

显然, 这种子矩阵共有 $n+1-p$ 个, 它们是 $\hat{B}(1, p+1), \dots, \hat{B}(n+1-p, n+1)$.

如 3.4.3 节所述, 矩阵 B 的有效秩为 p 意味着未知参数向量 x 中只有 p 个待定参数是独立的. 不妨令这些参数是 x 的前 p 个参数, 它们连同 1 一起构成 $(p+1) \times 1$ 维参数向量 $a = [1, x_1, \dots, x_p]^T$. 于是, 原方程组 (3.4.45b) 的求解就变成了下列 $n+1-p$ 个方程组的求解:

$$\hat{B}(k, p+k)a = 0, \quad k = 1, \dots, n+1-p \quad (3.4.51a)$$

或等价于下列方程的求解:

$$\begin{bmatrix} \hat{B}(1, p+1) \\ \hat{B}(2, p+2) \\ \vdots \\ \hat{B}(n+1-p, n+1) \end{bmatrix} a = 0 \quad (3.4.51b)$$

不难证明

$$\hat{B}(k, p+k) = \sum_{j=1}^p \sigma_{uj} u_j (v_j^k)^H \quad (3.4.52)$$

式中 u_j 是酉矩阵 U 的第 j 列; v_j^k 是酉矩阵 V 第 j 列的一个加窗段, 定义为

$$v_j^k = [v(k, j), v(k+1, j), \dots, v(k+p, j)]^T \quad (3.4.53)$$

其中 $v(k, j)$ 是矩阵 V 的第 k 行、第 j 列的元素.

根据最小二乘原理, 求方程组 (3.4.51) 的最小二乘解等价于使代价函数

$$\begin{aligned} f(a) &= [\hat{B}(1, p+1)a]^H \hat{B}(1, p+1)a + \dots + \\ &\quad [\hat{B}(n+1-p, n+1)a]^H \hat{B}(n+1-p, n+1)a \\ &= a^H \left[\sum_{i=1}^{n+1-p} \hat{B}(i, p+i) \hat{B}(i, p+i)^H \right] a \end{aligned} \quad (3.4.54)$$

最小化.

定义 $(p+1) \times (p+1)$ 维矩阵

$$S^{(p)} = \sum_{i=1}^{n+1-p} \hat{B}(i, p+i) \hat{B}(i, p+i)^H \quad (3.4.55)$$

将式 (3.4.52) 代入上式, 可以求得

$$\mathbf{S}^{(p)} = \sum_{j=1}^p \sum_{i=1}^{n-1-p} \sigma_{jj}^2 \mathbf{v}_j^i (\mathbf{v}_j^i)^H \quad (3.4.56)$$

另一方面, 将式 (3.4.55) 代入式 (3.4.54), 又可将代价函数简写作

$$f(\mathbf{a}) = \mathbf{a}^H \mathbf{S}^{(p)} \mathbf{a} \quad (3.4.57)$$

$f(\mathbf{a})$ 的最小化就是求解 $\frac{\partial f(\mathbf{a})}{\partial \mathbf{a}} = 0$, 其结果为

$$\mathbf{S}^{(p)} \mathbf{a} = \alpha \mathbf{e} \quad (3.4.58)$$

式中 $\mathbf{e} = [1, 0, \dots, 0]^T$, 归一化常数 α 的选择应使参数向量 $\mathbf{a}$ 的第一个元素为 1.

方程式 (3.4.58) 的求解是容易的. 若令 $\mathbf{S}^{-(p)}$ 是 $\mathbf{S}^{(p)}$ 的逆矩阵, 则解仅取决于逆矩阵 $\mathbf{S}^{-(p)}$ 的第一列. 易知, 待求的参数由

$$\hat{x}_i = \mathbf{S}^{-(p)}(i+1, 1) / \mathbf{S}^{-(p)}(1, 1), \quad i = 1, \dots, p \quad (3.4.59)$$

给出.

归纳起来, 求总体最小二乘解的算法由下列步骤组成.

