QTA2025 暑期求职笔试训练营-week2

欧岱松

2025 年 7 月 30 日

笔试题目

1. 随机数

小 U 有一个质地不均匀的硬币, 已知这枚硬币正面朝上的概率 p 不是 0 或 0.5, 但不知道确切的概率, 可以只用这枚硬币帮小 U 生成 100 个 0-15 之间均匀分布的随机整数吗? 小 U 是个追求严谨的小朋友, 如果可以, 需要聪明的你证明这种策略下生成的随机数的分布严格服从上述要求; 如果不可以, 也请给出证明。

这个问题的核心在于如何利用有篇的随机源来构建概率均等的随机数，我们考虑如下策略，将硬币连续抛掷两次：

- 如果结果是正-反 (HT)，我们就将其记为 0。
- 如果结果是反-正 (TH)，我们就将其记为 1。
- 如果结果是正-正 (HH) 或反-反 (TT)，我们就舍弃这次结果，重新抛掷两次。

第一步：验证该策略确实可以生成我们要求的随机数

设硬币正面朝上的概率为 p，反面朝上的概率为 $1-p$（其中 $p \neq 0, 0.5, 1$）。

计算各种情况的概率：

$$P(HT) = p \cdot (1-p) \tag{1}$$

$$P(TH) = (1-p) \cdot p = p \cdot (1-p) \tag{2}$$

$$P(HH) = p^2 \tag{3}$$

$$P(TT) = (1-p)^2 \tag{4}$$

注意到 $P(HT) = P(TH) = p(1-p)$

在条件概率下：

$$P(\text{输出 } 0 | \text{不拒绝}) = \frac{P(HT)}{P(HT) + P(TH)} = \frac{p(1-p)}{2p(1-p)} = \frac{1}{2} \tag{5}$$

$$P(\text{输出 } 1 | \text{不拒绝}) = \frac{P(TH)}{P(HT) + P(TH)} = \frac{p(1-p)}{2p(1-p)} = \frac{1}{2} \tag{6}$$

因此，每次生成的随机数都是公平的，即 $P(\text{输出 } 0) = P(\text{输出 } 1) = \frac{1}{2}$。

第二步：生成 0-15 的均匀分布

考虑到是生成 16 个数，我们可以二进制来处理，具体来说：

1. 使用上述方法生成 4 个概率均等的随机数 $b_3 b_2 b_1 b_0$

2. 将这上面的随机数转换为整数：$n = 8b_3 + 4b_2 + 2b_1 + b_0$

3. 由于 $n \in \{0, 1, 2, \ldots, 15\}$，直接输出 n

由于每次生成的单个随机数都是独立且公平的（$P(b_i = 0) = P(b_i = 1) = \frac{1}{2}$），所以：

$$P(n = k) = P(b_3b_2b_1b_0\text{的二进制表示为}k) = \left(\frac{1}{2}\right)^4 = \frac{1}{16}$$

对于所有 $k \in \{0, 1, 2, \ldots, 15\}$ 都成立。

第三步：生成 100 个随机数：

重复上述过程 100 次，每次生成一个 0-15 之间的随机整数。由于每次生成都是独立的，且每个整数的概率都是 $\frac{1}{16}$，所以生成的 100 个数严格服从 0-15 的均匀分布。

2. 剪纸转圈圈

(1) 小 U 在做剪纸小游戏, 他先剪了一大一小两个圆, 如图 1 所示, 其中圆 A 的半径为 1, 圆 B 的半径为 5。现在让小圆 A 绕着大圆 B 滚动旋转, 请问小圆 A 旋转多少圈后其圆心将再次到达起点?

我们可以将小圆的旋转进行分解

- **小圆滚动产生的自转**：这部分等于两圆周长之比，即 $C_B/C_A = (2\pi R)/(2\pi r) = R/r = 5/1 = 5$ 圈。
- **公转产生的自转**：小圆 A 的圆心绕大圆 B 的圆心完成了一整个圆周运动，这个公转本身会带来额外的 1 圈自转。

总圈数 = 滚动自转 + 公转自转 = $5 + 1 = 6$ 圈。

(2) 小 U 又剪了一个边长为 1 的小正方形 C, 现在让小正方形 C 在大圆 B 的圆周上移动一周 (保持正方形始终直立, 如图 2), 请聪明的你帮小 U 算一算: 小正方形 C 平移一周所覆盖区域的面积是多少?

我们考虑微元法进行求解

1. 将正方形的运动分解为当其中心沿着圆周移动一个无穷小的角度 $d\theta$ 时的过程。

2. 计算微元面积 dA 计算在这个无穷小角度 $d\theta$ 内，正方形所扫过的面积 dA。

3. 积分求和：将微元面积 dA 从 0 到 2π (一整圈) 进行定积分，得出总面积 A。

具体来说，我们可以把正方形扫过的这片极薄的“切片”区域，看作是一个极薄的环形扇区。这个环形扇区的面积，等于由外边界扫过的扇区面积减去由内边界扫过的扇区面积。一个半径为 r、圆心角为 $d\theta$ 的扇区面积是 $\frac{1}{2}r^2d\theta$。

