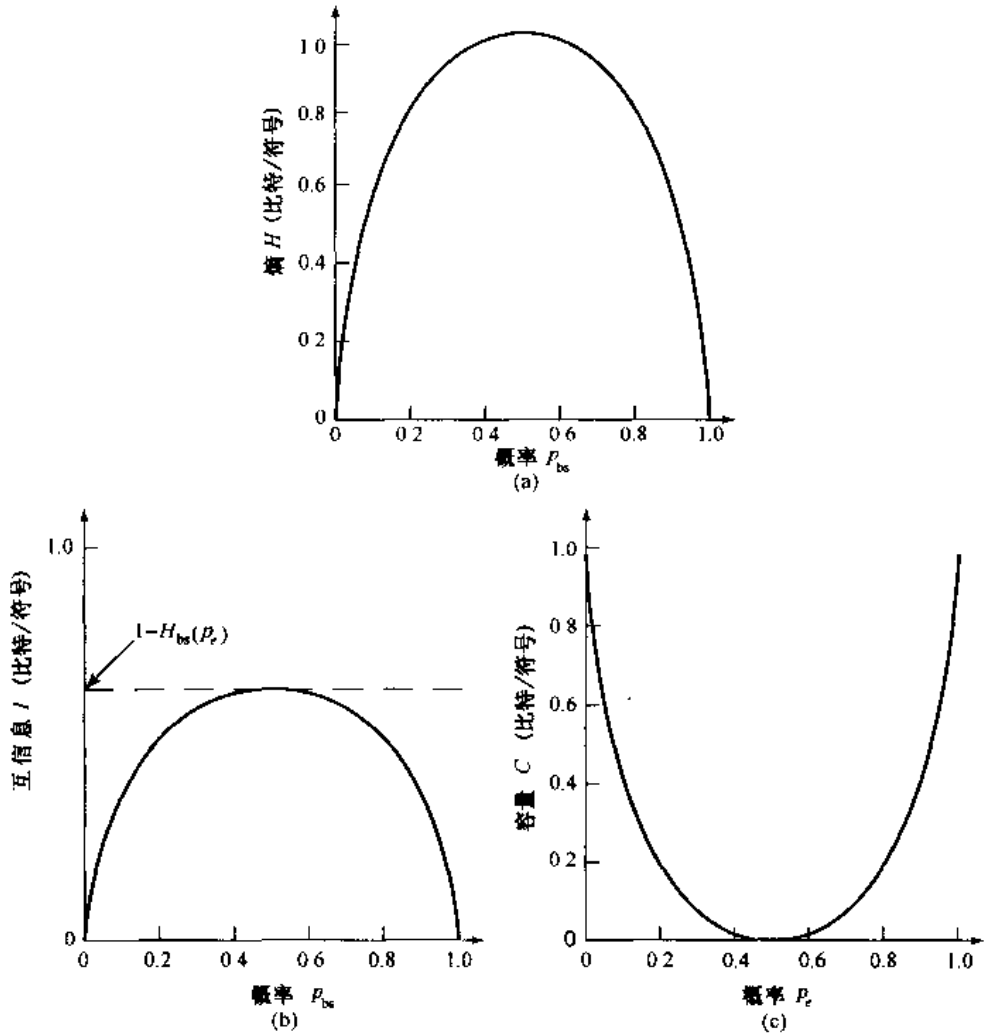

图 8.8 三个二值信息函数。(a)二值熵函数,(b)二值对称信道(BSC)的互信息,(c)BSC的容量

8.3.3 基本编码定理

8.3.2 节中介绍的总体数学框架是以图 8.7 中显示的模型为基础的。图 8.7 中包含一个信源、信道和信宿。这一节,在模型中增加了一个通信系统,并分析关于编码或信息表示的三个基本理论。如图 8.9 所示,通信系统插在信源和信宿之间,它包含一个编码器和一个解码器。