算法 3.4.1 (SVD-TLS 算法)

步骤 1 计算增广矩阵 B 的 SVD, 并存储奇异值和矩阵 V ;

步骤 2 确定增广矩阵 B 的有效秩 p ;

步骤 3 利用式 (3.4.56) 和式 (3.4.53) 计算矩阵 $\mathbf{S}^{(p)}$;

步骤 4 求 $\mathbf{S}^{(p)}$ 的逆矩阵 $\mathbf{S}^{-(p)}$, 并由式 (3.4.59) 计算未知参数的总体最小二乘估计.

将式 (3.4.43) 定义的自相关矩阵 $\mathbf{R}_e$ 视为上述算法中的增广矩阵 B , 即可确定出 ARMA 模型的 AR 阶数 p , 并求出 p 个 AR 参数的总体最小二乘解.

一旦 AR 阶数 p 和 AR 参数 $a_1, \dots, a_p$ 被估计出来后, 即可利用 Cadzow 谱估计子或 Kaveh 谱估计子得到 ARMA 过程 $\{x(n)\}$ 的功率谱密度.

3.5 ARMA 模型辨识

在一些应用中, 不仅希望得到 AR 阶数和 AR 参数, 而且还希望获得 MA 阶数和 MA 参数的估计, 最终完成对整个 ARMA 模型的辨识.

3.5.1 MA 阶数确定

对于一个 ARMA 模型而言, 其 MA 阶数的确定问题从理论上讲很简单, 但实际算法的有效性却并不像想象的那样简单。

修正 Yule-Walker 方程式 (3.4.19) 对所有 $l > q$ 均成立。该式还表明,

$$R_x(l) + \sum_{i=1}^p a_i R_x(l-i) \neq 0, \quad l = q \quad (3.5.1)$$

否则, 这将意味着式 (3.4.19) 对所有 $l > q-1$ 均成立, 即 ARMA 过程的阶数是 $q-1$, 这与 ARMA(p, q) 模型相矛盾。事实上, 式 (3.5.1) 对某些 $l < q$ 可能成立也可能不成立, 但对 $l = q$ 必定成立。这说明, MA 阶数是使

$$R_x(l) + \sum_{i=1}^p a_i R_x(l-i) \neq 0, \quad l \leq q \quad (3.5.2)$$

成立的最大整数 l 。确定 MA 阶数的这一原理是 Chow^[45] 于 1972 年提出的。问题是: 在数据比较短的情况下, 样本相关函数 $\hat{R}_x(l)$ 具有比较大的估计误差和方差, 因此, 式 (3.4.61) 的检验方法将缺乏数值稳定性。更好的方法是线性代数法, 它是 1993 年由本书作者等人提出的^[215]。这一方法的理论基础是: MA 阶数 q 的信息包含在一 Hankel 矩阵里。

命题 3.5.1 令 R_1 是一个 $(p+1) \times (p+1)$ 维 Hankel 矩阵, 即

$$R_1 = \begin{bmatrix} R_x(q) & R_x(q-1) & \cdots & R_x(q-p) \\ R_x(q+1) & R_x(q) & \cdots & R_x(q+1-p) \\ \vdots & \vdots & \ddots & \vdots \\ R_x(q+p) & R_x(q+p-1) & \cdots & R_x(q) \end{bmatrix} \quad (3.5.3)$$

若 $a_p \neq 0$, 则 $\text{rank}(R_1) = p+1$ 。

证明 如果矩阵 R_1 第 $p+1$ 列与其他 p 列线性相关, 那么子列向量

$$r_{p+1} = [R_x(q-p+1), R_x(q-p+2), \cdots, R_x(q)]^T$$

必定是下述 p 个子列向量的线性组合:

$$\begin{aligned} r_p &= [R_x(q-p+2), R_x(q-p+3), \cdots, R_x(q+1)]^T \\ &\vdots \\ r_1 &= [R_x(q+1), R_x(q+2), \cdots, R_x(q+p)]^T \end{aligned}$$

