EM算法参数估计

陈煜磊 ZY2103502

1 问题

一个袋子中三种硬币的混合比例为：s_1, s_2 与 $1-s_1-s_2$ $(0 \leq s_i \leq 1)$，三种硬币掷出正面的概率分别为：p, q, r 。指定系数 $s_1=0.2, s_2=0.3, p=0.1, q=0.9, r=0.5$，生成 N 个投掷硬币的结果（由 01 构成的序列，其中 1 为正面，0 为反面），利用最大期望算法（Expectation-maximization algorithm，EM 算法）来对参数进行估计并与预先假定的参数进行比较。

2 原理

EM 算法由 Dempster 等[1]在 1977 年提出，是在概率模型中寻找参数最大似然估计或者最大后验估计的算法，其中概率模型依赖于无法观测的隐变量。算法经过两个步骤交替进行计算：

1. 初始化分布参数
2. 重复直到收敛：
   1. E步骤：根据参数的假设值，给出未知变量的期望估计，应用于缺失值。
   2. M步骤：根据未知变量的估计值，给出当前的参数的极大似然估计。

Wu[2] 证明了 EM 算法是收敛的，但不能保证收敛到极大值点，因此算法中初值选择很重要。

设 y_j 是第 j 次实验抛硬币的观测数据，$z_i=(\alpha_i,\beta_i)$ 为第 i 次迭代中的隐变量，其中 α_i 表示摸到硬币 A 的概率，β_i 表示摸到硬币 B 的概率，模型参数 $\theta=(s_1,s_2,p,q,r)$，第 i 次迭代时参数估计为 $\theta^{(i)}=(s_1^{(i)},s_2^{(i)},p^{(i)},q^{(i)},r^{(i)})$ 。观测数据的似然函数：

$$P(Y|\theta)=\prod_{j=1}^{n}[s_1p^{y_j}(1-p)^{1-y_j}+s_2q^{y_j}(1-q)^{1-y_j}+(1-s_1-s_2)r^{y_j}(1-r)^{1-y_j}] \tag{1}$$

观测数据 Y 关于当前参数估计 $\theta^{(i)}$ 的对数似然函数为：

$$L(\theta)=\log P(Y|\theta)=\log\left(\sum_Z P\left(Y|Z,\theta\right)P\left(Z|\theta\right)\right) \tag{2}$$

我们希望迭代参数能使得 $L(\theta)$ 极大化，取两次迭代的差值：

$$\begin{aligned}
L(\theta)-L\left(\theta^{(i)}\right) &= \log\left(\sum_Z P\left(Z\mid Y,\theta^{(i)}\right)\frac{P(Y\mid Z,\theta)P(Z\mid\theta)}{P\left(Z\mid Y,\theta^{(i)}\right)}\right)-\log P\left(Y\mid\theta^{(i)}\right)\\
&\geqslant \sum_Z P\left(Z\mid Y,\theta^{(i)}\right)\log\frac{P(Y\mid Z,\theta)P(Z\mid\theta)}{P\left(Z\mid Y,\theta^{(i)}\right)}-\log P\left(Y\mid\theta^{(i)}\right)\\
&= \sum_Z P\left(Z\mid Y,\theta^{(i)}\right)\log\frac{P(Y\mid Z,\theta)P(Z\mid\theta)}{P\left(Z\mid Y,\theta^{(i)}\right)P\left(Y\mid\theta^{(i)}\right)}
\end{aligned} \tag{3}$$

则迭代过程可表示为：

$$\begin{aligned}
\theta^{(i+1)} &= \arg\max_\theta\left(L\left(\theta^{(i)}\right)+\sum_Z P\left(Z\mid Y,\theta^{(i)}\right)\log\frac{P(Y\mid Z,\theta)P(Z\mid\theta)}{P\left(Z\mid Y,\theta^{(i)}\right)P\left(Y\mid\theta^{(i)}\right)}\right)\\
&= \arg\max_\theta\left(\sum_Z P\left(Z\mid Y,\theta^{(i)}\right)\log\left(P(Y\mid Z,\theta)P(Z\mid\theta)\right)\right)\\
&= \arg\max_\theta\left(\sum_Z P\left(Z\mid Y,\theta^{(i)}\right)\log P(Y,Z\mid\theta)\right)
\end{aligned} \tag{4}$$

