# 《模式识别》课程

# 实 验 报 告



**姓 名： 金家耀**

**专 业：**  人工智能

**学 号： 1193210320**

**江南大学人工智能与计算机学院**

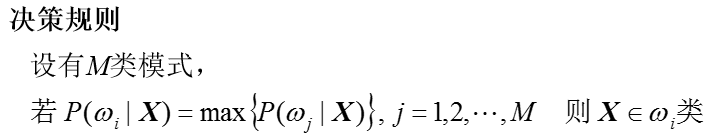
**贝叶斯分类器**

**1实验目的**

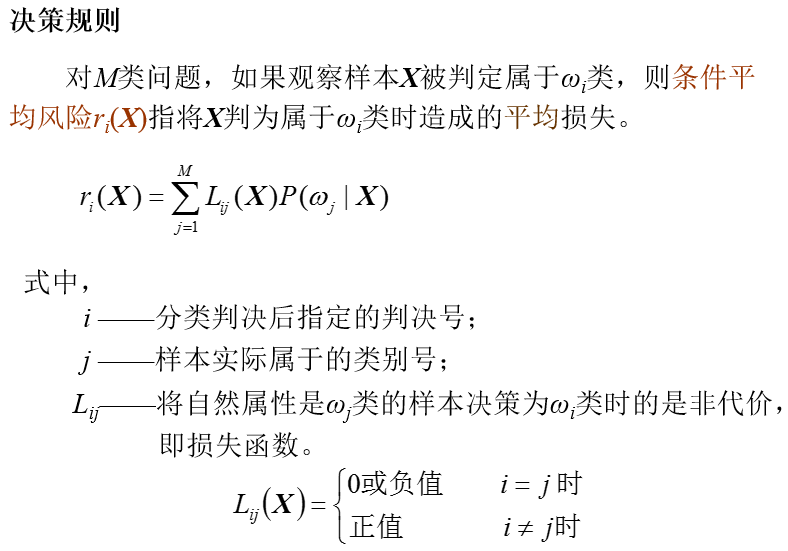
贝叶斯决策是统计决策理论中的一个基本方法，其应用非常广泛。本实验旨在让学生理解最小错误率贝叶斯决策和最小风险贝叶斯决策的基本原理，能够通过分类器的设计和实现对贝叶斯决策理论和算法有一个深刻地认识，体会贝叶斯决策在模式识别中的作用。

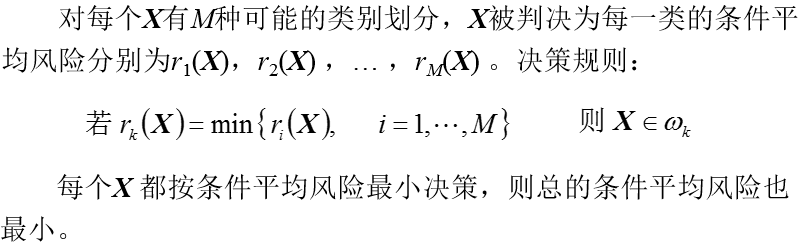
**2实验原理**

最小错误率贝叶斯决策基本思想：

 以各种错误分类所造成的错误率最小为规则，进行分类决策。

最小风险贝叶斯决策基本思想：

 以各种错误分类所造成的平均风险最小为规则，进行分类决策。



**3实验内容**

假定某个局部区域细胞识别中正常（）和非正常（）两类先验概率分别为

正常状态：P（）=0.9；

异常状态：P（）=0.1。

现有一系列待观察的细胞，其观察值为：

-3.9847 -3.5549 -1.2401 -0.9780 -0.7932 -2.8531

-2.7605 -3.7287 -3.5414 -2.2692 -3.4549 -3.0752

-3.9934 2.8792 -0.9780 0.7932 1.1882 3.0682

-1.5799 -1.4885 -0.7431 -0.4221 -1.1186 4.2532

已知类条件概率密度曲线如下图：

类条件概率密度、服从正态分布，分别为(-2,0.25)、(2,4)试对观察的结果进行分类。

**4实验要求**

* + - 1. 完成最小错误率贝叶斯决策分类器的设计，要求程序相应语句有说明文字。
      2. 根据例子画出后验概率的分布曲线和分类结果。
      3. 如果是最小风险贝叶斯决策，决策表如下：

|  |  |  |
| --- | --- | --- |
| 状态  决策 |  |  |
| α1 | 0 | 6 |
| α2 | 1 | 0 |

请重新设计程序，画出相应的后验概率的分布曲线和分类结果,并比较两个结果。

**5实验代码和结果**

（1）定义正态分布

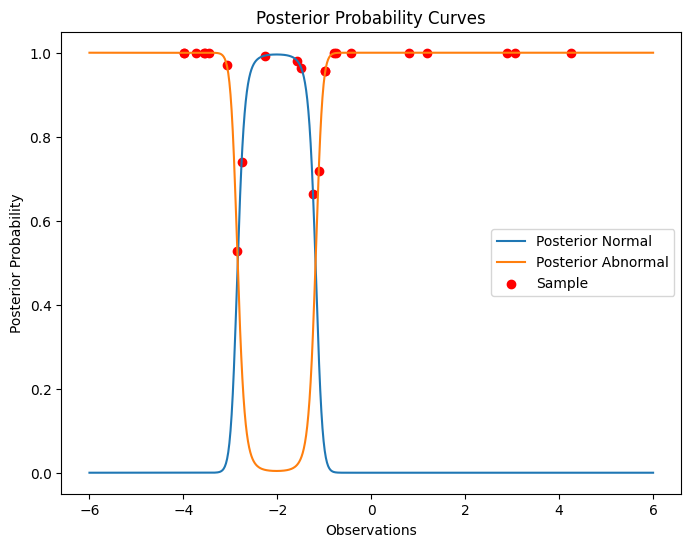
import numpy as np  
import matplotlib.pyplot as plt  
  
