# 《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**

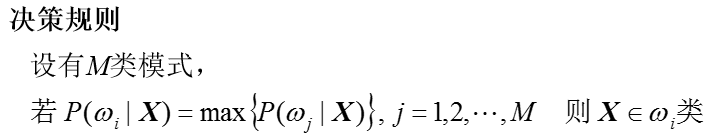
**贝叶斯分类器**

**1实验目的**

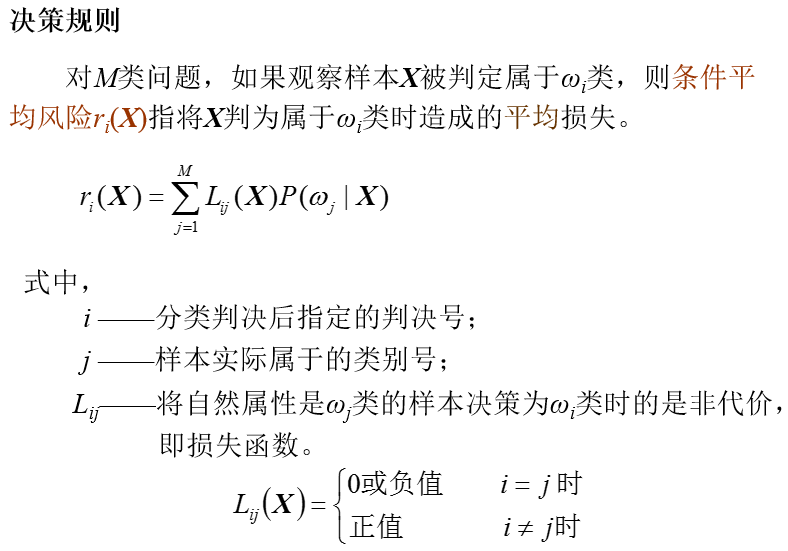
贝叶斯决策是统计决策理论中的一个基本方法，其应用非常广泛。本实验旨在让学生理解最小错误率贝叶斯决策和最小风险贝叶斯决策的基本原理，能够通过分类器的设计和实现对贝叶斯决策理论和算法有一个深刻地认识，体会贝叶斯决策在模式识别中的作用。

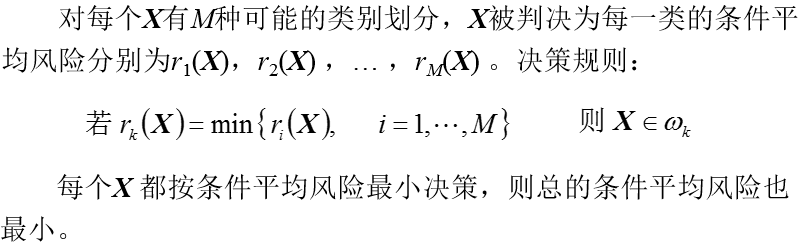
**2实验原理**

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**

代码已上传至<https://github.com/shinejjy/PatternRecognition/tree/master/sy4>

（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

这段代码实现了一个简单的贝叶斯分类器，用于对给定的一组观察值进行分类。首先，我们定义了观察值数组 observations，其中包含待分类的样本数据。

接下来，设定了两个类别的先验概率 prior\_normal 和 prior\_abnormal，这些概率表示在没有观察到任何数据信息的情况下，一个样本属于正常类别和异常类别的概率。在这里，正常类别的先验概率为 0.9，异常类别的先验概率为 0.1。

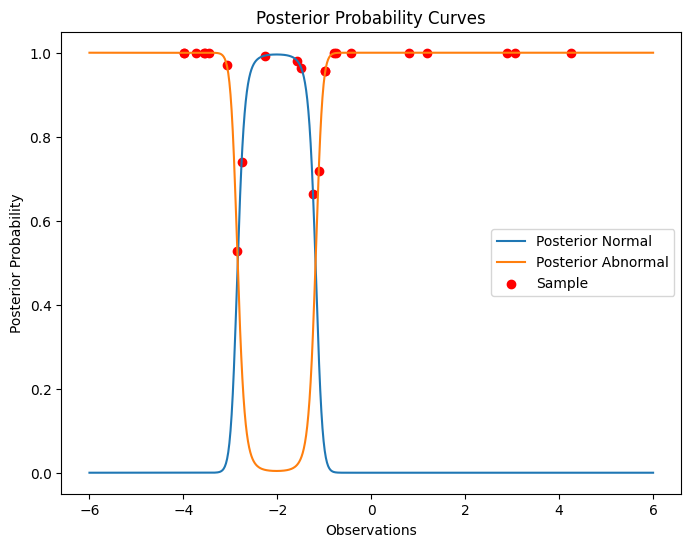
然后，通过设定正常和异常类别的高斯分布参数，包括均值 mean\_normal 和 mean\_abnormal，以及标准差 std\_normal 和 std\_abnormal，用于表示每个类别的数据分布。

