模型-机器学习-聚类-期望最大化算法EM【hxy】

1. 模型名称

期望最大算法（Expectation Maximization Algorithm，EM）

2. 模型评价

2.1 优点

算法简单，稳定上升的步骤能非常可靠地找到“最优的收敛值 ”

2.2 缺点

对初始值敏感，EM算法需要初始化参数θ，而参数θ的选择直接影响收敛效率以及能否得到全局最优解

3. 基本算法

输入：观测变量数据Y，隐变量数据Z，联合分布$P(Y,Z|\theta)$，条件分布$P(Z|Y,\theta)$

输出：模型参数θ

1. 选择参数的初值$\theta^{(0)}$，开始迭代
2. E步：记$\theta^{(i)}$为第i次迭代参数θ的估计值，在$i+1$次迭代的E步，计算

   $$Q(\theta,\theta^{(i)})=E_z[logP(Y,Z|\theta)|Y,\theta^{(i)}]=\sum_z logP(Y,Z|\theta)P(Z|Y,\theta^{(i)})$$

   此处$P(Z|Y,\theta^{(i)})$是在给定观测数据Y和当前的参数估计$\theta^{(i)}$下隐变量数据z的条件概率分布
3. M步：求使$Q(\theta,\theta^{(i)})$极大化的θ，确定第$i+1$次迭代的参数的估计值$\theta^{(i+1)}$

   $$\theta^{(i+1)}=arg\max_\theta Q(\theta,\theta^{(i)})$$

4. 重复第2步和第3步，直到收敛

4. 实例

4.1 数据介绍

假设有两枚硬币A、B，以相同的概率随机选择一个硬币，进行如下的抛硬币实验：共做5次实验，每次实验独立的抛十次，结果如图中所示，例如某次实验产生了H、T、T、T、H、H、T、H、T、H，H代表正面朝上。

实习生忘了记录每次试验选择的是A还是B，我们无法观测实验数据中选择的硬币是哪个

4.2 实验目的

如何估计两个硬币正面出现的概率

4.3 求解步骤

1. 随机初始化

$$\theta_A = 0.6 \quad \theta_B = 0.5$$

2. **E步**

   以第一轮为例

$$P_A = \frac{0.6^5 \times 0.4^5}{(0.6^5 \times 0.4^5) + (0.5^5 \times 0.5^5)} = 0.45 \quad P_B = \frac{0.5^5 \times 0.5^5}{(0.6^5 \times 0.4^5) + (0.5^5 \times 0.5^5)} = 0.55$$

