第六章课后作业

5-2 设三个总体 G_1,G_2 和 G_3 的分布分别为：$N(2,0.5^2)$，$N(0,2^2)$和 $N(3,1^2)$. 试问样品 $x=2.5$ 应判归哪一类？

(1) 按距离判别准则；

(2) 按贝叶斯判别准则 $\left(\text{取 } q_1=q_2=q_3=\frac{1}{3}, L(j|i)=\begin{cases}1, i\neq j\\0, i=j\end{cases}\right)$.

解：(1) 用样品到总体的相对平方距离作判别函数

样品到 G_1 的距离：$d_1^2(x)=\frac{(2.5-2)^2}{0.5^2}=1$

样品到 G_2 的距离：$d_2^2(x)=\frac{(2.5-0)^2}{2^2}=1.5625$

样品到 G_3 的距离：$d_3^2(x)=\frac{(2.5-3)^2}{1^2}=0.25$

$d_{min}^2=d_3^2(x)$，故 $x\in G_3$

(2) 按贝叶斯判别准则，判别函数为

$Z_i(x)=-\frac{1}{2}\ln|\Sigma_i|-\frac{1}{2}(x-\mu^{(i)})'\Sigma_i^{-1}(x-\mu^{(i)})$ $(q_1=q_2=q_3=\frac{1}{3})$

则 $Z_1(x)=-\frac{1}{2}\ln 0.5^2-\frac{(2.5-2)^2}{2\times 0.5^2}=0.1931$

$Z_2(x)=-\frac{1}{2}\ln 2^2-\frac{(2.5-0)^2}{2\times 2^2}=-1.4744$

$Z_3(x)=-\frac{1}{2}\ln 1^2-\frac{(2.5-3)^2}{2\times 1^2}=-0.125$

$\because \max\limits_{1\le i\le 3} Z_i(x)=Z_1(x)$，故 $x\in G_1$

5-10 已知某研究对象分为三类，每个样品考察 4 项指标，各类的观测样品数分别为 7，4，6；另外还有 3 个待判样品(所有观测数据见表 5.4). 假定样本均来自正态总体.

(1) 试用马氏距离判别法进行判别分析，并对 3 个待判样品进行判别归类.

表 5.4 判别分类的数据

样品号	X_1	X_2	X_3	X_4	类别号
1	6.0	−11.5	19.0	90.0	1
2	−11.0	−18.5	25.0	−36.0	3
3	90.2	−17.0	17.0	3.0	2
4	−4.0	−15.0	13.0	54.0	1
5	0.0	−14.0	20.0	35.0	2
6	0.5	−11.5	19.0	37.0	3
7	−10.0	−19.0	21.0	−42.0	3
8	0.0	−23.0	5.0	−35.0	1
9	20.0	−22.0	8.0	−20.0	3
10	−100.0	−21.4	7.0	−15.0	1
11	−100.0	−21.5	15.0	−40.0	2
12	13.0	−17.2	18.0	2.0	2
13	−5.0	−18.5	15.0	18.0	1
14	10.0	−18.0	14.0	50.0	1
15	−8.0	−14.0	16.0	56.0	1
16	0.6	−13.0	26.0	21.0	3
17	−40.0	−20.0	22.0	−50.0	3
1	−8.0	−14.0	16.0	56.0	
2	92.2	−17.0	18.0	3.0	
3	−14.0	−18.5	25.0	−36.0	

解:(1)通过提供的样本计算三类总体的样本均值和样本协方差矩阵,通过Python程序计算得:

```
X_bar = [-14.42857143 -17.34285714  12.71428571  31.14285714]
Y_bar = [  0.8    -17.425   17.5      0.    ]
Z_bar = [ -6.65       -17.33333333  20.16666667 -15.        ]
X_S =
[[1463.95238095   67.31190476   97.19047619  821.9047619 ]
 [  67.31190476   16.83952381   18.61904762  174.12380952]
 [  97.19047619   18.61904762   24.9047619   204.38095238]
 [ 821.9047619   174.12380952  204.38095238 1939.47619048]]
Y_S =
[[6.10082667e+03 1.49586667e+02 7.04666667e+01 1.43220000e+03]
 [1.49586667e+02 9.52250000e+00 6.21666667e+00 9.48666667e+01]
 [7.04666667e+01 6.21666667e+00 4.33333333e+00 6.23333333e+01]
 [1.43220000e+03 9.48666667e+01 6.23333333e+01 9.46000000e+02]]
Z_S =
[[ 391.255        9.67        -75.05        369.72      ]
 [   9.67         17.16666667   12.66666667  129.1       ]
 [ -75.05         12.66666667   42.16666667    4.4       ]
 [ 369.72        129.1          4.4        1284.        ]]
```

