# Homework 2

自硕 21 崔晏菲 2021210976

注：因为我不会 R 语言，所以代码都是用 Python 写的。代码文件见 homework2-code.ipynb

## 4.1

解：

a. 对于这些飞蛾，完整数据应该是

$$Y = \begin{pmatrix} n_{CC} \\ n_{CI} \\ n_{CT} \\ n_{II} \\ n_{IT} \\ n_{TT} \\ n_{U_{II}} \\ n_{U_{IT}} \\ n_{U_{TT}} \end{pmatrix}$$

而观察到的数据是

$$X = \begin{pmatrix} n_C \\ n_I \\ n_T \\ n_U \end{pmatrix} = \begin{pmatrix} n_{CC} + n_{CI} + n_{CT} \\ n_{II} + n_{IT} \\ n_{TT} \\ n_{U_{II}} + n_{U_{IT}} + n_{U_{TT}} \end{pmatrix}$$

先计算条件期望

$$\begin{cases} E\left[N_{CC}|X,p^{(t)}\right] = n_{CC}^{(t)} = \dfrac{n_C\left(p_C^{(t)}\right)^2}{\left(p_C^{(t)}\right)^2 + 2p_C^{(t)}p_I^{(t)} + 2p_C^{(t)}p_T^{(t)}} \\ E\left[N_{CI}|X,p^{(t)}\right] = n_{CI}^{(t)} = \dfrac{2n_C p_C^{(t)}p_I^{(t)}}{\left(p_C^{(t)}\right)^2 + 2p_C^{(t)}p_I^{(t)} + 2p_C^{(t)}p_T^{(t)}} \\ E\left[N_{CT}|X,p^{(t)}\right] = n_{CT}^{(t)} = \dfrac{2n_C p_C^{(t)}p_T^{(t)}}{\left(p_C^{(t)}\right)^2 + 2p_C^{(t)}p_I^{(t)} + 2p_C^{(t)}p_T^{(t)}} \\ E\left[N_{II}|X,p^{(t)}\right] = n_{II}^{(t)} + n_{U_{II}}^{(t)} = \dfrac{n_I\left(p_I^{(t)}\right)^2}{\left(p_I^{(t)}\right)^2 + 2p_I^{(t)}p_T^{(t)}} + \dfrac{n_U\left(p_I^{(t)}\right)^2}{\left(p_I^{(t)}\right)^2 + 2p_I^{(t)}p_T^{(t)} + \left(p_T^{(t)}\right)^2} \\ E\left[N_{IT}|X,p^{(t)}\right] = n_{IT}^{(t)} + n_{U_{IT}}^{(t)} = \dfrac{2n_I p_I^{(t)}p_T^{(t)}}{\left(p_I^{(t)}\right)^2 + 2p_I^{(t)}p_T^{(t)}} + \dfrac{2n_U p_I^{(t)}p_T^{(t)}}{\left(p_I^{(t)}\right)^2 + 2p_I^{(t)}p_T^{(t)} + \left(p_T^{(t)}\right)^2} \\ E\left[N_{TT}|X,p^{(t)}\right] = n_{TT}^{(t)} + n_{U_{TT}}^{(t)} = n_T + \dfrac{n_U\left(p_T^{(t)}\right)^2}{\left(p_I^{(t)}\right)^2 + 2p_I^{(t)}p_T^{(t)} + \left(p_T^{(t)}\right)^2} \end{cases}$$

E-step：

$Q$函数为：

$$Q(p|\,p^{(t)}) = n_{CC}^{(t)}\log p_C^2 + n_{CI}^{(t)}\log 2p_Cp_I + n_{CT}^{(t)}\log 2p_Cp_T + \left(n_{II}^{(t)} + n_{U_{II}}^{(t)}\right)\log p_I^2 + \left(n_{IT}^{(t)} + n_{U_{IT}}^{(t)}\right)\log 2p_Ip_T + \left(n_{TT}^{(t)} + n_{U_{TT}}^{(t)}\right)\log p_T^2 + k\left(n_C, n_I, n_T, n_U, p^{(t)}\right)$$

