点，一类用“$\ominus$”表示，称之为变量节点，或者等号节点，对应一个向量的各分量，故共有 n 个 $\ominus$ 节点；另一类用“$\oplus$”表示，称之为校验节点，或者加号节点，对应一个校验方程，故共有 m 个 $\oplus$ 节点。这两类节点之间用一些边连接起来，遵循的规则是：一个 $\ominus$ 节点与一个 $\oplus$ 节点相连，当且仅当该节点 $\ominus$ 对应的变量参与了该 $\oplus$ 节点对应的方程。

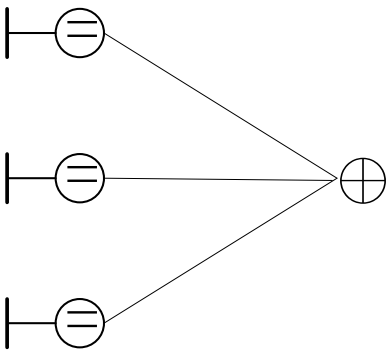
【例题 6.4】 重复码 $[3,1]$ 的校验矩阵是

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

其对应的正规图是



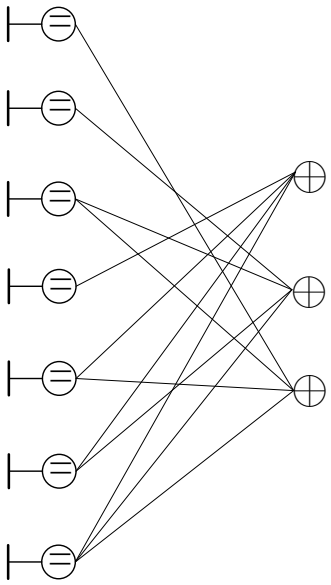
【例题 6.5】 奇偶校验码 $[3,1]$ 的校验矩阵是 $\mathbf{H}=[1\ 1\ 1]$，其对应的正规图是



【例题 6.6】 考虑汉明码 $\mathscr{C}[7,4]$，其校验矩阵是

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

其列恰好包含了所有的 3 维非零向量。该校验矩阵对应的正规图是



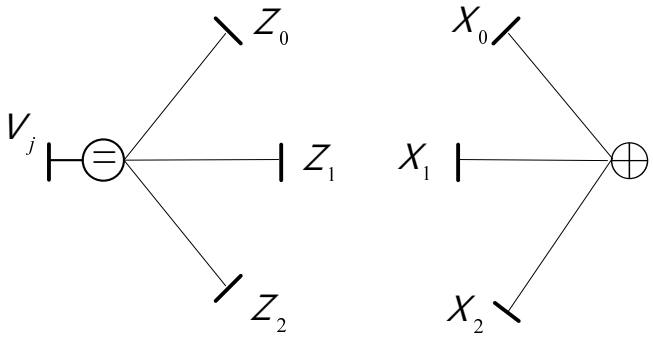
从上面三个简单的例子看到，一个等号与一个加号相连，当且仅当其对应的 $\mathbf{H}$ 元素为 1。还注意到在等号的左端有一条“半边”，这些半边可以认为是该二分图与外界（比如信道等）交换信息的接口。任何二元线性分组码都可以用类似这些例子中展示的正规图来表示，不过，只有当线性分组码具有稀疏的校验矩阵时，所对应的正规图才具有稀疏性，即对应较少的边。在这种情况下，迭代译码才具有较好的性能。

为描述译码算法，我们把一个向量 $(v_0, v_1, \cdots, v_{n-1})$ 看作是随机向量 $(V_0, V_1, \cdots, V_{n-1})$ 的一个实现，每个分量对应一个“半边”。译码的问题就是（准确地或近似地）计算每个分量的概率分布 $P_{V_j}(v), v \in \mathbb{F}_2$。一般地，我们把每条边都看作一个“独立”的随机变量，而把一个节点看作一个约束条件，约束所有与其相连的边所代表的随机变量。在 LDPC 码的正规图表示中，有两类节点：$\ominus$ 与 $\oplus$，其中

$\ominus$：要求所有与其相连的边必须取得相同的值；

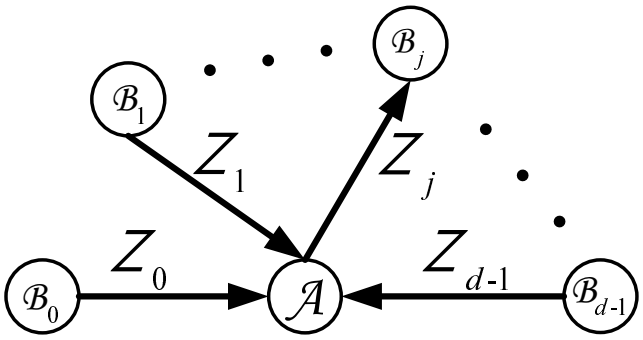
$\oplus$：要求所有与其相连的边必须相加等于 0。

我们看下面的例图，



在上面左图中，我们要求 $V_j = Z_0 = Z_1 = Z_2$，所以，可能的取值仅有两种，$(V_j, Z_0, Z_1, Z_2) = (0, 0, 0, 0)$ 或者 $(V_j, Z_0, Z_1, Z_2) = (1, 1, 1, 1)$。在上面右图中，我们要求 $X_0 + X_1 + X_2 = 0$，所以，可能的取值有四种，即 $(0, 0, 0), (1, 1, 0), (0, 1, 1), (1, 0, 1)$。

为了更清楚地描述译码算法，我们首先介绍在任意类型的节点上进行信息处理的一般规则。假设 $\mathcal{A}$ 是一个任意类型且度为 d 的节点，与它相连的节点为 $\mathcal{B}_0, \mathcal{B}_1, \cdots, \mathcal{B}_{d-1}$，对应的边是 d 个取值空间在 $\mathbb{F}_q$ 上的离散随机变量 $Z_j (0 \le j \le d-1)$，如下图所示。



假设所有输入的信息 $P_{Z_j}^{(\mathcal{B}_j\to\mathcal{A})}(z), z\in\mathbb{F}_q$ 都是可用的，则节点 $\mathcal{A}$ 可看作是一个信息处理器：任意给定变量 Z_j，流出节点 $\mathcal{A}$ 的信息可以通过计算下面的似然函数得到

$$P_{Z_j}^{(\mathcal{A}\to\mathcal{B}_j)}(z)\propto \Pr\{\mathcal{A}\text{ 的约束条件满足 }|Z_j=z\}, z\in\mathbb{F}_q.$$

由于上式的计算与输入信息 $P_{Z_j}^{(\mathcal{B}_j\to\mathcal{A})}(z)$ 是无关的，因此，在这种定义下，$P_{Z_j}^{(\mathcal{A}\to\mathcal{B}_j)}$ 即为所谓的“外信息”。

【例题 6.7】 考虑图 ?? 的两个图，设从其他子系统传向 $\ominus$ 节点的消息已知，例如，

$$P_{V_j}^{|\to\ominus}(0)=0.9,\ P_{V_j}^{|\to\ominus}(1)=0.1$$

$$P_{Z_0}^{|\to\ominus}(0)=0.8,\ P_{Z_0}^{|\to\ominus}(1)=0.2$$

$$P_{Z_1}^{|\to\ominus}(0)=0.4,\ P_{Z_1}^{|\to\ominus}(1)=0.6$$

$$P_{Z_2}^{|\to\ominus}(0)=0.7,\ P_{Z_2}^{|\to\ominus}(1)=0.3$$

则我们可以计算从 $\ominus$ 节点传向其他子系统的消息，比如

$$P_{Z_2}^{\ominus\to|}(0)\propto 0.9\cdot 0.8\cdot 0.4$$

$$P_{Z_2}^{\ominus\to|}(1)\propto 0.1\cdot 0.2\cdot 0.6$$

归一化之后，我们有

$$P_{Z_2}^{\ominus\to|}(0)=\frac{0.9\cdot 0.8\cdot 0.4}{0.9\cdot 0.8\cdot 0.4+0.1\cdot 0.2\cdot 0.6}=\frac{0.288}{0.3}=0.96$$

$$P_{Z_2}^{\ominus\to|}(1)=\frac{0.1\cdot 0.2\cdot 0.6}{0.9\cdot 0.8\cdot 0.4+0.1\cdot 0.2\cdot 0.6}=\frac{0.012}{0.3}=0.04$$

对于 $\oplus$ 节点，若

$$P_{X_0}^{|\to\oplus}(0)=0.9,\ P_{X_0}^{|\to\oplus}(1)=0.1$$

$$P_{X_1}^{|\to\oplus}(0)=0.2,\ P_{X_1}^{|\to\oplus}(1)=0.8$$

$$P_{X_2}^{|\to\oplus}(0)=0.3,\ P_{X_0}^{|\to\oplus}(1)=0.7$$

则我们在 $\oplus$ 节点可以计算，比如

$$P_{X_2}^{\oplus\to|}(0)\propto 0.9\cdot 0.2+0.1\cdot 0.8$$

$$P_{X_2}^{\oplus\to|}(1)\propto 0.9\cdot 0.8+0.2\cdot 0.1$$

归一化之后，有

$$P_{X_2}^{\oplus\to|}(0)=\frac{0.9\cdot 0.2+0.1\cdot 0.8}{0.9\cdot 0.2+0.1\cdot 0.8+0.9\cdot 0.8+0.2\cdot 0.1}=\frac{0.26}{1}=0.26$$

$$P_{X_2}^{\oplus\to|}(1)=\frac{0.9\cdot 0.8+0.2\cdot 0.1}{0.9\cdot 0.2+0.1\cdot 0.8+0.9\cdot 0.8+0.2\cdot 0.1}=\frac{0.74}{1}=0.74$$

