问题 1：

1 问题描述：

在一维模式特征空间的两类问题中，两类模式的概率密度分布函数分别为 $N(0,\sigma^2)$ 和 $N(1,\sigma^2)$。试证明最小平均风险的分类阈值为 $x_0=\frac{1}{2}-\sigma_2\ln\frac{C_{12}\Pr(\omega_2)}{C_{21}\Pr(\omega_1)}$，其中假设 $C_{11}=C_{22}=0$。

2 证明：

设两类模式分别为 ω_1 和 ω_2，其概率密度分布函数分别为 $N(0,\sigma^2)$ 和 $N(1,\sigma^2)$，均满足正态分布。

由最小化贝叶斯风险决策规则：

$$Decide \quad x\in\omega_1 \quad if \quad \frac{p(x\mid w_1)}{p(x\mid w_2)}>\frac{(C_{12}-C_{22})}{(C_{21}-C_{11})}\cdot\frac{\Pr(\omega_2)}{\Pr(\omega_1)}$$

可知当两边相等时求出的 x 即为最小平均风险的分类阈值 x_0。

又根据正态分布的概率密度函数：

$$p(x\mid w_i)=\frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(x-\mu)^2}{2\sigma^2})$$

可得

$$\frac{p(x\mid w_1)}{p(x\mid w_2)}=\frac{(C_{12}-C_{22})}{(C_{21}-C_{11})}\cdot\frac{\Pr(\omega_2)}{\Pr(\omega_1)}$$

$$\frac{\frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{x^2}{2\sigma^2})}{\frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(x-1)^2}{2\sigma^2})}=\frac{(C_{12}-C_{22})}{(C_{21}-C_{11})}\cdot\frac{\Pr(\omega_2)}{\Pr(\omega_1)}$$

由假设 $C_{11}=C_{22}=0$，并两边取对数得

$$-\frac{x^2}{2\sigma^2}+\frac{(x-1)^2}{2\sigma^2}=\ln\frac{C_{12}\Pr(\omega_2)}{C_{21}\Pr(\omega_1)}$$

$$\frac{1-2x}{2\sigma^2}=\ln\frac{C_{12}\Pr(\omega_2)}{C_{21}\Pr(\omega_1)}$$

整理后

$$x_0 = \frac{1}{2} - \sigma_2 \ln \frac{C_{12} \Pr(\omega_2)}{C_{21} \Pr(\omega_1)}$$ 得证。

问题 2：

1 问题描述：

1. 生成两个各包含 N=1000 个二维随机矢量的数据集合 X 和 X'。数据集合中随机矢量来自于三个分布模型，它们分别满足均值矢量 $m_1 = [1,1]^T$ 、 $m_2 = [4,4]^T$ 和 $m_3 = [8,1]^T$ 和协方差矩阵 $S_1 = S_2 = S_3 = 2I$ ，其中 I 是 2×2 的单位矩阵。在生成数据集合 X 时，假设来自三个分布模型的先验概率相同；而在生成数据集合 X'时，先验概率分别为 0.6、0.3 和 0.1。
2. 画出所生成的两个数据集合中随机矢量的散布图。
3. 在两个数据集合上分别应用“似然率测试规则”、“贝叶斯风险规则”（其中 $C_{12} = 2, C_{13} = 3, C_{23} = 2.5, C_{11} = C_{22} = C_{33} = 0, C_{21} = C_{31} = C_{32} = 1$）、“最大后验概率规则”和“最短欧氏距离规则”进行模式分类实验，给出实验过程设计和实验结果。
4. 对每个数据集合给出上述每种分类规则的分类错误率，分析结果并给出你的结论。

2 基本思路：

1. 生成数据集合 X：
来自三个分布模型的先验概率相同，来自三个分布模型的数据量相等，分别取 333,333,334.
生成数据集合 X'：
先验概率分别为 0.6、0.3 和 0.1，来自三个分布模型的数据量分别取 600,300,100.
2. 画出数据集合 X 和 X'的散点图。

3. 在 d 维模式特征空间中， ω_i 类模式特征 x 的似然函数遵循多元正态概率密度分布函数：

$$p(x \mid \omega_i) = \frac{1}{(2\pi)^{d/2} \mid \Sigma_i \mid^{1/2}} \exp(-\frac{1}{2}(x - \mu_i)^T \sum_i^{-1} (x - \mu_i))$$

