第四章无失真信源编码

第四节续：变长码

陈兴同

中国矿业大学 数学学院

2021 年 8 月

内容提要

内容提要

内容提要

1. Huffman 编码
2. Fano 编码
3. Elias 编码

内容提要

1. Huffman 编码
2. Fano 编码
3. Elias 编码
4. 算术码

编码算法:

1952 年 Huffman 提出了构造最优码的一个算法。设离散无记忆信源有概率分布（4-2），字符空间 $\mathcal{X}$ 上的二进 Huffman 编码算法如下：（1）将所有 N 个消息字符按它的概率从大到小降序排列

$$\begin{array}{ccccccccc} a_1 & & a_2 & & a_3 & & \cdots & & a_N \\ p_1 & \geq & p_2 & \geq & p_3 & \geq & \cdots & \geq & p_N \end{array}.$$

（2）将两个最小概率对应的字符 a_{N-1}, a_N 合并成一个字符 $\tilde{a}_{N-1}$，前面的字符不变，则得到一个新的字符空间及其上概率分布

$$\tilde{X} \sim p(x) = \begin{pmatrix} a_1 & a_2 & \cdots & a_{N-2} & \tilde{a}_{N-1} \\ p_1 & p_2 & \cdots & p_{N-2} & \tilde{p}_{N-1} \end{pmatrix}, \tilde{p}_{N-1} = p_{N-1} + p_N.$$

(4.16) ($\cancel{1.1}$)

续算法：

（3）对新随机变量 $\tilde{X}$ 的字符 $a_1, a_2, \cdots, a_{N-2}, \tilde{a}_{N-1}$ 进行即时码编码，得到码字

$$\tilde{c}_1, \tilde{c}_2, \cdots, \tilde{c}_{N-2}, \tilde{c}_{N-1}.$$

（4） 将最后一个码字 $\tilde{c}_{N-1}$ 分别添加 0 与 1 作为后缀，构造两个新的码字即 $c_{N-1} = \tilde{c}_{N-1}0, c_N = \tilde{c}_{N-1}1$ ，其它码字及顺序不变，则得原来离散无记忆信源的字符的编码 $c_1, c_2, \cdots, c_N$。
（5）采用递归方法，不断进行信源字符集缩减，直到剩下 2 个字符为止，这两个字符编码为 0、1。

定理 4.4.9：最优码定理

上述算法生成的二进 Huffman 编码是最优码。

证明：对 $N = 2, 3, 4, \cdots$ 使用数学归纳法。
Step 1: 当 $N = 2$ 时，字符空间 $\mathcal{X}$ 只有两个字符，这时对应 Huffman 编码只能是 $0, 1$，它们就是平均码长（等于 1bit）最小的即时码。

续证明：

Step 2: 当 $N>2$ 时，假设对有 $N-1$ 个字符的字符空间 $\tilde{\mathcal{X}}$ 进行 Huffman 编码能得到最优码。对有 N 个字符的字符空间 $\mathcal{X}$ 进行 Huffman 编码，记它的缩减字符空间为 $\tilde{\mathcal{X}}$，这个只有 $N-1$ 字符的字符空间上有概率分布 (4.16)，对它进行 Huffman 编码，得码字 $\tilde{c}_1,\tilde{c}_2,\cdots,\tilde{c}_{N-1}$，根据假设它就是缩减字符空间上的最优码，记码长序列为 $\tilde{l}_1,\tilde{l}_2,\cdots,\tilde{l}_{N-1}$，将最后一个码字 $\tilde{c}_{N-1}$ 分别添加 0 与 1 作为后缀，构造两个新的码字即 $c_{N-1}=\tilde{c}_{N-1}0, c_N=\tilde{c}_{N-1}1$，其它码字及顺序不变，则得原来字符空间 $\mathcal{X}$ 上 Huffman 编码 $c_1,c_2,\cdots,c_N$。它们的码长有关系

$$l_i=\tilde{l}_i \quad (i=1,\cdots,N-2), l_{N-1}=l_N=\tilde{l}_{N-1}+1.$$

续证明:

于是对于码 $c_1, c_2, \cdots, c_N$ 的平均码长 $\bar{L}_X$ 为

$$\begin{aligned}
\bar{L}_X = \sum_{i=1}^{N} p_i l_i &= \sum_{i=1}^{N-2} p_i \tilde{l}_i + p_{N-1}(\tilde{l}_{N-1} + 1) + p_N(\tilde{l}_{N-1} + 1) \\
&= \sum_{i=1}^{N-2} p_i \tilde{l}_i + (p_{N-1} + p_N)\tilde{l}_{N-1} + p_{N-1} + p_N \\
&= \sum_{i=1}^{N-1} p_i \tilde{l}_i + p_{N-1} + p_N \\
&= \bar{L}_{\tilde{X}} + p_{N-1} + p_N,
\end{aligned}$$

由于平均码长 $\bar{L}_{\tilde{X}}$ 最小，故 $\bar{L}_X$ 也最小。
这个定理说明，Huffman 编码算法也给出了前面关于码长的整数最优化问题的解。

例题 4.4.5：求哈夫曼编码

设随机变量 X 的分布如下，求二进 Huffman 编码。

$$X \sim p(x) = \begin{pmatrix} a_1 & a_2 & a_3 & a_4 & a_5 \\ 0.25 & 0.15 & 0.20 & 0.15 & 0.25 \end{pmatrix}.$$

