Biostatistics Program1

Task 1

目录

1 Problem Statement

已知袋中有 A 种硬币 a 枚，B 种硬币 b 枚，两种硬币除了抛掷出现正反面的概率不同以外没有其他区别，随机从中取出 1 枚，抛掷 10 次，有放回的取出 5 次，记录每次抛掷的结果得到如上数据，下求从袋中取出一枚硬币，该硬币为 i 种硬币的概率以及两种硬币各自抛掷正反面的概率的极大似然估计。其中，i=a，b.

2 Notation

符号说明

符号	说明
x_i	第 i 次取出硬币时，抛掷十次中正面朝上的次数。
π_k	从袋中随机取出一枚硬币，该硬币为 i 种硬币的概率，k = a, b.
θ_k	抛掷一枚 i 种硬币时，正面朝上的概率，k = a, b.
z_i	第 i 次取出的硬币种类，当硬币为 A 种时，该值为 0，否则为 1。

注：1. 当符号没有下标时，表示相应的向量表述。

2. 当符号为大写字母时表示相应随机变量，小写时表示相应的样本数据。

3. 带上标符号表示迭代次数为上标时，该符号代表的参数值。

3 EM derivation

目标:

$$\begin{pmatrix}\hat{\theta}\\ \hat{\pi}\end{pmatrix} = \underset{\theta,\pi}{\operatorname{argmax}} \log P(X;\theta,\pi) = \underset{\theta,\pi}{\operatorname{argmax}} \log(\prod_i P(X_i = x_i;\theta,\pi)$$

$$= \underset{\theta,\pi}{\operatorname{argmax}} \sum_i \log P(x_i;\theta,\pi) = \underset{\theta,\pi}{\operatorname{argmax}} \sum_i \log \sum_{z_i} P(X_i = x_i, Z_i = z_i;\theta,\pi)$$

步骤:

①随机化或根据专业知识等先验选取相应初始参数，记 ite=0（迭代次数），转步二。

②E-step：

$$E_z(\ln P(X,Z;\theta,\pi)) = \sum_Z Ln(P(X,Z;\theta,\pi))P(Z \mid X;\theta^{(ite)},\pi^{(ite)})$$

$$= \sum_Z \sum_i Ln(P(X_i,Z;\theta,\pi))P(Z \mid X_i;\theta^{(ite)},\pi^{(ite)})$$

已知：

$$P(X_i = x_i \mid Z_i = z_i;\theta,\pi) = C_n^{x_i}(\theta_A^{(1-z_i)}\theta_B^{z_i})^{x_i}(1-\theta_A^{(1-z_i)}\theta_B^{z_i})^{10-x_i}$$

$$P(Z_i = 0) = \pi_A \qquad P(Z_i = 1) = \pi_B$$

$$P(X_i = x_i, Z_i = z_i;\theta,\pi) = P(Z_i = z_i;\theta,\pi)P(X_i = x_i \mid Z_i = z_i;\theta,\pi)$$

$$P(Z_i = z_i \mid X_i = x_i;\theta,\pi) = \frac{P(X_i = x_i, Z_i = z_i;\theta,\pi)}{P(X_i = x_i;\theta,\pi)}$$

$$= \frac{P(X_i = x_i, Z_i = z_i;\theta,\pi)}{P(X_i = x_i, Z_i = 0;\theta,\pi) + P(X_i = x_i, Z_i = 1;\theta,\pi)}$$

将已知代入 $E_z(\ln P(X,Z;\theta,\pi))$，转步 3。

③M-step

$$\begin{pmatrix}\theta^{(ite+1)}\\ \pi^{(ite+1)}\end{pmatrix} = \arg\max_{\theta,\pi} E_{z;\theta^{(ite)},\pi^{(ite)}}(\ln P(X,Z;\theta,\pi))$$

$$= \arg\max_{\theta,\pi} \sum_i (\ln(\pi_A C_n^{x_i}\theta_A^{x_i}(1-\theta_A)^{10-x_i})$$

$$\frac{\pi_A^{(ite)}C_n^{x_i}\theta_A^{(ite)x_i}(1-\theta_A^{(ite)})^{10-x_i}}{\pi_A^{(ite)}C_n^{x_i}\theta_A^{(ite)x_i}(1-\theta_A^{(ite)})^{10-x_i} + \pi_B^{(ite)}C_n^{x_i}\theta_B^{(ite)x_i}(1-\theta_B^{(ite)})^{10-x_i}}$$

$$+ \ln(\pi_B C_n^{x_i}\theta_B^{x_i}(1-\theta_B)^{10-x_i})$$

$$\frac{\pi_B^{(ite)}C_n^{x_i}\theta_B^{(ite)x_i}(1-\theta_B^{(ite)})^{10-x_i}}{\pi_A^{(ite)}C_n^{x_i}\theta_A^{(ite)x_i}(1-\theta_A^{(ite)})^{10-x_i} + \pi_B^{(ite)}C_n^{x_i}\theta_B^{(ite)x_i}(1-\theta_B^{(ite)})^{10-x_i}})$$

