|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

|  |  |
| --- | --- |
| **ФАКУЛЬТЕТ** | **ИУК «Информатика и управление»** |
| **КАФЕДРА** | **ИУК4 «Программное обеспечение ЭВМ,** |
| **информационные технологии»** | |

**ДОМАШНЯЯ РАБОТА №1**

«Применение байесовского подхода принятия решений»

**ДИСЦИПЛИНА:** «Программные системы распознавания и обработки информации»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил: студент гр. ИУК4-31М | |  |  | ( | Сафронов Н.С, | ) |
|  |  |  | (подпись) |  | (Ф.И.О.) |  |
| Проверил: | |  |  | ( | Гагарин Ю.Е. | ) |
|  |  |  | (подпись) |  | (Ф.И.О.) |  |

|  |  |
| --- | --- |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | |
|  | - Балльная оценка:  - Оценка: |

Калуга, 2025

**Цель:**

Формирование практических навыков использования байесовского подхода принятия решений в случае классификации двух классов, условные плотности вероятностей которых соответствуют одномерному и двумерному нормальному закону распределения.

**Задачи:**

1. Определение границы разделения классов;

2. Построение графиков условных плотностей вероятностей и границ разделения классов;

3. Использование формулы Байеса для определения апостериорных вероятностей;

4. Определение вероятности ошибки классификации.

**Задание**

1. Использование байесовского классификатора для двух классов.

Для первого класса **,** из определенного диапазона, задать случайным образом 50 значений признака .

Для второго класса , из определенного диапазона, задать случайным образом 70 значений признака .

Предполагая, что условные плотности вероятности и соответствуют нормальному закону распределения ,

определить значения параметров для двух классов.

Найти границу разделения классов.

Построить графики условных плотностей вероятностей и границы разделения классов.

Задать значение признака и предполагая, что априорные вероятности равны , определить апостериорные вероятности по формуле Байеса

.

Определить к какому классу относится значение признака .

Рассчитать вероятность ошибки классификации.

1. Классификация двух классов по двум признакам.

Для первого класса , из определенного диапазона, задать случайным образом 50 значений признаков .

Для второго класса , из определенного диапазона, задать случайным образом 70 значений признаков .

Предполагая, что условные плотности вероятности и соответствуют двумерному нормальному закону распределения

и для случая, когда определить значения параметров для двух классов.

Определить границу разделения классов.

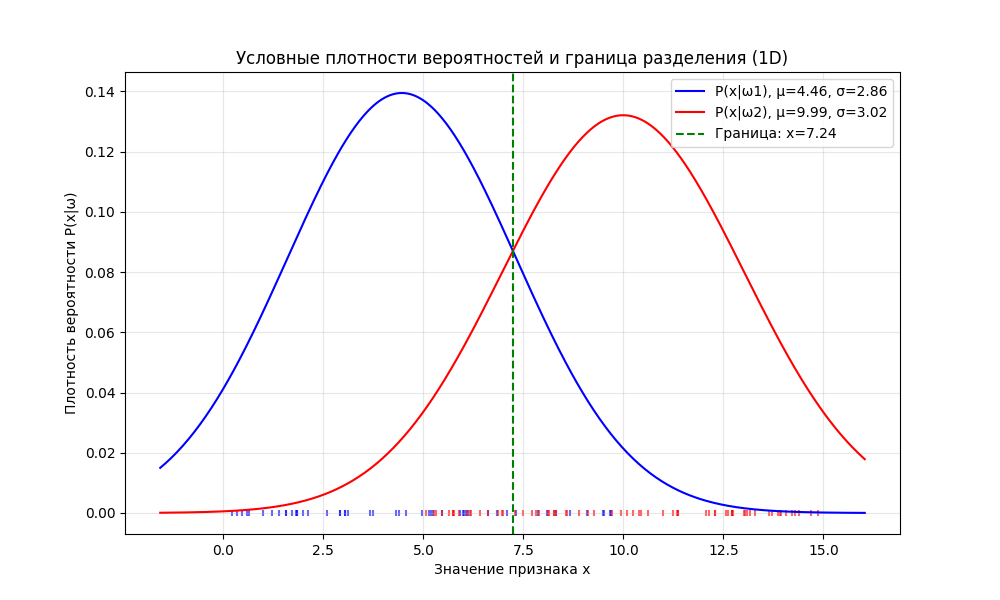
Построить графики условных плотностей вероятностей и границы разделения классов.