# 定义正态分布的概率密度函数  
def normal\_pdf(x, mean, std):  
 return 1 / (std \* np.sqrt(2 \* np.pi)) \* np.exp(-0.5 \* ((x - mean) / std) \*\* 2)

（2）最小错误率贝叶斯决策分类器

# 观察值  
observations = np.array([-3.9847, -3.5549, -1.2401, -0.9780, -0.7932, -2.8531,  
 -2.7605, -3.7287, -3.5414, -2.2692, -3.4549, -3.0752,  
 -3.9934, 2.8792, -0.9780, 0.7932, 1.1882, 3.0682,  
 -1.5799, -1.4885, -0.7431, -0.4221, -1.1186, 4.2532])  
  
# 先验概率  
prior\_normal = 0.9  
prior\_abnormal = 0.1  
  
# 类条件概率密度参数  
mean\_normal, std\_normal = -2, 0.25  
mean\_abnormal, std\_abnormal = 2, 4  
  
# 计算两个类别的p(x|wi)  
px\_normal = normal\_pdf(observations, mean\_normal, std\_normal)  
px\_abnormal = normal\_pdf(observations, mean\_abnormal, std\_abnormal)  
  
# 计算p(x)=sum(p(x|wi)\*wi)  
px = px\_normal \* prior\_normal + px\_abnormal \* prior\_abnormal  
  
# 计算后验概率p(wi)  
p\_normal = px\_normal \* prior\_normal / px  
p\_abnormal = px\_abnormal \* prior\_abnormal / px  
# 计算最小错误率下题目样本选取的后验概率  
judge = np.max(np.vstack([p\_normal, p\_abnormal]), axis=0)  
  
# 根据最小错误率预测类别  
results = np.zeros(observations.shape[0])  
results[p\_normal > p\_abnormal] = 1  
  
results

（3）最小错误率贝叶斯后验概率的分布曲线和分类结果

# 绘制后验概率图  
x = np.linspace(-6, 6, num=1000) # 选择适当的数量，这里选择1000个点  
  
# 计算两个类别的p(x|wi)  
px\_normal\_ = normal\_pdf(x, mean\_normal, std\_normal)  
px\_abnormal\_ = normal\_pdf(x, mean\_abnormal, std\_abnormal)  
  
# 计算p(x)=sum(p(x|wi)\*wi)  
px\_ = px\_normal\_ \* prior\_normal + px\_abnormal\_ \* prior\_abnormal  
  
# 计算后验概率p(wi)  
p\_normal\_ = px\_normal\_ \* prior\_normal / px\_  
p\_abnormal\_ = px\_abnormal\_ \* prior\_abnormal / px\_  
  
# 绘制后验概率曲线  
plt.figure(figsize=(8, 6))  
plt.plot(x, p\_normal\_, label='Posterior Normal')  
plt.plot(x, p\_abnormal\_, label='Posterior Abnormal')  
plt.scatter(observations, judge, label='Sample', c='r')  
plt.xlabel('Observations')  
plt.ylabel('Posterior Probability')  
plt.legend()  
plt.title('Posterior Probability Curves')  
plt.show()



（4）最小风险贝叶斯决策

# 最小风险贝叶斯决策的决策表  
cost\_matrix = np.array([[0, 6], [1, 0]])  
r1 = cost\_matrix[0, 0] \* p\_normal + cost\_matrix[0, 1] \* p\_abnormal  
r2 = cost\_matrix[1, 0] \* p\_normal + cost\_matrix[1, 1] \* p\_abnormal  
  
results2 = np.zeros(observations.shape[0])  
results2[r1 < r2] = 1  
  
results2

（5）最小风险贝叶斯决策后验概率的分布曲线和分类结果与前者的对比

# 计算最小风险值下题目样本选取的后验概率  
judge2 = np.where(r1 < r2, p\_normal, p\_abnormal)  
  
# 绘制后验概率曲线  
plt.figure(figsize=(8, 6))  
plt.plot(x, p\_normal\_, label='Posterior Normal')  
plt.plot(x, p\_abnormal\_, label='Posterior Abnormal')  
plt.scatter(observations, judge, label='Sample Min Error', c='r')  
plt.scatter(observations, judge2, label='Sample Min Risk', c='g', marker='x')  
plt.xlabel('Observations')  
plt.ylabel('Posterior Probability')  
plt.legend()  
plt.title('Posterior Probability Curves')  
plt.show()