对上式化简,可将 π_B 用 π_A 表示，并求极大值：

$$\frac{\partial E_z(\ln P(X,Z;\theta,\pi))}{\partial \pi_A} = 0$$

$$\frac{\partial E_z(\ln P(X,Z;\theta,\pi))}{\partial \theta_A} = 0$$

$$\frac{\partial E_z(\ln P(X,Z;\theta,\pi))}{\partial \theta_B} = 0$$

记 $\frac{\pi_A^{(ite)} C_n^{x_i} \theta_A^{(ite)^{x_i}} (1-\theta_A^{(ite)})^{10-x_i}}{\pi_A^{(ite)} C_n^{x_i} \theta_A^{(ite)^{x_i}} (1-\theta_A^{(ite)})^{10-x_i} + \pi_B^{(ite)} C_n^{x_i} \theta_B^{(ite)^{x_i}} (1-\theta_B^{(ite)})^{10-x_i}}$ 为 $\gamma_{iA}^{(\text{ite})}$，

记 $\frac{\pi_B^{(ite)} C_n^{x_i} \theta_B^{(ite)^{x_i}} (1-\theta_B^{(ite)})^{10-x_i}}{\pi_A^{(ite)} C_n^{x_i} \theta_A^{(ite)^{x_i}} (1-\theta_A^{(ite)})^{10-x_i} + \pi_B^{(ite)} C_n^{x_i} \theta_B^{(ite)^{x_i}} (1-\theta_B^{(ite)})^{10-x_i}}$ 为 $\gamma_{iB}^{(ite)}$

$$\pi_A^{(ite+1)} = \frac{\sum_i \gamma_{iA}^{(\text{ite})}}{\sum_i \gamma_{iA}^{(\text{ite})} + \sum_i \gamma_{iB}^{(\text{ite})}}$$

$$\pi_B^{(ite+1)} = \frac{\sum_i \gamma_{iB}^{(\text{ite})}}{\sum_i \gamma_{iA}^{(\text{ite})} + \sum_i \gamma_{iB}^{(\text{ite})}}$$

$$\theta_A^{(ite+1)} = \frac{\sum_{\text{i}} \gamma_{iA}^{(\text{ite})} x_i}{\sum_{\text{i}} \gamma_{iA}^{(\text{ite})} x_i + \sum_{\text{i}} \gamma_{iA}^{(\text{ite})} (10-x_i)}$$

$$\theta_B^{(ite+1)} = \frac{\sum_{\text{i}} \gamma_{iB}^{(\text{ite})} x_i}{\sum_{\text{i}} \gamma_{iB}^{(\text{ite})} x_i + \sum_{\text{i}} \gamma_{iA}^{(\text{ite})} (10-x_i)}$$

计算出相应的参数，ite=ite+1，转步四。

④计算是否满足收敛，若是输出参数，若否转步 2。

4 Simulating data

实验 1：

在 r 中设定参数：

```
thetaa=0.4
thetab=0.4
pia=0.5
pib=1-pia
#generate data
set.seed(1234)
```

生成相应的数据 X=（2 4 4 4 6 2 4）

实验 2：

在 r 中设定参数：

```
thetaa=0.4
thetab=0.8
pia=0.6
pib=1-pia
#generate data
set.seed(1234)
```

生成相应的数据 X=（9 2 4 4 4 6 5）

5 Result

5.1 运行实验

经过多步迭代分别得到如下结果：

实验一：

初值：

π_A：0.5　　π_B：0.5　　θ_A：0.4999　　θ_B：0.5

结果：

π_A：0.5001285　　π_B：0.4998715　　θ_A：0.3714267　　θ_B：0.3714304

实验二：

初值：

π_A：0.5　　π_B：0.5　　θ_A：0.4999　　θ_B：0.5

结果：

π_A：0.8595305　　π_B：0.1404695　　θ_A：0.4189718　　θ_B：0.8941107

5.2 用于分析的实验

实验 1：分析初值设置

```
thetaa=0.4
thetab=0.8
pia=0.6
pib=1-pia
#generate data
set.seed(1234)
num_data=7#数据总个数
```

（1）初值：

π_A：0.5　　π_B：0.5　　θ_A：0.5　　θ_B：0.5

结果：

π_A：0.5　　π_B：0.5　　θ_A：0.4857143　　θ_B：0.4857143

（2）初值：

π_A：0.5　　π_B：0.5　　θ_A：0.49　　θ_B：0.5

结果：

π_A：0.8595303　　π_B：0.1404697　　θ_A：0.4189717　　θ_B：0.8941104

实验 2：分析初值设置

```
thetaa=0.4
thetab=0.8
pia=0.6
pib=1-pia
#generate data
set.seed(1234)
```

