大数据机器学习第七次理论作业

2026 年 1 月 22 日

一、 题目: 如例 9.1 的三硬币模型。假设观测数据不变, 试选择不同的初值, 例如,$\pi^{(0)} = 0.46, p^{(0)} = 0.55, q^{(0)} = 0.67$, 求模型参数 $\sigma = (\pi, p, q)$ 的极大似然估计。

例 9.1 (三硬币模型) 假设有 3 枚硬币, 分别记作 A,B,C. 这些硬币正面出现的概率分别是 π, p 和 q. 进行如下掷硬币试验: 先掷硬币 A, 根据其结果选出硬币 B 或硬币 C, 正面选硬币 B, 反面选硬币 C; 然后掷选出的硬币, 掷硬币的结果, 出现正面记作 1, 出现反面记作 0: 独立地重复 n 次试验 (这里,n=10), 观测结果如下:

1,1,0,1,0,0,1,0,1,1

假设只能观测到掷硬币的结果, 不能观测掷硬币的过程. 问如何估计三硬币正面出现的概率, 即三硬币模型的参数.

1. 问题分析与相关知识点

1). EM 算法

EM 算法，用于含有隐变量的概率模型参数的极大似然或极大后验概率估计。每次迭代包括 E 步和 M 步。E 步：利用当前的参数估计计算隐变量的后验概率（期望）。M 步：利用 E 步计算的隐变量后验概率极大化完整数据的对数似然的新的参数估计，直到收敛。称为期望极大算法（expectation maximization algorithm）。

2). 三硬币模型

我们用数学公式表达的三硬币模型为：

$$\begin{aligned} p(y|\theta) &= \sum_z P(y, z|\theta) = \sum_z P(z|\theta)P(y|z, \theta) \\ &= \pi p^y (1 = p)^{1-y} + (1 - \pi)q^y(1 - q)^{1-y} \end{aligned} \tag{1}$$

其中，y 是观测变量，表示一次实验的结果是 1 或 0；z 是隐变量，表示未观测到的掷硬币 A 的结果；$\theta = (\pi, p, q)$ 是模型参数，这一模型是以上数据的生成模型。（随机变量 y 数据可观测，z 数据不可观测。）

将观测数据表示为 $Y = (Y_1, Y_2, \cdots, Y_n)^T$，未观测数据表示为 $Z = (Z_1, Z_2, \cdots, Z_n)^T$。观测数据的对数似然函数：

$$\begin{aligned} L &= \log P(Y|\theta) \\ &= \log \prod_{j=1}^{n} [\pi p_j^y (1 = p)^{1-y_j} + (1 - \pi)q_j^y(1 - q)^{1-y_j}] \end{aligned} \tag{2}$$

考虑求模型参数 $\theta = (\pi, p, q)$ 的极大似然估计，即：

$$\hat{\theta} = \arg\max_{\theta} \log P(Y|\theta) \tag{3}$$

由于隐变量的存在，无法直接求解，我们通过 EM 算法迭代求解。

1. 选取参数初值，$\theta^{(0)} = (\pi^{(0)}, p^{(0)}, q^{(0)})$，然后通过 E 步和 M 步迭代计算参数估计只，直至收敛。

2. 第 i 次的参数估计值为 $\theta^{(i)} = (\pi^{(i)}, p^{(i)}, q^{(i)})$。

3. EM 算法第 i+1 次迭代如下：

   (a) E 步：计算在参数 $\pi^{(i)}, p^{(i)}, q^{(i)}$ 下，观测数据 y_j 来自掷硬币 B 的概率。

$$\mu_j^{(i+1)} = \frac{\pi^{(i)}(p^{(i)})^{y_j}(1-p^{(i)})^{1-y_j}}{\pi^{(i)}(p^{(i)})^{y_j}(1-p^{(i)})^{1-y_j} + (1-\pi^{(i)})(q^{(i)})^{yj}(1-q^{(i)})^{1-y_j}} \tag{4}$$