计算三个待判样品到三类总体的马氏距离：

$$D=(Y-\mu)'\Sigma^{-1}(Y-\mu)$$

计算结果为：

```
d1_X_square = 1.3945678050562498
d1_Y_square = 895973089446986.0
d1_Z_square = 39.87898930679415

d2_X_square = 35.63219086990604
d2_Y_square = 195708362558.17847
d2_Z_square = 68.70624559237007

d3_X_square = 123.06721325590334
d3_Y_square = 1219817336488197.5
d3_Z_square = 1.3961442455577868
```

根据计算结果，

待判样品1归为第1类；

待判样品2归为第1类；

待判样品3归为第3类；

代码如下：

```python
import numpy as np

# 第一类观测值
X = np.array([[6.0, -11.5, 19.0, 90.0],
              [-4.0, -15.0, 13.0, 54.0],
              [0.0, -23.0, 5.0, -35.0],
              [-100.0, -21.4, 7.0, -15.0],
              [-5.0, -18.5, 15.0, 18.0],
              [10.0, -18.0, 14.0, 50.0],
              [-8.0, -14.0, 16.0, 56.0]
              ])

# 第二类观测值
Y = np.array([[90.2, -17.0, 17.0, 3.0],
              [0.0, -14.0, 20.0, 35.0],
              [-100.0, -21.5, 15.0, -40.0],
              [13.0, -17.2, 18.0, 2.0]
              ])

# 第三类观测值
Z = np.array([[-11.0, -18.5, 25.0, -36.0],
              [0.5, -11.5, 19.0, 37.0],
              [-10.0, -19.0, 21.0, -42.0],
              [20.0, -22.0, 8.0, -20.0],
              [0.6, -13.0, 26.0, 21.0],
              [-40.0, -20.0, 22.0, -50.0]
              ])

# 三类样本均值
X_bar = np.mean(X, axis=0)
Y_bar = np.mean(Y, axis=0)
Z_bar = np.mean(Z, axis=0)
print(f'X_bar = {X_bar}\nY_bar = {Y_bar}\nZ_bar = {Z_bar}')

# 三类样本协方差
X_S = np.cov(X, rowvar=False)
Y_S = np.cov(Y, rowvar=False)
Z_S = np.cov(Z, rowvar=False)
```

```python
# 方差求逆
X_S1 = np.linalg.inv(X_S)
Y_S1 = np.linalg.inv(Y_S)
Z_S1 = np.linalg.inv(Z_S)

# 待判样品1、2、3
P1 = np.array([-8.0, -14.0, 16.0, 56.0])
P2 = np.array([92.2, -17.0, 18.0, 3.0])
P3 = np.array([-14.0, -18.5, 25.0, -36.0])
```

```python
# 计算待判样品与三类观测样品的马氏距离
# 待判样品1
tmp1 = np.dot(P1 - X_bar, X_S1)
d1_X_square = np.dot(tmp1, P1 - X_bar)

tmp1 = np.dot(P1 - Y_bar, Y_S1)
d1_Y_square = np.dot(tmp1, P1 - Y_bar)

tmp1 = np.dot(P1 - Z_bar, Z_S1)
d1_Z_square = np.dot(tmp1, P1 - Z_bar)

print(f'd1_X_square = {d1_X_square}\nd1_Y_square = {d1_Y_square}\nd1_Z_square = {d1_Z_square}\n')

# 待判样品2
tmp2 = np.dot(P2 - X_bar, X_S1)
d2_X_square = np.dot(tmp2, P2 - X_bar)

tmp2 = np.dot(P2 - Y_bar, Y_S1)
d2_Y_square = np.dot(tmp2, P2 - Y_bar)

tmp2 = np.dot(P2 - Z_bar, Z_S1)
d2_Z_square = np.dot(tmp2, P2 - Z_bar)

print(f'd2_X_square = {d2_X_square}\nd2_Y_square = {d2_Y_square}\nd2_Z_square = {d2_Z_square}\n')

# 待判样品3
tmp3 = np.dot(P3 - X_bar, X_S1)
d3_X_square = np.dot(tmp3, P3 - X_bar)

tmp3 = np.dot(P3 - Y_bar, Y_S1)
d3_Y_square = np.dot(tmp3, P3 - Y_bar)

tmp3 = np.dot(P3 - Z_bar, Z_S1)
d3_Z_square = np.dot(tmp3, P3 - Z_bar)

print(f'd3_X_square = {d3_X_square}\nd3_Y_square = {d3_Y_square}\nd3_Z_square = {d3_Z_square}\n')
```