得到

$$\begin{cases} p_C^{(t+1)} = \dfrac{2n_{CC}^{(t)} + n_{CI}^{(t)} + n_{CT}^{(t)}}{2n} \\ p_I^{(t+1)} = \dfrac{2n_{II}^{(t)} + n_{IT}^{(t)} + n_{CI}^{(t)} + 2n_{U_{II}}^{(t)} + n_{U_{IT}}^{(t)}}{2n} \\ p_T^{(t+1)} = \dfrac{2n_{TT}^{(t)} + n_{IT}^{(t)} + n_{IT}^{(t)} + 2n_{U_{TT}}^{(t)} + n_{U_{IT}}^{(t)}}{2n} \end{cases}$$

b.

```
MLE_EM = moth.EM_algorithm(X)
print("EM算法得到的p的MLE为: p_c = %f, p_i = %f, p_t = %f"%(MLE_EM[0], MLE_EM[1], MLE_EM[2]))
✓ 0.0s
```

```
EM算法得到的p的MLE为: p_c = 0.036067, p_i = 0.195799, p_t = 0.768134
```

c. 作业不要求，略

d.

```
Boot strapping采样10000次后得到:
p_c, p_i, p_t的标准差分别为0.003865, 0.010963, 0.011401
p的相关系数矩阵为:
 [[ 1.         -0.06069471 -0.28067047]
 [-0.06069471  1.         -0.94099946]
 [-0.28067047 -0.94099946  1.        ]]
可见, corr(p_c, p_i) = -0.060695, corr(p_c, p_t) = -0.280670, corr(p_i, p_t) = -0.940999
```

e.

$$Q(p|\,p^{(t)})$$

$$= n_{CC}^{(t)}\log p_C^2 + n_{CI}^{(t)}\log 2p_Cp_I + n_{CT}^{(t)}\log 2p_Cp_T + \left(n_{II}^{(t)} + n_{U_{II}}^{(t)}\right)\log p_I^2 + \left(n_{IT}^{(t)} + n_{U_{IT}}^{(t)}\right)\log 2p_Ip_T + \left(n_{TT}^{(t)} + n_{U_{TT}}^{(t)}\right)\log p_T^2 + k\left(n_C, n_I, n_T, n_U, p^{(t)}\right)$$

$$= n_{CC}^{(t)}\log p_C^2 + n_{CI}^{(t)}\log 2p_Cp_I + n_{CT}^{(t)}\log\left(2p_c(1-p_C-p_I)\right) + \left(n_{II}^{(t)} + n_{U_{II}}^{(t)}\right)\log p_I^2 + \left(n_{IT}^{(t)} + n_{U_{IT}}^{(t)}\right)\log\left(2p_I(1-p_C-p_I)\right) + \left(n_{TT}^{(t)} + n_{U_{TT}}^{(t)}\right)\log(1-p_C-p_I)^2 + k\left(n_C, n_I, n_T, n_U, p^{(t)}\right)$$

故

$$\nabla Q=\begin{pmatrix}\dfrac{2n_{CC}^{(t)}+n_{CI}^{(t)}+n_{CT}^{(t)}}{p_c}-\dfrac{n_{CT}^{(t)}+n_{IT}^{(t)}+n_{UIT}^{(t)}+2\left(n_{TT}^{(t)}+n_{UTT}^{(t)}\right)}{1-p_c-p_I}\\ \dfrac{n_{CI}^{(t)}+2\left(n_{II}^{(t)}+n_{UII}^{(t)}\right)+n_{IT}^{(t)}+n_{UIT}^{(t)}}{p_I}-\dfrac{n_{CT}^{(t)}+n_{IT}^{(t)}+n_{UIT}^{(t)}+2\left(n_{TT}^{(t)}+n_{UTT}^{(t)}\right)}{1-p_c-p_I}\end{pmatrix}$$