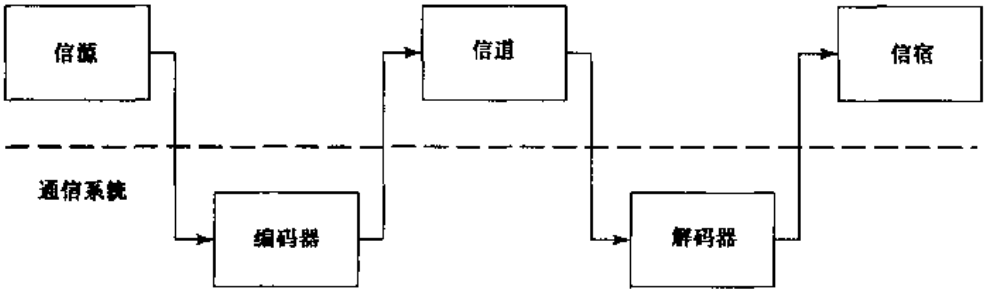

图 8.9 一个通信系统的模型

无噪声编码定理

当信道和通信系统中不存在噪声的时候，通信系统的主要功能是用尽可能简捷的方法表示信源。在这样的情况下，无噪声编码定理，又称香农第一定理(香农，Shannon[1948])，定义了可以达到的每个信源符号的最小平均码字长度。

一个具有有限集合$(A,\mathbf{z})$和在统计上独立的符号源的信息源被称为零记忆信源。如果将它的输出看做来自信源字母表的(而不是一个单一的符号)符号的一个 n 元组，信源输出就取自所有可能的 n 个元素序列集合 $A'=\{\alpha_1,\alpha_2,\cdots,\alpha_{J'}\}$ 的 J' 个可能值中的一个，表示为 α_i。换句话说，每个 α_i(称为一个块随机变量)由来自 A 的 n 个符号组成(符号 A' 区别于来自 A 的块符号集合，A 是单个符号集合)。一个给定的 α_i 的概率是 $P(\alpha_i)$，…这个概率是与单符号概率 $P(a_j)$ 通过下式相联系的：

$$P(\alpha_i)=P(a_{j1})P(a_{j2})\cdots P(a_{jn}) \tag{8.3.14}$$

这里，下标 $j1,j2,\cdots,jn$ 用于指示来自集合 A 的 n 个符号，它构成了一个 α_i。如前所述，向量 $\mathbf{z}'$(撇号表示使用了块随机变量)表示所有信源概率的集合 $\{P(\alpha_1),P(\alpha_2),\cdots,P(\alpha_{J'})\}$，信源的熵为：

$$H(\mathbf{z}')=-\sum_{i=1}^{J'}P(\alpha_i)\log P(\alpha_i)$$

用式(8.3.14)代替 $P(\alpha_i)$ 经过化简得到：

$$H(\mathbf{z}')=nH(\mathbf{z}) \tag{8.3.15}$$

因此，零记忆信息源(这个信源产生块随机变量)的熵为对应的单符号信源熵的 n 倍。这样一个信源被称为单一符号或非扩充信源的第 n 次扩充。注意信源的首次扩充都是非扩充信源本身。

因为信源输出 α_i 的自信息是 $\log[1/P(\alpha_i)]$，所以用一个整数长度 $l(\alpha_i)$ 的码字对 α_i 进行编码似乎是合理的。即：

$$\log\frac{1}{P(\alpha_i)}\leqslant l(\alpha_i)<\log\frac{1}{P(\alpha_i)}+1 \tag{8.3.16}$$

在直观上的提示：信源输出 α_i 是用一个码字表示的，这个码字的长度为超出 α_i 自信息的最小整数。[①] 用 $P(\alpha_i)$ 与这个结果相乘并将所有乘积相加得到：

$$\sum_{i=1}^{J'}P(\alpha_i)\log\frac{1}{P(\alpha_i)}\leqslant\sum_{i=1}^{J'}P(\alpha_i)l(\alpha_i)<\sum_{i=1}^{J'}P(\alpha_i)\log\frac{1}{P(\alpha_i)}+1$$

或

$$H(\mathbf{z}')\leqslant L'_{\text{avg}}<H(\mathbf{z}')+1 \tag{8.3.17}$$

这里 L'_{avg} 表示对应于非扩充信源的第 n 次扩充的编码平均字长。即，

$$L'_{\text{avg}}=\sum_{i=1}^{J'}P(\alpha_i)l(\alpha_i) \tag{8.3.18}$$

用 n 除式(8.3.17)并注意到式(8.3.15)中的 $H(\mathbf{z}')/n$ 等于 $H(\mathbf{z})$，得到：

① 在这种限制之下可以构造一个惟一的可解的码。

$$H(\mathbf{z}) \leqslant \frac{L'_{\text{avg}}}{n} < H(\mathbf{z}) + \frac{1}{n} \tag{8.3.19}$$