解：由下面码树（图 ??）中从根到叶子路径中的权值即构成 Huffman 码。

4.9

续解：码树及编码

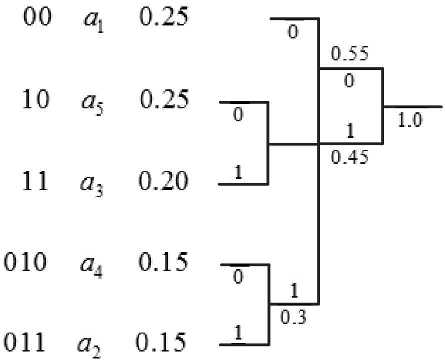


Figure: 图 4-9

其平均码长2.3 bits，$H(X)=2.285$ bits

二进 Huffman 编码性质：

（1） 概率较小的信源字符或消息有较长的码字，即 $p_i > p_j \Rightarrow l_i \leq l_j$。

（2） 概率最小的两个字符对应的码字具有相同的最大码长；并且两个码字只有最后一位不同，前面的每一位都相同。

（3） Huffman 码的码树是一颗完全树。如果一种编码的码树不是完全树，则这个编码一定不是 Huffman 编码。

例题 4.4.6:

设随机变量 X 的分布如下，求二进 Huffman 编码。

$$X \sim p(x) = \begin{pmatrix} a_1 & a_2 & a_3 & a_4 & a_5 \\ 0.4 & 0.2 & 0.20 & 0.1 & 0.1 \end{pmatrix}.$$

解:

由下面码树（图 4-10）中从根到叶子路径中的权值即构成 Huffman 码。其平均码长

$$\bar{L} = 1.4 + 2.2 + 3 \times 0.2 + 4 \times 0.1 + 4 \times 0.1 = 2.2 \quad \text{bits},$$

但二进熵为 $H(X) = 2.1219$bits，所以平均码长接近二进熵。

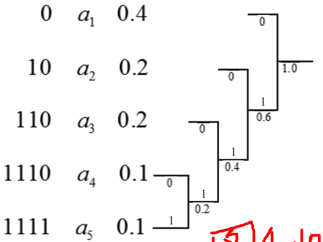


图4-10

续解：

但也可以如图 4-11 这样编码，其平均码长

$$\bar{L} = 2.4 + 2.2 + 2 \times 0.2 + 3 \times 0.1 + 3 \times 0.1 = 2.2 \quad \text{bits.}$$

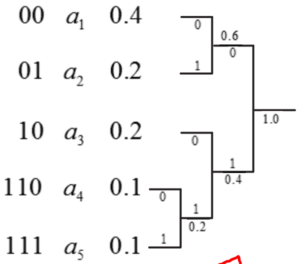


图4-11

两种编码比较：

这两个编码都是 Huffman 编码，平均码长也相同，对应的码树也都是完全树，但是它们还是有区别的。因为字符 X 的码长 $L(X)$ 可以看成离散无记忆信源的函数，平均码长正好是数学期望

$$\bar{L} = E\left[L(X)\right] = \sum_{i=1}^{N} p_i L(a_i) = \sum_{i=1}^{N} p_i l_i,$$

它当然也有方差

$$\sigma^2 = D\left[L(X)\right] = E\left[L^2(X)\right] - E^2\left[L(X)\right],$$

表示码字之间码长的偏差情况。

续比较：

对于本例题有

$$\sigma_1^2 = 0.4\times 1^2+0.2\times 2^2+0.2\times 3^2+0.1\times 4^2+0.1\times 4^2-2.2^2 = 1.36,$$

$$\sigma_2^2 = 0.4\times 2^2+0.2\times 2^2+0.2\times 2^2+0.1\times 3^2+0.1\times 3^2-2.2^2 = 0.16,$$

因此第二种 Huffman 码的码长方差较小。通常选用码长方差较小的 Huffman 码，因为它的码长分布较均匀，方便译码器设计。

D 进 Huffman 编码算法：

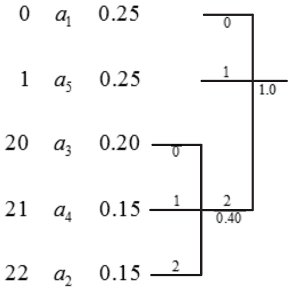
一般地也可以进行 D 进 Huffman 编码，其方法同二进 Huffman 编码的递归方法，在合并字符时每次要将 D 个概率最小的字符为一组。

(例题 4.4.7)

仍然用例题 4.4.5 中的信源的分布，进行三进 Huffman 编码。

解：

由下面码树（图 4-12）中从根到叶子路径中的权值即构成



Huffman 码。

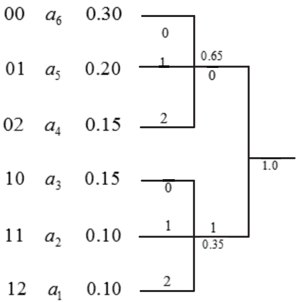
例题 4.4.8:

设随机变量 X 的分布如下，求三进 Huffman 编码。

$$X \sim p(x) = \begin{pmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 \\ 0.1 & 0.1 & 0.15 & 0.15 & 0.2 & 0.3 \end{pmatrix}.$$