由于 $a_p \neq 0$, 所以修正 Yule-Walker 方程的表示是惟一的, 这意味着上述线性组合也是惟一的, 可表示为

$$r_{p+1} = - \sum_{i=1}^p a_i r_{p+1-i} / a_p \quad (3.5.4)$$

然而, 当我们用 $-a_{p-k}/a_p (k=1, \dots, p)$ 去乘第 $p+1-k$ 列, 并将其求和向量的第一个元素同第 $p+1$ 列的第一个元素相比较时, 便会得到

$$R_x(q-p) \neq - \sum_{k=1}^p a_{p-k} R_x(q-p+k) / a_p$$

这是式 (3.5.2) 当 $l=q$ 时的等价结果。这就是说, 矩阵 R_1 的第 $p+1$ 列不可能是其他 p 列的线性组合。此外, 在定理 3.4.1 中, 已证明矩阵 R_1 最左边的 p 列是线性独立的。这两个性质告诉我们, 矩阵 R_1 的 $(p+1)$ 列是线性独立的。再由 R_1 的对称性可知, R_1 的所有 $(p+1)$ 行也是线性独立的, 从而 $\text{rank}(R_1) = p+1$ 。 ■

进一步地, 我们来考虑具有扩展阶 q_e 的矩阵 R_{1e} 的秩。

命题 3.5.2 假定 p 已经确定, 并令 R_{1e} 是一个 $(p+1) \times (p+1)$ 维矩阵, 定义为

$$R_{1e} \stackrel{\text{def}}{=} \begin{bmatrix} R_x(q_e) & R_x(q_e-1) & \cdots & R_x(q_e-p) \\ R_x(q_e+1) & R_x(q_e) & \cdots & R_x(q_e-p+1) \\ \vdots & \vdots & \ddots & \vdots \\ R_x(q_e+p) & R_x(q_e+p-1) & \cdots & R_x(q_e) \end{bmatrix} \quad (3.5.5)$$

则当 $q_e > q$ 时 $\text{rank}(R_{1e}) = p$, 而且仅当 $q = p$ 时才有 $\text{rank}(R_{1e}) = p+1$ 。

证明 先考虑 $q_e > q$ 的情况, 并记 $q_e = \bar{q}_e + 1$, 其中 $\bar{q}_e > q$ 。此时, 矩阵 R_{1e} 可写作

$$R_{1e} = \begin{bmatrix} R_x(\bar{q}_e+1) & R_x(\bar{q}_e) & \cdots & R_x(\bar{q}_e+1-p) \\ R_x(\bar{q}_e+2) & R_x(\bar{q}_e+1) & \cdots & R_x(\bar{q}_e+2-p) \\ \vdots & \vdots & \ddots & \vdots \\ R_x(\bar{q}_e+p+1) & R_x(\bar{q}_e+p) & \cdots & R_x(\bar{q}_e+1) \end{bmatrix}$$

即是说, 矩阵 R_{1e} 与在式 (3.4.25a) 中取 $p_e = p$ 和 $M = p$ 时的矩阵 R_e 完全相同, 并且命题 3.4.1 的条件满足。由命题 3.4.1 知 $\text{rank}(R_{1e}) = \text{rank}(R_e) = p$ 。再考虑 $q_e = q$ 的情况。此时, 矩阵 R_{1e} 与式 (3.5.3) 定义的矩阵 R_1 完全相同。由命题 3.5.1, 立即有 $\text{rank}(R_{1e}) = \text{rank}(R_1) = p+1$ 。 ■

命题 3.5.2 表明, 真实的 MA 阶数 q 隐含在矩阵 R_{1e} 内。理论上, 阶数 q 可以这样来确定: 从 $Q = q_e > q$ 开始, 并依次取 $Q \leftarrow Q-1$, 使用 SVD 确定 R_{1e} 的秩; 当秩从 p 变为 $p+1$ 的第一个转折发生在 $Q = q$ 的时候。然而, 在实际应

用中, 由于使用样本自相关函数, 所以阶数从 p 跳变为 $p+1$ 的转折点往往不是很明显。为了发展一种 MA 定阶的实际算法, 可以使用“超定的”矩阵 R_{2e} , 其元素 $R_{2e}[i, j] = \hat{R}_x(q_e + i - j), i = 1, \dots, M, j = 1, \dots, p_e + 1, M \gg p_e$ 。显然,

$$\text{rank}(R_{2e}) = \text{rank}(R_{1e}) \quad (3.5.6)$$

因为 R_{2e} 包含了整个 R_{1e} , 而任意第 $k(k \geq p+2)$ 列 (或行) 与其左边 p 列 (或上面 p 行) 线性相关。这并不会改变整个矩阵的秩。