3. 正态分布

概率论和微积分是一个 quant 需要熟练掌握的基础知识, 小 U 正在温习正态分布的相关内容。设 $\varphi(y)$ 和 $\Phi(y)$ 分别为标准正态分布的密度和分布函数, 求:

$$\int_{-\infty}^{+\infty} \left(y + \frac{1}{\sqrt{\pi}}\right)^2 (1 - \Phi(y))\varphi(y)dy$$

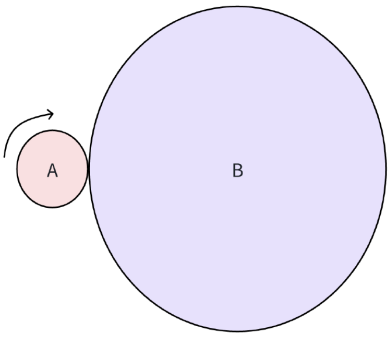

图 1: 小圆 A 绕大圆 B 滚动

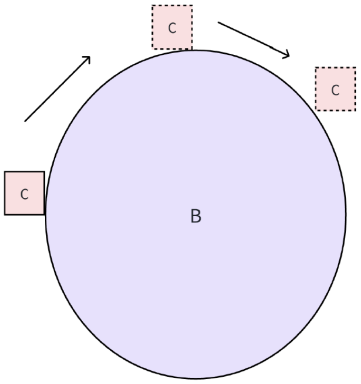

图 2: 正方形 C 沿大圆 B 移动

令要求解的积分为 I，我们将采用对称性和分部积分法来解决。:

$$I = \int_{-\infty}^{+\infty} \left(y + \frac{1}{\sqrt{\pi}}\right)^2 (1 - \Phi(y))\varphi(y)dy$$

1. 对称性换元

我们利用标准正态分布的两个关键性质：密度函数 $\varphi(y)$ 是一个偶函数以及分布函数满足 $\Phi(-y) = 1 - \Phi(y)$

在积分 I 中，令 $z = -y$，则 $y = -z$ ，得到:

$$I = \int_{-\infty}^{+\infty} \left(z - \frac{1}{\sqrt{\pi}}\right)^2 (1 - (1 - \Phi(z)))\varphi(z)dz$$

将变量 z 换回 y，我们得到一个与原式等价的积分表达式:

$$I = \int_{-\infty}^{+\infty} \left(y - \frac{1}{\sqrt{\pi}}\right)^2 \Phi(y)\varphi(y)dy$$

2. 求解

将两个等价的 I 表达式相加，得到 $2I$:

$$2I = \int_{-\infty}^{+\infty} \left[\left(y + \frac{1}{\sqrt{\pi}}\right)^2 (1 - \Phi(y)) + \left(y - \frac{1}{\sqrt{\pi}}\right)^2 \Phi(y)\right] \varphi(y)dy$$

化简表达式得到:

$$2I = \int_{-\infty}^{+\infty} \left[y^2 + \frac{1}{\pi} + \frac{2y}{\sqrt{\pi}}(1 - 2\Phi(y))\right] \varphi(y)dy$$

现在，我们将积分拆分为三个部分:

$$2I = \int_{-\infty}^{+\infty} y^2\varphi(y)dy + \frac{1}{\pi}\int_{-\infty}^{+\infty} \varphi(y)dy + \frac{2}{\sqrt{\pi}}\int_{-\infty}^{+\infty} y(1 - 2\Phi(y))\varphi(y)dy$$

3. 计算各部分积分

- **第一部分**: $\int_{-\infty}^{+\infty} y^2\varphi(y)dy$ 是标准正态分布的二阶矩，等于其方差，即 **1**。
- **第二部分**: $\frac{1}{\pi}\int_{-\infty}^{+\infty} \varphi(y)dy$。根据概率密度函数的定义，$\int_{-\infty}^{+\infty} \varphi(y)dy = 1$。所以，该项等于 $\frac{1}{\pi}$。

- **第三部分**: $\frac{2}{\sqrt{\pi}}\left[\int y\varphi(y)dy - 2\int y\Phi(y)\varphi(y)dy\right]$
  - 其中 $\int_{-\infty}^{+\infty} y\varphi(y)dy$ 是标准正态分布的均值，等于 0。
  - 对于 $\int_{-\infty}^{+\infty} y\Phi(y)\varphi(y)dy$，通过分部积分可得其值为 $\frac{1}{2\sqrt{\pi}}$，具体计算过程见附录。
  - 因此，第三部分的值为：$\frac{2}{\sqrt{\pi}}\left[0 - 2\left(\frac{1}{2\sqrt{\pi}}\right)\right] = -\frac{2}{\pi}$。