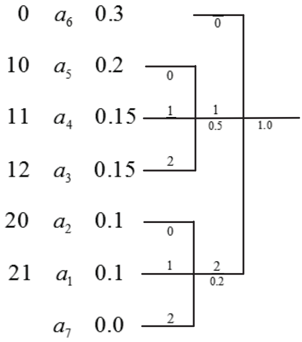
4.13

解：由下面码树（图 ??）中从根到叶子路径中的权值即构成码字。



续解：

它的平均码长为 2 个三进字符，但是码树不是完全树，因为根的分枝不是 3 个，因此它不是 Huffman 码。但是如果补充一个虚拟字符 a_7（它的概率为 0），再进行三进编码，则由下面码树（图 4-14）中从根到叶子路径中的权值即构成 Huffman 码。其



平均码长为 1.7 三进字符。

辅助字符（0 概率字符）：

由此可见添加一个辅助字符可以实现三进 Huffman 编码，使码树构成完全树。事先怎样知道补多少个字符？

设信源有 N 个消息要进行 D 进 Huffman 编码，因为每次要合并 D 个信源字符才能得到缩减信源，这时缩减信源消息数比原信源消息数减少了 $D-1$ 个；为了使码树成为完全树，若最后一次缩减之前已经进行了 k 次缩减，这时信源消息数个应当恰好剩 D 个，从而只要消息总数满足 $N=D+(D-1)k$，对信源进行 D 进 Huffman 编码就不需要添加任何虚拟字符！

命题 4.4.1:

设信源有 N 个消息要进行 D 进 Huffman 编码，要添加虚拟字符数有下面结论:

（1） *当 $N=(D-1)k+1$ 即 N 被 $D-1$ 除余 1 时，不需要添加任何虚拟字符。*

（2） *当 $N=(D-1)k$ 即 N 被 $D-1$ 恰好整除时，需要添加 1 个虚拟字符。*

（3） *当 $N=(D-1)k+M$ 并且 $M=2,\cdots,D-2$ 即 N 被 $D-1$ 除余数大于 1 时，需要添加 $D-M$ 虚拟字符。*

例题 4.4.9:

设随机变量 X 的分布如下，求四进 Huffman 编码。

$$X \sim p(x) = \begin{pmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 \\ 0.2 & 0.19 & 0.18 & 0.17 & 0.15 & 0.10 & 0.007 & 0.003 \end{pmatrix}.$$

解：它的 8 个字符，进行四进编码，要补充 2 个辅助字符，由码树图 4-15 中从根到叶子路径中的权值即构成 Huffman 码，辅助字符的码不要写。

图 4-15:

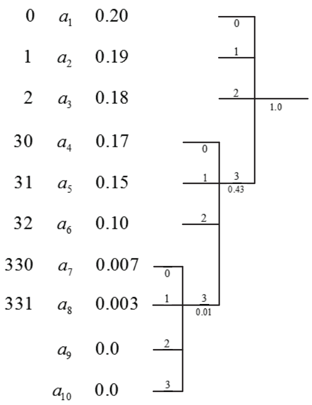


Figure: 图 4-15

解：

如果需要对离散无记忆信源的 n 长消息进行 Huffman 编码，则必须求出每个消息或弱典型序列的概率，将每个 n 长消息看成是一个字符，然后再利用 Huffman 编码。

二进 Fano 编码算法:

（1）求出所有 n 长消息的概率分布

$$r_i = p(\alpha_i), \alpha_i \in \mathcal{X}^n, i = 1, 2, \cdots, N^n,$$

并将所有 N^n 个 n 长消息按它的概率从大到小排序

$$\begin{array}{ccccccccc} \alpha_1 & & \alpha_2 & & \alpha_3 & & \cdots & & \alpha_{N^n} \\ r_1 & \geq & r_2 & \geq & r_3 & \geq & \cdots & \geq & r_{N^n} \end{array}.$$

续算法:

（2）将全体消息 $\alpha_1, \alpha_2, \cdots, \alpha_{N^n}$ 分成尽可能等概率的两部分，即选择一个 k 使

$$\left|\sum_{i=1}^{k} r_i - \sum_{i=k+1}^{N^n} r_i\right|,$$

尽量小，给第一部分指定码符 0 第二部分指定码符 1。
（3）再用 (2) 中方法对每部分进行分组，同时指定码符 0 与 1，重复进行下去，直到每部分中只剩一个消息为止，划分过程中出现的码符即构成码字。

例题 4.4.10：

设随机变量 X 的概率分布如下，求每个字符的二进 Fano 码。

$$X \sim p(x) = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ 0.20 & 0.15 & 0.1 & 0.3 & 0.15 & 0.1 \end{pmatrix}.$$