其中： ω_i 类样本的均值 μ_i ， $d \times d$ 维协方差矩阵 Σ_i ， $|\Sigma_i|$ 表示 Σ_i 的行列式。

4. 根据以下规则在三个分布模型下分别对数据集合 X 和 X'进行分类并统计错误率。
<u>似然率测试规则：</u>

$$Decide \quad x \in \omega_i \quad if \quad \Lambda(x) = \frac{p(x \mid w_i)}{p(x \mid w_j)} > \frac{\Pr(\omega_j)}{\Pr(\omega_i)}, \forall j \neq i$$

其中： $p(x \mid w_i)$ 为数据 x 的似然函数， $\Pr(\omega_i)$ 为类别 ω_i 的先验概率。

最小化贝叶斯风险决策规则：

$$Decide \quad x \in \omega_1 \quad if \quad \frac{p(x \mid w_1)}{p(x \mid w_2)} > \frac{(C_{12} - C_{22})}{(C_{21} - C_{11})} \cdot \frac{\Pr(\omega_2)}{\Pr(\omega_1)}$$

最大后验概率决策规则：

$$Decide \quad x \in \omega_i \quad if \quad \Pr(\omega_i \mid x) > \Pr(\omega_j \mid x), \forall j \neq i$$

其中： $p(w_i \mid x)$ 为类别 ω_i 的后验概率 $p(w_i \mid x) = \frac{p(x \mid \omega_i)\Pr(\omega_i)}{p(x)}$

最短欧氏距离规则：

$$\omega^* = \arg_{\omega_i} \min g_i(x)$$

其中， $g_i(x)$ 为数据 x 到类别 ω_i 的欧氏距离 $g_i(x) = (x - \mu_i)^T (x - \mu_i)$

3 算法

3.1 算法描述

1. 用 numpy.random.multivariate_normal 在三类分布模型下按照相应数量随机生成正态分布矩阵，组成集合 X 和 X′。
2. 用 plot 函数画出数据集合 X 和 X′的散点图。
3. 设三个类别模式分别为 ω_1， ω_2， ω_3。

集合 X 中来自三个分布模型的先验概率相同，即 $\Pr(\omega_1) = \Pr(\omega_2) = \Pr(\omega_3)$。

集合 X′中来自三个分布模型的先验概率分别为 0.6、0.3 和 0.1，即 $\Pr(\omega_1) = 0.6$，

$\Pr(\omega_2) = 0.3$， $\Pr(\omega_3) = 0.1$。

“似然率测试规则”实验：

集合 X 只需要比较似然函数的大小即可。对集合 X 中的一个数据 x，分别计算在三类分布模型下的似然函数值 $p(x \mid \omega_1)$、 $p(x \mid \omega_2)$、 $p(x \mid \omega_3)$。若 $p(x \mid \omega_1) > p(x \mid \omega_2)$ 且 $p(x \mid \omega_1) > p(x \mid \omega_3)$，则 x 属于 ω_1；若 $p(x \mid \omega_2) > p(x \mid \omega_1)$ 且 $p(x \mid \omega_2) > p(x \mid \omega_3)$，则 x 属于 ω_2；若 $p(x \mid \omega_3) > p(x \mid \omega_1)$ 且 $p(x \mid \omega_3) > p(x \mid \omega_2)$，则 x 属于 ω_3。

对集合 X′中的一个数据 x′，分别计算在三类分布模型下的似然函数值 $p(x' \mid \omega_1)$、$p(x' \mid \omega_2)$、$p(x' \mid \omega_3)$。若 $\frac{p(x' \mid w_1)}{p(x' \mid w_2)} > \frac{\Pr(\omega_2)}{\Pr(\omega_1)}$ 且 $\frac{p(x' \mid w_1)}{p(x' \mid w_3)} > \frac{\Pr(\omega_3)}{\Pr(\omega_1)}$，则 x′属于 ω_1；若 $\frac{p(x' \mid w_2)}{p(x' \mid w_1)} > \frac{\Pr(\omega_1)}{\Pr(\omega_2)}$ 且 $\frac{p(x' \mid w_2)}{p(x' \mid w_3)} > \frac{\Pr(\omega_3)}{\Pr(\omega_2)}$，则 x′属于 ω_2；若 $\frac{p(x' \mid w_3)}{p(x' \mid w_1)} > \frac{\Pr(\omega_1)}{\Pr(\omega_3)}$ 且