4. 汇总结果

将三部分的结果代入 $2I$ 的表达式：

$$2I = 1 + \frac{1}{\pi} - \frac{2}{\pi} = 1 - \frac{1}{\pi}$$

最终，求解 I：

$$I = \frac{1}{2} - \frac{1}{2\pi}$$

4. 打家劫舍

小 U 在复习完数学后觉得 coding 也不能落下，请聪明的你帮他设计一个时间复杂度尽可能小的算法，求一个整数数组 S 中不含相邻元素的子序列的最大和。

我们考虑动态规划的方法来解决当前问题，可以取得理论最优的时间复杂度 O(n) 和空间复杂度 O(1)。具体来说，我们从第一数组元素遍历每一个元素，每一次比较获取之前的元素和获取当前元素的数组和哪一个更大。

```python
def rob_optimized(S):
    """
    计算不含相邻元素的子序列的最大和
    param S: 一个整数数组
    return: 不含相邻元素的子序列的最大和
    """
    n = len(S)
    if n == 0:
        return 0

    # prev_max: 相当于 dp[i-2]
    # curr_max: 相当于 dp[i-1]
    prev_max = 0
    curr_max = 0

    for amount in S:
        # temp 用来临时存储 curr_max (即旧的 dp[i-1])
        temp = curr_max
        # 新的 curr_max (即 dp[i]) 是获取前一个元素的最大值(curr_max)与获得当前元素(amount + prev_max)的较大值
        curr_max = max(curr_max, amount + prev_max)
        # 更新 prev_max 为旧的 curr_max，为下一次迭代做准备
        prev_max = temp

    return curr_max
# --- 示例数组 ---
example_S = [2, 7, 9, 3, 1]

# --- 调用函数并打印结果 ---
max_sum = rob_optimized(example_S)
print(f"对于示例数组 S = {example_S}")
```

```
    # 对于示例数组 S = [2, 7, 9, 3, 1]
    print(f"不含相邻元素的最大和为: {max_sum}")
    # 不含相邻元素的最大和为: 12
```

Listing 1: O(1) 空间复杂度和 O(n) 时间复杂度的解法

5. Linear Regression Model

Xiao U is learning English and regression analysis. Please help him solve the following problem: Suppose we have a simple linear regression model,

$$y_j = \beta_0 + \beta_1 x_j + \epsilon_j, \quad j = 1, 2, ..., n$$

But the x_j are independently $N(\mu_x, \sigma_x^2)$. We assume further that $\epsilon_j \sim N(0, \sigma_e^2)$, independently of the x_j's. Suppose further that x_j is in fact measured with error, so the observations are (y_j, w_j), where

$$w_j = x_j + u_j, \quad u_j \sim N(0, \sigma_u^2)$$

and u is independent of both x and ϵ. Now, please show that $E(y_j|w_j) = \alpha_0 + \alpha_1 w_j$, and give expressions for α_0 and α_1 as functions of known constants.

因为 $E(y_j|w_j) = E(\beta_0 + \beta_1 x_j + \epsilon_j|w_j) = \beta_0 + E(\beta_1 x_j + \epsilon_j|w_j) = \beta_0 + \beta_1 E(x_j|w_j) + E(\epsilon_j|w_j)$
又因为 ϵ_j 与 w_j 独立，所以 $E(\epsilon_j|w_j) = E(\epsilon_j) = 0$

对于 $E(x_j|w_j)$，我们可以应用附录中证明的正态分布条件期望公式。在本题中，x_j 和 w_j 构成二元正态随机变量，其中：

- $E(x_j) = \mu_x$
- $Var(x_j) = \sigma_x^2$
- $E(w_j) = E(x_j + u_j) = \mu_x + 0 = \mu_x$
- $Var(w_j) = Var(x_j + u_j) = \sigma_x^2 + \sigma_u^2$
- $Cov(x_j, w_j) = Cov(x_j, x_j + u_j) = Cov(x_j, x_j) + Cov(x_j, u_j) = \sigma_x^2 + 0 = \sigma_x^2$

  根据附录中的条件期望公式：

$$E(X|Y = y) = E(X) + \frac{Cov(X, Y)}{Var(Y)}(y - E(Y))$$

  代入我们的参数，整理得到：

$$E(x_j|w_j) = \mu_x + \frac{\sigma_x^2}{\sigma_x^2 + \sigma_u^2}(w_j - \mu_x)$$

$$E(y_j|w_j) = \beta_0 + \beta_1 \left[\mu_x + \frac{\sigma_x^2}{\sigma_x^2 + \sigma_u^2}(w_j - \mu_x)\right]$$

$$E(y_j|w_j) = \beta_0 + \beta_1 \mu_x \left(1 - \frac{\sigma_x^2}{\sigma_x^2 + \sigma_u^2}\right) + \beta_1 \frac{\sigma_x^2}{\sigma_x^2 + \sigma_u^2} w_j$$

  令 $\alpha_0 = \beta_0 + \beta_1 \mu_x \left(1 - \frac{\sigma_x^2}{\sigma_x^2 + \sigma_u^2}\right)$，$\alpha_1 = \beta_1 \frac{\sigma_x^2}{\sigma_x^2 + \sigma_u^2}$
  则：

$$E(y_j|w_j) = \alpha_0 + \alpha_1 w_j$$

证明完毕。

6. 共半球问题

(1)

小 U 发现了一道经典的量化题: 圆上随机取 N 个点, 它们在同一个半圆内的概率有多大?

让我们先任意选择一个点，比如 P_1，以 P_1 为起点，顺时针画出一个半圆。我们称这个半圆为 H_1。要让所有其他 $N-1$ 个点都落在这个特定的半圆 H_1 内，概率是多少?