使用 SVD 确定 ARMA(p, q) 模型的 MA 阶数的算法如下 [215]。

算法 3.5.1 (MA 阶数确定算法)

步骤 1 用 AR 定阶与参数估计的 SVD-TLS 算法中的步骤 1 和 2 确定出 AR 阶数, 并取 $Q = q_e > q$;

步骤 2 令 $Q \leftarrow Q - 1$, 并构造样本自相关矩阵 R_{2e} , 计算其 SVD;

步骤 3 如果第 $(p+1)$ 个奇异值与上次计算结果相比, 有一个明显的转折, 则选择 $q = Q$; 否则返回步骤 2, 并继续以上步骤, 直到 q 被选出为止。

注释 1 由于 $p \geq q$, 所以选择 Q 初始值的最简单有效的方法是取 $Q = q_e = p+1$, 这样可以尽快找到 q 。

注释 2 上述 MA 定阶方法仅与 AR 阶数有关, 而与 AR 参数无关。换句话说, AR 阶数和 MA 阶数的确定可以在参数估计之前分别进行。

注释 3 算法的关键是在步骤 3 中决定第 $(p+1)$ 个奇异值的转折点。作为一个转折点的检验法则, 可以考虑使用比值

$$\alpha = \frac{\sigma_{p+1,p+1}^{(Q)} - \sigma_{p+1,p+1}^{(Q+1)}}{\sigma_{p+1,p+1}^{(Q+1)}} \quad (3.5.7)$$

式中 $\sigma_{p+1,p+1}^{(Q)}$ 是 R_{2e} 对应于 Q 值时的第 $(p+1)$ 个奇异值。如果对于某个 Q 值, 第 $p+1$ 个奇异值的相对变化率大于某个给定的阈值, 则接受此 Q 值为转折点。

例 3.5.1 一时间序列由

$$x(n) = \sqrt{20} \cos(2\pi 0.2n) + \sqrt{2} \cos(2\pi 0.213n) + v(n)$$

产生, 其中 $v(n)$ 是有关均值为零、方差为 1 的高斯白噪声。每个余弦波的信噪比 (SNR) 定义为该余弦波的功率与噪声平均功率即方差之比, 故频率为 0.2 和 0.213 的余弦波分别具有 10dB 和 0dB 的信噪比。共进行 10 次独立实验, 每从运行的数据长度取 300。在每次运行中, SVD 方法均给出 $p=4$ 的 AR 阶数估计结果, 而采用上述 MA 定阶的 SVD 方法也都给出 $q=4$ 的 MA 阶数估计, 其中计算出来的最小比值 $\alpha = 38.94\%$ 。以下是对应于 $\alpha = 38.94\%$ 时的各奇异值。

R_r 和 $Q=5$ 时 R_{2e} 的奇异值为

$$\sigma_{11} = 102.942, \quad \sigma_{22} = 102.349, \quad \sigma_{33} = 2.622, \quad \sigma_{44} = 2.508$$

$$\sigma_{55} = 0.588, \quad \sigma_{66} = 0.517, \quad \cdots, \quad \sigma_{15,15} = 0.216$$

易看出, 前4个奇异值占支配地位, 故给出 $p=4$ 的 AR 阶数估计.

$Q=4$ 时 R_{2e} 的奇异值:

$$\sigma_{11} = 103.918, \quad \sigma_{22} = 102.736, \quad \sigma_{33} = 2.621, \quad \sigma_{44} = 2.510$$

$$\sigma_{55} = 0.817, \quad \sigma_{66} = 0.575, \quad \cdots, \quad \sigma_{15,15} = 0.142$$