在取极限情况下,变成:

$$\lim_{n\to\infty}\left[\frac{L'_{\text{avg}}}{n}\right] = H(\mathbf{z}) \tag{8.3.20}$$

式(8.3.19)阐述了一个零记忆信源的香农第一定理。这个定理说明通过对无限扩充的信源进行编码可以令 L'_{avg}/n 无限地接近 $H(\mathbf{z})$。尽管这个结论是在假设信源符号在统计上相互独立的情况下推导出来的,但这个结论可以很容易推广到更一般的信源,这些信源的符号 a_j 的出现取决于位于它前面的有限个符号。这些类型的信源(称为马尔可夫信源)通常用于图像中像素间相关的模型。因为 $H(\mathbf{z})$ 是 L'_{avg}/n 的下限[即,随着 n 的增大,式(8.3.20)中 L'_{avg}/n 的极限为$H(\mathbf{z})$],所以任何编码策略的效率 η 可以定义为:

$$\eta = n\frac{H(\mathbf{z})}{L'_{\text{avg}}} \tag{8.3.21}$$

例 8.7 扩充编码

一个带有信源字母表 $A=\{a_1,a_2\}$ 的零记忆信息源具有符号概率 $P(a_1)=2/3$ 和 $P(a_2)=1/3$。根据式(8.3.3),这个信源的熵为 0.918 比特/符号。如果符号 a_1 和 a_2 由二进制码字 0 和 1 表示,则 $L'_{\text{avg}}=1$ 比特/符号,并且编码的效率为 $\eta=(1)(0.918)/1$ 或 0.918。

表 8.4 总结了刚才提到的编码及一个基于信源第二次扩充的可作为替代的编码过程。表 8.4的下半部分列出了信源的第二次扩充中的四个块符号分别为 a_1,a_2,a_3 和 a_4。根据式(8.3.14),它们的概率分别为 4/9,2/9,2/9 和 1/9。根据式(8.3.18),第二次编码的平均码长是 17/9 或1.89 比特/符号。第二次扩充的熵是非扩充源的熵的两倍,或者 1.83 比特/符号,因此,第二次编码的效率为 $\eta=1.83/1.89=0.97$。这比非扩充编码的效率0.92稍好。对信源的第二次扩充编码将每个信源符号的编码比特平均数从 1 比特/符号减少到 1.89/2 或 0.94 比特/符号。

噪声编码定理

如果图 8.9 的信道是有噪声的或容易出错的,则关注的重点从以尽可能紧凑的信息表示的编码转到尽可能稳定的通信。这样很自然地会提出这样的问题:在通信过程中到底允许出现多小的错误呢?

例 8.8 噪声二值信道

假设 BSC 有出错概率 $p_e=0.01$(即,99% 的信源符号可以通过信道正确传送)。一个简单的增强通信可靠性的方法是将每个信息或二进制符号重复几遍。假设传送 0 和 1,编码信息可以使用 000 和 111。在传送三符号信息的过程中无错误发生的概率或者是 $(1-p_e)^3$或者是 $\bar{p}_e^3$。单个错误发生的概率是 $3p_e\bar{p}_e^2$。两个错误的概率是 $3p_e^2\bar{p}_e$,三个错误的概率是 p_e^3。因为单一符号传通错误发生的概率小于 50%,可以通过三个接收符号进行多数投票的方法对接收信息进行解码。因此,对三符号码字进行错误解码的概率是两个符号错误和三个符号错误的概率之和,即 $p_e^3+3p_e^2\bar{p}_e$。当无错误或单一错误发生时,多数投票可以将信息正确解码。对 $p_e=0.01$,出现一个通信错误的概率是 0.0003。

表 8.4　扩充编码示例

a_i	信源符号	$P(a_i)$,式(8.3.14)	$I(a_i)$,式(8.3.1)	$l(a_i)$,式(8.3.16)	码字	码长
第一次扩充						
a_1	a_1	2/3	0.59	1	0	1
a_2	a_2	1/3	1.58	2	1	1
第二次扩充						
a_1	a_1a_1	4/9	1.17	2	0	1
a_2	a_1a_2	2/9	2.17	3	10	2
a_3	a_2a_1	2/9	2.17	3	110	3
a_4	a_2a_2	1/9	3.17	4	111	3