$\frac{p(x'|w_3)}{p(x'|w_2)} > \frac{\Pr(\omega_2)}{\Pr(\omega_3)}$，则 x′属于 ω_3。

5. “最小化贝叶斯风险决策规则”实验：

对集合 X 中的一个数据 x，分别计算在三类分布模式下的条件风险

$R_i = C_{i1}p(x|\omega_1) + C_{i2}p(x|\omega_2) + C_{i3}p(x|\omega_3)$，x 属于条件风险最小的那一类。

对集合 X′中的一个数据 x′，分别计算在三类分布模式下的条件风险

$R_i = C_{i1}p(x'|\omega_1)\Pr(\omega_1) + C_{i2}p(x'|\omega_2)\Pr(\omega_2) + C_{i3}p(x'|\omega_3)\Pr(\omega_3)$，x′属于条件风险最小的那一类。

6. “最大后验概率决策规则”实验：

对集合 X 中的一个数据 x，分别计算在三类分布模型下的判别式函数值

$g_i(x) = -\frac{1}{2}(x-\mu_i)^T\sum_i^{-1}(x-\mu_i) - \frac{1}{2}\ln(|\Sigma_i|) + \ln(\Pr(\omega_i))$，x 属于函数值最大的那一类。

同理计算集合 X′中的数据 x′的函数值。

7. “最短欧氏距离规则”实验：

对集合 X 中的一个数据 x，分别计算欧氏距离 $g_i(x) = (x-\mu_i)^T(x-\mu_i)$，x 属于距离最小的那一类。对集合 X′中的一个数据 x′，分别计算欧氏距离 $g_i(x) = (x'-\mu_i)^T(x'-\mu_i)$，x′属于距离最小的那一类。

8. 统计分类错误率：

在上述几种规则的实验中，对于类别模式 ω_1 中的数据 x，若被错误的归到类别模式 ω_2 或 ω_3 时记录一次错误 e_1，类别模式 ω_2 和 ω_3 中的数据类似，最后计算错误率 $P(error) = \frac{e_1+e_2+e_3}{1000}$。

3.2 算法实现

1. 正态概率密度分布函数：

由于三个分布模式的协方差矩阵相同，所以在计算时省略前面的系数 $\frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}}$。

```
def gauss(x, m, s):
    t = x - m
    return np.exp(-1/2 * np.dot(np.dot(np.transpose(t), np.linalg.inv(s)), t))
```

2. 似然率测试规则：

```python
def lrt(x, s, p, j, n):
    # 数据x，均值m，协方差s，先验概率p，模式j是正确的，第j组的个数n
    # 返回：第j组的错误个数
    e = 0
    for i in list(range(0, n)):
        g1 = p * gauss(x[i], m1, s)
        g2 = p * gauss(x[i], m2, s)
        g3 = p * gauss(x[i], m3, s)
        if g1 > g2 and g1 > g3:
            plt.plot(x[i][0], x[i][1], '.r')
            if j != 1:
                e = e + 1
        elif g1 < g2 and g3 < g2:
            plt.plot(x[i][0], x[i][1], '.g')
            if j != 2:
                e = e + 1
        else:
            plt.plot(x[i][0], x[i][1], '.b')
            if j != 3:
                e = e + 1
    return e
```

3. 贝叶斯风险规则：

```python
def bayes(x, s, p, j, n):
    # 数据x，协方差s，先验概率p，模式j是正确的，第j组的个数n
    # 返回：第j组的错误个数
    e = 0
    for i in list(range(0, n)):
        g1 = 0 * gauss(x[i], m1, s) * p + 2 * gauss(x[i], m2, s) * p + 3 * gauss(x[i], m3, s) * p
        g2 = 1 * gauss(x[i], m1, s) * p + 0 * gauss(x[i], m2, s) * p + 2.5 * gauss(x[i], m3, s) * p
        g3 = 1 * gauss(x[i], m1, s) * p + 1 * gauss(x[i], m2, s) * p + 0 * gauss(x[i], m3, s) * p
        if g1 < g2 and g1 < g3:
            plt.plot(x[i][0], x[i][1], '.r')
            if j != 1:
                e = e + 1
        elif g2 < g1 and g2 < g3:
            plt.plot(x[i][0], x[i][1], '.g')
            if j != 2:
                e = e + 1
        else:
            plt.plot(x[i][0], x[i][1], '.b')
            if j != 3:
                e = e + 1
    return e
```