$\nabla^2 Q$

$$=\begin{pmatrix}-\dfrac{2n_{CC}^{(t)}+n_{CI}^{(t)}+n_{CT}^{(t)}}{p_c^2}-\dfrac{n_{CT}^{(t)}+n_{IT}^{(t)}+n_{UIT}^{(t)}+2\left(n_{TT}^{(t)}+n_{UTT}^{(t)}\right)}{(1-p_c-p_I)^2} & -\dfrac{n_{CT}^{(t)}+n_{IT}^{(t)}+n_{UIT}^{(t)}+2\left(n_{TT}^{(t)}+n_{UTT}^{(t)}\right)}{(1-p_c-p_I)^2}\\ -\dfrac{n_{CT}^{(t)}+n_{IT}^{(t)}+n_{UIT}^{(t)}+2\left(n_{TT}^{(t)}+n_{UTT}^{(t)}\right)}{(1-p_c-p_I)^2} & \dfrac{n_{CI}^{(t)}+2\left(n_{II}^{(t)}+n_{UII}^{(t)}\right)+n_{IT}^{(t)}+n_{UIT}^{(t)}}{p_I^2}-\dfrac{n_{CT}^{(t)}+n_{IT}^{(t)}+n_{UIT}^{(t)}+2\left(n_{TT}^{(t)}+n_{UTT}^{(t)}\right)}{(1-p_c-p_I)^2}\end{pmatrix}$$

```
MLE_EM_grad = moth.EM_gradient_algorithm(X, inplace=False)
print("EM gradient算法得到的p的MLE为: p_c = %f, p_i = %f, p_t = %f"%(MLE_EM_grad[0], MLE_EM_grad[1], MLE_EM_grad[2]))
```

✓ 0.0s

```
EM gradient算法得到的p的MLE为: p_c = 0.036067, p_i = 0.195799, p_t = 0.768134
```

## 4.2

解：

a. 证明：

$$E\left[N_{z,0}\middle|\, n,\theta^{(t)}\right]=n_{z,0}^{(t)}=\frac{n_0 z_0\left(\theta^{(t)}\right)}{z_0\left(\theta^{(t)}\right)+t_0\left(\theta^{(t)}\right)+p_0\left(\theta^{(t)}\right)}$$

$$E\left[N_{t,0}\middle|\, n,\theta^{(t)}\right]=n_{t,0}^{(t)}=\frac{n_0 t_0\left(\theta^{(t)}\right)}{z_0\left(\theta^{(t)}\right)+t_0\left(\theta^{(t)}\right)+p_0\left(\theta^{(t)}\right)}$$

$$E\left[N_{p,0}\middle|\, n,\theta^{(t)}\right]=n_{p,0}^{(t)}=\frac{n_0 p_0\left(\theta^{(t)}\right)}{z_0\left(\theta^{(t)}\right)+t_0\left(\theta^{(t)}\right)+p_0\left(\theta^{(t)}\right)}$$

$$E\left[N_{t,i}\middle|\, n,\theta^{(t)}\right]=n_{t,i}^{(t)}=\frac{n_i t_i\left(\theta^{(t)}\right)}{t_i\left(\theta^{(t)}\right)+p_i\left(\theta^{(t)}\right)},i=1,\cdots,16$$

$$E\left[N_{p,i}\middle|\, n,\theta^{(t)}\right]=n_{p,i}^{(t)}=\frac{n_i p_i\left(\theta^{(t)}\right)}{t_i\left(\theta^{(t)}\right)+p_i\left(\theta^{(t)}\right)},i=1,\cdots,16$$

E-step：

$Q$函数为：

$$Q\left(\theta\middle|\,\theta^{(t)}\right)=n_{z,0}^{(t)}\log\alpha+\sum_{i=0}^{16}n_{t,i}^{(t)}\log\frac{\beta\mu^i e^{-\mu}}{i!}$$

$$+\sum_{i=0}^{16}n_{p,i}^{(t)}\log\frac{(1-\alpha-\beta)\lambda^i e^{-\lambda}}{i!}$$

M-step:

$\nabla Q=0$得

$$\begin{cases} \dfrac{\partial Q}{\partial \alpha} = \dfrac{n_{z,0}^{(t)}}{\alpha} + \sum_{i=0}^{16} \dfrac{n_{p,i}^{(t)}}{\alpha-\beta-1} = 0 \\ \dfrac{\partial Q}{\partial \beta} = \sum_{i=0}^{16} \left( \dfrac{n_{t,i}^{(t)}}{\alpha} + \dfrac{n_{p,i}^{(t)}}{\alpha-\beta-1} \right) = 0 \\ \dfrac{\partial Q}{\partial \mu} = \sum_{i=0}^{16} \left( -n_{t,i}^{(t)} + \dfrac{in_{t,i}^{(t)}}{\mu} \right) = 0 \\ \dfrac{\partial Q}{\partial \lambda} = \sum_{i=0}^{16} \left( -n_{p,i}^{(t)} + \dfrac{in_{p,i}^{(t)}}{\lambda} \right) = 0 \end{cases}$$

