

PRML 第一次编程作业

两类正态分布模式贝叶斯分类程序

黄磊 702 2022E8013282156

1. 程序为 [Bayes_of_Gaussian_with_2cls.py](#) 文件，具体测试说明见文档内注释。
2. 求解两类正态分布模式贝叶斯分界面方程的基本程序：

2.1 分别求出两类中各样本的均值向量：

$$\mathbf{m}_i = \frac{1}{N_i} \sum_{j=0}^{N_i} \mathbf{x}_{ij}, \quad i = 1, 2$$

2.2 求出各类协方差矩阵：

$$C_i = \frac{1}{N_i} \sum_{j=0}^{N_i} (\mathbf{x}_{ij} - \mathbf{m}_i)(\mathbf{x}_{ij} - \mathbf{m}_i)^T, \quad i = 1, 2$$

2.3 正态分布模式的贝叶斯判别函数：属于类别 ω_i 的判别函数：

$$g_i(\mathbf{x}) = \ln P(\omega_i) - \frac{1}{2} \ln |C_i| - \frac{1}{2} (\mathbf{x} - \mathbf{m}_i)^T C_i^{-1} (\mathbf{x} - \mathbf{m}_i), \quad i = 1, 2$$

2.4 判别函数即为：

$$g(\mathbf{x}): g_1(\mathbf{x}) - g_2(\mathbf{x}) = 0$$

当 $g(\mathbf{x}) > 0$ ，表示该测试点分为类 ω_1 ；

当 $g(\mathbf{x}) < 0$ ，表示该测试点分为类 ω_2 ；

当 $g(\mathbf{x}) = 0$ ，表示该测试点落在分界面上；

3. 程序主体为函数：`Bayes_Gaussian_with2cls(w1, w2, P_w1, P_w2, x)`，具体参数含义见代码注释。（主要函数截图见文档最后附件，具体函数请参照代码文件）

4. 测试

4.1 测例 1:作业题

设以下模式类别具有正态概率密度函数：

$$\omega_1: \{(0\ 0)^T, (2\ 0)^T, (2\ 2)^T, (0\ 2)^T\}$$

$$\omega_2: \{(4\ 4)^T, (6\ 4)^T, (6\ 6)^T, (4\ 6)^T\}$$

(1) 设 $P(\omega_1) = P(\omega_2) = 1/2$ ，求这两类模式之间的贝叶斯判别界面的方程式。

(2) 绘出判别界面。

测试结果见下图，可见其均值向量、协方差矩阵、分类面函数均与理论符合，求解正确。判别界面绘制正确。

带入测试点 $\mathbf{y} = [1, 2]$ ，理论可知属于 ω_1 类，可知分类正确。

```
/Users/huanglei/miniforge3/envs/tf/bin/python /Users/huanglei/Documents/PRML/作业/
第一次作业/Bayes_of_Gaussian_with_2cls.py
===== 测例 1 =====

The mean vector(m1) of the samples in category w1 is:
[1. 1.]

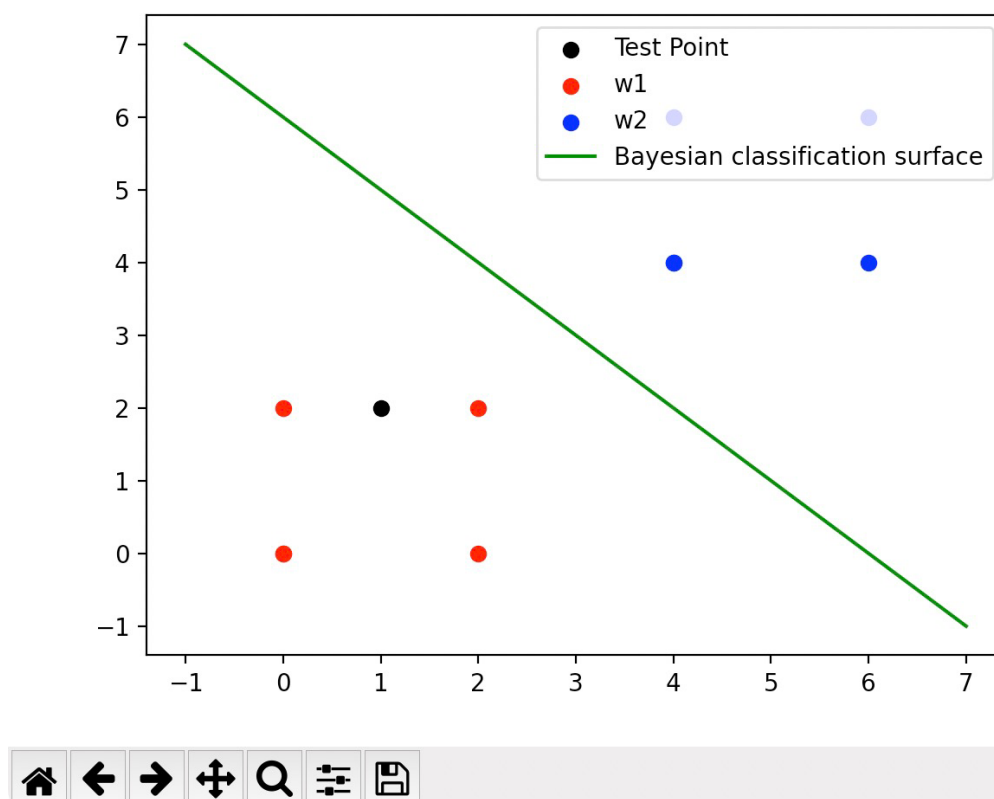
The mean vector(m2) of the samples in category w2 is:
[5. 5.]

The Covariance matrix(C1) in category w1 is:
[[1. 0.]
 [0. 1.]]

The Covariance matrix(C2) in category w2 is:
[[1. 0.]
 [0. 1.]]

The Bayesian classification surface equation is:
-4.0*x1 - 4.0*x2 + 24.0
```