   得到如下表格

No	Coin A	Coin B
1	0.45	0.55
2	0.80	0.20
3	0.73	0.27
4	0.35	0.65
5	0.65	0.35

3. **M步**

a) 以第一轮的A为例

$$H: 0.45 \times 5 = 2.2 \quad T: 0.45 \times 5 = 2.2$$

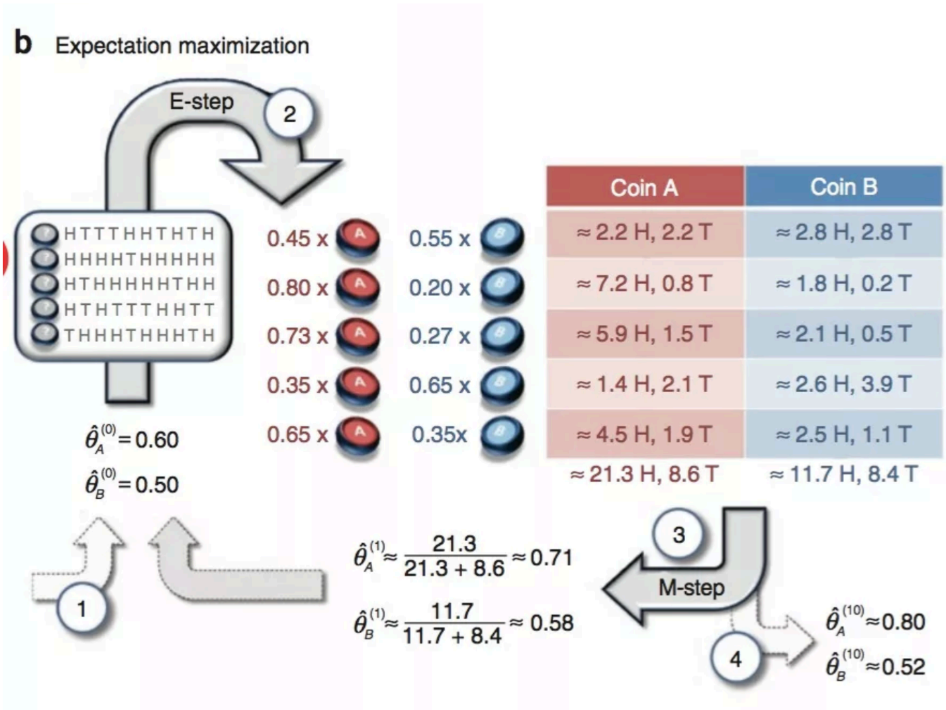
得到如下表格

No	Coin A	Coin B
1	2.2H, 2.2T	2.8H, 2.8T
2	7.2H, 0.8T	1.8H, 0.2T
3	5.9H, 1.5T	2.1H, 0.5T
4	1.4H, 2.1T	2.6H, 3.9T
5	4.5H, 1.9T	2.5H, 1.1T
Total	21.3H, 8.6T	11.7H, 8.4T

b) 用极大似然估计来估计新的P_A和P_B

$$P_A = \frac{21.3}{21.3 + 8.6} = 0.71 \quad P_B = \frac{11.7}{11.7 + 8.4} = 0.58$$

4. 反复迭代，算出最终的参数值



4.4 代码实现

em.py

代码：

```
"""
```

```
  This program achieves EM algorithm.

  Input: observations, theta
  Output: final theta, iterations

"""

import numpy as np
from scipy.stats import binom
import matplotlib.pyplot as plt
import math

# 建立数据集
observations = np.array([[1, 0, 0, 0, 1, 1, 0, 1, 0, 1],
                         [1, 1, 1, 1, 0, 1, 1, 1, 1, 1],
                         [1, 0, 1, 1, 1, 1, 1, 0, 1, 1],
                         [1, 0, 1, 0, 0, 0, 1, 1, 0, 0],
                         [0, 1, 1, 1, 0, 1, 1, 1, 0, 1]])
theta = [0.6, 0.5]

# 定义一次EM步
def em_single(observations, theta):
    # 计算每轮次数
    length = [0 for i in range(5)]
    for i in range(5):
        length[i] = len(observations[i])
    # 计算每轮H的个数和T的个数
    num_H = [0 for i in range(5)]
    num_T = [0 for i in range(5)]
    for i in range(5):
      num_H[i] = observations[i].sum()
      num_T[i] = length[i] - observations[i].sum()
    # E步
    # 计算PA
    old_theta_a = theta[0]
    pro_A = [0 for i in range(5)]
    for i in range(5):
        pro_A[i] = binom.pmf(num_H[i],length[i],old_theta_a)
    # 计算PB
    old_theta_b = theta[1]
    pro_B = [0 for i in range(5)]
    for i in range(5):
        pro_B[i] = binom.pmf(num_H[i],length[i],old_theta_b)
    # 计算硬币A的概率
    PA = [0 for i in range(5)]
    for i in range(5):
        PA[i] = pro_A[i] / (pro_A[i] + pro_B[i])
    # 计算硬币B的概率
    PB = [0 for i in range(5)]
```

```python
    for i in range(5):
        PB[i] = pro_B[i] / (pro_A[i] + pro_B[i])
    # 计算硬币A的H的期望，T的期望
    E_A_H = 0
    E_A_T = 0
    for i in range(5):
      E_A_H += num_H[i] * PA[i]
      E_A_T += num_T[i] * PA[i]
    # 计算硬币B的H的期望，T的期望
    E_B_H = 0
    E_B_T = 0
    for i in range(5):
      E_B_H += num_H[i] * PB[i]
      E_B_T += num_T[i] * PB[i]
    # M步
    # 重新计算
    new_theta_A = E_A_H / (E_A_H + E_A_T)
    new_theta_B = E_B_H / (E_B_H + E_B_T)
    return [new_theta_A, new_theta_B]

# EM主函数
def em(ovservations, theta, tol=1e-6, iterations=10000):
    iteration = 0
    while iteration < iterations:
        new_theta = em_single(observations, theta)
        delta = np.abs(theta[0] - new_theta[0])
        if delta < tol:
            break;
        else:
            theta = new_theta
            iteration += 1
    return [new_theta, iteration]

# 打印结果
print(em(observations, theta))
```

结果：

```
>>>
= RESTART: /Users/xinyuanhe/Desktop/working/2021美赛/模型/【正式】模型-机器学习-聚类-期望最大化算法EM【hxy】/em.py
[[0.7967887593831099, 0.5195839356752803], 14]
```

5. 参考资料

1. EM算法及其推广
2. 二项分布函数文档
3. 机器学习——EM算法及代码实现
4. 一文详尽系列之EM算法