   (b) M 步：计算模型参数的新估计值。

$$\pi^{(i+1)} = \frac{1}{n}\sum_{j=1}^{n}\mu_j^{(i+1)} \tag{5}$$

$$p^{(i+1)} = \frac{\sum_{j=1}^{n}\mu_j^{(i+1)} y_j}{\sum_{j=1}^{n}\mu_j^{(i+1)}} \tag{6}$$

$$q^{(i+1)} = \frac{\sum_{j=1}^{n}(1-\mu_j^{(i+1)}) y_j}{\sum_{j=1}^{n}(1-\mu_j^{(i+1)})} \tag{7}$$

2. 选择初值解答

根据题目的要求，选择不同初值进行实验。初值取：$\pi^{(0)} = 0.46, p^{(0)} = 0.55, q^{(0)} = 0.67$。

第一轮： 由公式（4），有:

$y_j = 1$ 时,

$$\begin{aligned}\mu_j^{(1)} &= \frac{0.46 \times 0.55}{0.46 \times 0.55 + (1-0.46) \times 0.67} \\ &= 0.4115\end{aligned}$$

$y_j = 0$ 时,

$$\begin{aligned}\mu_j^{(1)} &= \frac{0.46 \times (1-0.55)}{0.46 \times (1-0.55) + (1-0.46) \times (1-0.67)} \\ &= 0.5346\end{aligned}$$

由公式（5）（6）（7）, 有：

$$\pi^{(1)} = 0.4619, p^{(1)} = 0.5346, q^{(1)} = 0.6561$$

第二轮： 由公式（4），有,

$y_j = 1$ 时,

$$\mu_j^{(2)} = 0.4115$$

$y_j = 0$ 时,

$$\mu_j^{(2)} = 0.5346$$

继续迭代，得，

$$\pi^{(2)} = 0.4619, p^{(2)} = 0.5346, q^{(2)} = 0.6561$$

得到模型参数的极大似然估计：

$$\hat{\pi} = 0.4619, \hat{p} = 0.5346, \hat{q} = 0.6561.$$

3. 代码验证

代码实现思路：设置两层循环，外层循环进行 E 步和 M 步的迭代，内层循环在 E 步对观测数据遍历计算 μ_j。代码实现如下：

```
import numpy as np

Y = np.array([1,1,0,1,0,0,1,0,1,1])
def cal_em_3coins(Y,pi_0 =0.46 ,p_0= 0.55,q_0= 0.67,iter_max = 3):
    # 初值
    pi_i= pi_0
    p_i= p_0
    q_i= q_0
    # 迭代步骤
    n =  len(Y)
    # 存储j方便计算
    mu = np.zeros(n) # mu_j
    y_times_mu = np.zeros(n) # y_j*mu_j
    y_times_1_mu = np.zeros(n) # y_j*(1-mu_j)
    mu_1 = np.zeros(n) # (1-mu_j)

    for i in  range(iter_max):
        # E步 计算mu_j
        for j in  range(n):
            mu[j] = (pi_i*p_i**Y[j] * (1-p_i)**(1-Y[j]))/(pi_i*p_i**Y[j] * (1-p_i)**(1-Y[j])
                +(1-pi_i)*q_i**Y[j] * (1-q_i)**(1-Y[j]))
            y_times_mu[j] = mu[j]*Y[j]
            y_times_1_mu[j] = Y[j]*(1-mu[j])
            mu_1[j] = 1-mu[j]
        # M步 更新参数
        pi = (1/n)* sum(mu)
        p_i =  sum(y_times_mu)/ sum(mu)
        q_i =  sum(y_times_1_mu)/ sum(mu_1)

        print(f'第{i+1}轮迭代, y_j = 1,mu={mu[0]}.y_j=0,mu={mu[2]}')
        print(f'pi_{i+1}={pi_i},p_{i+1}={p_i},q_{i+1}= {q_i}')
    return pi_i, p_i, q_i

print('-----')
pi,p,q = cal_em_3coins(Y,0.5,0.5,0.5,2)
print('初值: pi= 0.5,p=0.5,q= 0.5')
print(f'pi= {pi},p={p},q= {q}')

print('-----')
pi,p,q = cal_em_3coins(Y,0.4,0.6,0.7,2)
print('初值: pi= 0.4,p=0.6,q= 0.7')
print(f'pi= {pi},p={p},q= {q}')

print('-----')
```

```
pi,p,q = cal_em_3coins(Y,0.46,0.55,0.67,2)
print('初值: pi= 0.4,p=0.6,q= 0.7')
print(f'pi= {pi},p={p},q= {q}')
print('-----')
```

