

## 课后作业

### 要求

1. 从两个高斯分布中采样。一组采样为正类，一组采样为负类。
2. 编程实现线性回归， $y_i$  为-1 或 1。保存参数，画出回归投影面，同时可视化显示结果。
3. 编程实现线性判别分析，保留参数，对测试数据做出预测，同时可视化显示结果，画出分类面。
4. 实验 1 两个高斯分布具有一样的方差，实验 2 两个高斯分布具有不同的方差，观察实验结果的不同。

### 数据集生成代码

```
import numpy as np
import matplotlib.pyplot as plt

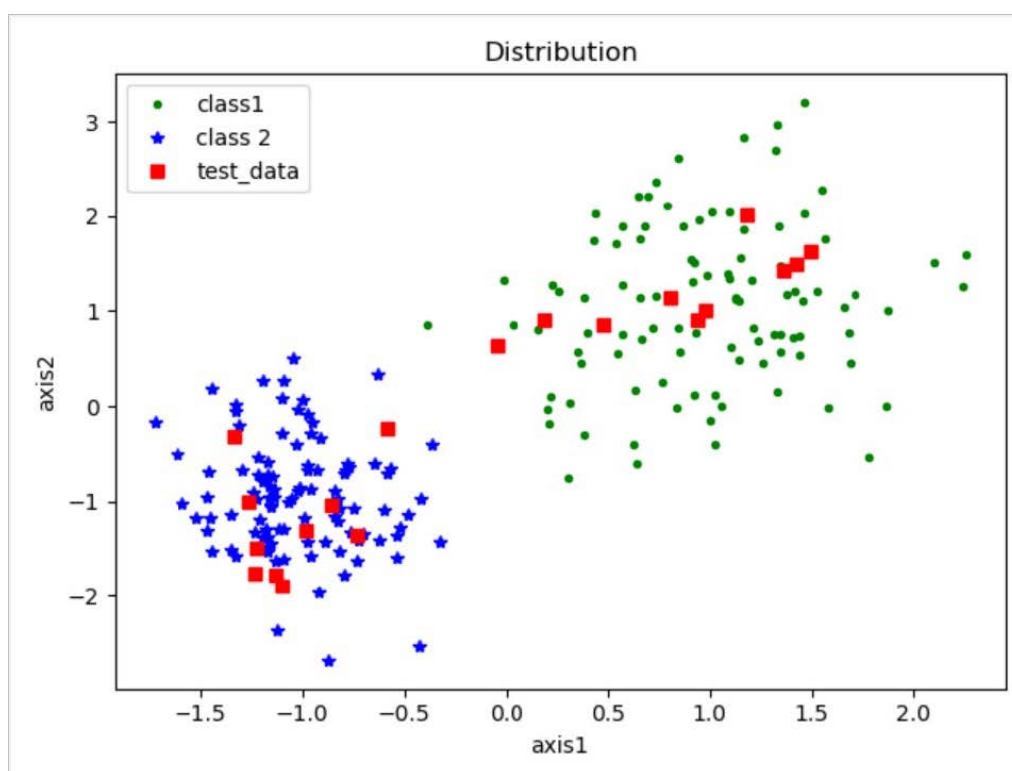
# 生成训练数据集，每类 100 个数据
def get_train_data(data_size=100):
    data_label = np.zeros((2 * data_size, 1)) # class 1
    x1 = np.reshape(np.random.normal(1, 0.6, data_size), (data_size, 1))
    y1 = np.reshape(np.random.normal(1, 0.8, data_size), (data_size, 1))
    data_train = np.concatenate((x1, y1), axis=1)
    data_label[0:data_size, :] = 0
    # class 2
    x2 = np.reshape(np.random.normal(-1, 0.3, data_size), (data_size, 1))
    y2 = np.reshape(np.random.normal(-1, 0.5, data_size), (data_size, 1))
    data_train = np.concatenate((data_train, np.concatenate((x2, y2), axis=1)), axis=0)
    data_label[data_size:2 * data_size, :] = 1

    return data_train, data_label

# 生成测试数据集，每类 10 个
def get_test_data(data_size=10):
    testdata_label = np.zeros((2 * data_size, 1))
    # class 1
    x1 = np.reshape(np.random.normal(1, 0.6, data_size), (data_size, 1))
    y1 = np.reshape(np.random.normal(1, 0.8, data_size), (data_size, 1))
    data_test = np.concatenate((x1, y1), axis=1)
    testdata_label[0:data_size, :] = 0
    # class 2
    x2 = np.reshape(np.random.normal(-1, 0.3, data_size), (data_size, 1))
    y2 = np.reshape(np.random.normal(-1, 0.5, data_size), (data_size, 1))
    data_test = np.concatenate((data_test, np.concatenate((x2, y2), axis=1)), axis=0)
    testdata_label[data_size:2 * data_size, :] = 1

    return data_test, testdata_label
```