Задать значения признаков и предполагая, что априорные вероятности равны , определить апостериорные вероятности по формуле Байеса

.

Определить к какому классу относится значение признака .

**Результаты выполнения работы**

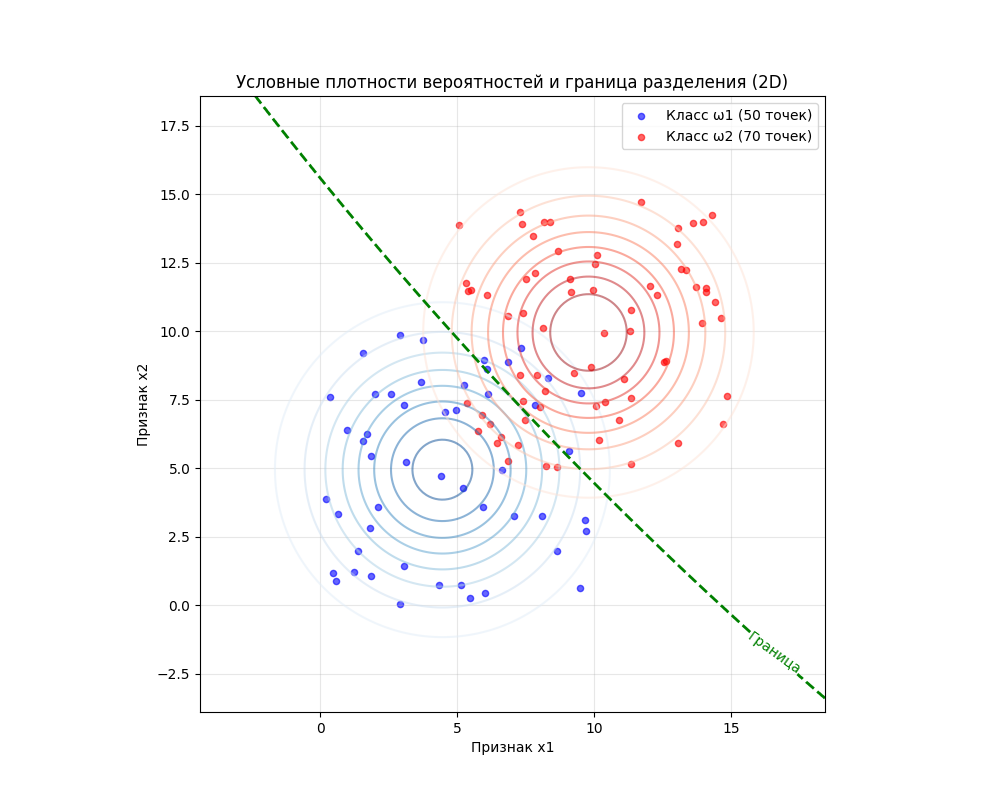
****

**Рисунок 1 –** Условные плотности вероятностей и граница разделения для классов

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

**Рисунок 2 –** Значения классов, граница распределения, вероятности для точки и ошибка классификации



**Рисунок 3 –** Условные плотности вероятностей и граница разделения для классов

Изображение выглядит как текст, Шрифт, снимок экрана, типография

Содержимое, созданное искусственным интеллектом, может быть неверным.

**Рисунок 4 –** Значения классов, граница распределения, вероятности для точки и ошибка классификации

**Вывод:** в результате выполнения домашней работы были сформированы практические навыки использования байесовского подхода принятия решений в случае классификации двух классов, условные плотности вероятностей которых соответствуют одномерному и двумерному нормальному закону распределения.

**Листинг программы**

**Задание 1**

import numpy as np

import matplotlib.pyplot as plt

from scipy.stats import norm

np.random.seed(42)

x1 = np.random.uniform(0, 10, 50)

x2 = np.random.uniform(5, 15, 70)

mu1, sigma1 = np.mean(x1), np.std(x1)

mu2, sigma2 = np.mean(x2), np.std(x2)

print(f"Класс ω1: μ = {mu1:.3f}, σ = {sigma1:.3f}")

print(f"Класс ω2: μ = {mu2:.3f}, σ = {sigma2:.3f}")

def find\_boundary(mu1: float, sigma1: float, mu2: float, sigma2: float, search\_range: tuple[float, float]) -> float:

"""

Нахождение границы разделения классов.