4. 最大后验概率规则：

```
# 最大后验概率决策规则 #
def mmap(x, s, p, j, n):
    # 数据x，均值m，协方差s，先验概率p，模式j是正确的，第j组的个数n
    # 返回：第j组的错误个数

    # 判别式函数
    def map_gauss(_x, _m, _s, _p):
        t = _x - _m
        return -1 / 2 * np.dot(np.dot(np.transpose(t), np.linalg.inv(_s)), t) - 1 / 2 * fast_logdet(_s) + math.log(
            _p, math.e)
```

```
    e = 0
    for i in list(range(0, n)):
        g1 = p * map_gauss(x[i], m1, s, p)
        g2 = p * map_gauss(x[i], m2, s, p)
        g3 = p * map_gauss(x[i], m3, s, p)
        if g1 > g2 and g1 > g3:
            plt.plot(x[i][0], x[i][1], '.r')
            if j != 1:
                e = e + 1
        elif g1 < g2 and g3 < g2:
            plt.plot(x[i][0], x[i][1], '.g')
            if j != 2:
                e = e + 1
        else:
            plt.plot(x[i][0], x[i][1], '.b')
            if j != 3:
                e = e + 1
    return e
```

5. 最短欧式距离规则：

```
def ed(x, j, n):
    # 数据x，模式j是正确的，第几组的个数n
    # 返回：第j组的错误个数
    e = 0
    for i in list(range(0, n)):
        t1 = x[i] - m1
        t2 = x[i] - m2
        t3 = x[i] - m3
        g1 = np.dot(t1, np.transpose(t1))
        g2 = np.dot(t2, np.transpose(t2))
        g3 = np.dot(t3, np.transpose(t3))
        if g1 < g2 and g1 < g3:
            plt.plot(x[i][0], x[i][1], '.r')
            if j != 1:
                e = e + 1
        elif g2 < g1 and g2 < g3:
            plt.plot(x[i][0], x[i][1], '.g')
            if j != 2:
                e = e + 1
        else:
            plt.plot(x[i][0], x[i][1], '.b')
            if j != 3:
                e = e + 1
    return e
```

完整代码见文件 p2. py。

4 结果与分析

4.1 实验步骤

按照要求生成数据集合 X 和 X’，并画出散点图。在两个集合上分别使用“似然率测试规则”、“贝叶斯风险规则”、“最大后验概率规则”和“最短欧氏距离规则”进行分类实验，并统计每种规则的分类错误率。以上步骤重复 5 次，计算平均错误率。