给定一个 LDPC 码，假定 $P_{V_j}^{|\to\ominus}(v), 0\le j\le n-1$，已知（比如，可以结合信道规律 $P(y_j|v_j)$，由接收符号 y_j 进行计算），则 LDPC 码的译码算法可以概括如下：

0. 初始化：所有从 $\oplus$ 到 $\ominus$ 的信息初始化为 $P^{\oplus\to\ominus}(0)=\frac{1}{2}$。

1. 在每个 $\ominus$ 节点，计算 $P^{\ominus\to\oplus}$ 的所有外信息。

2. 在每个 $\oplus$ 节点，计算 $P^{\oplus\to\ominus}$ 的所有外信息。

3. 在每个 $\ominus$ 节点，计算 V_j 的总信息，其正比于 $P^{|\to\ominus}\cdot P^{\oplus\to\ominus}$，根据该信息进行判决，得到 $\hat{v_j}$。

若 $\mathbf{H}\hat{\mathbf{v}}^{\mathrm{T}}=\mathbf{0}$，则输出 $\hat{\mathbf{v}}$，译码结束。否则，重复上述 1，2，3 步，直到预设的最大迭代次数。若找不到合法的 $\hat{\mathbf{v}}$，宣告译码失败。

需说明的是，为了推导公式的便利，我们假定在正规图中节点之间传送的消息是概率质量函数。而在工程实际中，可以用任何别的等价量取代。特别地，对于二元变量，我们可以用 $\frac{P_0}{P_1}$ 或者 $\ln\frac{P_0}{P_1}$ 进行表示。消息的表示不同，节点的处理公式也不同。节点的作用就是接收消息，处理并输出消息。

§6.4 仿真

由于 LDPC 码是从校验矩阵的角度定义的，所以仿真程序一般应该包括由校验矩阵变换为生成矩阵的过程，然后编码基于生成矩阵完成。对于有些标准采用的 LDPC 码，通常具有快速编码算法，这种情况下，就不必转化为生成矩阵。

LDPC 码的译码是基于校验矩阵的，而校验矩阵又对应一个稀疏正规图，比较方便的数据结构是双向链表，可以刻画正规图中节点的连结。

§6.5 习题

1. 利用高斯消元法将例 3.1.1 中的矩阵化为系统矩阵的形式，进而确定码的维数。注意，高斯消元法中用列的运算是在 $\mathbb{F}_2$ 意义下进行的。

2. 设 $(X_1, X_2, \cdots, X_d)$ 是 d 个独立的二元随机变量，记 $p_i = P_{X_i}(0), q_i = P_{X_i}(1)$，试着推导 $X = \sum_{i=1}^{d} X_i$ 的概率分布律。

3. 如何在计算机中表示稀疏矩阵？

第七章

卷积码

§7.1 多项式环与形式幂级数

粗略地讲，环是较域更弱的一个结构体，可以做“加、减、乘”三种运算，但未必可以做除法。换言之，域是环的一个特殊例子。我们常见的环有所有整数构成的集合 $\mathbb{Z}$。在 $\mathbb{Z}$ 中，我们可以定义 $x+y, x-y, x\cdot y$，但一般情况下没有办法定义除法，或者说没有合适的方式定义一个数 x 的乘法逆。也就是说，对于一般的整数 $x\neq 0$，我们找不到整数 y，使 $x\cdot y=1$。整数环 $(\mathbb{Z},+,\cdot)$ 是一类特殊的环，具有唯一分解性：任何一个非零整数可以唯一地分解成素数的连乘积。

在编码领域用得较多的是多项式环与形式幂级数环。我们讨论如下。

设 $\mathbb{F}$ 是一个域，形如 $f(D)=f_0+f_1D+\cdots+f_mD^m$ 的式子称为多项式，其中 $f_i(0\leq i\leq m)\in\mathbb{F}$ 称为系数，而 D 是“哑元”，在编码中暗示延迟（delay）算子。若 $f(D)=\sum_{i=0}^{m}f_iD^i$ 中 $f_m\neq 0$，我们称 f_m 为最高项系数或者首项系数，定义 $f(D)$ 的次数是 m，记为 $\deg(f)$。设 $f(D)$ 与 $g(D)$ 是两个多项式，则可以定义加法 $f(D)+g(D)$ 与乘法 $f(D)\cdot g(D)$。这些定义我们应该不陌生。

$$f(D)+g(D)=\sum_{i=0}^{\max(\deg(f),\deg(g))}(f_i+g_i)D^i,$$

$$f(D)\cdot g(D)=\sum_i h_iD^i,\ \text{其中 } h_i=\sum_\ell f_\ell g_{i-\ell}=\sum_\ell g_\ell f_{i-\ell}.$$

可以验证，两个多项式相加，次数不超过最大的那个（还有可能减小），而两个非零多项式相乘，次数是 $\deg(f)+\deg(g)$。我们约定零多项式的次数是 $-\infty$。

可以看出，次数不超过 m 的多项式 $f(D)$ 与有限长序列 $f=(f_0,f_1,\cdots,f_m)$ 是一一对应的。因而，我们可以把二者等同起来，只不过用多项式记号使我们更方便地定义加法与乘法。可以验证，域 $\mathbb{F}$ 上所有的多项式（记为 $\mathbb{F}[D]$）在上述定义的加法与乘法意义下构成一个环。

形式上，我们也可以把无穷序列 $a=(a_0,a_1,\cdots)$ 与无穷幂级数 $a(D)=\sum_{i=0}^{\infty}a_iD^i$ 对应起来，称为形式幂级数。注意这只是个形式，不要求其具有“收敛”性质。域 $\mathbb{F}$ 上所有的幂级数的全体记作 $\mathbb{F}[[D]]$。

对于两个形式幂级数 $a(D)$ 与 $b(D)$，我们可以形式地定义 $a(D)+b(D)$ 与 $a(D)\cdot b(D)$，定义方式与多项式运算的定义一致。同样可以验证，$\mathbb{F}[[D]]$ 是一个环。

与整数环 $\mathbb{Z}$ 类似，$\mathbb{F}[[D]]$ 可以扩展成一个域，即 Laurent（洛伦）级数全体，其中元素具有 $x(D)=\sum_{i=r}^{\infty}x_iD^i$ 的形式，r 是一个整数。也就是说，一个洛伦级数是包含至多有限项负幂次的幂级数。既然是域，就可以定义乘

法逆或者除法。这个除法的定义是按照长除法来定义的。我们按如下规则描述长除法：设 $f(D)$ 与 $g(D)$ 是两个洛伦级数（注意，多项式、幂级数都可以看作是洛伦级数的特殊情况），其中 $g(D)\neq 0$，我们可以把 $g(D)$ 写作 $D^{-r}g_0(D)$ 的形式，$g_0(D)$ 中无负幂次，这样 $\frac{f(D)}{g(D)}=\frac{D^r f(D)}{g_0(D)}$。要计算 $\frac{f(D)}{g_0(D)}$，我们把分子、分母均按照升次排序，然后试商也是按照升次排序。考虑如下 $\mathbb{F}_2((D))$ 中的例子，注意系数的运算按照 $\bmod 2$ 进行。

【例题 7.1】 $f(D)=1+D, g(D)=1+D+D^2$，求 $\frac{f(D)}{g(D)}$ 解：按照长除法

$$
\begin{array}{r|l}
1+D+D^2 & 1 \quad +D^2+D^3+D^4+D^5\cdots \\
\cline{2-2}
 & 1+D \\
 & 1+D+D^2 \\
\cline{2-2}
 & \quad D^2 \\
 & \quad D^2+D^3+D^4 \\
\cline{2-2}
 & \qquad D^3+D^4 \\
 & \qquad D^3+D^4+D^5 \\
\cline{2-2}
 & \qquad\qquad D^5
\end{array}
$$

依此类推，可以求出各项系数。当然，这是理解定义的方式，在实践中，或有些情况下是有闭式解的。

§7.2 卷积码

给定一个域 $\mathbb{F}$，我们定义了多项式环 $\mathbb{F}[D]$，进而可以定义有理函数域

$$\mathbb{F}(D)=\left\{\frac{f(D)}{g(D)}\mid f(D)\ \text{与}\ g(D)\ \text{是多项式，且}\ g(D)\neq 0\right\}.$$

我们也可以定义形式幂级数环 $\mathbb{F}[[D]]$，进而定义洛伦级数域 $\mathbb{F}((D))$。

从一般码的角度,我们可以定义 $\mathbb{F}((D))$ 上的分组码,其中一个码字具有形式 $c(D)=(c_0(D),c_1(D),\cdots,c_{n-1}(D))$。与常见的分组码相比，这里的分量（不失一般性）可以设为形式幂级数，代表的是序列，因而码长 n 通常比较小。

