

天津大学

机器学习实验报告



题目：机器学习实验——支持向量机

学 院 智能与计算学部
专 业 人工智能
年 级 2021
姓 名 666
学 号 最通用的 bassline

2023 年 3 月 29 日

机器学习实验---支持向量机

一、实验目的

1. 理解支持向量机算法原理，能实现支持向量机算法；
2. 针对特定应用场景及数据，实现二分类支持向量机算法，并尝试扩展到三分类任务重。

二、实验内容

1. 从 UCI 数据库中下载一个二分类数据集，进行数据说明；
2. 用 80%的数据训练，余下的做测试，计算分类准确度。

三、实验报告要求

1. 按实验内容撰写实验过程；
2. 报告中涉及到的代码，每一行需要有详细的注释；
3. 按自己的理解重新组织，禁止粘贴复制实验内容。

四、实验记录

因为原本的代码直接复制粘贴就能运行所以我不贴了，直接放三分类的代码。

正文：

```
#所需包
from sklearn import datasets
from sklearn import svm
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
import pandas as pd
#导入需要的数据
iris = datasets.load_iris()
# print(iris.keys())
print(iris.feature_names)
# print(iris.target_names)
X = iris.data
y = iris.target
# print(y)
#拆分数据
#20%测试数据
X_train, X_test, y_train, y_test = train_test_split(iris.data[:,2], iris.target,
test_size=0.2, random_state=0)
#三种核的 SVM
lin_svc = svm.SVC(kernel='linear').fit(X_train, y_train)
rbf_svc = svm.SVC(kernel='rbf').fit(X_train, y_train)
poly_svc = svm.SVC(kernel='poly', degree=3).fit(X_train, y_train)
print(lin_svc)
print(rbf_svc)
print(poly_svc)
#学习率
```

```

h = .08
#刻度上下限
x_min, x_max = X_train[:, 0].min() - 1, X_train[:, 0].max() + 1
y_min, y_max = X_train[:, 1].min() - 1, X_train[:, 1].max() + 1
xx, yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_min, y_max,
h))

```

```

titles = ['linear kernel', 'RBF kernel', 'polynomial (degree 3) kernel']

```

```

for i, clf in enumerate((lin_svc, rbf_svc, poly_svc)):
    plt.subplot(2,2, i + 1)
    # 画布间距
    plt.subplots_adjust(wspace=0.4, hspace=0.4)
    Z = clf.predict(np.c_[xx.ravel(), yy.ravel()])
    Z = Z.reshape(xx.shape) #(220, 280)
    plt.contourf(xx, yy, Z, cmap=plt.cm.Paired, alpha=0.9)
    plt.scatter(X[:, 0], X[:, 1], c=y, cmap=plt.cm.Paired)
    plt.xlabel('Sepal length')
    plt.ylabel('Sepal width')
    plt.xlim(xx.min(), xx.max())
    plt.ylim(yy.min(), yy.max())
    plt.xticks(())
    plt.yticks(())
    plt.title(titles[i])