- 对于任何一个随机点 P_i $(i \neq 1)$，它落在 H_1 内的概率是 $\frac{1}{2}$。
- 因为这 $N-1$ 个点的选择是相互独立的，所以它们全部落在 H_1 内的概率是 $(\frac{1}{2})^{N-1}$。

并计这个事件为事件 A_i，

- A_1: 所有点都落在以 P_1 为起点的顺时针半圆内。
- A_2: 所有点都落在以 P_2 为起点的顺时针半圆内。
- ...
- A_N: 所有点都落在以 P_N 为起点的顺时针半圆内。

注意到，这 N 个事件的互斥的，且根据对称性，其概率相同均为 $(\frac{1}{2})^{N-1}$。

(2)

求知欲很强的小 U 继续探索, 假如问题变成 3 维球体呢? 即在球面上随机取 N 个点, 它们在同一个半球内的概率有多大?

事件是"所有 N 个点都落在某一个半球内", 同样，我们先任意选择一个点 P_1。这个点 P_1 可以被看作是某个半球的"极点"。这个半球 H_1 就是由过球心且垂直于 $\vec{OP_1}$ 向量的平面所定义的、P_1 所在的那一半球面。要让所有其他 $N-1$ 个点都落在这个特定的半球 H_1 内，概率为 $(\frac{1}{2})^{N-1}$。我们同样定义 N 个事件 A_i: 所有其他 $N-1$ 个点都落在以 P_i 为"极点"的那个半球内。正如在二维情况中一样，这些事件也是相互排斥的。

$$P(\text{所有点在同一个半球内}) = \sum_{i=1}^{N} P(A_i) = N \times \left(\frac{1}{2}\right)^{N-1}$$

(3) 【拓展思考题】

在 n 维空间的球面上随机取 N 个点, 它们在同一个半球内的概率有多大?

类似的，维度的增加并没有改变问题的实质。在 n 维空间中，一个"超半球"(hyper-hemisphere) 是由一个通过球心的 $(n-1)$ 维超平面（hyperplane）将 $(n-1)$-球面切割成两半而得到的。。因此，一个随机点落在指定超半球内的概率永远是 $\frac{1}{2}$。类似的事件 A_i 也是相互排斥的，且根据对称性，其概率相同均为 $(\frac{1}{2})^{N-1}$。

$$P(\text{所有点在同一个超半球内}) = \sum_{i=1}^{N} P(A_i) = N \times \left(\frac{1}{2}\right)^{N-1}$$

7. Pattern Problem

小 U 在学习 stochastic process, 他手里有一个不均匀的硬币, 每次掷到正面 (H) 的概率为 p, 掷到反面 (T) 的概率为 q, $p+q=1$, 请问首次掷到以下 pattern 的期望投掷次数为多少?

1. HHHHHHTTTTTT
2. HHHHHHHHHHHH
3. HTHTHTTHTHT

7.1 鞅停时定理求解

假设我们有一枚不均匀的硬币，每次投掷得到正面（H）的概率为 p，得到反面（T）的概率为 q，其中 $p+q=1$。对于一个给定的、由 H 和 T 组成的模式 $S=s_1s_2...s_L$，我们的目标是计算首次观察到该模式所需的期望投掷次数，记为 E。

为了应用鞅理论，我们构造一个如下的假想公平赌局：

1. **赌徒入场**：在每次投掷 $n=1,2,\ldots$ 开始之前，都有一位新的赌徒 C_n 加入赌局，并下注 1 元。
2. **押注策略**：所有赌徒都采用相同的策略。赌徒 C_n 在他入场时（即第 n 次投掷），押注该次投掷结果为模式的第一个字符 s_1。如果他赢了，他将所有赢得的钱押在下一次（第 $n+1$ 次）投掷的结果是 s_2 上，以此类推。只要输一次，他就失去所有钱并出局。
3. **公平赔率**：为了使赌局对赌场和赌徒双方都公平，对于一次成功的押注，其赔付规则是：若押注的事件发生概率为 P_e，则 1 元的赌注在成功后会变为 $1/P_e$ 元。

在这个设定下，任何一次押注的期望净收益都为零，因此整个系统是公平的。

7.1.1 构造鞅过程：

令 $\mathcal{W}_n$ 表示在第 n 次投掷结束后，赌场的累计净收益。这个随机过程 $\{\mathcal{W}_n\}_{n\geq 0}$ 是一个鞅（Martingale），因为它满足鞅性质：

$$E[\mathcal{W}_{n+1}|\mathcal{F}_n]=\mathcal{W}_n$$

其中 $\mathcal{F}_n$ 代表直到时间 n 的所有历史信息。由于赌局是公平的，赌场的期望收益在任何时刻都应保持不变。初始时 $\mathcal{W}_0=0$，故 $E[\mathcal{W}_n]=0$ 对所有 n 成立。

7.1.2 应用鞅停时定理进行求解：

我们定义一个停时 τ 为模式 S 首次出现的时刻。即：

$$\tau=\min\{n\geq L: X_{n-L+1}\ldots X_n=S\}$$

其中 X_k 是第 k 次投掷的结果。根据鞅停时定理，在满足一定条件下（本问题的构造满足），鞅过程在停时的期望值等于其初始值：

$$E[\mathcal{W}_\tau]=E[\mathcal{W}_0]=0$$

这个等式是推导的关键。它告诉我们，当游戏结束时，赌场的期望净收益为零。这意味着：

$$E[\text{赌场在}\tau\text{ 时刻的总收入}]=E[\text{赌场在}\tau\text{ 时刻的总支出}]$$

我们来分别计算收入和支出。

- **总收入**：在停时 τ 到来时，从第 1 次到第 τ 次投掷，共有 τ 位赌徒入场，每位下注 1 元。所以赌场的总收入就是 τ。其期望为 $E[\tau]$。