§7.3 编码器

一般而言，卷积编码器接受 k 路序列，即 $\underline{u}(D)=(u_0(D),u_1(D),\cdots,u_{k-1}(D))$，输出 n 路序列，即 $\underline{c}(D)=(c_0(D),c_1(D),\cdots,c_{n-1}(D))$，而输入—输出的关系可以通过一个生成矩阵 $G(D)$ 来关联，即 $\underline{c}(D)=\underline{u}(D)G(D)$。生成矩阵 $G(D)$ 有如下形式 $G(D)=(g_{i,j}(D)),0\le i\le k-1,0\le j\le n-1$，其中 $g_{i,j}(D)$ 是有理函数的形式，即两个多项式相除的形式 $\frac{f(D)}{q(D)}$。前面已经提到，传统卷积码的 k,n 通常比较小。

【例题 7.2】 考虑一个码率为 $\frac{1}{2}$ 的卷积码，其生成矩阵 $G(D)=(1+D+D^2,\,1+D^2)$。我们知道，生成矩阵作为码字空间的一组基，可以做初等变换，得到的码字空间不变，但是信息序列与码字序列之间的对应关系会有变化。例如，$G(D)$ 可以等价变换为 $\tilde{G}(D)=\frac{1}{1+D+D^2}G(D)=(1,\,\frac{1+D^2}{1+D+D^2})$。从多项式环、有理分式域等的角度描述卷积码，提供了一些代数工具，可以研究卷积码的一些结构。在这里，我们这些描述主要是表明一个观点，卷积码也是一类“分组码”，只不过码字符号是形式级数罢了。这些描述有点儿抽象，更多时候是利用移位寄存器等直接描述编码。例如，对应 $G(D)$ 与 $\tilde{G}(D)$ 的编码器结构如图所示。

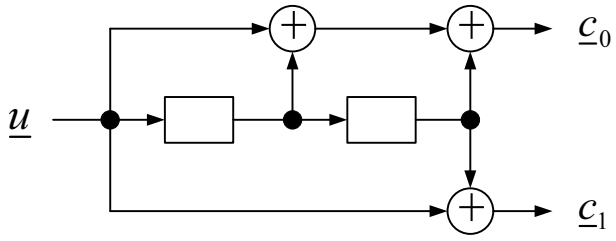


图 7.1: 非系统编码器 $G(D)=(1+D+D^2,\,1+D^2)$

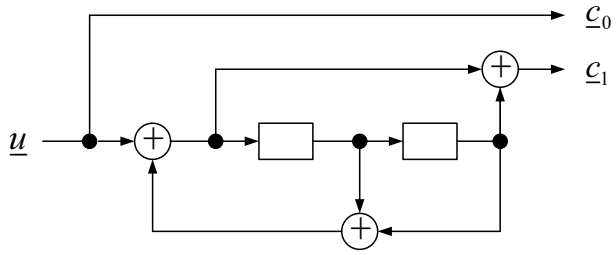


图 7.2: 递归系统编码器 $\tilde{G}(D)=(1,\,\frac{1+D^2}{1+D+D^2})$

对于 k 输入，n 输出的卷积码，我们需要 k 组寄存器，每组寄存器的个数是由生成矩阵的相应多项式的次数确定的。在给出更多例子之前，我们先给出一般的乘法与除法的实现框图，设 $f(D)=f_0+f_1D+\cdots+f_nD^n$，则一个输入序列 $u(D)$ 乘以 $f(D)$ 的输出可以按照下图实现。

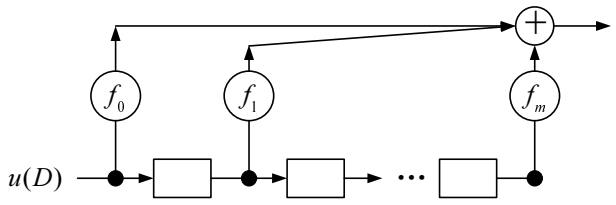


图 7.3: 多项式乘法器，$v(D)=u(D)f(D)$

注意，$u(D)$ 输入的顺序是 $u_0,u_1,\cdots,u_t,\cdots$。在时刻 t，寄存器中的内容是 $u_{t-1},u_{t-2},\cdots,u_{t-m}$。特别地，我们假定 u_t 在 $t<0$ 时是 0，这样，在 t 时刻的输出就表示为

$$v_t=f_0u_t+f_1u_{t-1}+\cdots+f_mu_{t-m},$$

与乘法定义中 D^t 的系数的定义一致。

乘法器的实现容易理解，而除法可以看作乘法的逆，但要求除式 $f(D)$ 的常数项必须为 1，即 $f(D) = 1 + f_1 D + f_2 D^2 + \cdots + f_m D^m$。从 $v(D) = u(D)f(D)$ 可以看出，$u(D)$ 在 t 时刻的系数 u_t 可以表示为 $u_t = v_t - f_1 u_{t-1} - f_2 u_{t-2} - \cdots - f_m u_{t-m}$，因此，除法器可以利用反馈实现。

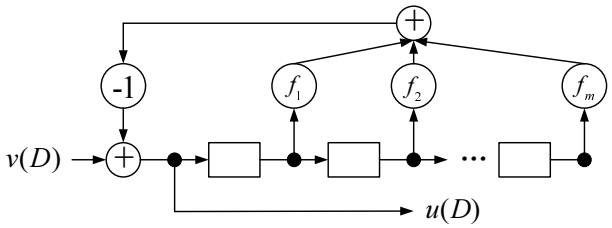


图 7.4: 多项式除法器，$u(D) = \frac{v(D)}{f(D)}$

【例题 7.3】 我们考虑一个码率为 $\frac{2}{3}$ 的卷积码，生成矩阵 $\mathbf{G} = \begin{bmatrix} 1 & 1+D & 0 \\ 1+D & 1+D^2 & 1+D+D^2 \end{bmatrix}$. 这个编码器的输入是两个序列 $u_1(D)$, $u_2(D)$，输出是三个序列 $v_1(D)$, $v_2(D)$, $v_3(D)$。我们需要两组移位寄存器，第一组是面向第一路输入的，有一个寄存器，而第二组是面向第二路输入的，有两个寄存器。编码器的实现如图 7.5 所示。

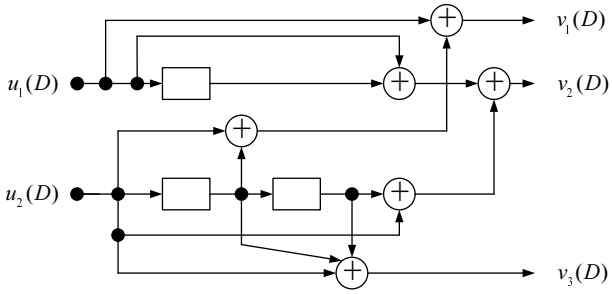


图 7.5: 码率为 $\frac{2}{3}$ 的卷积码对应的编码器实现图

我们前面定义了生成矩阵，其元素是有理函数 $\frac{f(D)}{g(D)}$ 的形成，是一种紧凑的表现形式，但看起来有点抽象。由于形式级数等价于序列，我们也可以考虑另外一种生成矩阵形式。我们以 $G(D) = (1+D+D^2, 1+D^2)$ 为例加以说明，其关键是考虑当输入序列为 $(1,0,0,\cdots,0,\cdots)$ 时，编码器输出什么？两路输出分别是 $(1,1,1,0,\cdots,0,\cdots)$ 与 $(1,0,1,0,\cdots,0,\cdots)$。我们把这两个序列交错排列得到 $(11,10,11,00,\cdots,00,\cdots)$，这就构成了生成矩阵的第一行。一般地，生成矩阵的第 i 行应该对应于当输入是 $(\underbrace{0,\cdots,0}_{i-1},1,0,\cdots,0,\cdots)$ 时的输出序列，这个序列就是第一行向右移位一个时刻，注意一个时刻对应两个输出符号，我们得到如下生成矩阵

$$\mathbf{G} = \begin{bmatrix} 11 & 10 & 11 & & \\ & 11 & 10 & 11 & \\ & & 11 & 10 & 11 \\ & & & \ddots & \end{bmatrix},$$

其中空白处是 0，可以看出 $\mathbf{G}$ 是一个带状矩阵。

一般地，一个卷积码的生成矩阵可以表示为如下形式，

$$\mathbf{G} = \begin{bmatrix} G_{0,0} & G_{0,1} & G_{0,2} & \cdots & G_{0,m} & & \\ & G_{1,1} & G_{1,2} & \cdots & G_{1,m-1} & G_{1,m} & \\ & & G_{2,2} & \cdots & G_{2,m-2} & G_{2,m-1} & G_{2,m} \\ & & & & & & \ddots \end{bmatrix},$$

其中 $G_{i,j}$ 是 $k \times n$ 矩阵，且 $G_{t,t}$ 是满秩的。如果第一行是 $(G_0, G_1, \cdots, G_m, 0, 0, \cdots)$，而其他行是上一行右移一个子块生成的，则对应的卷积码称为“时不变”的；否则，称为“时变”的。

上述生成矩阵是半无限矩阵，而在实际应用中，输入序列 u 是有限长的，输出序列 v 也应是有限长的，这种情况下，我们需把生成矩阵作相应处理。一种简单“粗暴”的方法是把列限制为有限，把其他部分略掉；另一种常用的方法是对编码器进行归零处理等价于把生成矩阵上边的行留下，而把其他行略掉。

§7.4

卷积码图表示与译码算法

从卷积码的编码器看出，卷积码与分组码的最大差别是，卷积码是面向流的，输入可以是连续不断地，而在 t 时刻的输出，不仅仅依赖于当前输入，还与之前的输入有关，这种记忆性是编码器的寄存器状态决定的，因此，可以用多种与有限状态机有关的方法描述：一种是用树的方式，另一种是用网格图的方式。我们以例 7.2 中非系统编码器为例进行说明。