При равных априорных вероятностях P(ω1)=P(ω2)=0.5:

Граница - это точка, где P(x|ω1) \* P(ω1) = P(x|ω2) \* P(ω2), т.е. P(x|ω1) = P(x|ω2)

"""

x\_vals = np.linspace(search\_range[0], search\_range[1], 1000)

pdf1 = norm.pdf(x\_vals, mu1, sigma1)

pdf2 = norm.pdf(x\_vals, mu2, sigma2)

diff = pdf1 - pdf2

idx = np.argmin(np.abs(diff))

return x\_vals[idx]

search\_start = min(mu1, mu2) - 2\*max(sigma1, sigma2)

search\_end = max(mu1, mu2) + 2\*max(sigma1, sigma2)

boundary = find\_boundary(mu1, sigma1, mu2, sigma2, (search\_start, search\_end))

print(f"Граница разделения классов: x = {boundary:.3f}")

x\_plot = np.linspace(search\_start, search\_end, 500)

pdf1\_plot = norm.pdf(x\_plot, mu1, sigma1)

pdf2\_plot = norm.pdf(x\_plot, mu2, sigma2)

plt.figure(figsize=(10, 6))

plt.plot(x\_plot, pdf1\_plot, label=f'P(x|ω1), μ={mu1:.2f}, σ={sigma1:.2f}', color='blue')

plt.plot(x\_plot, pdf2\_plot, label=f'P(x|ω2), μ={mu2:.2f}, σ={sigma2:.2f}', color='red')

plt.axvline(x=boundary, color='green', linestyle='--', label=f'Граница: x={boundary:.2f}')

plt.scatter(x1, [0]\*len(x1), alpha=0.6, color='blue', s=20, marker='|')

plt.scatter(x2, [0]\*len(x2), alpha=0.6, color='red', s=20, marker='|')

plt.title('Условные плотности вероятностей и граница разделения (1D)')

plt.xlabel('Значение признака x')

plt.ylabel('Плотность вероятности P(x|ω)')

plt.legend()

plt.grid(True, alpha=0.3)

plt.show()

X = 7.5

print(f"\nАнализ точки X = {X}")

P\_w1 = 0.5

P\_w2 = 0.5

P\_X\_w1 = norm.pdf(X, mu1, sigma1)

P\_X\_w2 = norm.pdf(X, mu2, sigma2)

P\_X = P\_X\_w1 \* P\_w1 + P\_X\_w2 \* P\_w2

P\_w1\_X = (P\_X\_w1 \* P\_w1) / P\_X

P\_w2\_X = (P\_X\_w2 \* P\_w2) / P\_X

print(f"Апостериорная вероятность P(ω1|X) = {P\_w1\_X:.4f}")

print(f"Апостериорная вероятность P(ω2|X) = {P\_w2\_X:.4f}")

predicted\_class = "ω1" if P\_w1\_X > P\_w2\_X else "ω2"

print(f"Точка X = {X} относится к классу {predicted\_class}")

P\_error\_w1 = 1 - norm.cdf(boundary, mu1, sigma1)

P\_error\_w2 = norm.cdf(boundary, mu2, sigma2)

P\_error = P\_w1 \* P\_error\_w1 + P\_w2 \* P\_error\_w2

print(f"\nВероятность ошибки классификации P(error) = {P\_error:.4f}")