输出结果:

```
-----
第1轮迭代, y_j = 1,mu=0.5.y_j=0,mu=0.5
pi_1=0.5,p_1=0.6,q_1= 0.6
第2轮迭代, y_j = 1,mu=0.5.y_j=0,mu=0.5
pi_2=0.5,p_2=0.6,q_2= 0.6
初值: pi= 0.5,p=0.5,q= 0.5
pi= 0.5,p=0.6,q= 0.6
-----
第1轮迭代, y_j = 1,mu=0.36363636363636365.y_j=0,mu=0.47058823529411764
pi_1=0.4064171122994653,p_1=0.5368421052631579,q_1= 0.6432432432432431
第2轮迭代, y_j = 1,mu=0.3636363636363638.y_j=0,mu=0.47058823529411764
pi_2=0.40641711229946537,p_2=0.536842105263158,q_2= 0.6432432432432431
初值: pi= 0.4,p=0.6,q= 0.7
pi= 0.40641711229946537,p=0.536842105263158,q= 0.6432432432432431
-----
第1轮迭代, y_j = 1,mu=0.41151594014313597.y_j=0,mu=0.5373831775700935
pi_1=0.461862835113919,p_1=0.5345950037850112,q_1= 0.6561346417857326
第2轮迭代, y_j = 1,mu=0.4115159401431359.y_j=0,mu=0.5373831775700936
pi_2=0.461862835113919,p_2=0.5345950037850112,q_2= 0.6561346417857326
初值: pi= 0.4,p=0.6,q= 0.7
pi= 0.461862835113919,p=0.5345950037850112,q= 0.6561346417857326
-----
```

4. 总结

可以发现 EM 算法与初值的选择有关，选择不同初值得到的参数估计值不同。

二、 已知观测数据 -67,-48,6,8,14,16,23,24,28,29,41,49,56,60,75。试估计两个分量的高斯混合模型的 5 个参数。

1. 问题分析与相关知识点

1). 高斯混合模型

高斯混合模型（Gaussian mixture model）是指具有如下概率分布的模型:

$$P(y|\theta) = \sum_{k=1}^{K} \alpha_k \phi(y|\theta_k) \tag{8}$$

其中，α_k 是系数，$\alpha_k \geq 0$ $\sum_{k=1}^{K} \alpha_k = 1$；$\phi(y|\theta_k)$ 是高斯分布密度，$\theta_k = (\mu_k, \sigma_k^2)$

$$\phi(y|\theta_k) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{(y-\mu_k)^2}{2\sigma_k^2}\right) \tag{9}$$

称为第 k 个分模型。

假设观测数据 $Y=(Y_1,Y_2,\cdots,Y_n)^T$ 由高斯混合模型生成，

$$P(y|\theta)=\sum_{k=1}^{K}\alpha_k\phi(y|\theta_k)$$

其中，模型参数 $\theta=(\alpha_1,\alpha_2,\cdots,\alpha_k;\theta_1,\theta_2,\cdots,\theta_k)$。