- **总支出**：在时刻 τ 游戏结束时，赌场需要向所有仍然“存活”的赌徒支付奖金。

  一位赌徒 $C_{\tau-k+1}$（在第 $\tau-k+1$ 次投掷前入场）若要存活到最后，他必须从入场开始连续赢 k 次。他的押注序列是 $s_1, s_2, \ldots, s_k$，而他经历的实际投掷结果序列是 $X_{\tau-k+1}, \ldots, X_\tau$。

  我们知道在停时 τ，实际序列 $X_{\tau-L+1} \ldots X_\tau$ 就是模式 S 本身。因此，序列 $X_{\tau-k+1} \ldots X_\tau$ 恰好是模式 S 的长度为 k 的后缀。

  所以，赌徒 $C_{\tau-k+1}$ 存活的充要条件是：S 的长度为 k 的前缀与后缀完全相同。即：

  $$s_1 s_2 \ldots s_k = s_{L-k+1} s_{L-k+2} \ldots s_L$$

  如果这个条件满足，该赌徒的初始 1 元赌注在连续赢了 k 次后，会变为：

  $$\frac{1}{P(s_1)} \times \frac{1}{P(s_2)} \times \cdots \times \frac{1}{P(s_k)} = \frac{1}{P(s_1 s_2 \ldots s_k)}$$

  赌场的总支出是所有这些存活赌徒的奖金之和。

将收入和支出代入平衡等式，我们得到最终的计算公式：

$$E[\tau] = \sum_{k=1}^{L} \frac{\mathbb{I}(s_1 \ldots s_k = s_{L-k+1} \ldots s_L)}{P(s_1 \ldots s_k)}$$

其中 $\mathbb{I}(\cdot)$ 是指示函数，当条件为真时取值为 1，否则为 0。

7.1.3 具体模式匹配问题求解

模式 1: `HHHHHHTTTTTT`

$S_1 = \text{HHHHHHTTTTTT}$, $L = 12$。我们检查其前缀和后缀的重叠。可以发现，除了 $k = 12$ 时整个模式自身与其自身重叠外，没有任何其他长度的重叠。

- $k = 1$: $\text{H} \neq \text{T}$
- $k = 2$: $\text{HH} \neq \text{TT}$
- ...
- $k = 11$: $\text{HHHHHHTTTTT} \neq \text{HHHHHTTTTTT}$

因此，只有 $k = 12$ 的项有贡献：

$$E_1 = \frac{1}{P(\text{HHHHHHTTTTTT})} = \frac{1}{p^6 q^6}$$

模式 2: `HHHHHHHHHHHH`

$S_2 = \text{HHHHHHHHHHHH}$, $L = 12$。这个模式具有完全的重叠性，任何长度为 k 的前缀都与其后缀相同。因此，从 $k = 1$ 到 $k = 12$ 的每一项都有贡献。

$$\begin{aligned}
E_2 &= \sum_{k=1}^{12} \frac{1}{P(\text{H}^k)} \\
&= \frac{1}{p} + \frac{1}{p^2} + \frac{1}{p^3} + \cdots + \frac{1}{p^{12}} \\
&= \frac{1/p \cdot ((1/p)^{12} - 1)}{1/p - 1} \quad \text{(等比数列求和)} \\
&= \frac{p^{-1}(p^{-12} - 1)}{(1-p)/p} = \frac{p^{-13} - p^{-1}}{q} \\
&= \frac{1 - p^{12}}{q \cdot p^{12}}
\end{aligned}$$

模式 3: `HTHTHTTHTHT` $S_3 = \text{HTHTHTTHTHT}$, $L = 11$。我们需要仔细检查其重叠情况：

- $k = 2$: HT == HT.
- $k = 4$: HTHT == HTHT.
- $k = 9$: HTHTHTTHT == HTHTHTTHT.
- $k = 11$: 整个模式自身重叠。

其他长度的 k 均不构成重叠。因此，期望值为这四项贡献之和：

$$\begin{aligned} E_3 &= \frac{1}{P(\text{HT})} + \frac{1}{P(\text{HTHT})} + \frac{1}{P(\text{HTHTHTTHT})} + \frac{1}{P(\text{HTHTHTTHTHT})} \\ &= \frac{1}{pq} + \frac{1}{p^2q^2} + \frac{1}{p^5q^4} + \frac{1}{p^6q^5} \end{aligned}$$

7.2 马尔可夫过程求解

我们也可以应用马尔可夫过程对原问题进行求解，具体来说，我们将模式匹配过程转化为一个吸收马尔可夫链中的状态转移问题，并利用迭代法建立并求解线性方程组。我们将寻找目标模式的过程，看作一个随机系统在一系列离散状态间的转移。当系统到达代表“模式完全匹配”的最终状态时，它将被“吸收”，过程结束。