## 训练集和测试集可视化结果



## 解答

源代码见 `code/data.py`

### 不同方差的两个高斯分布

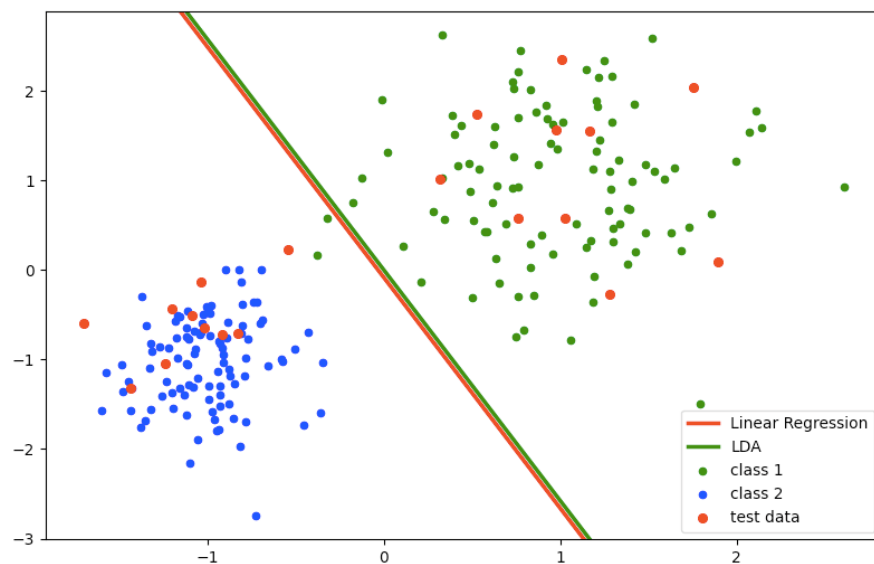


图 1: 训练

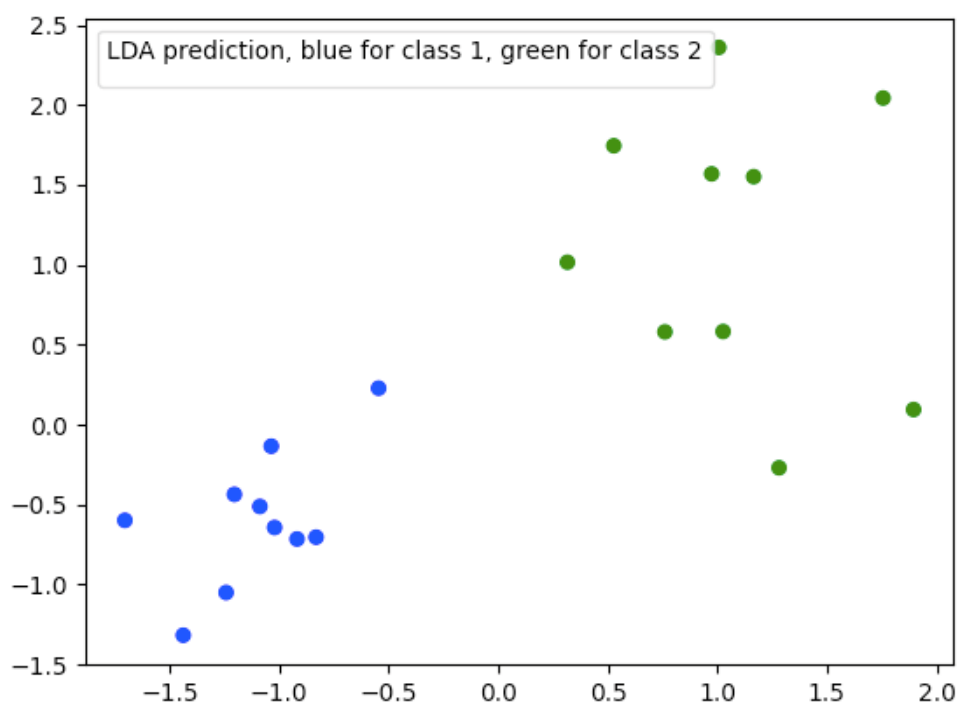