故

$$\begin{cases} \alpha^{(t+1)} = \dfrac{n_0 z_0\left(\theta^{(t)}\right)}{N} \\ \beta^{(t+1)} = \sum_{i=0}^{16} \dfrac{n_i t_i\left(\theta^{(t)}\right)}{N} \\ \mu^{(t+1)} = \dfrac{\sum_{i=0}^{16} in_i t_i\left(\theta^{(t)}\right)}{\sum_{i=0}^{16} n_i t_i\left(\theta^{(t)}\right)} \\ \lambda^{(t+1)} = \dfrac{\sum_{i=0}^{16} in_i p_i\left(\theta^{(t)}\right)}{\sum_{i=0}^{16} n_i p_i\left(\theta^{(t)}\right)} \end{cases}$$

证毕

b.

```
theta_pred = hiv.EM_algorithm(data, inplace=True)
print("预测的alpha = %f, beta = %f, mu = %f, lambda = %f"%theta_pred)
```

✓ 0.0s

```
预测的alpha = 0.122166, beta = 0.562542, mu = 1.467475, lambda = 5.938889
```

c.

```
Boot strapping采样2000次后得到:
alpha, beta, mu, lambda的标准差分别为0.044415, 0.089679, 1.325910, 1.452198
theta的相关系数矩阵为:
 [[ 1.         -0.59179758  0.39238327 -0.40846613]
 [-0.59179758  1.         -0.94386978  0.9611617 ]
 [ 0.39238327 -0.94386978  1.         -0.9805038 ]
 [-0.40846613  0.9611617  -0.9805038   1.        ]]
可见,
corr(alpha, beta) = -0.591798,  corr(alpha, mu) = 0.392383,     corr(alpha, lambda) = -0.408466,
corr(beta, mu) = -0.943870,     corr(beta, lambda) = 0.961162,  corr(mu, lambda) = -0.980504
```

## 4.3

解：

a. 因为是指数族分布，所以可以估计充分统计量的条件期望，即

$$S=\begin{pmatrix}\sum_{i=1}^{n} y_{ij}\,, j=1,\cdots,K \\ \sum_{i=1}^{n} y_{ij}y_{ik}\,, j,k=1,\cdots,K\end{pmatrix}$$

在 EM 算法的第$t$次迭代中，要估计的参数为

$$\theta^{(t)}=\left(\mu^{(t)},\Sigma^{(t)}\right)$$

因此，E-step:

$$E\left[\sum_{i=1}^{n} y_{ij} \mid Y_{\text{known}},\theta^{(t)}\right]=\sum_{i=1}^{n} y_{ij}^{(t)}\,, j=1,\cdots,K$$

$$E\left[\sum_{i=1}^{n} y_{ij}y_{ik} \mid Y_{\text{known}},\theta^{(t)}\right]=\sum_{i=1}^{n}\left(y_{ij}^{(t)}y_{ik}^{(t)}+b_{jki}^{(t)}\right)$$

其中

$$y_{ij}^{(t)}=\begin{cases} y_{ij} & \text{if } y_{ij} \text{ is konwn} \\ E\left[y_{ij}\mid y_{\text{known},i},\theta^{(t)}\right] & \text{else}\end{cases}$$

$$b_{jki}^{(t)}=\begin{cases} 0 & \text{if } y_{ij} \text{ is known or } y_{ik} \text{ is known} \\ \mathrm{Cov}\left(y_{ij},y_{ik}\mid y_{\text{known},i},\theta^{(t)}\right) & \text{else}\end{cases}$$