4.2 实验结果

第一次实验生成的数据集合散点图如图 1 所示，各规则分类结果如图 2~图 5 所示。5 次实验的错误率统计结果如表 1 所示。

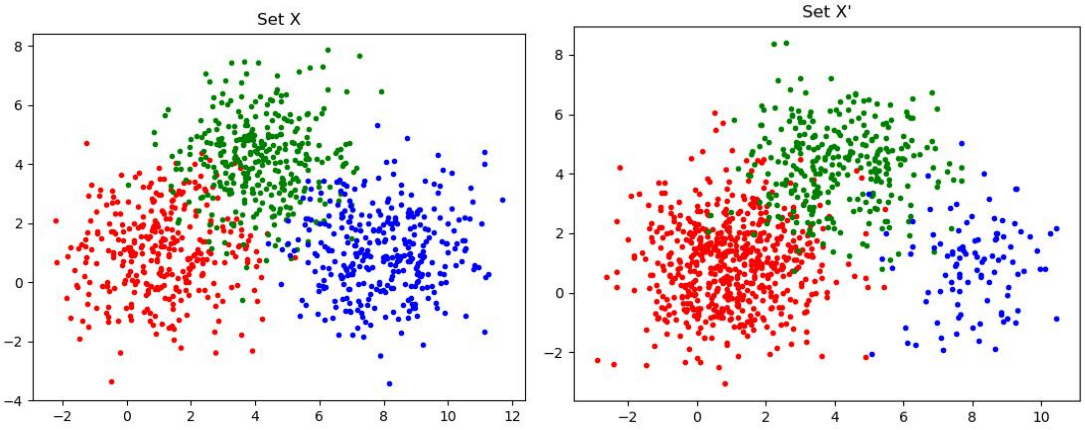

图 1 数据集合 X 和 X’的散点图

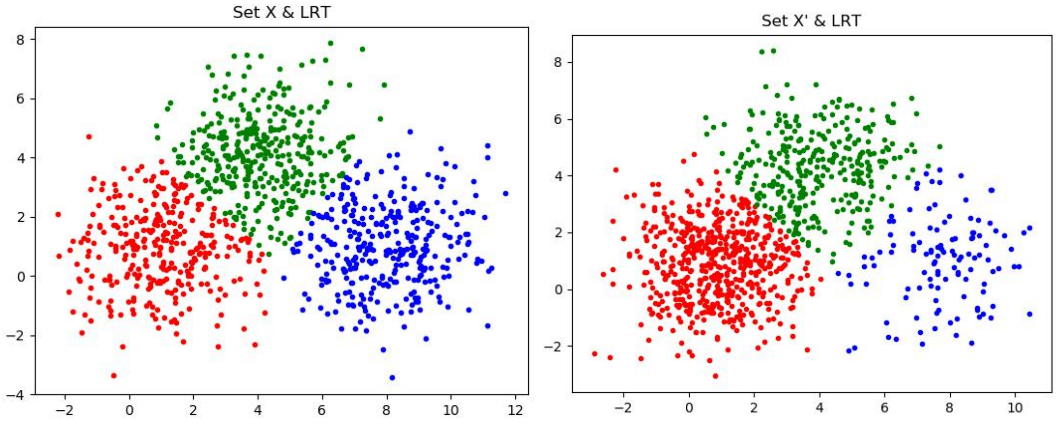

图 2 似然率测试规则分类结果散点图

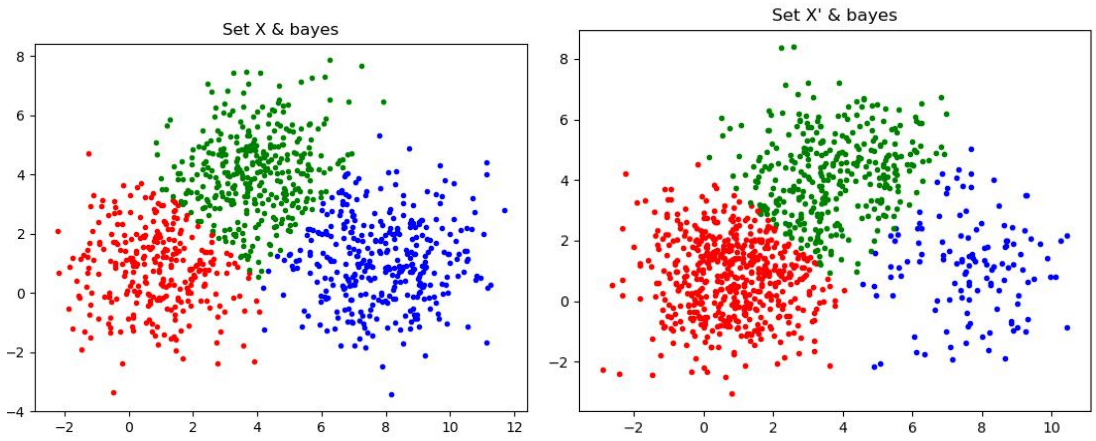

图 3 贝叶斯风险规则分类结果散点图

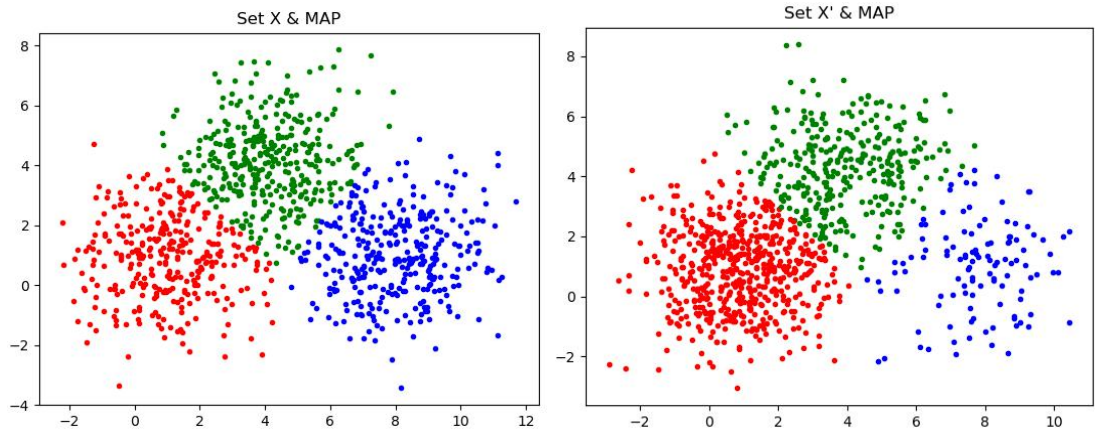

图 4 最大后验概率规则分类结果散点图

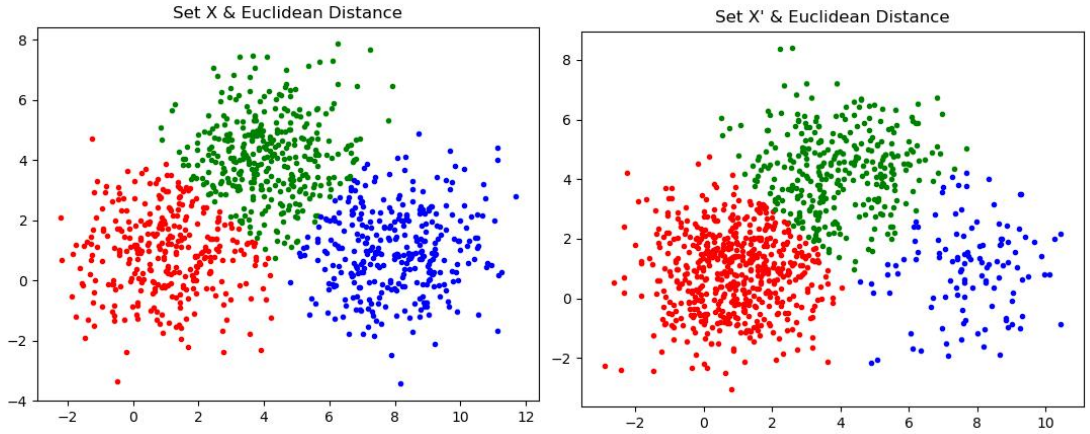

图 5 最短欧氏距离规则分类结果散点图

表 1 5 次实验的错误率统计结果

分类错误率		似然率测试规则	贝叶斯风险规则	最大后验概率规则	最短欧氏距离规则
数据集合 X	1	7.80%	7.60%	7.80%	7.80%
	2	8.20%	8.30%	8.20%	8.20%
	3	6.60%	6.60%	6.60%	6.60%
	4	7.10%	7.90%	7.10%	7.10%
	5	6.70%	7.30%	6.70%	6.70%
	平均	7.28%	7.45%	7.28%	7.28%
数据集合 X’	1	7.19%	7.39%	7.19%	7.70%