设编码器的初始状态是 $(0,0)$，我们可以用一棵二元树来描述，其根节点表示初始状态，第 t 层表示 t 时刻编码器状态，从第 $t-1$ 层到第 t 层的分支上对应标有 t 时刻的输出，习惯上一般把一个节点的左儿子对应输入 0，右儿子对应输入 1。

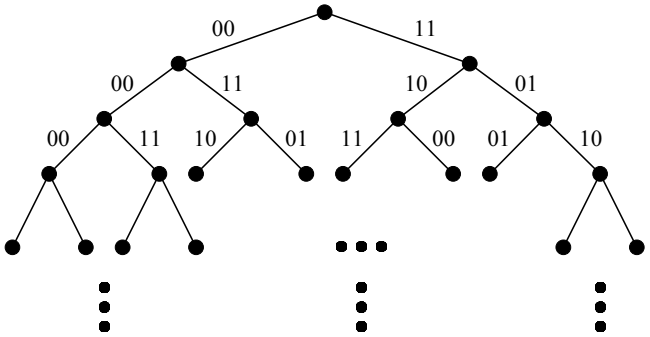


图 7.6: 卷积码 $G(D)=(1+D+D^2,1+D^2)$ 对应的树图

从编码树我们可以看出，一个有限长的码字序列对应从根节点到叶子节点的一条路径。例如，当输入序列是 $(0,1,0,\cdots)$ 时，对应的输出序列是“左, 右, 左, $\cdots$”，即 $(00,11,10,\cdots)$。当一个码字序列在信道中传输，收到一个有噪序列 $\underline{y}$，则译码问题就是在树中寻找一条路径，使得其对应的码字序列的似然值最大。当信道无记忆时，我们可以定义每条边的度量，进而定义每条路径的度量。有很多算法可用于搜索树上度量最大的路径，包括深度优先算法和宽度优先算法等，编码领域中基于树的译码算法通常称为序列译码算法，比较有名的是堆栈算法、Fano 算法等。

卷积码的另一种表示图是网格图，可以看作是马尔科夫状态转移图在时间上展开，其中的状态就是寄存器中的内容。以生成矩阵 $G(D)=(1+D+D^2,1+D^2)$ 为例，其有 4 个状态，分别是 $00,01,10,11$。初始状态一般假设为 00，输入序列 $\underline{u}$ 导引状态在各个状态中转换，对应的网格图如下所示。

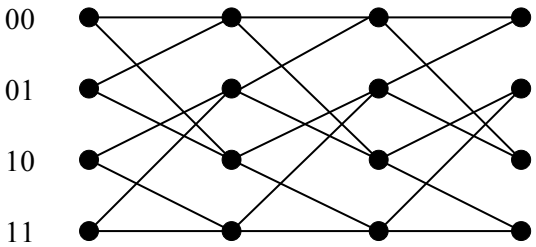


图 7.7: 卷积码 $G(D)=(1+D+D^2,1+D^2)$ 对应的网格图

编码器 $G(D)=(1+D+D^2,1+D^2)$ 对应的网格图有 4 个状态，记为 $00,01,10,11$，每条边由左状态输入、输出以及右状态刻画。我们可以用下表来表示这个网格图，为了方便起见，我们把状态用 0, 1, 2, 3 表示，即二进制对应的十进制表示。

与树图类似，对于给定的接收序列，可以给每条边赋予一个度量值，一条路径的度量值可以由边的度量值“累积”得到。最大似然译码问题因而转化为最短路径算法，这就是著名的 Viterbi 算法。

§7.5 仿真

卷积码的仿真实现主要是译码部分。仿真程序应定义生成矩阵（链接多项式）、“截断或结尾”方式等。若实现序列译码算法，则动态生成树，也就是说树图并不需要预先存储，而是边搜索边构造树。若要实现 Viterbi 算法，可以预先构造网格图。对于卷积码而言，我们只需要定义一节，包括左状态、输入、输出以及右状态等。

§7.6 习题

1. 画出如下运算的电路图：设 $a(D), b(D)$ 是无穷序列的幂级数表示，$f(D)$ 是首一多项式，$g(D)$ 是多项式。

(1) $a(D)+b(D)$；

(2) $a(D)\cdot D$；

(3) $a(D)\cdot g(D)$；

(4) $\frac{a(D)}{f(D)}$；

(5) $a(D)\cdot\frac{g(D)}{f(D)}$。

2. 设 $G(D)=(1+D, 1+D+D^3)$ 是卷积码的生成矩阵。

(1) 画出 $G(D)$ 的编码电路图；

(2) 若输入序列 $u=(0,1,1,0,0,1)$，初始状态是 0，编码输出序列是什么？末状态是什么？

(3) 写出与 $G(D)$ 等价的一种系统码形式，并画出编码电路图，回答 (2) 中的问题。

(4) 对于非系统码而言，可以通过输入序列来控制编码器的状态。那么，对于非系统码而言，如何做到这一点？

第八章

忒拨（Turbo）码

§8.1 用短码构造长码的方式

通常来讲，短码是指码长较短的分组码或者约束度较小的卷积码，其编译码算法可以在工程中实现。但从信道编码定理看，要逼近信道容量，码长需要充分长。在编码领域，一种实用的方法就是从简单码出发，构造性能好的长码。比较著名的例子有乘积码、级联码等，分别简述如下。

乘积码最早由 Elias 提出。设 $\mathscr{C}_1[n_1,k_1]$ 与 $\mathscr{C}_2[n_2,k_2]$ 是两个线性分组码，则乘积码的构造（也即编码）可以结合如下示意图进行描述。

$k_2\times k_1$ 信息位	行码校验
列码校验	校验之校验

图 8.1

第一步，把信息位（共 $k_1\times k_2$）排列成一个矩阵，每行 k_1 位，每列 k_2 位。

第二步，按行编码，得到行校验位，换句话讲，得到 k_2 个码字，取自 $\mathscr{C}_1$，这些码字排列在码字矩阵的前 k_2 行。

第三步，按列编码，即，对于上述码字矩阵的每一列，按照 $\mathscr{C}_2$ 的规则进行编码，得到列校验或者校验之校验。

可以看出，乘积码是分组码，其维数是 $k_1\times k_2$，码长是 $n_1\times n_2$。

【例题 8.1】 以汉明码 $[7,4,3]$ 为例进行说明，我们可以构造一个乘积码，其维数是 16。设输入信息序列 $\underline{u}=(1,1,0,0,0,0,0,0,0,1,0,1,0,0,0,1)$，按照上述步骤，可以得到如图 8.2 所示的码字矩阵，其中，汉明码的生成矩阵是：

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

可以验证，先按列编码，再按行编码，可以得到相同的码字矩阵。

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

图 8.2

级联码由 Forney 提出，由一个分组码（通常是代数码、RS 码和 BCH 码等）和一个卷积码按照串行的方式构成，如图 8.3 所示。

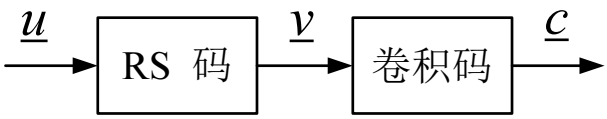


图 8.3

信息序列 $\underline{u}$ 分组之后进入代数编码器，如 RS 编码器，得到序列 $\underline{v}$，然后进入卷积码编码器，得到码字序列 $\underline{c}$。我们称卷积码是内码，RS 码是外码。译码时，先从接收序列估计内码的码字 $\hat{c}$，再用外码译码器纠正 $\hat{c}$ 中可能剩余的错误。按照 Forney 的观点，外码把内码的编译器连同信道一起看作一个“人工信道”。设外码的码率是 $R_o = \frac{k_o}{n_o}$，内码的码率是 $R_i = \frac{k_i}{n_i}$，则系统的码率是 $R = R_o \cdot R_i$。Forney 提出的级联码已在卫星通信标准中得到应用。

§8.2 级联码

现代的级联码始于 1993 年的忒拨（Turbo）码，与传统级联码的主要差别是其利用了迭代译码算法。目前，迭代译码的原理没有完全解释清楚，人们习惯称之为“Turbo 原理”，已在多个领域得到应用。

1993 年 Berrou 等提出的忒拨码基于卷积码构造，后已被推广到很多类似构造。这里编码的一个主要部件是随机交织器，与乘积码类似，信息序列被编码两次，一次是原序列，另一次是经过交织器的信息序列。以卷积码作为例子，我们有如下的忒拨码构造。

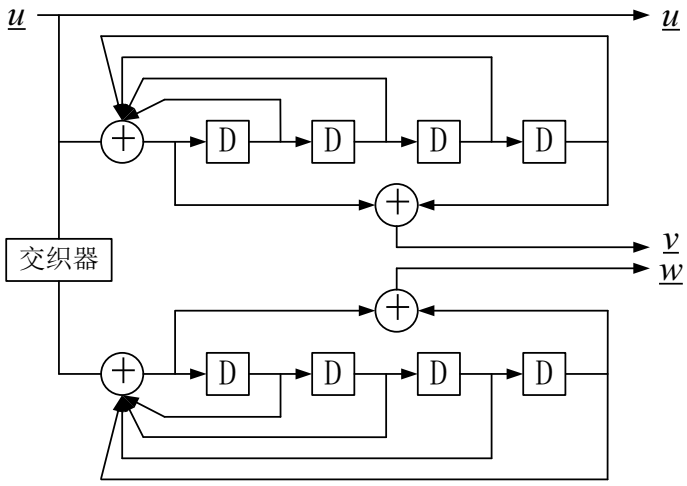


图 8.4

从上图看出，信息序列 $\underline{u}$ 可以产生校验序列 $\underline{v}$，经过交织器后又产生序列 $\underline{w}$，最终的码字由 $(\underline{u},\underline{v},\underline{w})$ 构成。其中，可以“打掉”（puncture）$\underline{v}$ 与 $\underline{w}$ 序列中某些比特以调整速率。比如，如果 $\underline{v}$ 取偶数位，$\underline{w}$ 取奇数位，则得到码率是 $\frac{1}{2}$。上述 8.4 图是忒拨码最初的架构，显然可以推广更一般的形式，如下图。

