实验六：支持向量机

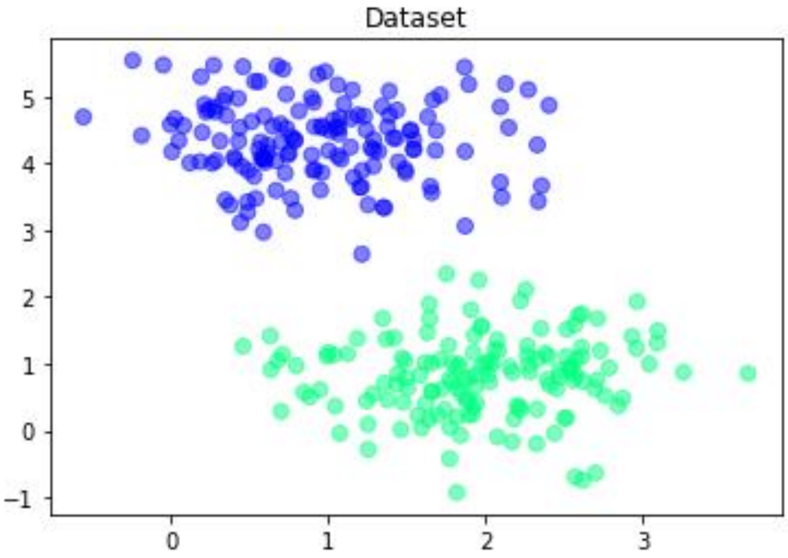
姓名：　　　　　　　　　　　　　　　　　学号：

- ## 实验目的及内容

理解和掌握支持向量机基本原理和方法，理解数据线性可分性，理解支持向量机对偶问题，理解支持向量机核函数等概念，掌握间隔、支持向量、对偶、核函数等概念及计算方法。

- ## 实验要求

基于给定数据集，编程实现线性支持向量机模型，采用约束优化方法对模型进行训练。模型初始参数随机生成。绘制训练过程中损失函数迭代曲线，绘制最终分类超平面，给出最终支持向量机表达式和模型在测试集上的精度。



- ## 实验环境

python、numpy、matplotlib、scipy

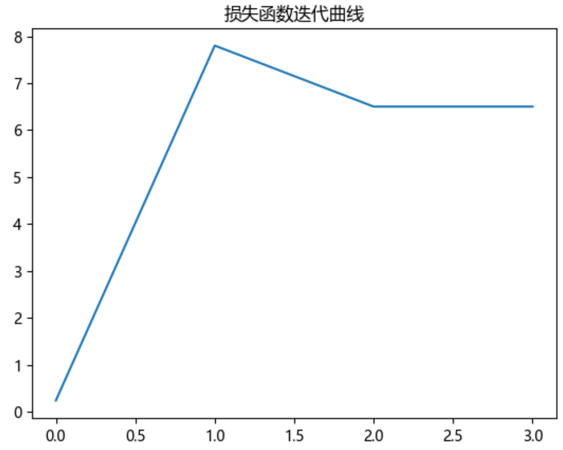
- ## 实验代码

```
import numpy as np

from scipy.optimize import LinearConstraint, minimize

import matplotlib.pyplot as plt


training_data = np.loadtxt('experiment_06_training_set.csv', delimiter=',')

testing_data = np.loadtxt('experiment_06_testing_set.csv', delimiter=',')
```