对于一个长度为 L 的目标模式 $S = s_1s_2...s_L$，我们定义 $L + 1$ 个状态：

- **状态 i (S_i)**，其中 $0 \le i < L$：代表最近的 i 次投掷结果恰好与模式 S 的前 i 个字符 ($s_1...s_i$) 相同。状态 S_0 是初始状态，表示没有任何前缀被匹配。
- **状态 L (S_L)**：代表整个模式 S 已经完全匹配。这是一个吸收状态，一旦进入，过程便告终止。

目标：计算期望首达时间

我们的目标是计算从初始状态 S_0 出发，首次到达吸收状态 S_L 所需的期望步数。我们令 E_i 为从状态 S_i 出发，到达吸收状态 S_L 所需要的额外投掷次数的期望值。根据定义，我们最终要求解的是 E_0。显然，如果我们已经身处吸收状态，那么不再需要任何额外步骤，因此：

$$E_L = 0$$

我们对任意非吸收状态 S_i ($0 \le i < L$) 进行迭代分析。考虑从状态 S_i 出发，进行一次投掷，系统会转移到某个新状态 S_j。E_i 的值等于这次投掷的成本（1 步）加上到达新状态 S_j 后剩余路程的期望值 E_j。

$$E_i = 1 + \sum_{j=0}^{L} P(i \to j) \cdot E_j$$

其中 $P(i \to j)$ 是从状态 i 转移到状态 j 的概率。具体来说，我们可以写成：

$$E_i = 1 + p \cdot E_{\text{next state if H}} + q \cdot E_{\text{next state if T}}$$

这里的关键在于确定“下一个状态”是什么。规则如下：

- 假设当前在状态 S_i（已匹配 $s_1...s_i$），下一次投掷结果为 $c \in \{H, T\}$。
- 形成新序列 $s_1...s_i + c$。
- 在这个新序列的所有后缀中，寻找一个能与模式 S 的前缀相匹配的，并取其最长的匹配长度 j。系统则转移到状态 S_j。

程序实现思路我们可以将 L 个关于 $E_0, \ldots, E_{L-1}$ 的线性方程组表示为矩阵形式 $\mathbf{Ax} = \mathbf{b}$。

- $\mathbf{x} = [E_0, E_1, \ldots, E_{L-1}]^T$ 是我们要求的解向量。
- $\mathbf{b} = [1, 1, \ldots, 1]^T$ 是长度为 L 的常数向量。
- $\mathbf{A}$ 是一个 $L \times L$ 的系数矩阵。对于第 i 行（代表 E_i 的方程），我们将方程 $E_i = 1 + pE_j + qE_k$ 整理为 $E_i - pE_j - qE_k = 1$。
  - 对角线元素 $A_{i,i}$ 初始为 1。
  - 若从状态 i 投 H 后转移到状态 $j < L$，则 $A_{i,j}$ 减去 p。
  - 若从状态 i 投 T 后转移到状态 $k < L$，则 $A_{i,k}$ 减去 q。

