EM 算法仿真

ZY2103518 吕晔

题目：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。 （1）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。 截至日期：4 月 22 日晚 12 点前

1 参数设置

自己指定的参数如下：

参数	值
S1	0.4
S2	0.4
P	0.3
q	0.3
r	0.8

2 生成随机样本

然后随机生成 1000 个样本，由数字 1 代表正面，数字 0 代表反面。生成训练样本的代码如下：

```
%% Prepare training data
Pi = [pi1, pi2, pi3];
pqr=[p, q, r];

for k=1:N
    r1 = rand;
    if(r1<Pi(1))
        coin_Flag = 1;
    elseif(r1<sum(Pi(1:2)))
        coin_Flag = 2;
    else
        coin_Flag = 3;
    end

    for kk=1:M
        r2 = rand;
        if r2<pqr(coin_Flag)
            train_Data(k, kk) = 1;
        else
            train_Data(k, kk) = 0;
        end

    end


end
```

3 EM 算法

3.1 E-Step

开始 EM 算法的迭代。首先进行 E-Step，计算出每一个样本是由某一枚硬币掷出的后验概率。分别用 u1,u2,u3 表示，其表达式为：

$$u_1(x^{(i)}) = \frac{p^{x^{(i)}}(1-p)^{1-x^{(i)}}\pi_1}{p^{x^{(i)}}(1-p)^{1-x^{(i)}}\pi_1 + q^{x^{(i)}}(1-q)^{1-x^{(i)}}\pi_2 + r^{x^{(i)}}(1-r)^{1-x^{(i)}}(1-\pi_1-\pi_2)} \tag{3.1}$$

$$u_2(x^{(i)}) = \frac{q^{x^{(i)}}(1-q)^{1-x^{(i)}}\pi_2}{p^{x^{(i)}}(1-p)^{1-x^{(i)}}\pi_1 + q^{x^{(i)}}(1-q)^{1-x^{(i)}}\pi_2 + r^{x^{(i)}}(1-r)^{1-x^{(i)}}(1-\pi_1-\pi_2)} \tag{3.2}$$

$$u_3(x^{(i)}) = 1 - u_1(x^{(i)}) - u_2(x^{(i)}) \tag{3.3}$$

代码如下：

```
%% start EM algorithm
for i=1:J-1


    %% E-Step : calculating u1,u2 and u3
%     u1=(theta(1,3).^train_Data.*(1-theta(1,3)).^(1-train_Data).*theta(1,1))./(theta(1,3).^train_Data.*(1-theta(1,3)).^(1-train_Data).*theta(1,
%     u2=(theta(1,4).^train_Data.*(1-theta(1,4)).^(1-train_Data).*theta(1,2))./(theta(1,3).^train_Data.*(1-theta(1,3)).^(1-train_Data).*theta(1,
%     u3=(theta(1,5).^train_Data.*(1-theta(1,5)).^(1-train_Data).*(1-theta(1,1)-theta(1,2)))./(theta(1,3).^train_Data.*(1-theta(1,3)).^(1-train_
%
%     u1=(theta(1,3).^train_Data.*(1-theta(1,3)).^(1-train_Data))./(theta(1,3).^train_Data.*(1-theta(1,3)).^(1-train_Data)+theta(1,4).^train_Dat
%     u2=(theta(1,4).^train_Data.*(1-theta(1,4)).^(1-train_Data))./(theta(1,3).^train_Data.*(1-theta(1,3)).^(1-train_Data)+theta(1,4).^train_Dat
%     u3=1-u1-u2;

%     u1=(theta(1,3).^sum(train_Data,2).*(1-theta(1,3)).^(M-sum(train_Data,2)))./(theta(1,3).^sum(train_Data,2).*(1-theta(1,3)).^(M-sum(train_Da
%     u2=(theta(1,4).^sum(train_Data,2).*(1-theta(1,4)).^(M-sum(train_Data,2)))./(theta(1,3).^sum(train_Data,2).*(1-theta(1,3)).^(M-sum(train_Da
%     u3=1-u1-u2;
%
    u1=(theta(1,3).^sum(train_Data,2).*(1-theta(1,3)).^(M-sum(train_Data,2)).*theta(1,1))./(theta(1,3).^sum(train_Data,2).*(1-theta(1,3)).^(M-su
    u2=(theta(1,4).^sum(train_Data,2).*(1-theta(1,4)).^(M-sum(train_Data,2)).*theta(1,2))./(theta(1,3).^sum(train_Data,2).*(1-theta(1,3)).^(M-su
    u3=1-u1-u2;
%
```