```

```

plt.show()
#利用测试数据集进行精准度测试
lin_svc_pre = lin_svc.predict(X_test)
acc_lin_svc = sum(lin_svc_pre==y_test)/len(y_test)

rbf_svc_pre = rbf_svc.predict(X_test)
acc_rbf_svc = sum(rbf_svc_pre==y_test)/len(y_test)

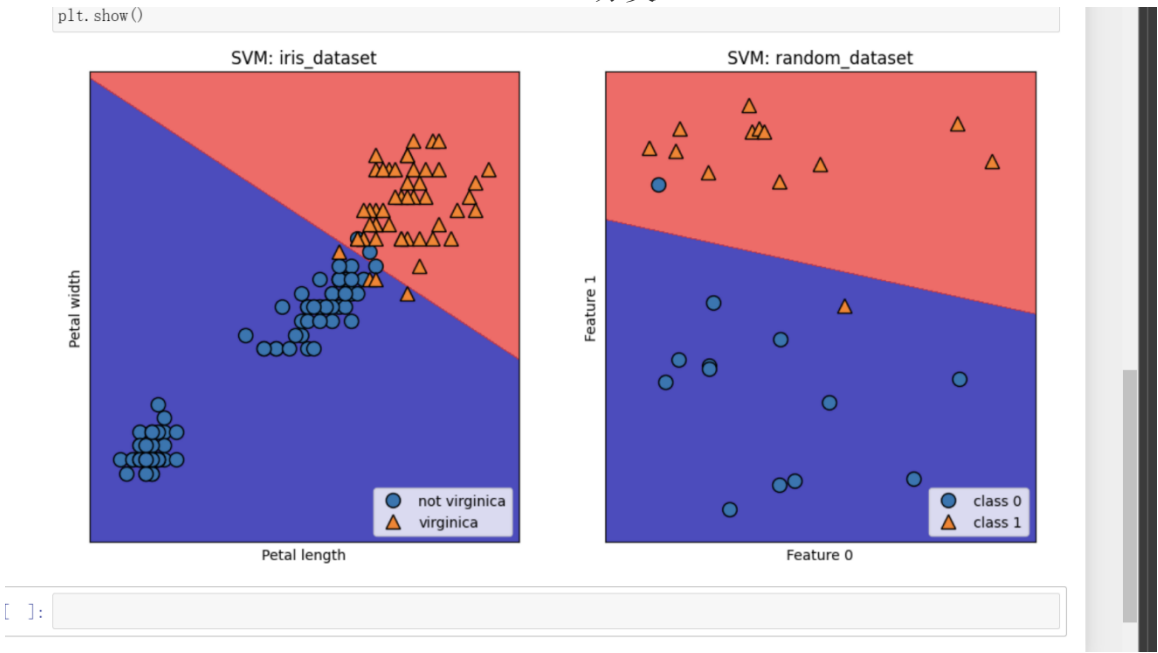
poly_svc_pre = poly_svc.predict(X_test)
acc_poly_svc = sum(poly_svc_pre==y_test)/len(y_test)

print(acc_lin_svc)
print(acc_rbf_svc)
print(acc_poly_svc)

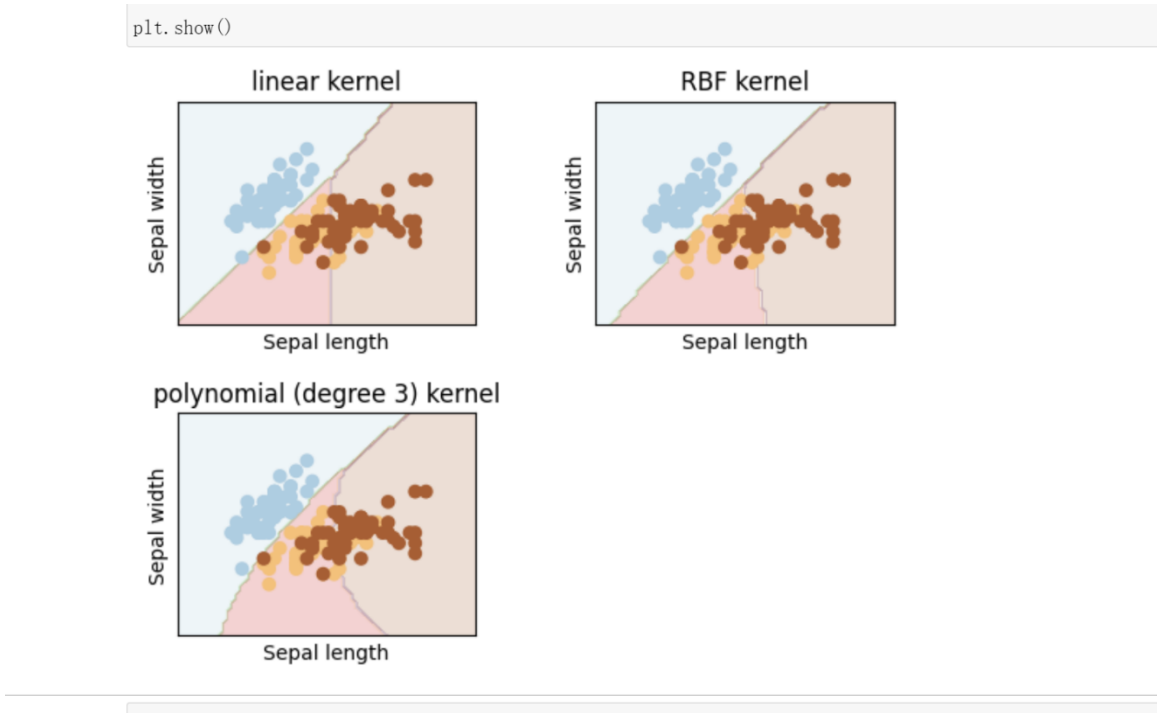
```

五、运行结果

二分类



三分类



数据测试精准度

```
1 [62]: lin_svc_pre = lin_svc.predict(X_test)
        acc_lin_svc = sum(lin_svc_pre==y_test)/len(y_test)

        rbf_svc_pre = rbf_svc.predict(X_test)
        acc_rbf_svc = sum(rbf_svc_pre==y_test)/len(y_test)

        poly_svc_pre = poly_svc.predict(X_test)
        acc_poly_svc = sum(poly_svc_pre==y_test)/len(y_test)

        print(acc_lin_svc)
        print(acc_rbf_svc)
        print(acc_poly_svc)

0.7333333333333333
0.7333333333333333
0.7333333333333333
```

六、实验小结

本次实验是理解支持向量机算法的原理并实现支持向量机算法，其输入是实例的特征向量，输出为实例的分类类别，以产生二分类输出，用于将数据分为两部分，因此也称为线性二分类器。实验中使用 `jupyterbook` 进行实验，并使用到了 `pandas`、`numpy`、`Matplotlib`、`sklearn` 等机器学习库，可对机器学习有初步理解。

python 大家都没学过，一点一点摸索还挺难的，幸好有源码可以学，要不然光靠库和文档就要死了，为了写这个玩意研究了两三天，最后还是搞不太清楚。

支持向量机的数学原理，我已经推过两遍了，但是一来到代码环节就开始发怵，这东西也没说从底层直接实现，以至于各种函数的参数特别不清楚，看文档看的眼花缭乱，报各种奇怪的错，甚至连教程都很少。

而且就我感觉，生成散点图这一关特别难，搞不清楚需要什么参数，以至于最后只能照猫画虎别人的代码，自己没法游刃有余的写，这个东西看来还需要仔细研究一下。