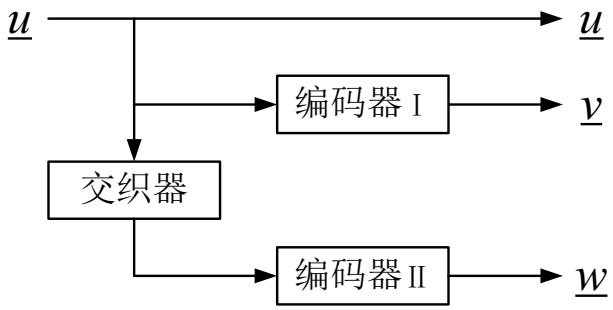


图 8.5

我们对编码器 I 与 II 的要求是，要有有效的软输入软输出（soft input soft output, SISO）译码算法。

上述忒拨码可以看作是并行级联码，我们也可以构造串行级联码，如下图。

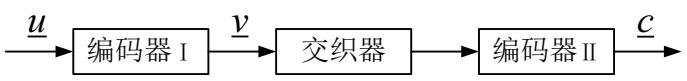


图 8.6

编码的过程与 Forney 经典的级联码一致，即信息序列 $\underline{u}$ 进入编码器 得到中间序列 $\underline{v}$，经过交织器后进入编码器 ，得到码字序列 $\underline{c}$。

§8.3 译码框架

忒拨码，无论是串行级联的，还是并行级联的，其译码算法的本质与 LDPC 码的译码算法具有相同的译码框架。前面已经指出，忒拨码的构造需要成份码具有有效的软输入软输出译码算法。我们先看如何描述成份码的译码算法，再给出忒拨码的译码框架。

一个码可以用一个简单的约束图来描述，左端对应输入 u，右端对应输出 v。

图 8.7

我们可以认为 u 连着“信源”，v 连着“信道”，u 和 v 都是序列（可以认为是随机序列）。设 u_i 是 u 的第 i 个分量，我们用它的概率质量函数 $(P_{u_i}(0), P_{u_i}(1))$ 或者其他等价的量表示 u_i 的信息或者消息。所有分量的信息表示为 $P_u^a(u)$，同样地，$P_v^a(v)$ 表示序列 v 的信息。初始地，我们用 $P_u^a(u), P_v^a(v)$ 表示相应的先验信息，这些量的计算可以从“信源”（相应地，“信道”）的统计规律获得，我们假定 u, v 的各个分量是“独立”的，不受编码约束。所谓软输入-软输出译码算法，就是根据 $P_u^a(u), P_v^a(v)$，并考虑编码约束来计算外信息，分别记作 $P_u^e(u), P_v^e(v)$。需要指出的是，如果一个随机变量先验未知，我们可以假定对应的信息是 $(\frac{1}{2}, \frac{1}{2})$，外信息的计算可以用精确的算法，也可以考虑低复杂度的近似计算。从原理上讲，精确的计算公式如下，

$$P_{u_t}^e(0) \propto \sum_{u_t=0} P_{u'}^a(u) \cdot P_v^a(v)$$

$$P_{u_t}^e(1) \propto \sum_{u_t=1} P_{u'}^a(u) \cdot P_v^a(v),$$

在公式 $P_{u_t}^e(0)$ 中，求和是对所有满足编码约束，且 $u_t = 0$ 的 u, v 进行的，而 $P_{u'}^a(u)$ 中的 u' 表示不含 $P_{u_t}^a(0)$, $P_{u_t}^a(1)$。类似地，v_t 的外信息计算如下：

$$P_{v_t}^e(0) \propto \sum_{v_t=0} P_u^a(u) \cdot P_{v'}^a(v)$$

$$P_{v_t}^e(1) \propto \sum_{v_t=1} P_u^a(u) \cdot P_{v'}^a(v).$$

并行级联类的忒拨码可以用如下正规图表示，

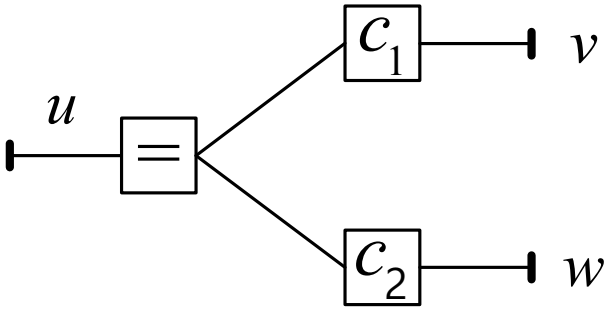


图 8.8

其中，$\boxed{=}$ 节点表示所连边的变量必须取相同的值，$\boxed{c_1}$ 表示成份码 I 的约束，意指 u 对应的码字必须是 v，$\boxed{c_2}$ 类似。并行级联类忒拨码的译码框架如下，

初始化：设序列 (u, v, w) 的消息均已获得，记作 $P_u^{|\to=}, P_v^{|\to c_1}, P_w^{|\to c_2}$，中间边 $\boxed{=}$—$\boxed{c_1}$ 与 $\boxed{=}$—$\boxed{c_2}$ 上的消息初始化为 $(\frac{1}{2}, \frac{1}{2})$。

迭代：做如下迭代，最大次数 $I_{\max}$ 满足某个预设条件。

1. 在 $\boxed{=}$ 节点，利用输入信息，按照外信息的方式计算 $P^{=\to c_1}$，即 $(P_u^{|\to=}, P^{c_2\to=}|=) \xrightarrow{\text{计算}} P^{=\to c_1}$；
2. 在 $\boxed{c_1}$ 节点，按照外信息的方式计算 $(P^{=\to c_1}, P_v^{|\to c_1}|c_1) \xrightarrow{\text{计算}} P^{c_1\to=}$；
3. 在 $\boxed{=}$ 节点，按照外信息的方式计算 $(P_u^{|\to=}, P^{c_1\to=}|=) \xrightarrow{\text{计算}} P^{=\to c_2}$；
4. 在 $\boxed{c_2}$ 节点，按照外信息的方式计算 $(P^{=\to c_2}, P_w^{|\to c_2}|c_2) \xrightarrow{\text{计算}} P^{c_2\to=}$。

判决：在 $\boxed{=}$ 节点，按照全信息方式，计算 $(P_u^{|\to=}, P_{c_1}^{|\to=}, P_{c_2}^{|\to=}) \xrightarrow{\text{计算}} P_u$，根据 P_u 进行判决，得到 $\hat{u}$。

我们在上述图 8.8 中略去了交织器，其信息处理较为简单，即把输入信息经过交织、解交织后得到输出信息。类似地，串行级联式拨码的正规图如下，

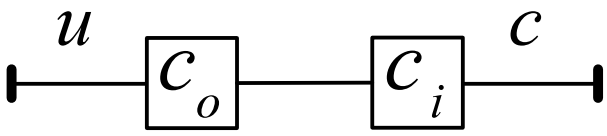


图 8.9

译码框架如下，

初始化：设 $P_u^{|\to c_o}, P_c^{|\to c_i}$ 已经获得，边 $\boxed{c_o}$—$\boxed{c_i}$ 上的消息初始化为均匀随机变量序列。

迭代：做如下步骤，最大迭代次数 $I_{\max}$ 满足某个预设条件。

1. 在 $\boxed{c_i}$ 节点，计算 $(P^{c_o\to c_i}, P_c^{|\to c_i}|c_i) \xrightarrow{\text{计算}} P^{c_i\to c_o}$；
2. 在 $\boxed{c_o}$ 节点，计算 $(P_u^{|\to c_o}, P^{c_i\to c_o}|c_o) \xrightarrow{\text{计算}} P^{c_o\to c_i}$；

判决，在 $\boxed{c_o}$ 节点，计算全信息 $(P_u^{|\to c_o}, P^{c_i\to c_o}|c_o) \xrightarrow{\text{计算}} P_u$，根据 P_u，判决得到 $\hat{u}$。

§8.4 仿真

仿真程序的主要模块包括交织器，软输入软输出译码器。交织器的实现可以有随机分组交织器、卷积交织器、确定型的交织器等。一个软输入软输出译码算法的有效实现，我们在以后的章节中讨论，当前需要明白的是一个编码模块接收的是从“信源”来的序列，输出的是面向“信道”的序列。译码模块的软输入包括信源的先验信息，也包括信道的似然信息；软输出是指两个变量的外信息，即给定变量某个特定的取值，其所参与的系统约束满足的概率。软输入既可以是针对信源符号的，也可以是针对信道符号的。