2)． 高斯混合模型参数估计 EM 算法：

1. 取参数初值开始迭代：

2. E 步：根据当前的参数，计算分模型 k 对数据 y_j 的响应度 $\hat{\gamma}_{jk}$:

$$\begin{aligned}\hat{\gamma}_{jk}&=E(\gamma_{jk}|y,\theta)=P(\gamma_{jk}=1|y,\theta)\\&=\frac{P(\gamma_{jk}=1,y_j|\theta)}{\sum_{k=1}^{K}P(\gamma_{jk}=1,y_j|\theta)}\\&=\frac{\alpha_k\phi(y_j|\theta_k)}{\sum_{k=1}^{K}\alpha_k\phi(y_j|\theta_k)}\\j&=1,2,\cdots,N;k=1,2,\cdots,K\end{aligned}\tag{10}$$

3. M 步：求新一轮迭代参数估计值。

$$\hat{\mu}_k=\frac{\sum_{j=1}^{N}\hat{\gamma}_{jk}y_j}{\sum_{j=1}^{N}\hat{\gamma}_{jk}},k=1,2,\cdots,K\tag{11}$$

$$\hat{\sigma}_k^2=\frac{\sum_{j=1}^{N}\hat{\gamma}_{jk}(y_j-\mu_k)^2}{\sum_{j=1}^{N}\hat{\gamma}_{jk}},k=1,2,\cdots,K\tag{12}$$

$$\hat{\alpha}_k=\frac{n_k}{N}=\frac{\sum_{j=1}^{N}\hat{\gamma}_{jk}}{N},k=1,2,\cdots,K\tag{13}$$

4. 重复 (2)E 步、(3)M 步，直到收敛。

3)． 代入问题

问题要求估计两个分量的高斯混合模型的 5 个参数，共有 15 个数据，即 $N=15,K=2$，由 $\sum_{k=1}^{K}\alpha_k=1$ 可得，若第一个模型系数为 α_1，则第二个模型系数为 $\alpha_2=1-\alpha_1$。故需要求解的五个参数为，$\theta=(\alpha_1,\sigma_1^2,\sigma_2^2,\mu_1,\mu_2)$。