3.2 M-Step

在 M-Step 中需要对参数进行一次极大似然估计，实现对参数的更新，待更新的参数的表达式为：$\theta = [\pi_1, \pi_2, p, q, r]$，其中包含三枚硬币出现的概率以及三枚硬币掷出正面的概率。通过以下公式直接计算出θ的极大似然估计。

$$\hat{\pi}_1 = \sum_{i=1}^{N} u_1(x^{(i)}) / N \tag{3.4}$$

$$\hat{\pi}_2 = \sum_{i=1}^{N} u_2(x^{(i)}) / N \tag{3.5}$$

$$\hat{p} = \sum_{i=1}^{N} u_1(x^{(i)})x^{(i)} / \sum_{i=1}^{N} u_1(x^{(i)})M \tag{3.6}$$

$$\hat{q} = \sum_{i=1}^{N} u_2(x^{(i)})x^{(i)} / \sum_{i=1}^{N} u_2(x^{(i)})M \tag{3.7}$$

$$\hat{r} = \sum_{i=1}^{N} (1-u_1(x^{(i)})-u_2(x^{(i)}))x^{(i)} / \sum_{i=1}^{N} (1-u_1(x^{(i)})-u_2(x^{(i)}))M \tag{3.8}$$

其中的 M 为每一个样本的维度，在本次实验中取 M=100。即每次取出一枚硬币后连续掷 100 次。

代码如下：

```
%% M-Step : update theta
theta(i+1,1) = sum(u1)/N;
theta(i+1,2) = sum(u2)/N;
theta(i+1,3) = sum(u1.*sum(train_Data,2))/sum(u1.*M);
theta(i+1,4) = sum(u2.*sum(train_Data,2))/sum(u2.*M);
theta(i+1,5) = sum(u3.*sum(train_Data,2))/sum(u3.*M);
```

3.3 θ 的初始参数

随意选定的初始参数数据如下表所示

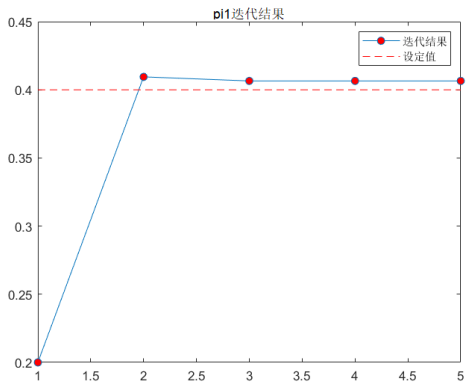
初始参数	值
$\pi_1^{(0)}$	0.2
$\pi_1^{(0)}$	0.2
$p^{(0)}$	0.5
$q^{(0)}$	0.5
$r^{(0)}$	0.9

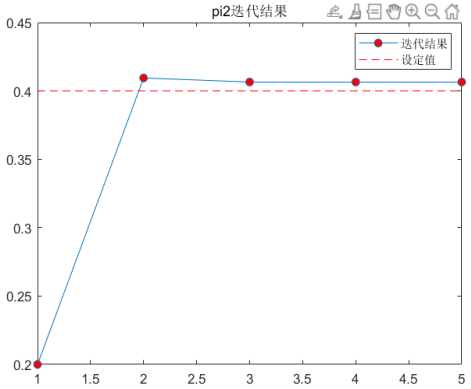
每一次 E-Step 和 M-Step 作为一步迭代，一共迭代 5 此，然后比较最终结果。

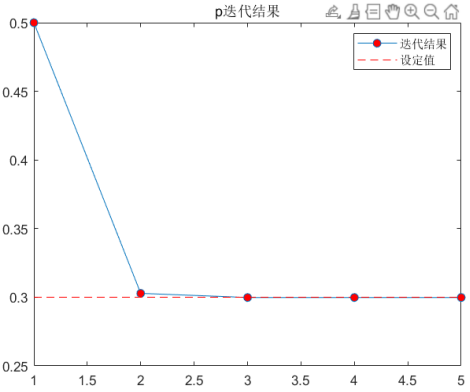
4 实验结果

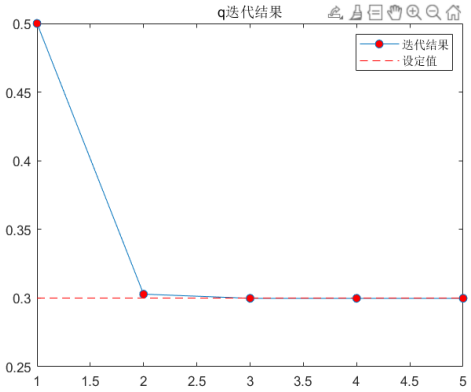
	1	2	3	4	5
1	0.2000	0.2000	0.5000	0.5000	0.9000
2	0.4095	0.4095	0.3028	0.3028	0.8050
3	0.4065	0.4065	0.2998	0.2998	0.8021
4	0.4065	0.4065	0.2998	0.2998	0.8021
5	0.4065	0.4065	0.2998	0.2998	0.8021