我们看到 σ_{55} 在 $Q=4$ 时有一个比较明显的转折, 即

$$\alpha = \frac{0.817 - 0.588}{0.588} = 39.94\%$$

故选择 $q=4$ 作为 MA 阶数估计结果. ■

3.5.2 MA 参数估计

在推导 Karch 谱估计子时, 曾得到了式 (3.4.10). 比较该式左右两边相同幂次项的系数, 即可得到一组重要的方程

$$\left. \begin{aligned} \sigma^2(b_0^2 + b_1^2 + \cdots + b_q^2) &= c_0 \\ \sigma^2(b_0 b_1 + \cdots + b_{q-1} b_q) &= c_1 \\ &\vdots \\ \sigma^2 b_0 b_q &= c_q \end{aligned} \right\} \quad (3.5.8)$$

观察上述非线性方程组, 可以看出共有 $q+2$ 个未知参数 $b_0, b_1, \cdots, b_q$ 和 σ^2 , 但方程却只有 $q+1$ 个. 为了保证解的惟一性, 通常约定 $\sigma^2=1$ 或 $b_0=1$. 事实上, 这两种约定是相包容的, 因为在 $\sigma^2=1$ 的约定下, 只需将 MA 参数归一化为 $b_0=1$, 仍然可以得到 $\sigma^2=b_0^2$. 为方便计, 这里约定 $\sigma^2=1$.

下面介绍求解非线性方程组 (3.5.8) 的 Newton-Raphson 算法 [24, pp.498~500].

首先定义拟合误差函数

$$f_k = \sum_{i=0}^q b_i b_{i+k} - c_k, \quad k=0, 1, \cdots, q \quad (3.5.9)$$

及 $(q+1) \times 1$ 维向量

$$\mathbf{b} = [b_0, b_1, \dots, b_q]^T \quad (3.5.10a)$$

$$\mathbf{f} = [f_0, f_1, \dots, f_q]^T \quad (3.5.10b)$$

再定义 $(q+1) \times (q+1)$ 维矩阵

$$\mathbf{F} = \frac{\partial \mathbf{f}}{\partial \mathbf{b}^T} = \begin{bmatrix} \frac{\partial f_0}{\partial b_0} & \frac{\partial f_0}{\partial b_1} & \dots & \frac{\partial f_0}{\partial b_q} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_q}{\partial b_0} & \frac{\partial f_q}{\partial b_1} & \dots & \frac{\partial f_q}{\partial b_q} \end{bmatrix} \quad (3.5.11)$$

由式 (3.5.9) 求偏导 $\partial f_k / \partial b_j$, 再代入式 (3.5.11) 中, 即可得到

$$\mathbf{F} = \begin{bmatrix} b_0 & b_1 & \dots & b_q \\ b_1 & \dots & b_q & \\ \vdots & \ddots & & \\ b_q & & & 0 \end{bmatrix} + \begin{bmatrix} b_0 & b_1 & \dots & b_q \\ & b_0 & \dots & b_{q-1} \\ & & \ddots & \vdots \\ 0 & & & b_0 \end{bmatrix} \quad (3.5.12)$$

根据 Newton-Raphson 方法的原则, 若在第 i 次迭代中求得的 MA 参数向量为 $\mathbf{b}^{(i)}$, 则第 $(i+1)$ 次迭代的 MA 参数向量估计值由下式给出:

$$\mathbf{b}^{(i+1)} = \mathbf{b}^{(i)} - \mathbf{F}^{-(i)} \mathbf{f}^{(i)} \quad (3.5.13)$$

式中 $\mathbf{F}^{-(i)}$ 是矩阵 $\mathbf{F}$ 在第 i 次迭代中的矩阵 $\mathbf{F}^{(i)}$ 的逆矩阵。

归纳起来, 估计 MA 参数的 Newton-Raphson 算法由下列步骤组成。

算法 3.5.2 (Newton-Raphson 算法)

初始化 利用式 (3.4.12) 计算 MA 谱系数 $c_k, k = 0, 1, \dots, q$, 并令初始值 $b_0^{(0)} = \sqrt{c_0}$ 和 $b_i^{(0)} = 0, i = 1, \dots, q$ 。

步骤 1 由式 (3.5.9) 计算拟合误差函数 $f_k^{(i)}, k = 0, 1, \dots, q$, 并用式 (3.5.12) 计算 $\mathbf{F}^{(i)}$;