通过对刚才描述的重复编码方案的扩充,将通信中发生的错误控制在希望的尽可能小的程度内。在一般情况下,通过使用长度为 r 的 K 元编码序列对信源进行第 n 次扩充来编码,这里 $K^r \leqslant J^n$。这种方法的关键是,在 K^r 个可能的编码序列中选择合适的 φ 作为有效的码字并建立一条规则使正确解码的概率得到最优化。在前面的例子中,将每个信源符号重复三次等同于,使用 8 个之中的两个可能出现的二进制码字,对非扩充二值信源进行块编码。两个有效的码字是 000 和 111。如果一个无效码字输送到解码器中,则由三个编码比特的多数投票决定输出。

一个零记忆信息源以一定速率(以每个符号的信息单元数表示)生成信息,速率等于这个信源的熵 $H(\mathbf{z})$。信源的第 n 次扩充以每个符号 $H(\mathbf{z}')/n$ 信息单元的速度提供信息。如果信息是进行过编码的,就像前面的例子那样,则当用于对信源编码的 φ 个有效码字等概率出现时,编码信息的最大速率是 $\log(\varphi/r)$。因此,尺寸为 φ 且块长度为 r 的编码具有:

$$R = \log \frac{\varphi}{r} \quad \text{信息单元 / 符号} \tag{8.3.22}$$

的速率。香农第二定理(Shannon[1948])也称为噪声编码定理,告诉我们对任意 $R < C$,存在一个整数 r 和块长度为 r、速率为 R 的编码,这个编码的块解码误差的概率小于或等于任意的 $\varepsilon > 0$,这里 C 是具有矩阵 $\mathbf{Q}$ 的零记忆信道[①] 的容量。因此,只要编码信息率小于信道容量,则出现误差的概率就可以为任意小。

信源编码定理

这个定理描述了迄今为止在可靠信道和不可靠信道上进行无误差通信时建立的基本限制。在这一节中,转向信道无误差而通信过程本身有损耗的情况上来。在这样的环境下,通信系统的主要功能是"信息压缩"。在大多数情况下,由压缩引入的平均差错受到某个最大允许级别 D 的限制。我们希望判定在给定的保真度准则限制下,信源向信宿传送信息的最小速率。这个问题在称为"率失真理论"的信息论分支中进行了明确的讲述。

分别使用有限集合 $(A,\mathbf{z})$ 和 $(B,\mathbf{z})$ 定义图 8.9 中的信源和解码输出。假设图 8.9 的信道

① 一个零记忆信道是指信道对当前输入的响应与对前一个输入符号的响应是彼此独立的。

是不出现错误的,因此,根据式(8.3.6)将 $\mathbf{z}$ 和 $\mathbf{v}$ 相联系的信道矩阵 $\mathbf{Q}$ 可以被认为是将编解码处理进行模型化的过程。因为编解码处理是确定性的,$\mathbf{Q}$ 描述了一个人工零记忆信道,它模型化了信息压缩和解压缩的功能。每次信源生成信源符号 a_j 时,都用一个编码符号表示,这个编码符号以概率 q_{kj} 解码为输出符号 b_k(见 8.3.2 节)。

解决信源编码问题,使其平均失真小于 D,需要对每个可能的信源输出的近似值指定一个数量的失真值。对非扩充信源的简单情况,一个称为失真度量的非负值开销函数 $\rho(a_j, b_k)$ 可用于定义再次生成解码输出 b_k 的信源输出 a_j 时进行的相应处理。信源的输出是随机的,因此,其失真同样是随机变量,这个随机变量用 $d(\mathbf{Q})$ 表示为:

$$\begin{aligned} d(\mathbf{Q}) &= \sum_{j=1}^{J}\sum_{k=1}^{K}\rho(a_j, b_k)P(a_j, b_k) \\ &= \sum_{j=1}^{J}\sum_{k=1}^{K}\rho(a_j, b_k)P(a_j)q_{kj} \end{aligned} \tag{8.3.23}$$

符号 $d(\mathbf{Q})$ 强调平均失真是编解码处理过程的一个函数。编解码过程用 $\mathbf{Q}$(前面提到过的)来模型化。当且仅当与 $\mathbf{Q}$ 对应的平均失真小于或等于 D 时,特定的编解码过程才被称为是 D 所接纳的。因此,所有 D 所接纳的编解码过程的集合表示为:

$$\mathbf{Q}_D = \{q_{kj} \mid d(\mathbf{Q}) \leqslant D\} \tag{8.3.24}$$

因为每个编解码过程都是用一个人工信道矩阵 $\mathbf{Q}$ 定义的,所以根据观察单一解码器的输出得到的平均信息可以根据式(8.3.12)进行计算。因此,可以定义一个率失真函数:

$$R(D) = \min_{\mathbf{Q}\in\mathbf{Q}_D}[I(\mathbf{z}, \mathbf{v})] \tag{8.3.25}$$