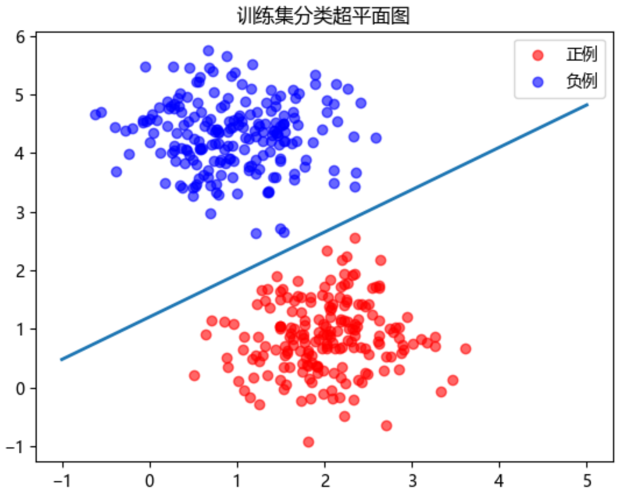
```
train_point = training_data[:, 0:2]
train_tag = training_data[:, 2].reshape(-1, 1)

# wb = np.zeros((3,))
wb = np.array([np.random.rand() for _ in range(3)])
y_x = train_point * train_tag
a = np.hstack((y_x, train_tag))


def objective(w):
    return 0.5 * (w[0] ** 2 + w[1] ** 2)


iteration = 1
loss_iteration = []
loss_iteration.append(objective(wb))


def print_loss(intermediate_result):
    global iteration
    global loss_iteration
    parameter = intermediate_result    # .x
    loss_iteration.append(objective(parameter))
    print('iteration', iteration, 'loss', objective(parameter))
    iteration += 1


lc = LinearConstraint(a, lb=1, ub=np.inf)
aa = minimize(objective, wb, constraints=lc, callback=print_loss)
print(aa)
print(aa.x)

plt.rcParams['font.sans-serif'] = ['Microsoft YaHei']
plt.plot(loss_iteration)
plt.title("损失函数迭代曲线")
plt.show()

indices_positive = np.where(train_tag == 1)[0]
indices_negative = np.where(train_tag == -1)[0]
positive_points = train_point[indices_positive]
negative_points = train_point[indices_negative]

plt.rcParams['font.sans-serif'] = ['Microsoft YaHei']
plt.scatter(positive_points[:, 0], positive_points[:, 1], c='red', label='正例', alpha=0.6)
plt.scatter(negative_points[:, 0], negative_points[:, 1], c='blue', label='负例', alpha=0.6)
plt.title("训练集分类超平面图")
x = np.linspace(-1, 5, 400)
y = - aa.x[0] / aa.x[1] * x - aa.x[2] / aa.x[1]
plt.plot(x, y, lw=2.0)
plt.legend()
plt.show()


testing_point = testing_data[:, 0:2]
testing_tag = testing_data[:, 2]
predict_tag = aa.x[0] * testing_point[:, 0] + aa.x[1] * testing_point[:, 1] + aa.x[2]
predict_tag = np.sign(predict_tag)
predict_tag = np.where(predict_tag == 0, 1, predict_tag)
correct_predictions = np.sum(testing_tag == predict_tag)
print('Test Accuracy', correct_predictions / predict_tag.shape[0])


indices_positive = np.where(testing_tag == 1)[0]
indices_negative = np.where(testing_tag == -1)[0]
positive_points = testing_point[indices_positive]
negative_points = testing_point[indices_negative]


plt.rcParams['font.sans-serif'] = ['Microsoft YaHei']
plt.scatter(positive_points[:, 0], positive_points[:, 1], c='red', label='正例', alpha=0.6)
plt.scatter(negative_points[:, 0], negative_points[:, 1], c='blue', label='负例',
alpha=0.6)

plt.title("测试集分类超平面图")

x = np.linspace(-1, 5, 400)

y = - aa.x[0] / aa.x[1] * x - aa.x[2] / aa.x[1]

plt.plot(x, y, lw=2.0)

plt.legend()

plt.show()
```

- **结果分析**

（1）损失函数迭代曲线



（2）分类超平面图

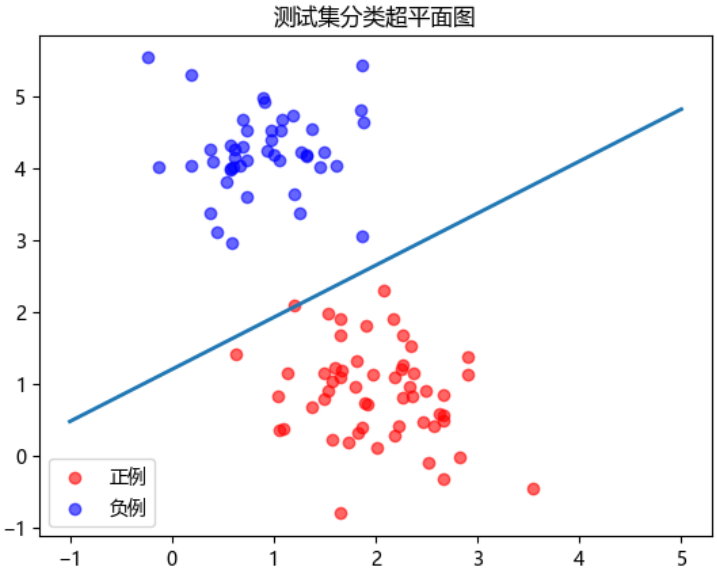


（3）支持向量机表达式

$$0 = 2.115x - 2.920y + 3.512$$

(4) 测试集上精度

Test Accuracy：0.99


测试集分类超平面图
正例
负例