**Задание 2**

import numpy as np

import matplotlib.pyplot as plt

from scipy.stats import multivariate\_normal

np.random.seed(42)

x1\_class1 = np.random.uniform(0, 10, 50)

x2\_class1 = np.random.uniform(0, 10, 50)

data\_class1 = np.column\_stack((x1\_class1, x2\_class1))

x1\_class2 = np.random.uniform(5, 15, 70)

x2\_class2 = np.random.uniform(5, 15, 70)

data\_class2 = np.column\_stack((x1\_class2, x2\_class2))

mu1\_2d = np.mean(data\_class1, axis=0)

mu2\_2d = np.mean(data\_class2, axis=0)

sigma1\_2d = np.sqrt(np.mean(np.var(data\_class1, axis=0)))

sigma2\_2d = np.sqrt(np.mean(np.var(data\_class2, axis=0)))

cov1 = np.array([[sigma1\_2d\*\*2, 0],

[0, sigma1\_2d\*\*2]])

cov2 = np.array([[sigma2\_2d\*\*2, 0],

[0, sigma2\_2d\*\*2]])

print(f"Класс ω1: μ = [{mu1\_2d[0]:.3f}, {mu1\_2d[1]:.3f}], σ = {sigma1\_2d:.3f}")

print(f"Класс ω2: μ = [{mu2\_2d[0]:.3f}, {mu2\_2d[1]:.3f}], σ = {sigma2\_2d:.3f}")

x\_min, x\_max = min(mu1\_2d[0]-3\*sigma1\_2d, mu2\_2d[0]-3\*sigma2\_2d), max(mu1\_2d[0]+3\*sigma1\_2d, mu2\_2d[0]+3\*sigma2\_2d)

y\_min, y\_max = min(mu1\_2d[1]-3\*sigma1\_2d, mu2\_2d[1]-3\*sigma2\_2d), max(mu1\_2d[1]+3\*sigma1\_2d, mu2\_2d[1]+3\*sigma2\_2d)

xx, yy = np.meshgrid(np.linspace(x\_min, x\_max, 200),

np.linspace(y\_min, y\_max, 200))

pos = np.dstack((xx, yy))

rv1 = multivariate\_normal(mu1\_2d, cov1)

rv2 = multivariate\_normal(mu2\_2d, cov2)

pdf1\_2d = rv1.pdf(pos)

pdf2\_2d = rv2.pdf(pos)

fig, ax = plt.subplots(1, 1, figsize=(10, 8))

levels = np.linspace(0, max(np.max(pdf1\_2d), np.max(pdf2\_2d)), 10)

cs1 = ax.contour(xx, yy, pdf1\_2d, levels=levels, cmap='Blues', alpha=0.5)

cs2 = ax.contour(xx, yy, pdf2\_2d, levels=levels, cmap='Reds', alpha=0.5)

diff = pdf1\_2d - pdf2\_2d

boundary\_contour = ax.contour(xx, yy, diff, levels=[0], colors='green', linewidths=2, linestyles='--')

plt.clabel(boundary\_contour, inline=True, fontsize=10, fmt='Граница')

ax.scatter(data\_class1[:, 0], data\_class1[:, 1], c='blue', s=20, alpha=0.6, label='Класс ω1 (50 точек)')

ax.scatter(data\_class2[:, 0], data\_class2[:, 1], c='red', s=20, alpha=0.6, label='Класс ω2 (70 точек)')

ax.set\_title('Условные плотности вероятностей и граница разделения (2D)')

ax.set\_xlabel('Признак x1')

ax.set\_ylabel('Признак x2')

ax.legend()

ax.grid(True, alpha=0.3)

ax.set\_aspect('equal', 'box')

plt.show()

X\_2d = np.array([7.5, 7.5])

print(f"\nАнализ точки X = ({X\_2d[0]}, {X\_2d[1]})")

P\_w1 = 0.5

P\_w2 = 0.5

P\_X\_w1 = rv1.pdf(X\_2d)

P\_X\_w2 = rv2.pdf(X\_2d)

P\_X = P\_X\_w1 \* P\_w1 + P\_X\_w2 \* P\_w2

P\_w1\_X = (P\_X\_w1 \* P\_w1) / P\_X

P\_w2\_X = (P\_X\_w2 \* P\_w2) / P\_X

print(f"Апостериорная вероятность P(ω1|X) = {P\_w1\_X:.4f}")

print(f"Апостериорная вероятность P(ω2|X) = {P\_w2\_X:.4f}")

predicted\_class\_2d = "ω1" if P\_w1\_X > P\_w2\_X else "ω2"

print(f"Точка X = ({X\_2d[0]}, {X\_2d[1]}) относится к классу {predicted\_class\_2d}")