	2	6.80%	7.39%	6.80%	7.50%
	3	6.10%	6.20%	6.10%	6.10%
	4	7.39%	6.90%	7.39%	7.00%
	5	7.19%	7.70%	7.19%	7.80%
	平均	6.93%	7.12%	6.93%	7.22%

4.3 结果分析

数据集合 X 和 X’是随机生成的，因此每次实验的结果是不同的，整体错误率都在 6%~8%之间。由表 1 实验结果可以看出，在先验概率不变的情况下，“似然率测试规则”和“最大后验概率规则”是等价的，因此错误率也相同，集合 X 和 X’分别为 7.28%和 6.93%。

因为集合 X 在三个分布模式下的先验概率和协方差矩阵均相同，所以“似然率测试规则”、“最大后验概率规则”和“最短欧氏距离规则”也是等价的。集合 X’在三个分布模式下虽然协方差矩阵均相同，但是先验概率不同，因此“最短欧氏距离规则”与“似然率测试规则”和“最大后验概率规则”的错误率不相同。

综合集合 X 和 X’的结果来看，“似然率测试规则”和“最大后验概率规则”的错误率是最优的。因为“贝叶斯风险规则”是追求风险最小，而不是错误次数最小，各类出错的代价不同，所以错误率会略大一些，集合 X 和 X’分别为 7.45%和 7.12%。

在代码方面，我的实现是一边判断每个数据的分类一边统计错误次数，若能采取先画出决策平面，在根据决策平面判断分类的方法可能会更好，但由于时间有限未能来得及实现。