§8.5 习题

1. 以奇偶校验码 $[3,2]$ 为基本码，构造一个乘积码，试写出该乘积码对应的生成矩阵以及校验矩阵。设输入信息是 (1001)，给出对应的码字阵列。

2. 考虑图 8.5中的并行级联编码器。设编码器的生成矩阵分别是 $\mathbf{G}_1, \mathbf{G}_2$，交织器的矩阵形式是 $\mathbf{\Pi}$，$\mathbf{\Pi}$ 是一个置换矩阵。试写出该级联码的生成矩阵，并讨论其码率。

3. 考虑图 8.6中的串行级联编码器。设两个编码器的生成矩阵是 $\mathbf{G}_1, \mathbf{G}_2$，交织器对应的矩阵是 $\mathbf{\Pi}$，试写出该级联码的生成矩阵。

4. 在级联码中，若交织器是均匀随机交织器，则级联码可以看作一类随机码。试从简单码的重量谱分析级联码的重量谱。

第九章

网格图与译码算法

我们在前几章多次提到迭代译码算法的关键部件是软输入软输出译码算法。从原理上讲，其算法机理是贝叶斯分析，包括全概率公式与条件概率公式，以及如何由先验概率计算后验概率，并根据最小错误原则进行判决推断等。这一章，我们通过引入网格图来描述这类算法。

§9.1 网格图表示

毫不夸张地说，任何时间离散的有限字符系统，如果不考虑网格图是否时变，状态数是否太多等情况，都可以用网格图来描述。

网格图（Trellis）又称为篱笆图，可以看作是由一节一节的子图拼接而成的。网格图在 t 时刻的一节可以描述为一个二分图，其顶点称为状态。左状态集合记为 $\mathcal{S}_{t-1}$，而右状态集合记为 $\mathcal{S}_t$。一个左状态与一个右状态之间可以有一条边或多条边连结，也可以没有边连结。一个边 $\underline{b}$ 可以由一个四元组刻画 $\underline{b}=(\sigma_-,i,o,\sigma_+)$，其中 σ_- 表示左状态，σ_+ 表示右状态，i 表示这条边所对应的输入，o 表示这条边所对应的输出。在有些情况下，i 与 o 可以相等，此时，四元组就可以简化成三元组。

线性分组码的网格图既可以从生成矩阵构造，也可以从校验矩阵构造。从校验矩阵构造比较容易理解，简述如下。我们知道，如果二元线性分组码 $\mathscr{C}[n,k]$ 以 $\mathbf{H}$ 为校验矩阵，则一个向量 $\mathbf{c}\in\mathbb{F}_2^n$ 是一个码字当且仅当 $\mathbf{c}$ 满足 $\mathbf{H}\mathbf{c}^{\mathrm{T}}=\mathbf{0}$。设 $\mathbf{H}_t$ 是 $\mathbf{H}$ 的第 t 列，我们有 $\sum_{t=0}^{n-1}c_t\mathbf{H}_t=\mathbf{0}$，即，$\mathbf{H}_t$ 的加权和等于 $\mathbf{0}$。显然，一个求和形式可以逐次累加计算，设 $s_0=0\in\mathbb{F}_2^{n-k}$，对于 $t\geq 0$，可以定义 $s_{t+1}=s_t+c_t\mathbf{H}_t$。这个累加递推运算可以用网格图表示，其状态空间是 $\mathbb{F}_2^{n-k}$，$s_t\in\mathbb{F}_2^{n-k}$ 是 t 时刻的一个状态，$s_{t+1}\in\mathbb{F}_2^{n-k}$ 是 $t+1$ 时刻的一个状态，从 s_t 出发可有两条边，一条边对应码元符号 $c_t=0$，另一条边对应码元符号 $c_t=1$，这两条边分别连到 $s_{t+1}=s_t$ 与 $s_{t+1}=s_t+\mathbf{H}_t$。一个向量 $\mathbf{v}$ 是一个码字，当且仅当 $\mathbf{v}$ 引导的路径始于 0 状态，终于 0 状态。

【例题 9.1】 汉明码 $[7,4,3]$ 的校验矩阵是

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

其对应的网格图是，

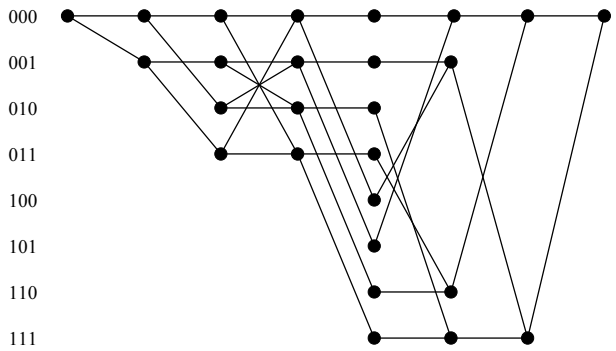


图 9.1

卷积码的网格图表示是比较自然的，在第 4 章已经提到。网格图也可以用来表示码间串扰信道（intersymbol interference, ISI）的输入与其无噪输出之间的关系。我们看下面的例子。

【例题 9.2】 一个码间串扰信道输入序列 $\mathbf{x} = (x_0, x_1, \cdots, x_{n-1})$，其中 x_t 是实数或者复数，无噪输出序列 $\mathbf{y} = (y_0, y_1, \cdots, y_{n-1})$ 可以描述为

$$y_t = \sum_{i=0}^{l} h_i x_{t-i},$$

其中 $h_i (0 \le i \le l)$ 称为信道系数。这个表达式看起来与卷积码的输入-输出转换类似，只是其中的运算在实数域或复数域中进行。假设 $x_t = +1$ 或 -1，而 $h_0 = 1, h_1 = -1, l = 1$，即 $h(D) = 1 - D$ 是信道的响应。此时，网格图可以表示如下。

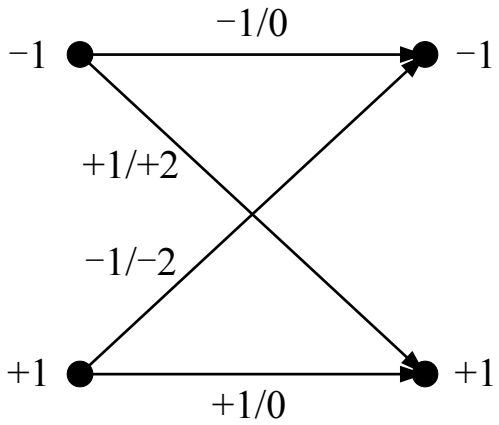


图 9.2

事实上，网格图提供了一种描述输入序列与输出序列之间对应关系的一种方式，这种方式与马尔科夫（Markov）性质有关联。给定状态条件下，输入符号与输出符号之间的对应与其他因素无关。比如，系统是按照什么路径到达当前状态的，是不影响下一个时刻的网格结构的。当然，若允许平行边存在，最少状态网格图即“查表”映射，见如下例子。

【例题 9.3】 4 PAM（pulse amplitude modulation）在 Gray 映射下，可以用下图表示。

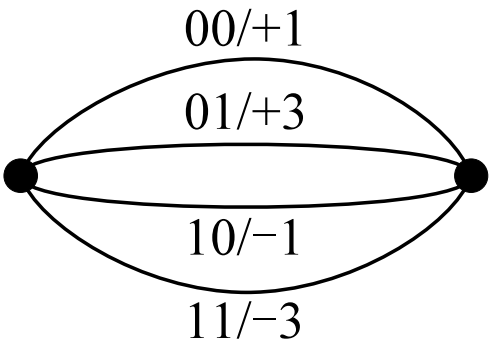


图 9.3

综上，如果我们写一个面向网格图的程序，则通过配置网格图，可以实现很多不同场景下的编译码算法或检测算法。

为描述这些算法的原理，我们考虑离散无记忆信源与信道。给定一个篱笆图。设 s_0 是初始状态，若 $\mathbf{u} = (u_0, u_1, \cdots, u_{n-1})$ 是信源的输出，其概率分布律记作 $P_U(u), u \in \mathcal{U}$，由 s_0 与 $\mathbf{u}$ 一起，可以引导一条路径，使得这条路径的初始状态是 s_0，在 t 时刻对应的边是 (s_t, u_t, v_t, s_{t+1})，其中，v_t 是 t 时刻的输出。顺着这条路径，得到一个输出序列 $\mathbf{v} = (v_0, v_1, \cdots, v_{n-1})$。若 $\mathbf{v}$ 在信道上传输，得到一个接收序列 $\mathbf{y} = (y_0, y_1, \cdots, y_{n-1})$。译码问题就是根据 $\mathbf{y}$，推断信源输出序列 $\hat{\mathbf{u}}$。

§9.2

维特比算法

最大似然算法是在篱笆图上找一条路径，使其对应的输出序列具有最大的似然函数值，即：

寻找 $\mathbf{v}^*$，使得 $P(\mathbf{v}^*)P(\mathbf{y}|\mathbf{v}^*) \geq P(\mathbf{u})P(\mathbf{y}|\mathbf{u})$，对于所有路径 $\mathbf{u}$ 成立。

为此，对于每一条边，定义一个度量

$$\gamma(s_t, u_t, v_t, s_{t+1}) = \log P_U(u_t) + \log P(y_t|v_t).$$

一条路径的度量是其所有边的度量之和。设 $\underline{b} = (b_0, b_1, \cdots, b_{L-1})$ 是一条路径，其度量记作 $\Gamma(\underline{b}) = \sum_{t=0}^{L-1} \gamma(b_t)$。因此，最大似然译码是在篱笆图中寻找一条度量最大的路径。这是典型的动态规划问题，可以用维特比算法予以解决，其基本原理是：全局最优的路径，局部一定是最优的。更具体地，我们有如下问题，如图 9.4 所示。