步骤 2 利用式 (3.5.13) 更新 MA 参数估计向量 $\mathbf{b}^{(i+1)}$;

步骤 3 检验 MA 参数估计向量是否收敛。若收敛, 则停止迭代, 输出 MA 参数估计结果; 否则, 令 $i \leftarrow i+1$, 并返回步骤 1, 重复以上步骤, 直至 MA 参数估计收敛。

作为终止迭代的一个法则, 可采用比较各参数前、后二次迭代估计值的绝对误差。但是, 这一方法对绝对值较大的 MA 参数比较合适, 却不适合绝对值比较小的

MA 参数。更好的方法是采用相对误差衡量某个参数在迭代过程中是否收敛^[211], 比如

$$\left| \frac{b_k^{(i+1)} - b_k^{(i)}}{b_k^{(i-1)}} \right| \leq \alpha \quad (3.5.14)$$

表示参数 b_k 收敛。式中, 阈值 α 可选择 0.05 等。

当存在加性 AR 有色噪声时, ARMA 功率谱估计需要使用广义最小二乘算法, 利用自举方法从 AR 有色噪声下的修正 Yule-Walker 方程求 ARMA 模型的 AR 参数。对这一算法感兴趣的读者可参考文献 [212]。

3.6 最大熵谱估计

ARMA 谱估计的理论基础是随机过程的建模。本节从信息论的观点介绍另一种功率谱估计方法——最大熵方法。这一方法是 Burg 于 1967 年提出的^[29]。由于最大熵方法的一系列优点, 它成了现代谱估计中的一个重要分支。有意思的是, 最大熵谱估计在不同条件下与 AR 谱估计、ARMA 谱估计之间存在着等价关系。

3.6.1 Burg 最大熵谱估计

在信息论中, 常常对在观测到一个以概率 p_k 发生的事件 $X = x_k$ 后能够得到多少信息量感兴趣。这一信息量用符号 $I(x_k)$ 表示, 定义为

$$I(x_k) \stackrel{\text{def}}{=} I(X = x_k) = \log \left(\frac{1}{p_k} \right) = -\log p_k \quad (3.6.1)$$

式中, 对数的底可任意选取。当使用自然对数时, 信息量的单位为 nat(奈特); 而当使用以 2 为底的对数时, 其单位为 bit(比特)。在后面的叙述中, 均使用以 2 为底的对数, 除非另有说明。无论采用何种对数, 由定义式 (3.6.1) 都容易证明信息量具有以下性质。

性质 1 肯定发生的事件不含任何信息, 即

$$I(x_k) = 0, \quad \forall p_k = 1 \quad (3.6.2)$$

性质 2 信息量是非负的, 即

$$I(x_k) \geq 0, \quad 0 \leq p_k \leq 1 \quad (3.6.3)$$

这一性质称为信息的非负值性。它表明: 一个随机事件的发生要么带来信息, 要么不带来任何信息, 但决不会造成信息的丢失或损失。

性质 3 概率越小的事件发生时, 我们从中得到的信息越多, 即

$$I(x_k) > I(x_i), \quad \text{若 } p_k < p_i \quad (3.6.4)$$

考察离散随机变量 X , 其取值的字符集为 $\mathcal{X}$. 令随机变量 X 取值 x_k 的概率为 $p_k = \Pr\{X = x_k\}, x_k \in \mathcal{X}$.

定义 3.6.1 信息量 $I(x)$ 在字符集合 $\mathcal{X}$ 内的平均值称为离散随机变量 X 的熵, 记作 $H(X)$, 定义为

$$H(X) \stackrel{\text{def}}{=} E\{I(x)\} = \sum_{x_k \in \mathcal{X}} p_k I(x_k) = - \sum_{x_k \in \mathcal{X}} p_k \log p_k \quad (3.6.5)$$

我们约定 $0 \log 0 = 0$, 这很容易从极限 $\lim_{x \rightarrow 0} x \log x = 0$ 得到证明。这表明, 在熵的定义式中增加一个零概率项, 不会对熵产生任何影响。