建立好矩阵 $\mathbf{A}$ 和向量 $\mathbf{b}$ 后，即可调用数值计算库求解 $\mathbf{x}$。

```python


import numpy as np

def get_next_state(pattern: str, current_len: int, next_char: str) -> int:
    """
    计算在给定当前匹配长度和下一个字符时，马尔可夫链的下一个状态。

    Args:
        pattern (str): 目标模式，例如 "HTH"。
        current_len (int): 当前已匹配的模式前缀的长度（即当前状态）。
        next_char (str): 下一次投掷的实际结果 ('H' 或 'T')。

    Returns:
        int: 转移后的新状态（即新的匹配长度）。
    """
    # 构造当前匹配序列加上新字符后的完整序列
    sequence = pattern[:current_len] + next_char

    # 从最长的可能性开始，寻找新序列的后缀与模式前缀的最长匹配
    # 这是马尔可夫链状态转移的核心逻辑
    for length in range(len(sequence), 0, -1):
        suffix = sequence[-length:]
        prefix = pattern[:length]
        if suffix == prefix:
            return length  # 返回最长的匹配长度作为新状态

    return 0  # 如果没有任何匹配，回到初始状态 0

def calculate_expected_flips_markov(pattern: str, p_heads: float) -> float:
    """
    使用吸收马尔可夫链模型计算首次出现特定模式的期望投掷次数。

    Args:
        pattern (str): 由 'H' 和 'T' 组成的目标模式。
        p_heads (float): 掷出正面 (H) 的概率。

    Returns:
        float: 首次出现该模式的期望投掷次数。
    """
    L = len(pattern)
    if L == 0:
        return 0.0
```

```
44
45          p_tails = 1.0 - p_heads
46
47          # 我们要求解 E_0, E_1, ..., E_{L-1} 这 L 个变量。
48          # 吸收状态 E_L = 0。
49          # 方程组形式为 A * x = b, 其中 x 是 [E_0, E_1, ..., E_{L-1}]^T
50          n_states = L
51          A = np.zeros((n_states, n_states))
52          b = np.ones(n_states)
53
54          # 遍历每个状态 i (从 0 到 L-1), 为 E_i 建立一个方程
55          for i in range(n_states):
56              # 方程 E_i = 1 + p * E_{next_H} + q * E_{next_T}
57              # 整理为标准形式: E_i - p*E_{next_H} - q*E_{next_T} = 1
58
59              # 1. 设置 E_i 的系数
60              A[i, i] = 1.0
61
62              # 2. 计算掷出 H 后的转移
63              next_state_H = get_next_state(pattern, i, 'H')
64              if next_state_H < L:  # 如果不是吸收状态
65                  A[i, next_state_H] -= p_heads
66
67              # 3. 计算掷出 T 后的转移
68              next_state_T = get_next_state(pattern, i, 'T')
69              if next_state_T < L:  # 如果不是吸收状态
70                  A[i, next_state_T] -= p_tails
71
72          # 使用 numpy 求解线性方程组 A * x = b
73          try:
74              expected_values = np.linalg.solve(A, b)
75          except np.linalg.LinAlgError:
76              return float('inf') # 如果矩阵是奇异的, 则无法求解
77
78          # 我们要求的是从状态 0 开始的期望次数 E_0
79          return expected_values[0]
80
81      ### 程序使用示例
82
83
84      if __name__ == "__main__":
85          # 设定硬币的概率
86          # 为了演示, 我们假设是一个均匀硬币, p=0.5
87          # 你可以修改为任意 0 < p < 1 的值
88          p_h = 0.6
89          print(f"设定硬币掷出正面的概率 p = {p_h}\n")
90
91          # 定义三个模式
92          patterns = [
93              "HHHHHHTTTTTT", # 1. 几乎不重叠
94              "HHHHHHHHHHHH", # 2. 高度重叠
95              "HTHTHTTHTHT"   # 3. 复杂重叠
96          ]
97
98          for p_str in patterns:
99              print(f"--- 正在计算模式: {p_str} ---")
100
101             # 使用马尔可夫链方法计算
102             expected_flips = calculate_expected_flips_markov(p_str, p_h)
```

```
103
104             print(f"期望投掷次数为: {expected_flips:.2f}")
105
106             # 为了对比，我们也可以用鞅方法的公式直接计算来验证
107             # (这部分仅为验证，核心是上面的马尔可夫方法)
108             martingale_result = 0
109             p_t = 1.0 - p_h
110             for k in range(1, len(p_str) + 1):
111                 prefix = p_str[:k]
112                 suffix = p_str[-k:]
113                 if prefix == suffix:
114                     prob_prefix = 1.0
115                     for char in prefix:
116                         prob_prefix *= p_h if char == 'H' else p_t
117                     martingale_result += 1 / prob_prefix
118
119             print(f"（使用鞅方法验证结果: {martingale_result:.2f}）")
120             print("-" * 30 + "\n")
121
122         # 你也可以测试一个更短的、更有趣的模式
123         print("--- 正在计算模式: HTHT ---")
124         expected_flips_ht_ht = calculate_expected_flips_markov("HTHT", 0.5)
125         print(f"期望投掷次数为: {expected_flips_ht_ht:.2f}")
126         print("-" * 30 + "\n")
```

Listing 2: 求解期望投掷次数的函数核心逻辑

附录

附录 1: 正态分布概率密度函数推导

首先我们先证明正态分布概率密度函数有如下性质:

$$\begin{aligned}
\varphi'(y) &= \frac{d}{dy}\left(\frac{1}{\sqrt{2\pi}}e^{-\frac{y^2}{2}}\right) \\
&= \frac{1}{\sqrt{2\pi}}\cdot\frac{d}{dy}\left(e^{-\frac{y^2}{2}}\right) \\
&= \frac{1}{\sqrt{2\pi}}\cdot e^{-\frac{y^2}{2}}\cdot(-y) \\
&= -y\cdot\frac{1}{\sqrt{2\pi}}e^{-\frac{y^2}{2}} \\
&= -y\cdot\varphi(y)
\end{aligned}$$

接着我们可以推导出：

$$\begin{aligned}\int_{-\infty}^{+\infty} y\Phi(y)\varphi(y)dy &= -\int_{-\infty}^{+\infty} \Phi(y)d(-\varphi(y)) \\ &= -\left[\Phi(y)(-\varphi(y))\right]_{-\infty}^{+\infty} + \int_{-\infty}^{+\infty} \varphi(y)d\Phi(y) \\ &= -\left[\Phi(y)(-\varphi(y))\right]_{-\infty}^{+\infty} + \int_{-\infty}^{+\infty} \varphi(y)^2 dy \\ &= 0 + \int_{-\infty}^{+\infty} \frac{1}{2\pi} e^{-y^2} dy \\ &= \frac{1}{2\pi} \cdot \sqrt{\pi} \\ &= \frac{1}{2\sqrt{\pi}}\end{aligned}$$

附录 2: 证明正态分布的条件期望公式

证明过程的核心思想是：

1. 写出 X 和 Y 的联合概率密度函数 $f(x, y)$。
2. 写出 Y 的边缘概率密度函数 $f_Y(y)$。
3. 利用公式 $f_{X|Y}(x|y) = \frac{f(x,y)}{f_Y(y)}$ 求出在 Y=y 的条件下 X 的条件概率密度函数。
4. 这个条件概率密度函数本身也是一个正态分布，它的期望值就是我们要求的 $E(X|Y = y)$。