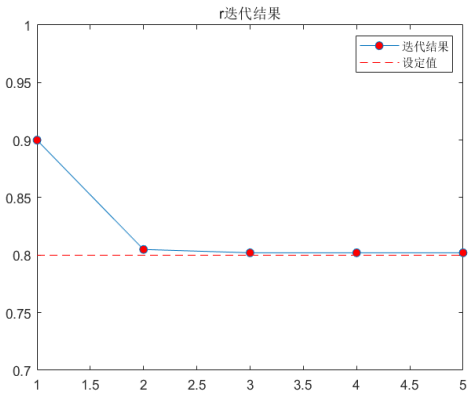
上表是 5 个参数分别迭代 5 次的结果。由表中的最后一行可以看出，实验结果基本与用于生成训练数据的参数一致，EM 算法迭代正确。











5 附录

```
%% 深度学习与自然语言处理2022第二次大作业
%   吕晔
%   ZY2103518
%   EM算法的仿真验证
%   2022年4月15日

%% 自定义各项训练参数
%   define  pi1 = 0.4   第一枚硬币出现的概率
%   define  pi2 = 0.4   第二枚硬币出现的概率
%   define  pi3 = 0.2   第三枚硬币出现的概率
%   define  p = 0.3     第一枚硬币出现正面的概率
%   define  q = 0.3     第二枚硬币出现正面的概率
%   define  r = 0.8     第三枚硬币出现正面的概率
%   训练集样本数 N = 1000
%   EM循环次数  J = 5

%% 初始化数据
clc
clear

pi1=0.4;
pi2=0.4;
pi3=0.2;
p=0.3;
q=0.3;
r=0.8;
N=1000;
M=100;
J=5;
train_Data = zeros(N,M);
u1 = zeros(N,1);
u2 = zeros(N,1);
u3 = zeros(N,1);
theta = zeros(J,5);
theta(1,1)=0.2;
theta(1,2)=0.2;
theta(1,3)=0.5;
theta(1,4)=0.5;
theta(1,5)=0.9;

%% Prepare training data
Pi = [pi1,pi2,pi3];
pqr=[p,q,r];

for k=1:N
    r1 = rand;
    if(r1<Pi(1))
        coin_Flag = 1;
    elseif(r1<sum(Pi(1:2)))
        coin_Flag = 2;
    else
        coin_Flag = 3;
    end
```

```
    for kk=1:M
        r2 = rand;
        if r2<pqr(coin_Flag)
            train_Data(k,kk) = 1;
        else
            train_Data(k,kk) = 0;
        end

    end


end




%% start EM algorithm
for i=1:J-1


   %% E-Step : calculating u1,u2 and u3
%    u1=(theta(i,3).^train_Data.*(1-theta(i,3)).^(1-train_Data).*theta(i,1))./(theta(i,3).^train_Data.*(1-
theta(i,3)).^(1-train_Data).*theta(i,1)+theta(i,4).^train_Data.*(1-theta(i,4)).^(1-
train_Data).*theta(i,2)+theta(i,5).^train_Data.*(1-theta(i,5)).^(1-train_Data).*(1-theta(i,1)-
theta(i,2)));
%    u2=(theta(i,4).^train_Data.*(1-theta(i,4)).^(1-train_Data).*theta(i,2))./(theta(i,3).^train_Data.*(1-
theta(i,3)).^(1-train_Data).*theta(i,1)+theta(i,4).^train_Data.*(1-theta(i,4)).^(1-
train_Data).*theta(i,2)+theta(i,5).^train_Data.*(1-theta(i,5)).^(1-train_Data).*(1-theta(i,1)-
theta(i,2)));
%    u3=(theta(i,5).^train_Data.*(1-theta(i,5)).^(1-train_Data).*(1-theta(i,1)-
theta(i,2)))./(theta(i,3).^train_Data.*(1-theta(i,3)).^(1-
train_Data).*theta(i,1)+theta(i,4).^train_Data.*(1-theta(i,4)).^(1-
train_Data).*theta(i,2)+theta(i,5).^train_Data.*(1-theta(i,5)).^(1-train_Data).*(1-theta(i,1)-
theta(i,2)));
%
%    u1=(theta(i,3).^train_Data.*(1-theta(i,3)).^(1-train_Data))./(theta(i,3).^train_Data.*(1-
theta(i,3)).^(1-train_Data)+theta(i,4).^train_Data.*(1-theta(i,4)).^(1-
train_Data)+theta(i,5).^train_Data.*(1-theta(i,5)).^(1-train_Data));
%    u2=(theta(i,4).^train_Data.*(1-theta(i,4)).^(1-train_Data))./(theta(i,3).^train_Data.*(1-
theta(i,3)).^(1-train_Data)+theta(i,4).^train_Data.*(1-theta(i,4)).^(1-
train_Data)+theta(i,5).^train_Data.*(1-theta(i,5)).^(1-train_Data));
%    u3=1-u1-u2;