熵是随机变量 X 的分布的函数, 它与随机变量 X 的实际取值无关, 而只与 X 取值的概率有关。

若字符集合 $\mathcal{X}$ 由 $2K + 1$ 个字符组成, 则熵可表示为

$$H(x) = \sum_{k=-K}^K p_k I(x_k) = - \sum_{k=-K}^K p_k \log p_k \quad (3.6.6)$$

它是有界函数, 即

$$0 \leq H(x) \leq \log(2K + 1) \quad (3.6.7)$$

下面是熵的下界和上界的性质:

(1) $H(x) = 0$, 当且仅当对某个 $X = x_k$ 有 $p_k = 1$, 从而 X 取集合 $\mathcal{X}$ 中其他数值的概率全部为零; 换句话说, 熵的下界 0 对应为没有任何不确定性;

(2) $H(x) = \log(2K + 1)$, 当且仅当对所有 k 恒有 $p_k = 1/(2K + 1)$, 即所有离散取值是等概率的。因此, 熵的上界对应为最大的不确定性。

性质 (2) 的证明可参考文献 [81]。

例 3.6.1 令

$$X = \begin{cases} 1, & \text{以概率 } p \\ 0, & \text{以概率 } 1 - p \end{cases} \quad (3.6.8)$$

由式 (3.6.5) 可计算出 X 的熵为

$$H(X) = -p \log p - (1 - p) \log(1 - p) \quad (3.6.9)$$

特别地, 若 $p = 1/2$, 则 $H(X) = 1 \text{ bit}$. ■

例 3.6.2 令

$$X = \begin{cases} a, & \text{以概率 } 1/2 \\ b, & \text{以概率 } 1/8 \\ c, & \text{以概率 } 1/4 \\ d, & \text{以概率 } 1/8 \end{cases} \quad (3.6.10)$$

则 X 的熵为

$$H(X) = -\frac{1}{2} \log \frac{1}{2} - \frac{1}{8} \log \frac{1}{8} - \frac{1}{4} \log \frac{1}{4} - \frac{1}{8} \log \frac{1}{8} = \frac{7}{4} \text{ bit} \quad (3.6.11)$$

将定义 3.6.1 推广到连续随机变量, 则有以下定义.

定义 3.6.2 (连续随机变量的熵) 设连续随机变量 x 的分布密度函数为 $p(x)$, 则其熵定义为

$$H(x) \stackrel{\text{def}}{=} - \int_{-\infty}^{\infty} p(x) \ln p(x) dx \quad (3.6.12a)$$

$$= -E\{\ln p(x)\} \quad (3.6.12b)$$

1967 年, Burg^[29] 仿照连续随机变量的熵的定义, 将

$$H[P(\omega)] = \frac{1}{2\pi} \int_{-\pi}^{\pi} \ln P(\omega) d\omega \quad (3.6.13)$$

称为功率谱 $P(\omega)$ 的熵 (简称谱熵), 并提出利用给定的 $2p+1$ 个样本自相关函数 $\hat{R}_x(k), k=0, \pm 1, \dots, \pm p$ 估计功率谱时, 应该使谱熵为最大. 这就是著名的 Burg 最大熵谱估计方法. 当然, 所估计的功率谱的 Fourier 反变换还应该能够还原出原来的 $2p+1$ 个样本自相关函数 $\hat{R}_x(k), k=0, \pm 1, \dots, \pm p$. 具体说来, Burg 最大熵谱估计可以叙述为, 求功率谱密度 $P(\omega)$, 使 $P(\omega)$ 在约束条件

$$\hat{R}_x(m) = \frac{1}{2\pi} \int_{-\pi}^{\pi} P(\omega) e^{j\omega m} d\omega, \quad m=0, \pm 1, \dots, \pm p \quad (3.6.14)$$

之下, 能够使谱熵 $H[P(\omega)]$ 为最大. 这一具有约束条件的优化问题很容易用 Lagrange 乘子法求解.

构造目标函数

$$J[P(\omega)] = \frac{1}{2\pi} \int_{-\pi}^{\pi} \ln P(\omega) d\omega + \sum_{k=-p}^p \lambda_k \left[\hat{R}_x(k) - \frac{1}{2\pi} \int_{-\pi}^{\pi} P(\omega) e^{j\omega k} d\omega \right] \quad (3.6.15)$$