这里,假设式(8.3.12)的最小值超出了 D 所接纳的编码。注意,最小值可以在 $\mathbf{Q}$ 上取得,因为 $I(\mathbf{z}, \mathbf{v})$ 是向量 $\mathbf{z}$ 中的概率和矩阵 $\mathbf{Q}$ 中元素的函数。如果 $D = 0$,则 $R(D)$ 小于或等于信源的熵,或 $R(0) \leqslant H(\mathbf{z})$。

式(8.3.25)定义了在限制平均失真小于或等于 D 的条件下,信源可以传送给信宿的信息的最小速率。为了计算这个速率[即,$R(D)$],可以简单地通过对受下列限制的 $\mathbf{Q}$(或 q_{kj})进行适当的选择最小化 $I(\mathbf{z}, \mathbf{v})$。

$$q_{kj} \geqslant 0 \tag{8.3.26}$$

$$\sum_{k=1}^{K} q_{kj} = 1 \tag{8.3.27}$$

和

$$d(\mathbf{Q}) = D \tag{8.3.28}$$

式(8.3.26)和式(8.3.27)是信道矩阵 $\mathbf{Q}$ 的基本性质。$\mathbf{Q}$ 的元素必须为正,且因为对于任何产生的输入符号有些输出是必然会收到的,所以 $\mathbf{Q}$ 中每一列的和必须为 1。式(8.3.28)指出当允许出现最大可能失真的时候,就会得到最小信息速率。

例 8.9 计算一个零记忆二值信源的率失真函数

考虑一个具有等概率信源符号{0,1}的零记忆二值信源,其简单失真量度为:

$$\rho(a_j, b_k) = 1 - \delta_{jk}$$

这里 δ_{jk} 是单位德尔塔函数。因为如果 $a_j \neq b_k$,则 $\rho(a_j, b_k)$ 等于 1,否则为 0,所以每个编

码解码误差被记为一个单位的失真。这个变量的微分可以用于计算 $R(D)$。令 $\mu_1, \mu_2, \cdots, \mu_{J+1}$ 为拉格朗日乘数，构造增强的准则函数：

$$J(\mathbf{Q}) = I(\mathbf{z},\mathbf{v}) - \sum_{j=1}^{J} \mu_j \sum_{k=1}^{K} q_{kj} - \mu_{J+1} d(\mathbf{Q})$$

令它的 JK 关于 q_{kj} 的导数为 0(即，$\mathrm{d}J/\mathrm{d}q_{kj} = 0$)，并连同与式(8.3.27)和式(8.3.28)相关的 $J+1$ 个方程式对未知量 q_{kj} 和 $\mu_1, \mu_2, \cdots, \mu_{J+1}$ 求解，得出方程式。如果得到的 q_{kj} 是非负值[或满足式(8.3.26)]，就找到了一个有效的解。对于上述定义的信源和失真对，得到下列 7 个公式(带有 7 个未知量)：

$$2q_{11} = (q_{11} + q_{12})\exp[2\mu_1] \qquad 2q_{22} = (q_{21} + q_{22})\exp[2\mu_2]$$
$$2q_{12} = (q_{11} + q_{12})\exp[2\mu_1 + \mu_3] \qquad 2q_{21} = (q_{21} + q_{22})\exp[2\mu_2 + \mu_3]$$
$$q_{11} + q_{21} = 1 \qquad q_{12} + q_{22} = 1$$
$$q_{21} + q_{12} = 2D$$

由此生成了一系列冗长的但形式简单的代数运算步骤：

$$q_{12} = q_{21} = D$$
$$q_{11} = q_{22} = 1 - D$$
$$\mu_1 = \mu_2 = \log\sqrt{2(1-D)}$$
$$\mu_3 = \log\frac{D}{1-D}$$