解：编码过程如图 4-16。于是可得 Fano 编码如下：

x_3 x_5

消息	x_4	x_1	x_2	x_5	x_3	x_6
码字	00	01	100	101	110	111

Fano 码的码树是完全树，平均码长为

$$\bar{L} = \frac{19}{8} = 2.375 \text{ bits} > H_2(X) = 2.28 \text{ bits}.$$

图 4-16：

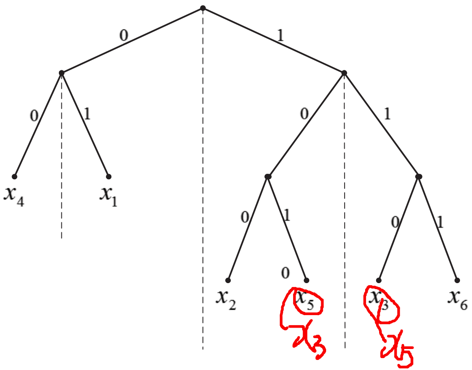


Figure: 图 4-16

引言：

Elias

前面讲的不论是哪种编码，均需要对概率进行排序，然后再进行编码，当消息长度较大时，要编码的消息非常多，此时不仅需要求出消息的概率而且要进行排序，这将非常费时，编码效率较低。能不能设计一种方法不进行概率排序就可以编码？Elias 编码方法就是这样的编码方法。

假设：

设离散无记忆信源分布为（4-2），其中概率处于无序状态，但是信源字符是事先进行编号的，即信源的字符是有序的，这主要是为了求每个字符的累积概率。记字符 $a_1, a_2, \cdots, a_N$ 的累积概率分别为

$$F_1 = 0, F_k = \sum_{i=1}^{k-1} p_i = p_1 + \cdots + p_{k-1}, k = 2, 3, \cdots, N+1. \quad (\cancel{3.1}) \; 4.17$$

这些累积概率将半开区间 $[0, 1)$ 分成 N 个小区间：$I_k = [F_k, F_{k+1}), k = 1, 2, \cdots, N$，如图 4-17。

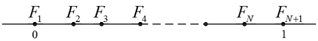


Figure: 图 4-17

编码区间：

这些小区间有如下特点：

（1） 各个小区间互不相交，它们的并集构成整个区间 $[0,1)$。

（2） 每个小区间可以与信源字符建立一一对应关系 $a_k \leftrightarrow I_k, k=1,2,\cdots,N$。

（3） 第 k 个小区间的长度就是第 k 个字符 a_k 的概率 p_k。

Elias 码就是利用累积概率决定的小区间与字符之间的一一对应关系，取每个小区间中的某个有限长度的 D 进小数作为相应字符的编码。如果能选择小数点后有 L 位的 D 进小数 $0.c$，使得小区间 $[0.c, 0.c+D^{-L})$ 被包含在某个小区间 I_k 中，则就可以取字符串 c 作为字符 a_k 的码字，通常选择小区间 I_k 的中点进行 D 进小数表示。

编码算法：

（1） 求累积概率 $F_1, F_2, \cdots, F_N$ 及小区间 I_k 中点

$$\tilde{F}_1 = \frac{p_1}{2}, \tilde{F}_k = F_k + \frac{p_k}{2}, k = 2, \cdots, N. \qquad (\cancel{3.2})$$

4.18

（2） 取码长

4.19

$$l_k = \left\lceil \log_D \frac{1}{p_k} \right\rceil + 1, k = 1, 2, \cdots, N, \qquad (\cancel{3.3})$$

它比仙农码码长多 1 位。

（3） 求 $\tilde{F}_k$ 的 D 进小数，然后取小数点后的 l_k 个字符作为字符 a_k 的码字 c_k。

性质：

定理4.4.13

上述算法生成的 Elias 码具有如下性质：

（1） *Elias 码是即时码。*

（2） *Elias 码的平均码长满足：*

4.20

$$H_D(X)+1 \le \bar{L} < H_D(X)+2. \tag{3.4}$$

证明：设 a_k 的 D 进码字为 c_k，则它对应的小数 $\tilde{c}_k = 0.c_k$ 是 $\tilde{F}_k$ 的 D 进小数的前 l_k 位，即 $\tilde{F}_k = 0.c_k\cdots$，于是就有

$$\tilde{c}_k \le \tilde{F}_k < \tilde{c}_k + D^{-l_k}. \tag{3.5}$$

4.21

续证明：

现在来证明半开小区间 $\tilde{I}_k = [\tilde{c}_k, \tilde{c}_k + D^{-l_k})$ 被完全包含在半开区间 $I_k = [F_k, F_{k+1})$ 中。由（4-19）可得

$$D^{-l_k} \leq \frac{p_k}{D} \leq \frac{p_k}{2},$$

再利用（4-12,4-18），可得

$$\tilde{c}_k + D^{-l_k} \leq \tilde{c}_k + \frac{p_k}{D} \leq \tilde{F}_k + \frac{p_k}{2} = F_{k+1},$$

$$\tilde{c}_k > \tilde{F}_k - D^{-l_k} \geq \tilde{F}_k - \frac{p_k}{D} \geq \tilde{F}_k - \frac{p_k}{2} = F_k,$$

于是半开小区间 $\tilde{I}_k$ 被完全包含在半开区间 $[F_k, F_{k+1})$ 中，但是以小数 $\tilde{c}_k$ 开头的所有小数全部在小区间 $\tilde{I}_k$ 内部，因此码字 c_k 不可能是另一个码字的前缀，从而 Elias 码是即时码。