定义 Q 函数：

$$Q\left(\theta,\theta^{(i)}\right)\hat{=}\sum_Z P\left(Z\mid Y,\theta^{(i)}\right)\log P(Y,Z\mid\theta) \tag{5}$$

则问题转化为 $\arg\max_\theta Q(\theta,\theta_i)$. 代入本问题，得：

$$Q(\theta,\theta_i)=\sum_{j}^{n}\{\alpha_i^{(i+1)}[\log s_1+y_j\log p+(1-y_i)\log(1-p)]+\beta_i^{(i+1)}[\log s_2+y_j\log q+(1-y_j)\log(1-q)]+(1-\alpha_i^{(i}$$

$j=1$

2.1 E 步骤

已知第 i 次迭代得参数估计为 $\theta^{(i)}$ ，在该参数下观测数据 y_j 来自硬币 A 的概率为：

$$\alpha_j^{(i+1)}=\frac{s_1^{(i)}(p^{(i)})^{y_j}(1-p^{(i)})^{1-y_j}}{s_1^{(i)}(p^{(i)})^{y_j}(1-p^{(i)})^{1-y_j}+s_2^{(i)}(q^{(i)})^{y_j}(1-q^{(i)})^{1-y_j}+(1-s_1^{(i)}-s_2^{(i)})(r^{(i)})^{y_j}(1-r^{(i)})^{1-y_j}} \tag{7}$$

来自硬币 B 的概率为：

$$\beta_j^{(i+1)}=\frac{s_2^{(i)}(q^{(i)})^{y_j}(1-q^{(i)})^{1-y_j}}{s_1^{(i)}(p^{(i)})^{y_j}(1-p^{(i)})^{1-y_j}+s_2^{(i)}(q^{(i)})^{y_j}(1-q^{(i)})^{1-y_j}+(1-s_1^{(i)}-s_2^{(i)})(r^{(i)})^{y_j}(1-r^{(i)})^{1-y_j}} \tag{8}$$

2.2 M 步骤

要极大化 $Q(\theta,\theta_i)$，需对参数求偏导。对 s_1,s_2 ：

$$\frac{\partial Q}{\partial s_1}=\sum_{j=1}^{n}[\frac{\alpha_j^{(i+1)}}{s_1}-\frac{1-\alpha_j^{(i+1)}-\beta_j^{(i+1)}}{1-s_1-s_2}]=0 \tag{9}$$

$$\frac{\partial Q}{\partial s_2}=\sum_{j=1}^{n}[\frac{\beta_j^{(i+1)}}{s_2}-\frac{1-\alpha_j^{(i+1)}-\beta_j^{(i+1)}}{1-s_1-s_2}]=0 \tag{10}$$

解得$s_1=\frac{1}{n}\sum_{j=1}^{n}\alpha_j^{(i+1)}, s_2=\frac{1}{n}\sum_{j=1}^{n}\beta_j^{(i+1)}$

再对 p,q,r 求偏导，由：

$$\frac{\partial Q}{\partial p}=\sum_{j=1}^{n}\alpha_j^{(i+1)}[\frac{y_j}{p}-\frac{1-y_j}{1-p}]=0 \tag{11}$$

得 $p^{(i+1)}=\frac{\sum_{j=1}^{n}\alpha_j^{(i+1)}y_j}{\sum_{j=1}^{n}\alpha_j^{(i+1)}}$. 同理有 $q^{(i+1)}=\frac{\sum_{j=1}^{n}\beta_j^{(i+1)}y_j}{\sum_{j=1}^{n}\beta_j^{(i+1)}}, r^{(i+1)}=\frac{\sum_{j=1}^{n}(1-\alpha_j^{(i+1)}-\beta_j^{(i+1)})y_j}{\sum_{j=1}^{n}(1-\alpha_j^{(i+1)}-\beta_j^{(i+1)})}$

再由迭代得参数重复进行 E-M 步骤，直到达到最大迭代次数或参数收敛（即 $\left\|\theta^{(i+1)}-\theta^{(i)}\right\|<\varepsilon$）.