2. 代码实现

代码实现思路：与三硬币模型类似，外层循环进行迭代，E 步内层循环遍历数据计算 $\hat{\gamma}_{jk}$，M 步更新参数估计值，直至收敛。代码实现如下：

```
import numpy as np
import math
Y = np.array([-67, -48, 6, 8, 14, 16, 23, 24, 28, 29, 41, 49, 56, 60, 75])
# 辅助函数 高斯分布密度
def cal_gaussian_phi_k(y, sigma2_k, mu_k):
    res = 1 / (math.sqrt(2 * math.pi * sigma2_k))
    res = res * math.exp(-((y - mu_k) * (y - mu_k)) / (2 * sigma2_k))
    return res
# em估计gmm参数
def gmm_em(Y, alpha_1, sigma2_1, sigma2_2, mu_1, mu_2, iter_max=5):
```

```
    print(f'初值: alpha_1 = {alpha_1},sigma2_1 = {sigma2_1},sigma2_2 = {sigma2_2},mu_1 = {mu_1
        },mu_2 = {mu_2}')
    N =  len(Y)
    # 参数迭代值
    alpha_1i = alpha_1
    sigma2_1i = sigma2_1  # 是sigma^2
    sigma2_2i = sigma2_2
    mu_1i = mu_1
    mu_2i = mu_2
    # 更新参数时便于计算j1和j2
    gamma_j1_N = np.ones(N)
    gamma_j2_N = np.ones(N)
    gamma_j2_yj_N = np.ones(N)
    gamma_j1_yj_N = np.ones(N)
    gamma_yj_mu1_2_N = np.ones(N)
    gamma_yj_mu2_2_N = np.ones(N)
    for i in  range(iter_max):
        # E步
        for j in  range(N):
            # 循环计算gamma_jk
            yj = Y[j]
            gamma_j1_N[j] = (alpha_1i * cal_gaussian_phi_k(yj, sigma2_1i, mu_1i))/((alpha_1i *
                 cal_gaussian_phi_k(yj, sigma2_1i, mu_1i)) +
                            (1 - alpha_1i) * cal_gaussian_phi_k(yj,sigma2_2i,mu_2i))
            gamma_j2_N[j] = ((1 - alpha_1i) * cal_gaussian_phi_k(yj, sigma2_2i, mu_2i))/((
                alpha_1i * cal_gaussian_phi_k(yj, sigma2_1i, mu_1i)) +
                            (1 - alpha_1i) * cal_gaussian_phi_k(yj,sigma2_2i,mu_2i))
            gamma_j1_yj_N[j] = yj * gamma_j1_N[j]
            gamma_j2_yj_N[j] = yj * gamma_j2_N[j]
            gamma_yj_mu1_2_N[j] = gamma_j1_N[j] * ((yj - mu_1i) ** 2)
            gamma_yj_mu2_2_N[j] = gamma_j1_N[j] * ((yj - mu_2i) ** 2)
        # M步 更新参数
        mu_1i =  sum(gamma_j1_yj_N) /  sum(gamma_j1_N)
        mu_2i =  sum(gamma_j2_yj_N) /  sum(gamma_j2_N)
        sigma2_1i =  sum(gamma_yj_mu1_2_N) /  sum(gamma_j1_N)
        sigma2_2i =  sum(gamma_yj_mu2_2_N) /  sum(gamma_j2_N)
        alpha_1i =  sum(gamma_j1_N) / N
        # 打印
        print(f'第{i + 1}轮: mu_1i = {mu_1i},mu_2i = {mu_2i},sigma2_1i = {sigma2_1i},sigma2_2i
             = {sigma2_2i},alpha_1i = {alpha_1i}')
    return alpha_1i, sigma2_1i, sigma2_2i, mu_1i, mu_2i
alpha_1i, sigma2_1i, sigma2_2i, mu_1i, mu_2i = gmm_em(Y, 0.5, 1000, 1000, 100, 100, 100)
print(f'mu_1i = {mu_1i},mu_2i = {mu_2i},\sigma2_1i = {sigma2_1i},sigma2_2i = {sigma2_2i},
    alpha_1i = {alpha_1i}')
print('----------')
alpha_1i, sigma2_1i, sigma2_2i, mu_1i, mu_2i = gmm_em(Y, 0.4, 100, 100, 10, 10, 1000)
print(f'mu_1i = {mu_1i},mu_2i = {mu_2i},sigma2_1i = {sigma2_1i},sigma2_2i = {sigma2_2i},
    alpha_1i = {alpha_1i}')
```

输出结果:

```
初值: alpha_1 = 0.5,sigma2_1 = 1000,sigma2_2 = 1000,mu_1 = 100,mu_2 = 100
......
```

```
mu_1i = 20.933333333333334,mu_2i = 20.933333333333334,sigma2_1i = 1329.6622222222222,sigma2_2i
     = 1329.6622222222222,alpha_1i = 0.5
----------
初值: alpha_1 = 0.4,sigma2_1 = 100,sigma2_2 = 100,mu_1 = 10,mu_2 = 10
......
mu_1i = -58.23518803877892,mu_2i = 31.63300232231747,sigma2_1i = 89.70949872756016,sigma2_2i =
     1103.6395226368115,alpha_1i = 0.11905957988017955
```

3. 总结

初值为 $\alpha_1 = 0.5, \sigma_1^2 = 1000, \sigma_2^2 = 1000, \mu_1 = 100, \mu_2 = 100$ 时，估计的高斯混合模型的五个参数为：

$$\alpha_1 = 0.5, \sigma_1^2 = 1329.6622, \sigma_2^2 = 1329.6622, \mu_1 = 20.9333, \mu_2 = 20.9333$$

初值为 $\alpha_1 = 0.4, \sigma_1^2 = 100, \sigma_2^2 = 100, \mu_1 = 10, \mu_2 = 10$ 时，估计的高斯混合模型的五个参数为：

$$\alpha_1 = 0.1191, \sigma_1^2 = 89.7095, \sigma_2^2 = 1103.6395, \mu_1 = -58.2352, \mu_2 = 31.6330$$

选择不同初值，参数估计的结果不同。在计算时，也有一些初值会导致错误。