第一步：定义联合概率密度函数

设随机变量 (X, Y) 服从双变量正态分布，其参数如下：

- 均值：$E(X) = \mu_X$, $E(Y) = \mu_Y$
- 方差：$Var(X) = \sigma_X^2$, $Var(Y) = \sigma_Y^2$
- 相关系数：$Corr(X, Y) = \rho$

其联合概率密度函数 $f(x, y)$ 的核心在于指数部分。我们可以将指数部分记为 $-\frac{1}{2}Q(x, y)$，其中：

$$Q(x, y) = \frac{1}{1-\rho^2}\left[\left(\frac{x-\mu_X}{\sigma_X}\right)^2 - 2\rho\left(\frac{x-\mu_X}{\sigma_X}\right)\left(\frac{y-\mu_Y}{\sigma_Y}\right) + \left(\frac{y-\mu_Y}{\sigma_Y}\right)^2\right]$$

第二步：分解指数项

在这里，我们把所有和 x 相关的项整理成一个平方项。$Q(x, y)$ 中和 x 相关的部分可以看作是关于变量 $(\frac{x-\mu_X}{\sigma_X})$ 的二次方程。我们把它配成一个完全平方：

$$\begin{aligned}Q(x, y) &= \frac{1}{1-\rho^2}\left\{\left[\left(\frac{x-\mu_X}{\sigma_X}\right) - \rho\left(\frac{y-\mu_Y}{\sigma_Y}\right)\right]^2 + (1-\rho^2)\left(\frac{y-\mu_Y}{\sigma_Y}\right)^2\right\} \\ &= \frac{1}{1-\rho^2}\left[\left(\frac{x-\mu_X}{\sigma_X}\right) - \rho\left(\frac{y-\mu_Y}{\sigma_Y}\right)\right]^2 + \left(\frac{y-\mu_Y}{\sigma_Y}\right)^2\end{aligned}$$

这个形式非常重要，因为它成功地将表达式分离成两部分：一个同时包含 x 和 y 的平方项，以及一个只包含 y 的项。

第三步：导出条件概率密度函数

现在我们可以重写整个联合概率密度函数 $f(x,y)$。代入我们分解后的 $Q(x,y)$，并利用指数性质 $e^{A+B}=e^Ae^B$，我们可以把表达式拆开：

$$\begin{aligned} f(x,y) &= \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho^2}}\exp\left(-\frac{1}{2}Q(x,y)\right) \\ &= \underbrace{\left[\frac{1}{\sqrt{2\pi}\sigma_Y}\exp\left(-\frac{(y-\mu_Y)^2}{2\sigma_Y^2}\right)\right]}_{f_Y(y)} \\ &\quad \times \underbrace{\left[\frac{1}{\sqrt{2\pi}\sigma_X\sqrt{1-\rho^2}}\exp\left(-\frac{\left[\left(\frac{x-\mu_X}{\sigma_X}\right)-\rho\left(\frac{y-\mu_Y}{\sigma_Y}\right)\right]^2}{2(1-\rho^2)}\right)\right]}_{f_{X|Y}(x|y)} \end{aligned}$$

第四步：条件分布的期望

现在我们仔细观察条件概率密度函数 $f_{X|Y}(x|y)$。我们可以把它整理成一个标准正态分布 PDF 的形式 $\frac{1}{\sqrt{2\pi}\sigma_{\text{new}}}\exp(-\frac{(x-\mu_{\text{new}})^2}{2\sigma_{\text{new}}^2})$。

$$f_{X|Y}(x|y)=\frac{1}{\sqrt{2\pi}\sigma_X\sqrt{1-\rho^2}}\exp\left(-\frac{\left(x-\left[\mu_X+\rho\frac{\sigma_X}{\sigma_Y}(y-\mu_Y)\right]\right)^2}{2\sigma_X^2(1-\rho^2)}\right)$$

通过与标准形式对比，我们可以直接读出这个条件分布的期望（均值）和方差：

- **条件期望 (均值):** $\mu_{\text{new}}=E(X|Y=y)=\mu_X+\rho\frac{\sigma_X}{\sigma_Y}(y-\mu_Y)$
- **条件方差:** $\sigma_{\text{new}}^2=Var(X|Y=y)=\sigma_X^2(1-\rho^2)$

第五步：代入协方差

我们已经得到了条件期望的表达式，现在只需将其转换成用协方差表示的形式。根据定义，相关系数 $\rho=\frac{Cov(X,Y)}{\sigma_X\sigma_Y}$。将其代入我们得到的期望公式中：

$$\begin{aligned} E(X|Y=y) &= \mu_X+\left(\frac{Cov(X,Y)}{\sigma_X\sigma_Y}\right)\frac{\sigma_X}{\sigma_Y}(y-\mu_Y) \\ &= \mu_X+\frac{Cov(X,Y)}{\sigma_Y^2}(y-\mu_Y) \end{aligned}$$

最后，将 μ_X, μ_Y, σ_Y^2 替换为期望和方差的符号即可：

$$E(X|Y=y)=E(X)+\frac{Cov(X,Y)}{Var(Y)}(y-E(Y))$$

从而完成证明。