（1）number_data=7

初值：

π_A：0.5　　π_B：0.5　　θ_A：0.49　　θ_B：0.5

结果：

π_A：0.8595303　　π_B：0.1404697　　θ_A：0.4189717　　θ_B：0.8941104

（2）number_data=2000

初值：

π_A：0.5　　π_B：0.5　　θ_A：0.5　　θ_B：0.5

结果：

π_A：0.6021486　　π_B：0.3978514　　θ_A：0.3952088　　θ_B：0.7948334

6 Analysis

6.1 初值设置

我们通过数次实验发现，当我们令$\pi_A=\pi_B$且$\theta_A=\theta_B$时，结果将会始终保持$\pi_A=\pi_B$且$\theta_A=\theta_B$，我们分析 EM 算法的迭代公式发现由于π_i的迭代基于θ_A与θ_B，如果两者相同，π_k将不会发生变化，同样，当π_k固定不变，θ_A与θ_B也不会发生

变化。而只需将π_A，π_B设为不同的值，或将θ_A，θ_B设为不同的值，即使差别很小，结果也往往能迭代到更为接近真实情况的值。我们可以认为 EM 算法利用了先验知识计算后验期望的最大似然，而将$\pi_A=\pi_B$且$\theta_A=\theta_B$，则可认为我们将 A、B 视为同一类，先验即两者来自同一个总体，因此后验期望也会认为他们来自同一总体。

6.2 迭代结果

我们可以看到上述部分结果并不精准，但如果我们增大样本容量，我们可以发现迭代误差会变得很小，即使我们的先验偏差较大，实验结果也能收敛到很接近预设初值的结果，EM 算法是利用已知的样本数据，在先验的基础下求解参数的方法，因此即使先验偏差较大，但只要样本数据足够多且准确，我们就可以得到较好的迭代结果，但前提是先验不陷入某些 EM 算法的局限，如 6.1 中分析的 EM 算法的先验为样本均来自同一总体，则后验也将认为其来自同一总体。

6.3 后续分析

由于部分分析与 Task2 重合，因此我们将这部分分析放到 Task2 中。

7 Code

```
#task 1


#set parameter
thetaa=0.4
thetab=0.8
pia=0.6
pib=1-pia
#generate data
set.seed(1234)
num_data=2000#数据总个数
num_data1=rbinom(1,num_data,pia)
set.seed(1234)
data1=rbinom(num_data1, 10, thetaa)
set.seed(1234)
data2=rbinom(num_data-num_data1,10,thetab)
data=c(data2,data1)
```

```
data_0=rep(10,time=num_data)



#em
Expect=function(param,data)#条件期望
{
   params=param
   a=1:params[1]
   b=1:params[1]
   for (i in 1:params[1]) {
      di=data[i]
      k=(params[2]*choose(10,di)*((params[3])^(di)*((1-params[3])^(10-di))))
      l=((1-params[2])*choose(10,di)*((params[4])^(di))*((1-params[4])^(10-di)))
      a[i]=k/(k+l)
      b[i]=l/(k+l)

   }
   Ec=c(a,b)
   return(c(a,b))
}

EM=function(param,data)
{
   #param 初始参数，data 数据   param[num_data,pia,thetaa,thetab]
   num_cyc=10000
   parameter=matrix(c(1:num_cyc*4),num_cyc,4,1)
   #paramter=data.frame(paramter_matrix,colnames = c("num_data","pia","thetaa","thetab"))
   Ec=Expect(param,data)
   Ea=Ec[1:param[1]]
   Eb=Ec[(param[1]+1):(2*(param[1]))]
   parameter[1,1]=num_data
   parameter[1,2]=sum(Ea)/(sum(Ea)+sum(Eb))
   parameter[1,3]=data%*%Ea/(data%*%Ea+(data_0-data)%*%Ea)
   parameter[1,4]=data%*%Eb/(data%*%Eb+(data_0-data)%*%Eb)
   q=0
   error=0.000001
   for (i in 2:num_cyc) {
```

```
        Ec=Expect(parameter[i-1,],data)
        Ea=Ec[1:param[1]]
        Eb=Ec[(param[1]+1):(2*(param[1]))]
        parameter[i,1]=num_data
        parameter[i,2]=sum(Ea)/(sum(Ea)+sum(Eb))
        parameter[i,3]=data%*%Ea/(data%*%Ea+(data_0-data)%*%Ea)
        parameter[i,4]=data%*%Eb/(data%*%Eb+(data_0-data)%*%Eb)
        q=q+1

        if(parameter[i,2]-parameter[i-1,2]<error)
        {
           return(parameter[i,])
        }
    }
}

#main test
param=c(num_data,0.5,0.4,0.5)
parameter_fin=EM(param,data)
print(parameter_fin)
```