Figure 1



代码主要函数附件:

求解分类界面方程:

```
23 def Bayes_Gaussian_with2cls(w1, w2, P_w1, P_w2, x)\
24     -> sympy.tensor.array.dense_ndim_array.ImmutableDenseNDimArray:
25     m1 = np.mean(w1, axis=0)
26     m2 = np.mean(w2, axis=0)
27     print("\n The mean vector(m1) of the samples in category w1 is: \n", m1)
28     print("\n The mean vector(m2) of the samples in category w2 is: \n", m2)
29     C1 = np.zeros((len(m1), len(m1)))
30     C2 = np.zeros((len(m2), len(m2)))
31     for i in range(len(w1)):
32         c_i = np.dot((w1 - m1)[i].reshape(len(m1), 1), [(w1 - m1)[i]])
33         C1 += c_i
34     ## w1 和 w2 中的学习示例个数可能不同
35     for i in range(len(w2)):
36         c_i = np.dot((w2 - m2)[i].reshape(len(m2), 1), [(w2 - m2)[i]])
37         C2 += c_i
38     C1 = C1 / len(w1)
39     C2 = C2 / len(w2)
40     print("\n The Covariance matrix(C1) in category w1 is: \n", C1)
41     print("\n The Covariance matrix(C2) in category w2 is: \n", C2)
42     inv_C1 = np.linalg.inv(C1)
43     inv_C2 = np.linalg.inv(C2)
44     det_C1 = np.abs(np.linalg.det(C1))
45     det_C2 = np.abs(np.linalg.det(C2))
46     g1 = np.log(P_w1) - np.log(det_C1) / 2 - ((x - m1).dot(inv_C1)).dot((x - m1).reshape(len(m2), 1))/2
47     g2 = np.log(P_w2) - np.log(det_C2) / 2 - ((x - m2).dot(inv_C2)).dot((x - m2).reshape(len(m2), 1))/2
48     Bayes_Classfier = sympy.simplify(g1-g2)[0]
49     if(type(x[0]) == int or type(x[0]) == float):
50         print("\n The Value of Bayesian classification surface equation is: \n", Bayes_Classfier)
51         if (Bayes_Classfier > 0): print("This Vector belongs to w1 \n")
52         if (Bayes_Classfier < 0): print("This Vector belongs to w2 \n")
53         if (Bayes_Classfier == 0): print("This Vector lays on the classification surface\n")
54     else:
55         print("\n The Bayesian classification surface equation is: \n", sympy.simplify(Bayes_Classfier))
56     return Bayes_Classfier
```

绘图函数:

```
56 def plot_2d(w1, w2, bc, new_p=None):
57     low = min(min(min(w1)), min(min(w2)))
58     high = max(max(max(w1)), max(max(w2)))
59     fig = plt.figure()
60     if(new_p): plt.scatter(new_p[0], new_p[1], c="black", label="Test Point")
61     for i in range(len(w1)):
62         p1 = w1[i]
63         p2 = w2[i]
64         if (i == 0):
65             plt.scatter(p1[0], p1[1], c="r", label="w1")
66             plt.scatter(p2[0], p2[1], c="b", label="w2")
67         plt.scatter(p1[0], p1[1], c="r")
68         plt.scatter(p2[0], p2[1], c="b")
69
70     y1 = x1 - (low - 1)
71     y2 = x1 - (high + 1)
72     i1, j1 = sympy.solve([bc, y1], [x1, x2]).items()
73     i2, j2 = sympy.solve([bc, y2], [x1, x2]).items()
74     plt.plot([i1[1], j1[1]], [i2[1], j2[1]], "g-", label="Bayesian classification surface")
75     plt.legend(loc="best")
76     plt.show()
77     plt.close()
```

测试代码:

```
79 print("===== 测试 1 =====")
80 w1 = [[0,0],[2,0],[2,2],[0,2]]
81 w2 = [[4,4],[6,4],[6,6],[4,6]]
82 P_w1, P_w2 = 1/2, 1/2
83 x1, x2 = sympy.symbols("x1 x2")
84 x = [x1, x2]
85 bc = Bayes_Gaussian_with2cls(w1, w2, P_w1, P_w2, x)
86 print("===== 测试 y = [1,2] 属于哪类 =====")
87 y = [1, 2]
88 Bayes_Gaussian_with2cls(w1, w2, P_w1, P_w2, y)
89 ## 画出决策面
90 plot_2d(w1, w2, bc, y)
```