续证明：

由码长的表达式（4-19）可以得到

$$\log_D \frac{1}{p_k} + 1 \le l_k < \log_D \frac{1}{p_k} + 2,$$

$$p_k \log_D \frac{1}{p_k} + p_k \le p_k l_k < p_k \log_D \frac{1}{p_k} + 2p_k,$$

$$\sum_{k=1}^{N} p_k \log_D \frac{1}{p_k} + \sum_{k=1}^{N} p_k \le \sum_{k=1}^{N} p_k l_k < \sum_{k=1}^{N} p_k \log_D \frac{1}{p_k} + \sum_{k=1}^{N} 2p_k,$$

$$H_D(X) + 1 \le \bar{L} < H_D(X) + 2.$$

这个定理说明：Elias 编码牺牲了码长来换取了不必对概率进行降序排列的要求。

例题 4.4.11：

已知离散无记忆信源

$$X \sim p(x) = \begin{pmatrix} a_1 & a_2 & a_3 & a_4 & a_5 \\ 0.25 & 0.5 & 0.125 & 0.0625 & 0.0625 \end{pmatrix},$$

试求每个字符的二进 Elias 编码。
解：信源的熵是 $H(X) = 1.875$ bits 平均码长 $\bar{L} = 2.875$ bits。
编码过程如下表所示。

续解：编码过程

字符	概率	累积概率	小区间	中点	二进表示	码长
a_1	0.25	0	[0,0.25)	0.125	0.001	3
a_2	0.5	0.25	[0.25,0.75)	0.5	0.1	2
a_3	0.125	0.75	[0.75,0.875)	0.8125	0.1101	4
a_4	0.0625	0.875	[0.875,0.9375)	0.90625	0.11101	5
a_5	0.0625	0.9375	[0.9375,1)	0.96875	0.11111	5

长信息编码思想：

消息

Elias 编码属于分组码。如果要对信源输出的长消息序列进行编码，就要对这个长消息序列进行分组，设分组长度为 n 个字符，全体 n 长消息构成字符空间 $\mathcal{X}^n$。根据 Elias 编码的基本思想，要先确定每个 n 长消息的累积概率，再象单个字符那样对每个 n 长消息进行编码，其中关键是怎样求 n 长消息的累积概率。

长信息字典顺序：

对字符空间 $\mathcal{X}=\{a_1,a_2,\cdots,a_N\}$，规定单个字符的顺序就是它们出现的先后顺序，写成

$$a_1<a_2<\cdots<a_N.$$

两个 n 长消息 $x^{(n)}=(x_1,x_2,\cdots,x_n),y^{(n)}=(y_1,y_2,\cdots,y_n)$ 的顺序就按它们第一个不相同的字符顺序来规定，即

$x^{(n)}<y^{(n)}\Leftrightarrow$ 有一个下标 i 使 $\quad x_1=y_1,\cdots,x_{i-1}=y_{i-1},x_i<y_i.$

这样所有 n 长消息就具有字典顺序，被编号在 $1\sim N^n$ 之间。比如 2 长消息的字典顺序为

$$a_1a_1,a_1a_2,\cdots,a_1a_N,a_2a_1,a_2a_2,\cdots,a_2a_N,\cdots,a_Na_1,a_Na_2,\cdots,a_Na_N.$$

长信息累积概率推导：

现在记 n 长消息 $x^{(n)}$ 的概率为 $p(x^{(n)})$, 则此消息的累积概率计算公式为

$$F(x^{(n)}) = \sum_{y^{(n)} < x^{(n)}} p(y^{(n)}).$$

如果 n 长消息 $x^{(n)}$ 再增加一个字符 x_{n+1} 变成 $n+1$ 长消息 $x^{(n+1)} = x^{(n)} x_{n+1}$，小于消息 $x^{(n+1)}$ 的 $n+1$ 长消息 $y^{(n+1)}$ 可以分成两组：一组以 n 长消息 $y^{(n)} : y^{(n)} < x^{(n)}$ 开头，末尾字符 y_{n+1} 可以是任意字符的消息；另一组是以 $x^{(n)}$ 开头，末尾字符为 $y_{n+1} : y_{n+1} < x_{n+1}$ 的消息。

续长信息累积概率推导：

从而 $n+1$ 长消息 $x^{(n+1)}$ 对应的概率及累积概率为

$$
\begin{aligned}
p(x^{(n+1)}) &= p(x^{(n)})p(x_{n+1}|x^{(n)}), \\
F(x^{(n+1)}) &= \sum_{y^{(n+1)}<x^{(n+1)}} p(y^{(n+1)}) \\
&= \sum_{y^{(n)}<x^{(n)}} \sum_{y_{n+1}} p(y^{(n)}, y_{n+1}) + \sum_{y_{n+1}<x_{n+1}} p(x^{(n)} y_{n+1}) \\
&= F(x^{(n)}) + p(x^{(n)}) \sum_{y_{n+1}<x_{n+1}} p(y_{n+1}|x^{(n)}).
\end{aligned}
$$

特别对于离散无记忆信源，条件概率中的条件可以去掉，上面的概率及累积概率计算会更简单些。

4.22

$$
\begin{aligned}
p(x^{(n+1)}) &= p(x^{(n)})p(x_{n+1}), && (\cancel{3.6}) \\
F(x^{(n+1)})) &= F(x^{(n)}) + p(x^{(n)})F(x_{n+1}), && (3.7)
\end{aligned}
$$