这样得到：

$$\mathbf{Q} = \begin{bmatrix} 1-D & D \\ D & 1-D \end{bmatrix}$$

由于给出的信源符号是等概率的，所以最大可能失真是 1/2。因此 $0 \leqslant D \leqslant 1/2$ 和 $\mathbf{Q}$ 的元素对所有的 D 都满足式(8.3.12)。与 $\mathbf{Q}$ 相关的互信息和前面定义的二值信源是用式(8.3.12)计算的。然而，注意到在 $\mathbf{Q}$ 和二值对称信道矩阵之间的相似性，可以很快写出：

$$I(\mathbf{z},\mathbf{v}) = 1 - H_{bs}(D)$$

这个结果遵循例 8.6，将 $p_{bs} = 1/2$ 和 $p_e = D$ 代入 $I(\mathbf{z},\mathbf{v}) = H_{bs}(p_{bs}p_e + \bar{p}_{bs}\bar{p}_e) - H_{bs}(p_e)$。根据式(8.3.25)率失真函数为：

$$R(D) = \min_{\mathbf{Q}\in\mathbf{Q}_D}[1 - H_{bs}(D)] = 1 - H_{bs}(D)$$

最终的简化是基于以下条件，即对给定的 D，$1 - H_{bs}(D)$ 假定为单值，默认情况下为最小值。得到的函数即图 8.10 中所示的曲线。这条曲线的形状是典型的最大率失真函数。注意，D 的最大值表示为 D_{max}。这个值对于所有的 $D \geqslant D_{max}$ 有 $R(D) = 0$。另外，$R(D)$ 总是正值，且单调递减，在区间 $(0, D_{max})$ 内为凸形的。

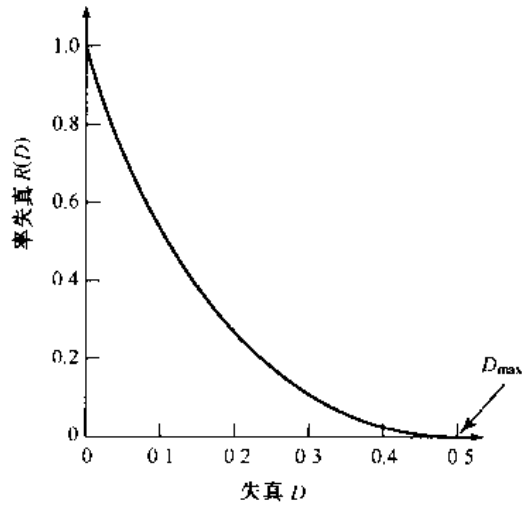

图 8.10 二值对称信源的率失真函数

正如前面的例子,对简单信源和失真量度可计算率失真函数。再有,在分析方法失败或方法实现起来不现实的时候,可以使用适于在数字计算机上实现的收敛的迭代算法。计算出 $R(D)$之后(对任何零记忆信源和单字符失真量度[①]),信源编码定理告诉我们对任意的 $\varepsilon>0$,存在一个 r 和块长度为 r、传送速率 $R<R(D)+\varepsilon$ 的编码,使得每个字符的平均失真满足 $d(\mathbf{Q})\leqslant D+\varepsilon$的条件。这个定理和噪声编码定理的一个重要的实用结果是,在容量为 $C>R(D)+\varepsilon$的信道所提供的误差概率为任意小的解码器中,可以对信源输出进行复原。这个结果就是众所周知的信息传输定理。

8.3.4 信息论的应用

信息论提供了处理信息表达和直接定量操作的基本工具。在本小节中,将探寻这些工具在特定的图像压缩问题中的应用。因为信息论的基本前提是生成的信息可以被模型化为一个概率过程,所以首先建立图像生成过程的统计模型。

例 8.10 计算图像的熵

考虑简单的 8 比特图像信息内容(或熵)的估计问题:

21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243

一种相对简单的方法是假设一个特定的信源模型,并计算基于这个模型的图像的熵。例如,可以假设图像由虚构的"8 比特灰度级信源"生成,这个信源根据事先定义的概率定律产生在统计上相互独立的像素。在这种情况下,信源符号是有灰度级的,并且信源字符表由 256 个可能的符号构成。如果符号出现的概率是已知的,图像中每个像素的平均信息量或熵可以使用式(8.3.3)计算得到。例如,在具有均匀概率密度的情况下,信源符号是

① 单字符失真量度是指与一个字符(或符号)块中每个字符(或符号)的失真之和相对应的失真。

等概率的，且信源由一个 8 比特/像素的熵进行表征。即，每个信源输出(像素)的平均信息是8 比特。这样一来，前述 4×8 大小的图像的总平均信息量是 256 比特。但这幅特殊图像仅仅是由这个信源产生的 $2^{8\times4\times8}$ 或 2^{256}(~ 10^{77})种等概率的 4×8 大小图像的一种。