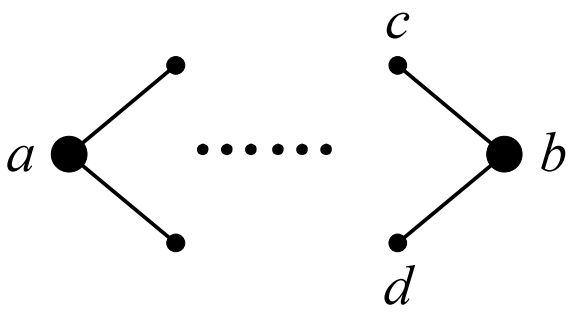


图 9.4

从 a 到 b 的最优路径，要么经过 c，要么经过 d。若经过 c，则从 a 到 c 的路径必须是最优的，否则，我们可以选择从 a 到 c 的最优路径予以替换。同理，若经过 d，则从 a 到 d 的路径必须是最优的。综上，求从 a 到 b 的最优路径可以约简为计算从 a 到 c 以及从 a 到 d 的最优路径。

根据这一原理，得到如下算法，即维特比算法。为简单起见，假定初始状态与末状态均已知，记为 0 状态。

1. 初始化：令 $t = 0$ 时刻的累积度量是 $\Gamma_0(s_0) = 0$，其中，$s_0 = 0$ 是已知的初始状态。若在网格图的第一节还有其他左状态 s，则置 $\Gamma(s) = -\infty$。

2. 递推：对于每个状态 $s_t \in \mathcal{S}_t$，其中，$t = 1, 2, \cdots$，按如下方式计算进入 s_t 的最优路径的累积度量，

$$\Gamma(s_t) = \max\{\Gamma_{t-1}(s_{t-1}) + \gamma(s_{t-1}, u_t, v_t, s_t)\}.$$

其中，max 操作是对所有进入 s_t 的边进行的，并记录最优的边 $b_t^* = (s_t - 1, u_t, v_t, s_t)$。

3. 回溯：在最后时刻，我们得到 $\Gamma_L(S_L = 0)$，并得到产生此最优累积度量的边 b_L^*。由此，可以得到 b_L^* 的左端点 s_{L-1}^*，而进入 s_{L-1}^* 的边也可“顺藤摸瓜”找出。于是，就得到了最优路径。

§9.3 列表维特比算法

维特比算法的输出是一条具有最大度量的路径，而在有些场景下，我们需要得到度量最大的 $l_{\max}$ 条路径。有两种方法实现这个功能，一种是串行的，另一种是并行的。并行的列表维特比算法很容易理解，且可以在维特比算法的基础上稍加修正即可。在每个时刻，每个状态存储至多 $l_{\max}$ 条进入的路径。若一个状态有 M 条进入的边，则进入该状态的 $l_{\max}$ 路径可以由上一时刻的最优路径拓展而来。事实上，每条边对应 $l_{\max}$ 条路径，所以，共有 $M \cdot l_{\max}$ 条路径，然后从中选出 $l_{\max}$ 条存活路径即可。下面给出一个示意图。

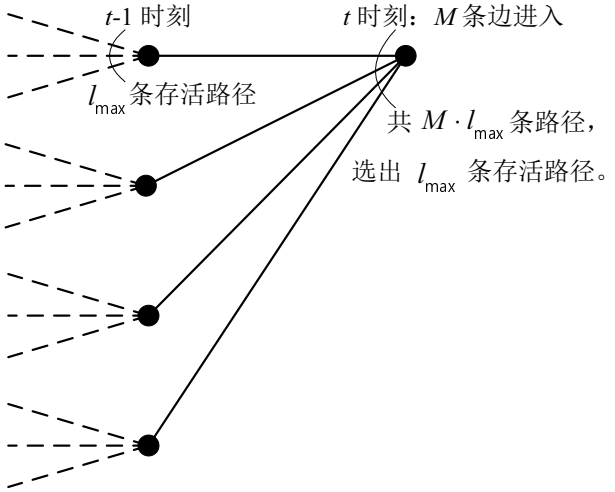


图 9.5

§9.4 BCJR 算法

维特比算法实现了最大似然序列译码（检测）算法，稍作修正，则可以实现最大后验概率序列译码算法。从符号角度，维特比算法不是最优的。最小符号错误概率算法是 1974 年提出的 BCJR（Bahl, Cocke, Jelinek, Raviv）算法，与维特比算法不同的是，BCJR 算法从概率域描述更为简单直接。为此，对于每一条边，定义一个度量

$$P(s_t, u_t, v_t, s_{t+1}) = P_U(u_t)P(y_t|v_t).$$

一条路径的度量定义为该路径所有边的度量连乘积。最大后验概率译码准则是计算 $P(u_t|y), u_t \in \mathcal{X}, 0 \le t < L$; 然后选择 $\hat{u}_t$ 使得 $P(u_t|y)$ 达到最大。由贝叶斯分析得到，

$$P(u_t|y) \propto \sum_{u:u_t} P(u)P(y|u).$$

换句话讲，将 t 时刻经过的边标为 u_t，把状态为 u_t 的所有路径的度量全部相加。BCJR 算法是实现这个目标的有效算法，其计算了“连乘积的和”，所以是“和积”算法（sum-product algorithm, SPA）。其基本单元包括前向递推、后向递推，以及中间提取等。前向递推是计算 $P(*, s_t)$，即进入状态 s_t 的所有路径的度量和。后向递推是计算 $P(s_t, *)$，即从状态 s_t 出发的所有路径的度量和。计算规则如下图所示。

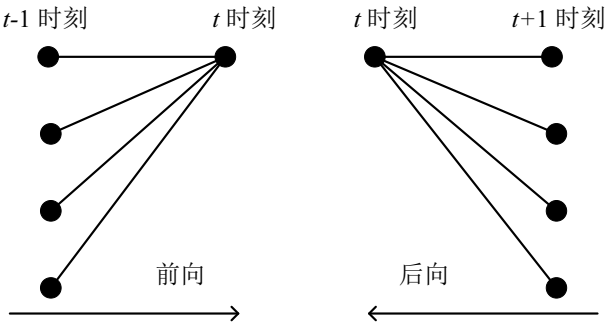


图 9.6

前向：$t-1$ 时刻的状态度量，经过乘、加运算后得到 t 时刻的状态度量。

后向：$t+1$ 时刻的状态度量，经过乘、加运算后得到 t 时刻的状态度量。

具体地，记 $\alpha_t(s)$ 表示在 t 时刻进入状态 s 的所有部分路径的度量和。类似地，记 $\beta_t(s)$ 表示在 t 时刻从状态 s 出发的所有部分路径的度量和，我们有如下算法。

1. 初始化：

$$\alpha_0(s)=\begin{cases}1, & s=0\\ 0, & s\neq 0\end{cases}$$

$$\beta_L(s)=\begin{cases}1, & s=0\\ 0, & s\neq 0\end{cases}$$

2. 前向递推

对于 $t\geq 1$，计算进入 s_t 的路径度量和，

$$\alpha_t(s_t)=\sum \alpha_{t-1}(s_{t-1})P_u(u_t)P(y_t|v_t),$$

其中，上面的求和是对于所有右状态是 s_t 的边进行的。

3. 后向递推

对于 $t<L$，计算从 s_t 出发的路径度量和，

$$\beta_t(s_t)=\sum \beta_{t+1}(s_{t+1})P_u(u_t)P(y_t|v_t),$$

其中，上面的求和是对于所有左状态是 s_t 的边进行的。

4. 中间提取

对于 t 时刻的一条边 $\underline{b}=(s_{t-1},u_t,v_t,s_t)$，其后验概率可以计算为

$$\Gamma(\underline{b})\propto \alpha_{t-1}(s_{t-1})P_u(u_t)P(y_t|v_t)\beta_t(s_t),$$

即对于每条边求上述乘积，然后进行归一化。

注：得到 t 时刻每条边的后验概率之后，根据需要可以计算相关概率。比如，要想知道某个符号的概率，把所有边标等于该符号的边的概率求和即可。由于最后计算边的概率时要归一化，所以，在前向递推/后向递推时，把 α、β 进行归一化是不影响中间提取的结果的，这是在计算中采取的必要手段，以防止计算向下溢出。

§9.5 仿真

编一个面向网格图的译码算法程序可以为研究带来许多便利。算法要以网格图作为一个数据结构，或者定义一个类，其基本成员包括边数、状态数、输入、输出边标等，用 4 个指针（动态数组，空间大小均为边数）表示即可。对于该结构，可以实现维特比算法、BCJR 算法以及相应的一些次优简化算法等。在这些算法中，基本的操作都是在网格图上逐节更新一些度量，这个更新以边为循环变量最为方便。另外，前面已经提到，要特别注意数值的溢出问题，基于理论作些必要的处理。比如，执行维特比算法时，可以把所有部分路径的度量同减去一个共同的数，是不影响最优路径的选择的。

§9.6 习题

1. 以某个重复码为例，画出其网格图。以某个奇偶校验码为例画出网格图。以某个卷积码为例画出其网格图。

2. 设 $X_i \in \{-1,+1\}, i \geq 1$，是一个随机变量序列。已知 X_i 的概率质量函数为 $(p_i, 1-p_i)$，其中 $p_i = \Pr\{X_i = -1\}$。利用网格图表示部分和变量 $S_t = \sum_{i \leq t} X_i, t \geq 1$，给出计算 S_t 概率质量函数的递推算法。