在计算观察值在正常和异常类别下的条件概率密度函数之后，我们计算每个观察值的边际概率密度，进而得到每个观察值在两个类别下的后验概率。这里使用了贝叶斯公式来更新分类的概率。

最后，通过比较每个观察值在正常和异常类别下的后验概率，选择概率较大的类别作为最终的分类结果。将分类结果存储在 results 中，其中1表示正常，0表示异常。

（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()



如上图所示，绘制出后验概率的分布曲线和分类结果。从图中可以看出，分类结果基本上按照后验概率来进行划分。

（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

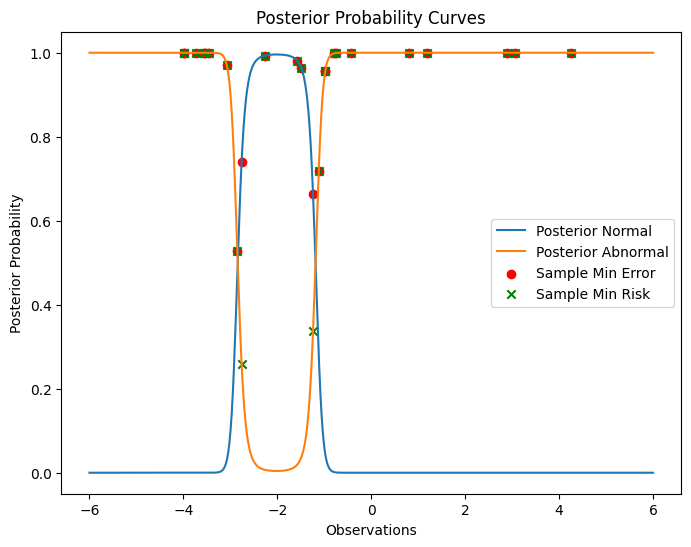
这段代码实现了最小风险贝叶斯决策的过程。首先，定义了一个代价矩阵 cost\_matrix，其中指定了将真实类别为 0 的样本误分类为类别 1 的代价为 6，将真实类别为 1 的样本误分类为类别 0 的代价为 1。

接下来，根据已经计算得到的正常和异常类别下每个数据点的后验概率，计算了每个数据点的期望风险。这里 r1 表示将样本分类为正常的期望风险，r2 表示将样本分类为异常的期望风险。期望风险的计算考虑了代价矩阵和后验概率的影响。

最后，根据最小期望风险原则，对每个数据点进行决策，将期望风险较小的类别作为最终的决策结果。决策结果存储在 results2 中，其中元素为 0 表示正常，元素为 1 表示异常。

（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()



如上图所示，分别绘制出最小错误率和最小风险的后验概率的分布曲线和分类结果，图中红色的圆表示最小错误率贝叶斯分类结果，绿色叉表示最小风险贝叶斯分类结果。由于将异常细胞识别为正常细胞的风险远远大于其他风险，从图中很容易可以看出最小风险贝叶斯分类器识别正常细胞的范围缩小了。

**6 实验心得**

在这次实验中，我首次深入学习了最小错误率和最小风险的贝叶斯决策分类器。通过实现这两种分类器，我对贝叶斯决策理论有了更为深刻的理解。这种理论的核心思想是通过考虑观察值在不同类别下的条件概率以及代价矩阵，以最小化错误率或期望风险为目标进行决策。这为处理实际问题中不同类别代价不同的情况提供了一种更灵活的决策方式。

在实验过程中，通过绘制后验概率的分布曲线，我直观地了解了不同观察值在正常和异常类别下的概率分布情况。这为我理解分类器如何根据后验概率进行决策提供了直观的参考。

通过比较最小错误率和最小风险两种贝叶斯决策方法，我进一步认识到最小风险决策的优势。最小风险决策能够更好地适应实际需求，因为它考虑了不同类别的代价，可以在某些特殊情况下灵活地调整决策边界。

在编写实验代码的过程中，我提高了对贝叶斯决策理论的实际操作能力。通过可视化工具展示结果，我更清晰地理解了算法的输出和决策过程。

最后，通过对比两种不同决策方法的结果，我更全面地了解了它们的优势和劣势。最小错误率适用于不同类别代价相对平等的情况，而最小风险贝叶斯决策则更适用于代价不同的情况，体现了决策问题的灵活性。