一种可作为替代的估计信息量的方法是，构造一个基于图像中灰度级出现频率的信源模型。即，被观察的图像可以简单解释为生成这幅图像的灰度级信源的特性。因为被观察图像是信源特性惟一有效的指示器，用取样图像的灰度级直方图对信源符号的概率模型化是较为合理的。

灰 度 级	计　数	概　率
21	12	3/8
95	4	1/8
169	4	1/8
243	12	3/8

称为一阶估计的信源熵的估计可以用式(8.3.3)计算。本例子的一阶估计为 1.81 比特/像素。信源和/或图像的熵就是 1.81 比特/像素，或总共 58 比特。

产生取样图像的灰度级信源熵的更好估计，可以通过分析取样图像中像素块的相对频率来计算，这里，一个块就是邻接的像素组。当块的大小接近无限大时，这个估计也接近信源真实的熵(这个结果可以用 8.3.3 节中证明无噪声编码定理的有效性时所使用的过程进行说明)。因此，通过假设取样图像是线与线首尾相接的，能够计算像素对的相对频率(即，信源的第二次扩充)：

灰度级对	计　数	概　率
(21,21)	8	1/4
(21,95)	4	1/8
(95,169)	4	1/8
(169,243)	4	1/8
(243,243)	8	1/4
(243,21)	4	1/8

得到的熵估计[再次使用式(8.3.3)]是 2.5/2，或 1.25 比特/像素。这里用 2 除的原因是由于一次同时考虑了两个像素。这个估计被称为信源熵的二阶估计，因为这个估计是通过计算两个像素块的相对频率得到的。尽管三阶、四阶和更高阶的估计可以提供信源熵的更好近似，但它们收敛于真实信源熵的速度太慢并且陷于大量计算之中。例如，普通的 8 比特图像具有 $(2^8)^2$ 或 65 536 种可能的符号对，这些符号对的相对频率都必须进行计算。如果考虑一个 5 像素的块，可能的 5 元组的数目是 $(2^8)^5$ 种，或 10^{12} 种。

尽管计算一幅图像实际的熵是困难的，但像前述例子那样对图像进行的估计，提供了一种对图像压缩能力的认识。例如，熵的一阶估计是通过变长编码可以实现压缩的下界(在 8.1.1 节，变长编码用于减少编码冗余)。另外，熵的高阶估计和一阶估计之间的差异表明了是否存在像素间冗余。即，它们显示出图像中的像素在统计上是否是彼此独立的。如果像素

在统计上是独立的(即,不存在像素间冗余),则高阶估计等于一阶估计,且变长编码提供了最佳的压缩。对于前述例子中考虑的图像,在一阶和二阶估计之间用数字表示的差异表明,可以建立一种映射,以便允许从图像表达中去掉额外的 $1.81-1.25=0.56$ 比特/像素。

例 8.11 使用映射减少熵

考虑前述例子的图像中像素的映射,构造一种表达方法:

21	0	0	74	74	74	0	0
21	0	0	74	74	74	0	0
21	0	0	74	74	74	0	0
21	0	0	74	74	74	0	0

这里,通过复制原图像的第一列并使用相邻列的余下元素的算术差构造一个差异阵列。例如,在新的表示方法中,第 1 行第 2 列的元素是(21 - 21),或 0。得到的差异分布是:

灰度级或差异	计 数	概 率
0	12	1/2
21	4	1/8
74	12	3/8

如果现在考虑"差异信源"生成的映射阵列,可以再次使用式(8.3.3)计算阵列熵的一阶估计,值为 1.41 比特/像素。因此通过对映射差异图像进行变长编码,原图像可以用 1.41 比特/像素或总共 46 比特表示。这个值比前述例子中计算的 1.25 比特/像素的熵的二阶估计要大,因此,我们知道可以找到更好的映射。

前述例子说明一幅图像熵的一阶估计不一定必然是图像的最小码率。原因是图像中的像素一般在统计上不是彼此独立的。正如 8.2 节中注意到的,一幅图像实际熵的最小化过程被称为信源编码。在无错误的情况下,这个过程包括映射和符号编码两种操作。如果信息损失是允许的,这个过程还包括第三步量化操作。