4.23

其中 $F(x_{n+1})$ 是单个字符的累积概率。

续长信息累积概率推导：

利用（3.6，3.7）可以构造如下计算累积概率的算法：

（1） 令消息的概率与累积概率的初值为

$$F(x^{(0)}) = 0, p(x^{(0)}) = 1,$$

其中 $x^{(0)}$ 是空消息。

（2） 设 $j = 1, 2, \cdots$， 则利用下面公式

$$\left.\begin{array}{l} x^{(j)} = x^{(j-1)}x_j, \\ p(x^{(j)}) = p(x^{(j-1)})p(x_j, \\ F(x^{(j)}) = F(x^{(j-1)}) + p(x^{(j-1)})F(x_j) \end{array}\right\} \quad (3.8)$$

4.24

反复迭代就可以求出任意长消息的累积概率。

构造编码区间:

现在构造左闭右开区间 $[0,1)$ 中的左闭右开小区间

$$I(x^{(j)}) = \left[F(x^{(j)}), F(x^{(j)}) + p(x^{(j)})\right), \quad j=1,2,3\cdots$$

则可以证明这些小区间有如下性质:

（1） 如果两个消息具有前缀关系 $x^{(n)} = x^{(m)}x_{m+1}\cdots x_n$，则 $I(x^{(m)}) \supseteq I(x^{(n)})$。因此对 $x^{(n)} = x_1x_2\cdots x_n$ 有

$$I(x_1) \supseteq I(x^{(2)}) \supseteq \cdots \supseteq I(x^{(n)}) \supseteq \cdots.$$

（2） 小区间 $I(x^{(n)})$ 的长度正好是消息 $x^{(n)}$ 的概率 $p(x^{(n)})$，消息长度越长对应概率就越小，区间长度也越小。

并集

（3） 所有 n 长消息对应的小区间 $I(x^{(n)})$ 互不相交，它们的并正好是区间 $[0,1)$。

求码字：

$x^{(n)}$

可以选择小区间 $I(x^{(n)})$ 的中点

$$\tilde{F}(x^{(n)}) = F(x^{(n)}) + \frac{p(x^{(n)})}{2},$$

并将它用 D 进小数表示，取小数点后长度为

$$l(x^{(n)}) = \left\lceil \log_D \frac{1}{p(x^{(n)})} \right\rceil + 1$$

的小数位作为消息 $x^{(n)}$ 的码字即可。

例题 4.4.12：

已知离散无记忆信源具有分布

$$X \sim p(x) = \begin{pmatrix} a_1 & a_2 & a_3 & a_4 \\ 0.5 & 0.3 & 0.15 & 0.05 \end{pmatrix},$$

试求消息 $a_2a_1a_1a_4a_3$ 的二元及三元 Elias 码字。
解：单个字符的累积概率为

$$F(a_1) = 0, F(a_2) = 0.5, F(a_3) = 0.8, F(a_4) = 0.95,$$

根据联合概率与累积概率的递推计算公式（4-24），可以求得下表：

续解：编码过程表

消息长度	消息	概率	累积概率	编码区间
j	$x^{(j)}$	$p(x^{(j)})$	$F(x^{(j)})$	$I(x^{(j)})$
1	a_2	0.3	0.5	[0.5,0.8)
2	a_2a_1	0.15	0.5	[0.5,0.65)
3	$a_2a_1a_1$	0.075	0.5	[0.5,0.575)
4	$a_2a_1a_1a_4$	0.00375	0.57125	[0.57125,0.575)
5	$a_2a_1a_1a_4a_3$	0.0005625	0.57425	[0.57425,0.5748125)

续解：

编码区间的中点为 $c = 0.57453125$，二进码长为 $l = 12$，化成二进小数为 $0.100100110001010\cdots$，因此二进 Elias 码为 $c = 100100110001$；三进码长为 $l = 8$，化成三进小数为 $0.12011121111\cdots$，因此三进 Elias 码为 $c = 12011121$。编码之所以较长是因为消息 $a_2a_1a_1a_4a_3$ 的概率太小 $p(a_2a_1a_1a_4a_3) = 0.0005625$。

求码字：

不讲，自学

算术码不属于分组码范围，是一种试图对信源输出的完整消息序列直接进行编码的编码方法。它是对 Elias 编码稍作改进后得到的编码方法。设离散无记忆信源的概率分布为（??）4.2，输出消息总长度为 n ，记为 $x^{(n)} = x_1x_2\ldots x_n$。算术编码方法首先要求出消息序列的编码区间，然后取这个编码区间中的某个数的 D 进制小数的一定位数作为码字。为描述编码与译码算法，引进如下记号：

（1） 信源输出的前 $j = 1, 2, \cdots, n$ 个字符构成消息序列，记为 $x^{(j)} = x_1x_2\cdots x_j$。

（2） 消息序列 $x^{(j)}$ 所对应的 Elias 编码区间记为 $I_j = [L_j, H_j)$，区间长度记为 $\Delta_j = H_j - L_j$。