3. 网格图上算法的复杂度与边数、状态数都有关系。试着分析 Viterbi 算法、BCJR 算法的计算复杂度与空间复杂度（即存储所需空间）。

第十章

极化码

§10.1 编码树

给定整数 $n > 0$，定义如下一棵有根树，这棵树有 n 层，其根用 $\boldsymbol{v} \in \mathbb{F}_2^N, N = 2^n$ 表征，第 i 层的节点用 $\boldsymbol{v} \in \mathbb{F}_2^{N_i}, N_i = 2^{n-i}$ 表征。一个节点若不是叶子节点，则其与两个子节点的关系定义如下（如图 10.1），

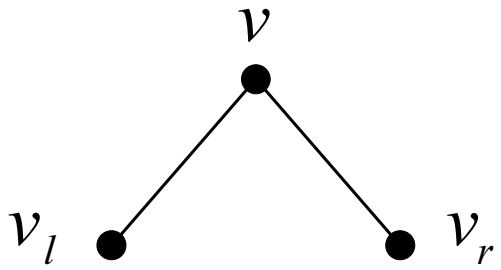


图 10.1

$$\boldsymbol{v} = (\boldsymbol{v}_l \oplus \boldsymbol{v}_r, \boldsymbol{v}_r)$$

其中 $\boldsymbol{v}$ 是父节点，$\boldsymbol{v}_l$ 与 $\boldsymbol{v}_r$ 分别是左右两个子节点。

上面定义的树共有 $N = 2^n$ 个叶子节点。所谓码的构造就是从中选取 $K < N$ 个叶子节点记为 $\mathcal{A}$，其余叶子节点记为 $\mathcal{F}$。编码时，将 K 个信息比特放置在 $\mathcal{A}$ 中的叶子节点之上，而 $\mathcal{F}$ 中的叶子节点置为比特 0。因此，$\mathcal{A}$ 中的节点称为活跃节点，而 $\mathcal{F}$ 中的节点称为冻结节点。编码的过程即是从叶子节点逐层向上计算，直至计算出根节点对应的码字向量。编码的复杂度是 $O(N \log N)$。

§10.2 译码

极化码的原始译码算法是依次递消算法。该算法可以沿用上一节定义的有根树表示，分成计算节点的软信息 $\boldsymbol{\alpha} \in \mathbb{R}^{N_i}$ 和判决节点的估计 $\boldsymbol{\beta} \in \mathbb{F}_2^{N_i}$ 两部分，N_i 表示节点的长度，详述如下。首先，当接收到序列 $\boldsymbol{y} = (y_0, y_1, \ldots, y_{N-1})$ 时，我们根据 $\alpha_j = \ln \frac{P(y_j|0)}{P(y_j|1)}$ 计算出根节点的软信息 $\boldsymbol{\alpha} = (\alpha_0, \alpha_1, \ldots, \alpha_{N-1})$，完成初始化。此后，我们根据如下规则计算节点的软信息和判决节点的比特估计。软信息的计算顺序为：父节点 $\to$ 左孩子节点 $\to$ 右孩子节点，比特估计的判决顺序为：左孩子节点 $\to$ 右孩子节点 $\to$ 父节点。

- 计算节点的软信息：

1. 当第 i 层的父节点得到软信息 $\alpha_j, 0 \le j \le N_i - 1$ 时，我们计算它的左孩子节点的软信息

$$\alpha_j^l = f(\alpha_j, \alpha_{N_i/2+j}), 0 \le j \le N_i/2 - 1.$$

其中，$f(x, y) \triangleq \ln \frac{e^{x+y}+1}{e^x+e^y} \approx \text{sign}(x)\text{sign}(y)\min(|x|, |y|)$。

2. 当第 i 层的父节点得到软信息 $\alpha_j, 0 \le j \le N_i - 1$ 并且它的左孩子节点得到比特估计 $\beta_j^l, 0 \le j \le N_i/2 - 1$ 时，我们计算它的右孩子节点的软信息

$$\alpha_j^r = g(\alpha_j, \alpha_{N_i/2+j}, \beta_j^l), 0 \le j \le N_i/2 - 1.$$

其中，$g(x, y, z) \triangleq (-1)^z \cdot x + y$。

- 判决节点的比特估计：

1. 对于冻结的叶子节点，我们直接判决它的估计 $\beta_0 = 0$.
2. 对于活跃的叶子节点，我们根据它的软信息 α_0 判决它的估计

$$\beta_0 = \begin{cases} 0, & \alpha_0 \ge 0 \\ 1, & \alpha_0 < 0 \end{cases}.$$

3. 当第 i 层的父节点的左右孩子节点分别得到估计 $\beta_j^l, 0 \le j \le N_i/2 - 1$ 和 $\beta_j^r, 0 \le j \le N_i/2 - 1$ 时，我们判决父节点的估计

$$\beta_j = \begin{cases} \beta_j^l \oplus \beta_j^r, & 0 \le j \le N_i/2 - 1 \\ \beta_j^r, & N_i/2 \le j \le N_i - 1 \end{cases}.$$

当得到所有叶子节点的比特估计后，译码结束。依次抵消译码的复杂度是 $O(N \log N)$。上述规则如图 10.2所示。

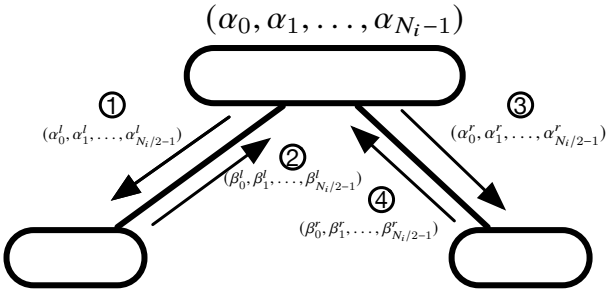


图 10.2

Arikan 提出的依次抵消译码算法可以保证极化码在码长趋于无穷时容量可达。但是，有限码长的极化码在依次抵消译码下的性能并不好。为了进一步提升性能，Tal 和 Vardy 提出了依次抵消列表译码算法。

在依次抵消列表译码算法中，我们需要事先设置一个列表大小 L，表示整个译码过程中最多可以保留 L 条译码路径，分别对应一棵译码树。准确地说，译码路径就是一棵树上已经得到的叶子比特估计组成的序列。首先，我们建立一颗译码树并根据接收序列 $\boldsymbol{y} = (y_0, y_1, \ldots, y_{N-1})$ 完成根节点的初始化。在这颗树上，我们按照依次抵消算法的规则计算节点的软信息，判决冻结节点。当判决活跃节点时，我们对这棵树进行复制，并在两棵树上对该叶子节点分别判决为 0 和 1。由此，一棵树分裂成两棵树。此后，我们对于每棵树分别继续进行软信息的计算，冻结节点的判决。当判决活跃节点时，每棵树再次一分为二。当译码路径的总数超过 L 时，我们需要保留 L 条最可靠的译码路径，删除其他的译码路径。因此，我们需要定义译码路径的可靠性度量，具体如下。假设一棵树已经得到前 t 个叶子节点的判决结果和软信息，这里分别记作 $(\hat{u}_0, \hat{u}_1, \ldots, \hat{u}_{t-1})$ 和 $(\lambda_0, \lambda_1, \ldots, \lambda_{t-1})$。此时，这棵树对应译码路径就是 $(\hat{u}_0, \hat{u}_1, \ldots, \hat{u}_{t-1})$，而路径的可靠性度量为

$$\text{PM}_t \triangleq \text{PM}_{t-1} + \ln(1 + e^{-(1-2\hat{u}_{t-1})\lambda_{t-1}}).$$

其中，$\text{PM}_{-1} = 0$。本质上，$\text{PM}_t = -\ln P(\hat{u}_0^{t-1}|\boldsymbol{y})$。$\text{PM}_t$ 越小，代表这条路径越可靠。当完成所有叶子节点的判决后，我们选择最可靠的路径作为译码结果。依次抵消列表译码的复杂度是 $O(LN \log N)$。

经过统计，我们发现在对极化码采用依次抵消列表译码时，会出现正确路径在最终列表中，但它并不是最可靠的路径的情况，从而导致译码失败。进一步地，Tal 和 Vardy 提出了循环冗余校验辅助的依次抵消列表译码算法。假设循环冗余校验比特的个数为 r。在编码时，对于 K 个信息比特，先进行循环冗余校验编码，再将 $K + r$ 个比特进行极化码编码。在译码时，对于 $K + r$ 个比特先进行列表大小为 L 的依次抵消列表译码，再从 L 个候选码字中选择通过循环冗余校验且最可靠的路径作为译码结果。

参考文献

[1] C. E. Shannon, "A mathematical theory of communication," The Bell System Technical Journal, vol. 27, no. 3, pp. 379–423, 623–656, 1948.

[2] I. Csiszár and J. Körner, Information theory: coding theorems for discrete memoryless systems. Cambridge University Press, 2011.

[3] T. Cover and J. Thomas, Elements of information theory. Wiley, 1991.

[4] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," Communications of the ACM, vol. 30, no. 6, pp. 520–540, 1987.

[5] R. G. Gallager, Information theory and reliable communication. Springer, 1968, vol. 2.