关于有损图像压缩的更为复杂一些的问题同样可以使用信息论这一工具解决。然而,在这种情况下,主要应用信源编码定理。正如 8.3.3 节中指出的那样,这个定理揭示,任何零记忆信源可以用一个速率 $R<R(D)$ 的码进行编码,以便每个符号失真的平均值小于 D。为了将这个结果正确地应用于有损图像压缩,需要确定适当的信源模型,设计一个有意义的失真度量,并计算率失真函数 $R(D)$。这个处理过程的第一步已经研究过了。第二步可以用来自 8.1.4 节的客观保真度准则方便地解决。最后一步涉及寻找某个矩阵 $\mathbf{Q}$,这个矩阵是由式(8.3.24)到式(8.3.28)的限制条件下使式(8.3.12)最小化的元素所组成的。遗憾的是,这个工作非常困难——在实际感兴趣的所有情况中,只有一小部分得到了解决。其中一个是图像为高斯型随机场并且失真度量是一个加权的平方误差函数的情况。此时,最佳编码器必须将图像扩展为它的卡南-洛耶夫(Karhunen-Loève)分量(见 11.4 节)且用相等的均方误差表示每个分量(Davisson[1972])。

8.4 无误差压缩

在众多的应用中，无误差压缩是仅有的可以接受的数据压缩方法。这类应用中的一种是医疗或商业文件的归档。在这些应用场合，有损压缩通常因为法律原因而被禁止。另一种应用是卫星成像的处理。在这类应用中，收集到的数据的使用和所花的费用都使得人们不希望有任何的数据损失。还有一类应用是数字 X 光照相术，这种应用中信息的丢失会导致损坏诊断的精确性。在这一类情况下，无误差压缩的需要是由预期用户和图像性质所推动的。

在本节中，我们的注意力将放在当前应用中的无误差压缩策略上。这些策略通常提供的压缩率为 2 到 10。另外，它们对二值图像和灰度级图像的适用性是相同的。正如 8.2 节中所指出的那样，无误差压缩技术通常由两种彼此独立的操作组成：(1)为减少像素间冗余，建立一种可替代的图像表达方式；(2)对这种表达方式进行编码以便消除编码冗余。这些步骤与图 8.6讨论的信源编码模型的映射和符号编码操作相对应。

8.4.1 变长编码

无误差图像压缩的最简单方法就是减少仅有的编码冗余。编码冗余通常存在于图像灰度级的自然二进制编码过程中。如 8.1.1 节中提到的，它可以通过对灰度级进行编码使式(8.1.4)最小化得到消除。这样做需要变长编码结构，它可把最短的码字赋予出现概率最大的灰度级。这里，对于构造这样的码字分析几种最佳的和接近最佳的编码技术。这些技术都是使用信息论的语言进行表达的。实际上，信源符号既可能是图像灰度级，也可能是灰度级映射操作的输出(像素差异、行程宽度，等等)。

霍夫曼编码

消除编码冗余的最常用技术要归功于霍夫曼(Huffman[1952])。当对独立信息源的符号进行编码的时候，霍夫曼编码对每个信源符号生成可能的最小数量的编码符号。无噪声编码定理(见 8.3.3 节)对于 n 的一个固定值生成的编码是最佳的，但受限于每次只能对一个信源符号进行编码。

霍夫曼方法的第一步是将需要考虑的符号概率进行排序，并将具有最低概率的符号联结为一个单一的符号，用这个符号在信源化简的下一步中替代联结之前的两个符号。图8.11说明了在二值编码情况下的这种处理过程(也可以构造 K 元的霍夫曼编码)。在最左边，一个假设的信源符号集合和它们的概率从上到下按概率值减少的顺序排列。进行第一次信源化简时，底部的两个概率 0.06 和 0.04 联结成为一个概率值为 0.1 的“复合符号”。这个复合符号和它对应的概率被置于第一次信源化简的列中，以便化简后的信源概率仍旧是从大到小排列。这个过程一直重复持续到信源只有两个符号(在最右边)。

霍夫曼编码过程的第二个步骤是对每个化简后的信源进行编码，从最小的信源开始，一直工作到原始的信源。当然，对于一个两个符号信源的最小长度，二值编码是符号 0 和 1。如图 8.12所示，这些符号被分配给最右边的两个符号(这种分配是任意的；掉转 0 和 1 的顺序并无妨碍)。当简化信源中，合并的两个符号产生的概率为 0.6 的简化信源符号赋值时，用 0 去编码，而后将一个 0 和 1 任意地附加在这两个符号编码上面，用于将两个符号彼此区分开。