（3） 单个信源字符 $a_i, i = 1, 2, \cdots, N$ 所对应的 Elias 编码区间记为 $[l_i, h_i)$。

编码区间上下界递推关系：

4.24

沿用 Elias 编码中联合概率与累积概率的计算方法（3.8），可得输出消息序列编码区间上下界的递推关系如下：

$$
\begin{aligned}
L_1 &= F(x_1), H_1 = F(x_1) + p(x_1), \Delta_1 = p(x_1), \\
&\quad \text{对} j = 2, 3, \cdots, n, \\
\Delta_j &= p(x^{(j)}) = p(x_1 x_2 \cdots x_j), \\
L_{j+1} &= L_j + \Delta_j F(x_{j+1}), \\
H_{j+1} &= L_j + \Delta_j \left[F(x_{j+1}) + p(x_{j+1})\right].
\end{aligned}
$$

编码算法：

（1）先求出每个信源字符的编码区间，字符 $a_i, i = 1, 2, \cdots, N$ 所对应的编码区间 $[l_i, h_i)$ 的上下界分别为

$$\begin{aligned} l_1 &= 0, \\ l_i &= p_1 + \cdots + p_{i-1}, i = 2, \cdots, N, \\ h_i &= l_i + p_i, i = 1, \cdots, N. \end{aligned}$$

（2）初始化消息的编码区间的上下界及长度：

$$j = 0, L_j = 0, H_j = 1, \Delta_j = 1.$$

续编码算法：

（3）作循环：对 $j=0,1,2,3,\cdots$ 计算 $j+1$ 长消息的编码区间的上下界及长度，设消息序列中第 $j+1$ 个字符 $x_{j+1}=a_i$ 则

$$\begin{aligned} L_{j+1} &= L_j+\Delta_j l_i, \\ H_{j+1} &= L_j+\Delta_j h_i, \\ \Delta_{j+1} &= H_{j+1}-L_{j+1}. \end{aligned}$$

（4）循环结束后输出信源消息的编码区间的上下界及区间长度 L_n, H_n, Δ_n，求编码区间 $[L_n, H_n)$ 中点的 D 进小数及码长

$$l=\left\lceil \log_D \frac{1}{\Delta_n} \right\rceil+1,$$

取小数点后面 l 位字符串 $c_1c_2\cdots c_l$ 作为这个信源消息序列的算术码。

译码算法：

对编好的算术码字 $c = c_1c_2\cdots c_l$，译码器只要知道信源字符的概率分布及信源输出消息的长度，就可以通过简单的算术运算进行译码。译码算法如下：

（1）先求出每个信源字符 $a_i, i = 1, 2, \cdots, N$ 所对应的编码区间，记为 $I_i = [l_i, h_i)$，计算方法如下：

$$\begin{aligned} l_1 &= 0, \\ l_i &= F(a_i) = p_1 + \cdots + p_{i-1}, i = 2, \cdots, N, \\ h_i &= l_i + p_i, i = 1, \cdots, N. \end{aligned}$$

（2）初始化 j 长消息的编码区间的上下界及长度

$$j = 0, L_j = 0, H_j = 1, \Delta_j = 1.$$

续译码算法：

（3）将码字转换成小数 $0.c = 0.c_1c_2\cdots c_l$。对 $j = 0, 1, 2, \cdots, n-1$，如果商

$$\frac{0.c - L_j}{\Delta_j}$$

在某个小区间 I_i 内部，就可以确定信源消息序列的第 $j+1$ 个字符必为 a_i! 因此译码方法是：若序号 $i(i = 1, 2, \cdots, N)$ 使

$$\frac{0.c - L_j}{\Delta_j} \in I_i,$$

则 $x_{j+1} = a_i$；然后计算

$$L_{j+1} = L_j + \Delta_j l_i; H_{j+1} = L_j + \Delta_j h_i; \Delta_{j+1} = H_{j+1} - L_{j+1}$$

（4）循环结束后即完成译码，输出信源消息 $x_1x_2\cdots x_n$。
要注意这些计算中要用相同的进位制，D 进码序列常常要转换成默认的十进制小数。

例题 4.4.13:

仍然用例题 4.4.12 中的信源，已知信源输出了一串消息 $a_3a_2a_4a_1a_1a_2a_3$，试求它的二进算术码并进行译码。
解：按照算术码的编码算法，将求编码区间的过程列表如下。

解：求码字过程表

消息长度 j	区间下界 L_j	区间上界 H_j	区间长度 Δ_j
0	0	1.0000000000	1.0000000000
1	0.8000000000	0.9500000000	0.1500000000
2	0.8750000000	0.9200000000	0.0450000000
3	0.9177500000	0.9200000000	0.0022500000
4	0.9177500000	0.9188750000	0.0011250000
5	0.9177500000	0.9183125000	0.0005625000
6	0.9180312500	0.9182000000	0.0001687500
7	0.9181662500	0.9181915625	0.0000253125

编码区间的中点为 0.91817890625000，二进码长为 $l = 17$，化成二进小数为
$0.11101011000011011100\cdots$，因此二进算术码为
$c = 11101011000011011.$