图 2: 预测

## 相同方差的两个高斯分布

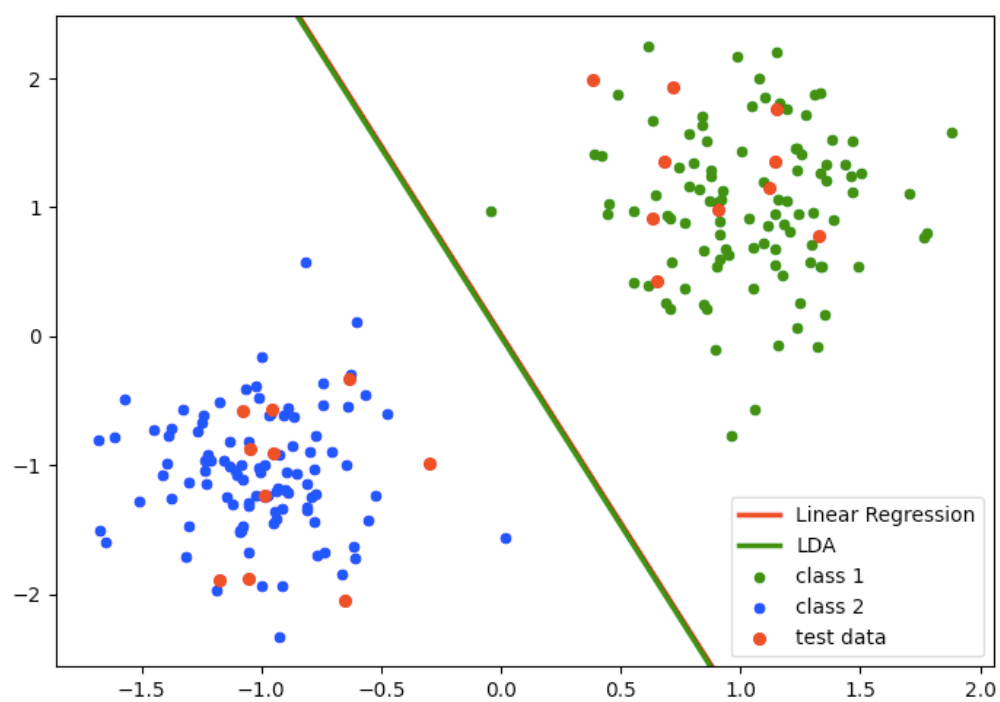


图 3: 训练

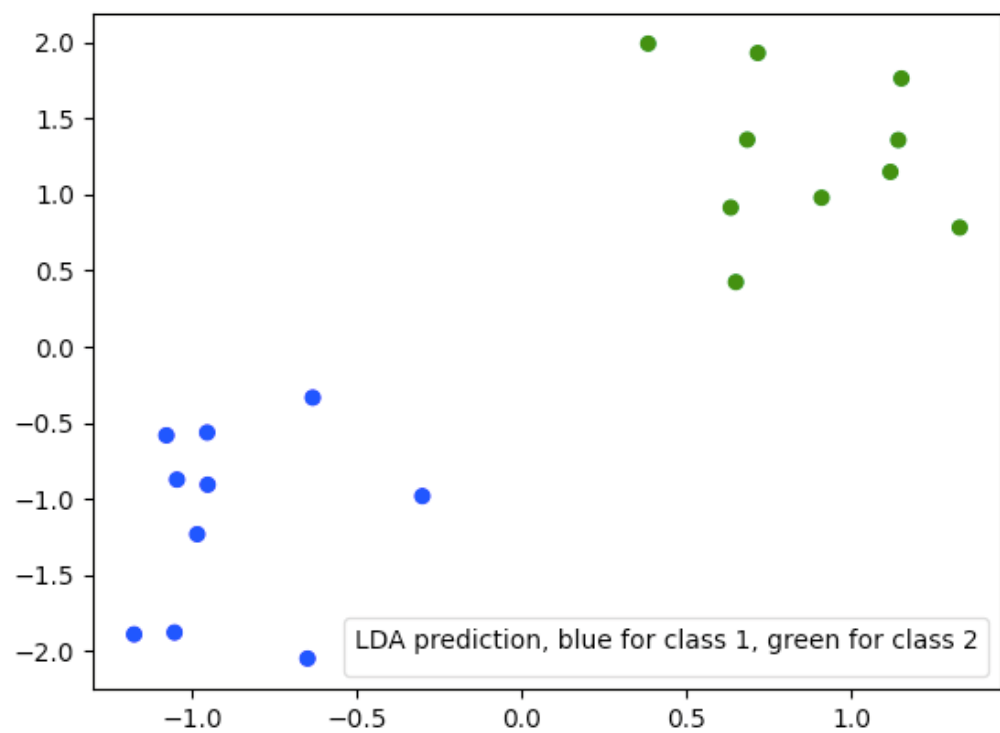


图 4: 预测