M-step:

$$\mu_j^{(t+1)}=\frac{1}{n}\sum_{i=i}^{n} y_{ij}^{(t)}\,, j=1,\cdots,K$$

$$\sigma_{jk}^{(t+1)}=\frac{1}{n}E\left[\sum_{i=1}^{n} y_{ij}y_{ik} \mid Y_{\text{known}}\right]-\mu_j^{(t+1)}\mu_k^{(t+1)}$$

$$=\frac{1}{n}\sum_{i=1}^{n}\left(y_{ij}^{(t)}y_{ik}^{(t)}+b_{jki}^{(t)}\right)-\mu_j^{(t+1)}\mu_k^{(t+1)}, j,k=1,\cdots,K$$

b.

```
预测的mu为:  [[0.87883229 2.85094528 9.02942437]]
预测的sigma为:
 [[1.3511929  0.95683854 1.28486395]
 [0.95683854 0.73441359 0.69129236]
 [1.28486395 0.69129236 2.36967305]]
```

c. 联合概率后验分布为

$$L(y,\mu)=\prod_{i=1}^{n}\frac{1}{(2\pi)^{\frac{K}{2}}|\Sigma|^{\frac{1}{2}}}e^{-\frac{1}{2}(y_i-\mu)^T\Sigma^{-1}(y_i-\mu)}\cdot\prod_{j=1}^{K}\frac{e^{-\frac{(\mu_j-\alpha_j)}{\beta_j}}}{\beta_j\left(1+e^{-\frac{(\mu_j-\alpha_j)}{\beta_j}}\right)^2}$$

故对数似然函数为

$$l(y,\mu)=-\frac{1}{2}\sum_{i=1}^{n}\left(K\ln(2\pi)+\ln|\Sigma|+(y_i-\mu)^T\Sigma^{-1}(y_i-\mu)\right)$$
$$+\sum_{j=1}^{K}\left(-\frac{(\mu_j-\alpha_j)}{\beta_j}-\ln\beta_j-2\ln\left(1+e^{-\frac{(\mu_j-\alpha_j)}{\beta_j}}\right)\right)$$

故

$$\nabla l=-\sum_{i=1}^{n}\left(\Sigma^{-1}(y_i-\mu)\right)-\frac{1}{\beta}+2\frac{1}{\beta\left(1+e^{\frac{(\mu-\alpha)}{\beta}}\right)}$$

$$\nabla^2 l=n\Sigma^{-1}-2I\left(\frac{e^{\frac{(\mu-\alpha)}{\beta}}}{\beta^2\left(1+e^{\frac{(\mu-\alpha)}{\beta}}\right)^2}\right)$$

故 EM gradient 的结果为

```
EM gradient方法预测的mu为:  [[0.8790627  2.8457106  9.03311281]]
预测的sigma为:
 [[1.  0.6 1.2]
 [0.6 0.5 0.5]
 [1.2 0.5 3. ]]
```

d. 如果Σ未知并且服从均匀的先验分布，那么 gradient EM 算法在Σ方向上的梯度和 Hessian 矩阵为

$$\nabla_\Sigma l=-\frac{1}{2}\sum_{i=1}^{n}(\Sigma^{-1}+(y_i-\mu)(y_i-\mu)^T)$$

故可以用 gradient EM 算法逼近。

## 4.4

解：

$$f(x)=abx^{b-1}e^{-ax^b}$$

$$l=\log L(a,b,x)=n(\ln a+\ln b)+(b-1)\sum_{i=1}^{n}\ln x_i-a\sum_{i=1}^{n}x_i^b$$

$$E\left[x_i^{(t)} \mid a^{(t)}, b^{(t)}\right] = \begin{cases} x_i & \text{if } x_i \text{ isn't right censored} \\ x_i\left(a^{(t)} x_i^{b^{(t)}}\right)^{-\frac{1}{b^{(t)}}} \Gamma\left(1+\frac{1}{b^{(t)}}, a^{(t)} x_i^{b^{(t)}}\right) & \text{else} \end{cases}$$

其中Γ()是不完全 Gamma 函数。

M-step:
由于

$$E[x^b] = \frac{1}{a}$$

故

$$a^{(t+1)} = \frac{1}{E\left[{x^{(t)}}^{b^{(t)}}\right]}$$

```
theta_coup = coupling.EM_algorithm(data, mask)
print("EM算法得到的预测值为, a = %f, b = %f"%(theta_coup[0,0], theta_coup[1,0]))
```

✓ 0.0s

```
EM算法得到的预测值为, a = 0.013325, b = 2.484870
```

因为可以用 Gamma 函数来计算 E-step，所以不需要蒙特卡洛采样，速度很快。