续解：编码过程图示

求编码区间的过程也可以用图 4-18 来表示。

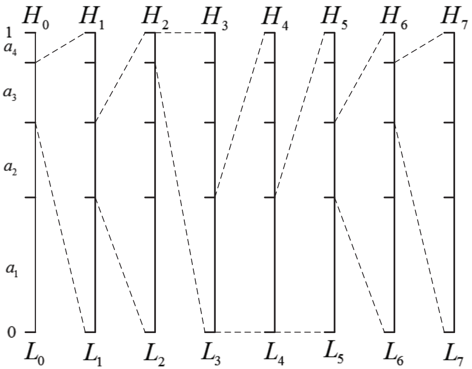


Figure: 图 4-18

续解：译码

在进行译码时，要先将码字序列转换成小数 $0.c$，只要

$$\frac{(0.c-L_j)}{\Delta_j} \in I_i,$$

则第 $j+1$ 个字符就是 a_i。译码过程如下表所示。最终译出的消息为 $a_3a_2a_4a_1a_1a_2a_3$，恰好就是信源的消息。

续解：译码过程表

译码顺序	区间左端	区间右端	区间长度	译出消
0	0.00000000000	1.00000000000	1.00000000000	
1	0.80000000000	0.95000000000	0.15000000000	a_3
2	0.87500000000	0.92000000000	0.04500000000	a_2
3	0.91775000000	0.92000000000	0.00225000000	a_4
4	0.91775000000	0.91887500000	0.00112500000	a_1
5	0.91775000000	0.91831250000	0.00056250000	a_1
6	0.91803125000	0.91820000000	0.00016875000	a_2
7	0.91816625000	0.91819156250	0.00002531250	a_3

二进信源的二进算术码算法：

（1） 设信源字符 0 的概率为 p_0，信源字符 1 的概率为 p_1。

（2） 初始化 j 长消息的编码区间的上下界及长度

$$j=0: L_j=0, H_j=1, \Delta_j=1.$$

（3） 作循环：对 $j=0,1,2,\cdots,n-1$ 计算 $lh=L_j+p_0\Delta_j$，如果 $x_{j+1}=0$，则 $L_{j+1}=L_j, H_{j+1}=lh$。否则 $L_{j+1}=lh, H_{j+1}=H_j$。再求区间长度：$\Delta_{j+1}=H_{j+1}-L_{j+1}$。进入下一步循环。

（4） 循环结束后输出信源消息的编码区间的上下界及区间长度 L_n, H_n, Δ_n，求编码区间 $[L_n, H_n)$ 中点的 2 进小数及码长

$$l=\left\lceil \log_2 \frac{1}{\Delta_n} \right\rceil + 1,$$

取小数点后面 l 位部分作为这个消息的算术码。

二进信源的二进算术码译码算法：

（1） 设信源字符 0 的概率为 p_0，信源字符 1 的概率为 p_1。

（2） 初始化 j 长消息的编码区间的上下界及长度

$$j=0, L_j=0, H_j=1, \Delta_j=1.$$

（3） 现在译码，作循环：对 $j=0,1,2,\cdots,n-1$ 判断

$$\frac{0.c-L_j}{\Delta_j}>p_0.$$

如果成立，则输出信源字符 $x_{j+1}=1$，否则输出信源字符 $x_{j+1}=0$。然后计算 $lh=L_j+p_0\Delta_j$，如果 $x_{j+1}=0$ 则 $L_{j+1}=L_j, H_{j+1}=lh$。否则 $L_{j+1}=lh, H_{j+1}=H_j$。再求区间长度：$\Delta_{j+1}=H_{j+1}-L_{j+1}$。进入下一步循环。

（4） 循环结束后即完成译码，输出信源消息 $x_1x_2\cdots x_n$。

例题 4.4.14：

已知二进信源只有两个字符“0，1”，它们概率为 $p_0 = 0.25, p_1 = 0.75$，求二元消息序列 $x = 11111100$ 的二进算术码并译码。

解：按照算术码的编码算法，可以将求编码区间的过程列表如下。编码区间的中点为 0.82758331298828，二进码长为 $l = 8$，化成二进小数为

$0.11010011\cdots$，因此二进算术码为 $c = 11010011$。

续解：编码过程表

消息长度	区间下界	区间上界	区间长度
j	L_j	H_j	Δ_j
0	0.0	1.00000000000000	1.0000000000000
1	0.25000000000000	1.00000000000000	0.7500000000000
2	0.43750000000000	1.00000000000000	0.5625000000000
3	0.57812500000000	1.00000000000000	0.4218750000000
4	0.68359375000000	1.00000000000000	0.3164062500000
5	0.76269531250000	1.00000000000000	0.2373046875000
6	0.82202148437500	1.00000000000000	0.1779785156250
7	0.82202148437500	0.86651611328125	0.0444946289062
8	0.82202148437500	0.83314514160156	0.0111236572265

续解：译码过程表

序号 j	区间左端	区间右端	区间长度
1	0.250000000000	1.000000000000000	0.750000000000000
2	0.437500000000	1.000000000000000	0.562500000000000
3	0.578125000000	1.000000000000000	0.421875000000000
4	0.683593750000	1.000000000000000	0.316406250000000
5	0.762695312500	1.000000000000000	0.237304687500000
6	0.822021484375	1.000000000000000	0.177978515625000
7	0.822021484375	0.866516113281250	0.044494628906250
8	0.822021484375	0.833145141601563	0.011123657226563

最终译出的消息为 1111100，恰好就是信源的消息。

练习：

模仿例题 4.4.13 绘制例 4.4.14 中算术码的图示。