DL-NLP 第二次实验——EM 算法

一、问题描述

一个袋子中三种硬币的混合比例为：s1, s2 与 1-s1-s2 (0<=si<=1), 三种硬币掷出正面的概率分别为：p, q, r。自己指定系数 s1, s2, p, q, r，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 EM 算法来对参数进行估计并与预先假定的参数进行比较。

二、实验原理

有限混合模型是一个可以用来表示在总体分布中含有多个子分布的概率模型。子分部可以是各种经典的分布模型，包括高斯模型，伯努利模型，多项式模型等。其公式为：

$$P(y|\theta,\pi)=\sum_{k=1}^{K}\pi_k P(y|\theta_k)$$

其中π_k为第k个子分布的混合系数，其总和为 1，θ_k为第k个子分布的参数，$Y=\{y_1,y_2,\dots,y_N\}$为包含N个样本的观测数据组成的集合。本次实验涉及到的伯努利混合分布便是由多个伯努利分布组合而成的混合模型。

在已知观测数据集Y的情况下，求参数θ,π的极大似然估计，可以考虑利用 EM 算法。

首先在 E 步，考虑已知参数的情况下观测数据y_j来自于第k个分布的概率μ_{jk}：

$$\mu_{jk}=\frac{\pi_k P(y_j|\theta_k)}{\sum_{k=1}^{K}\pi_k P(y_j|\theta_k)}$$

然后 M 步，考虑参数的更新，建立带惩罚项的混合模型（惩罚项约束混合系数总和为 1）的似然函数：

$$logL(Y|\theta,\pi)=\sum_{n=1}^{N}log\sum_{k=1}^{K}\pi_k\theta_k^{y_n}(1-\theta_k)^{1-y_n}+\lambda(1-\sum_{n=1}^{N}\pi_k)$$

上式对θ_k求偏导使其偏导为 0，得到：

$$\theta_k = \frac{\sum_{n=1}^{N} \mu_{nk} y_n}{\sum_{n=1}^{N} \mu_{nk}}$$

对λ和π_k求偏导令其为 0，解得：

$$\lambda = N$$

$$\pi_k = \frac{\sum_{n=1}^{N} \mu_{nk}}{N}$$

因此，可以选择参数初值，利用上述公式计算μ_{jk}，更新π_k和θ_k，重复执行直到收敛，完成模型参数的预估。

三、实验步骤与结果分析

本次实验中，针对给定的参数生成投掷结果，利用 EM 算法估计参数，讨论了初值和样本数量对结果的影响以及结果的偏差。

1. 程序分析

```
import numpy as np
#参数设定与硬币投掷
n = 10000
s1, s2 = 0.2, 0.1
p, q, r = 0.3, 0.1, 0.9
[n1, n2, n3] = np.random.multinomial(n, [s1, s2, 1-s1-s2])
h1 = np.random.binomial(n1, p)
h2 = np.random.binomial(n2, q)
h3 = np.random.binomial(n3, r)
h = h1 + h2 + h3
t = n - h
print("h={}, t={}, f={}".format(h, t, h/n))  #打印正反面数量与正面概率
#设置预估参数初值
p_est, q_est, r_est = 0.2, 0.1, 0.6
s1_est, s2_est = 0.1, 0.05
delta, eps = 1, 1e-16
#EM 算法
while delta > eps:
    po, qo, ro = p_est, q_est, r_est  #记录旧值，用于判断是否收敛
    htmp = s1_est*p_est + s2_est*q_est + (1-s1_est-s2_est)*r_est
    ttmp = 1 - htmp  #μ分为两部分，分母只有两种，提前计算
    sumh1, sumt1 = s1_est*p_est/htmp*h, s1_est*(1-p_est)/ttmp*t
    sumh2, sumt2 = s2_est*q_est/htmp*h, s2_est*(1-q_est)/ttmp*t
    summiu1, summiu2 = sumh1+sumt1, sumh2+sumt2  #计算μ
```

```
    s1_est, s2_est = summiu1/n, summiu2/n  #计算混合系数预估
    p_est, q_est, r_est = sumh1/summiu1, sumh2/summiu2, (h-sumh1-sumh2)/(n-summiu1-summiu2)  #计算分布参数预估
    delta = abs(p_est-po)+abs(q_est-qo)+abs(r_est-ro)  #计算变化判断是否收敛
    print("p={}, q={}, r={}, s1={}, s2={}, E={}".format(p_est, q_est, r_est, s1_est, s2_est, p_est*s1_est+q_est*s2_est+r_est*(1-s1_est-s2_est)))
```

主要程序分为两个部分，参数设定与初始化、EM 算法。由于预估参数时仅与正反面数量有关而与其出现的顺序无关，因此只统计正反面数量，用于 EM 算法计算。EM 算法中由于伯努利分布观测值只有两种取值，可以考虑计算时分别针对两种情况进行计算然后求和。

2. 结果与分析

针对样本数量：

条件	s1	s2	p	q	r
设定值(E=0.64)	0.3	0.2	0.4	0.6	0.8
预估初值	0.2	0.1	0.3	0.5	0.7
N=10(f=0.5)	0.225000	0.104167	0.222222	0.4	0.608696
N=100(f=0.59)	0.2025	0.100417	0.291358	0.489627	0.691213
N=10000(f=0.6452)	0.1887	0.098117	0.341918	0.547987	0.738819

说明：E=0.64 指在该设定分布下，出现正面的期望为 0.64，f=0.5 指的是，在 N=10 的情况下，有 10*0.5=5 个正面。

可以发现，当提高样本数量时，采样的数据中出现正面的频率趋近于设定值的分布的期望，于是这种情况下针对参数的拟合会更好些，从表中可以看到，分布参数随着样本量增加逐渐趋近于真实参数，而分布参数并没有很好的提升。

针对初值和最终偏差：（N=10000）

条件	s1	s2	p	q	r
设定值	0.3	0.2	0.4	0.6	0.8
预估初值与终值	(0.2)0.1887	(0.1)0.0981	(0.3)0.3419	(0.5)0.5480	(0.7)0.7388
	(0.1)0.0870	(0.05)0.0472	(0.2)0.2685	(0.4)0.4947	(0.6)0.6878
	(0.3)0.2998	(0.2)0.1999	(0.4)0.4006	(0.6)0.6006	(0.8)0.8004

可以发现，EM 算法针对该问题的初值设定比较敏感，靠近真实值的参数最

终离真实值更近。实际上我认为 EM 算法相当于只能利用采样数据中正反面出现的比例，于是该算法会调整各参数使得期望逼近这个比例，然而参数的量相比于已知信息要多，举个例子，当获得了 10000 个数据，6000 个正面，我如何判断(s1,s2,p,q,r)为(1,0,0.6,0,0)还是(0.2,0.2,0.75,0.75,0.5)还是其它？这些组合都能保证其期望与采样数据相匹配。于是 EM 算法会从初值出发，逐渐收敛到初值附近的一组参数，这组参数能保证期望与采样数据匹配，处于一个极大似然估计中的似然函数极大值，但并不能说一定会趋近于真实值。这种情况在混合高斯模型中会好一些，因为其可能的取值，也就相当于 EM 算法能获得的信息，相比于各子分布的参数要充分一些，于是逼近的效果会好一些。

四、参考文献

1. https://zhuanlan.zhihu.com/p/93513123