%    u1=(theta(i,3).^sum(train_Data,2).*(1-theta(i,3)).^(M-
sum(train_Data,2)))./(theta(i,3).^sum(train_Data,2).*(1-theta(i,3)).^(M-
sum(train_Data,2))+theta(i,4).^sum(train_Data,2).*(1-theta(i,4)).^(M-
sum(train_Data,2))+theta(i,5).^sum(train_Data,2).*(1-theta(i,5)).^(M-sum(train_Data,2)));
%    u2=(theta(i,4).^sum(train_Data,2).*(1-theta(i,4)).^(M-
sum(train_Data,2)))./(theta(i,3).^sum(train_Data,2).*(1-theta(i,3)).^(M-
sum(train_Data,2))+theta(i,4).^sum(train_Data,2).*(1-theta(i,4)).^(M-
sum(train_Data,2))+theta(i,5).^sum(train_Data,2).*(1-theta(i,5)).^(M-sum(train_Data,2)));
%    u3=1-u1-u2;
%
   u1=(theta(i,3).^sum(train_Data,2).*(1-theta(i,3)).^(M-
sum(train_Data,2)).*theta(i,1))./(theta(i,3).^sum(train_Data,2).*(1-theta(i,3)).^(M-
sum(train_Data,2)).*theta(i,1)+theta(i,4).^sum(train_Data,2).*(1-theta(i,4)).^(M-
sum(train_Data,2)).*theta(i,2)+theta(i,5).^sum(train_Data,2).*(1-theta(i,5)).^(M-sum(train_Data,2)).*(1-
```

```
theta(i,1)-theta(i,2)));
   u2=(theta(i,4).^sum(train_Data,2).*(1-theta(i,4)).^(M-
sum(train_Data,2)).*theta(i,2))./(theta(i,3).^sum(train_Data,2).*(1-theta(i,3)).^(M-
sum(train_Data,2)).*theta(i,1)+theta(i,4).^sum(train_Data,2).*(1-theta(i,4)).^(M-
sum(train_Data,2)).*theta(i,2)+theta(i,5).^sum(train_Data,2).*(1-theta(i,5)).^(M-sum(train_Data,2)).*(1-
theta(i,1)-theta(i,2)));
   u3=1-u1-u2;
%

%   for j=1:N
%       u1(j)=(theta(i,3).^train_Data(j).*(1-theta(i,3)).^(1-
train_Data(j)).*theta(i,1)).//(theta(i,3).^train_Data(j).*(1-theta(i,3)).^(1-
train_Data(j)).*theta(i,1)+theta(i,4).^train_Data(j).*(1-theta(i,4)).^(1-
train_Data(j)).*theta(i,2)+theta(i,5).^train_Data(j).*(1-theta(i,5)).^(1-train_Data(j)).*(1-theta(i,1)-
theta(i,2)));
%       u2(j)=(theta(i,4).^train_Data(j).*(1-theta(i,4)).^(1-
train_Data(j)).*theta(i,2)).//(theta(i,3).^train_Data(j).*(1-theta(i,3)).^(1-
train_Data(j)).*theta(i,1)+theta(i,4).^train_Data(j).*(1-theta(i,4)).^(1-
train_Data(j)).*theta(i,2)+theta(i,5).^train_Data(j).*(1-theta(i,5)).^(1-train_Data(j)).*(1-theta(i,1)-
theta(i,2)));
%       u3(j)=(theta(i,5).^train_Data(j).*(1-theta(i,5)).^(1-train_Data(j)).*(1-theta(i,1)-
theta(i,2))).//(theta(i,3).^train_Data(j).*(1-theta(i,3)).^(1-
train_Data(j)).*theta(i,1)+theta(i,4).^train_Data(j).*(1-theta(i,4)).^(1-
train_Data(j)).*theta(i,2)+theta(i,5).^train_Data(j).*(1-theta(i,5)).^(1-train_Data(j)).*(1-theta(i,1)-
theta(i,2)));
%
%   end
%

   %% M-Step : update theta
   theta(i+1,1) = sum(u1)/N;
   theta(i+1,2) = sum(u2)/N;
   theta(i+1,3) = sum(u1.*sum(train_Data,2))/sum(u1.*M);
   theta(i+1,4) = sum(u2.*sum(train_Data,2))/sum(u2.*M);
   theta(i+1,5) = sum(u3.*sum(train_Data,2))/sum(u3.*M);




end
```