3 代码

首先根据参数生成 N 次投掷硬币的观测结果：

```
def data_gen(s1, s2, p, q, r, N):
    data = []
    for i in range(N):
        coin = random.random()
        if 0 <= coin < s1:
            side = np.random.binomial(1,p)
        elif s1 <= coin < s1 + s2:
            side = np.random.binomial(1,q)
        else:
            side = np.random.binomial(1,r)
        data.append(side)
    return data
```

再给定初始参数估计，观测数据和迭代终止条件，运行 EM 算法：

```
def EM(theta, e, y, max_epoch):
    s1 = theta[0]
    s2 = theta[1]
    p  = theta[2]
    q  = theta[3]
    r  = theta[4]
    N  = len(y)
    i = 0
    while(i < max_epoch and threshold >= e):
        # Expectation
        a = np.random.rand(N)
        b = np.random.rand(N)
        for j in range(N):
```

```
            a[j] = (s1*pow(p,y[j])*pow(1-p,1-y[j]))/(s1*pow(p,y[j])*pow(1-p,1-
y[j])+s2*pow(q,y[j])*pow(1-q,1-y[j])+(1-s1-s2)*pow(r,y[j])*pow(1-r,1-y[j]))
            b[j] = (s2*pow(q,y[j])*pow(1-q,1-y[j]))/(s1*pow(p,y[j])*pow(1-p,1-
y[j])+s2*pow(q,y[j])*pow(1-q,1-y[j])+(1-s1-s2)*pow(r,y[j])*pow(1-r,1-y[j]))
        # Maximization
        s1_next = 1/N * sum(a)
        s2_next = 1/N * sum(b)
        p_next = sum([a[j]*y[j] for j in range(N)]) / sum(a)
        q_next = sum([b[j]*y[j] for j in range(N)]) / sum(b)
        r_next = sum([(1-a[j]-b[j])*y[j] for j in range(N)]) / sum([(1-a[j]-b[j]) for j in
range(N)])
        # Threshold
        threshold = np.linalg.norm(np.array([s1-s1_next,s2-s2_next,p-p_next,q-q_next,r-
r_next]),ord = 2)
        s1 = s1_next
        s2 = s2_next
        p  = p_next
        q  = q_next
        r  = r_next
        i += 1
        print(i,[s1,s2,p,q,r])
    return s1,s2,p,q,r
```

4 实验结果

给定初值 $\theta^{(0)} = (0.4, 0.5, 0.2, 0.6, 0.8)$，取最大迭代次数 10，终止阈值 1×10^{-20}，得

```
1 [0.42512077294685996, 0.48309178743961356, 0.16363636363636366, 0.54, 0.7578947368421054]
2 [0.42512077294685996, 0.48309178743961356, 0.16363636363636364, 0.54, 0.7578947368421064]
3 [0.4251207729468601, 0.48309178743961356, 0.16363636363636355, 0.54, 0.7578947368421068]
4 [0.4251207729468601, 0.48309178743961356, 0.16363636363636358, 0.54, 0.7578947368421071]
5 [0.4251207729468601, 0.48309178743961356, 0.16363636363636355, 0.54, 0.7578947368421072]
6 [0.4251207729468601, 0.48309178743961356, 0.16363636363636355, 0.54, 0.7578947368421071]
7 [0.4251207729468601, 0.48309178743961356, 0.16363636363636355, 0.54, 0.7578947368421071]
```

与真值 $\theta = (0.2, 0.3, 0.1, 0.9, 0.5)$ 相去甚远。这是因为EM算法只能保证参数估计序列收敛到对数似然函数序列的稳定点，不能保证收敛到极大值点。

5 参考文献

[1] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” Journal of the Royal Statistical Society: Series B (Methodological), vol. 39, no. 1, pp. 1–22, 1977, doi: 10.1111/j.2517-6161.1977.tb01600.x.
[2] C. F. J. Wu, “On the Convergence Properties of the EM Algorithm,” The Annals of Statistics, vol. 11, no. 1, pp. 95–103, 1983, doi: 10.1214/aos/1176346060.
[3] 李航, 统计学习方法. 清华大学出版社, 2012.