

实验六 支持向量机

一、实验目的

1. 掌握支持向量机的基本原理
2. 编程实现简单的支持向量机
3. 掌握支持向量机里的关键语句
4. 能够用支持向量机解决实际问题

二、实验步骤

1. 生成数据集
2. 调用支持向量机函数
3. 绘图
4. 得出结果并分析不同参数对结果的影响

三、实验内容

1. 运行、调试源代码
2. 分析支持向量机的基本步骤
3. 分析运行结果中输出参数的含义
4. 编程解决实际问题

注意：所有图名都要体现“班级-姓名-学号”。

四、实验原理

SVM的工作原理详细分析见教材或者课件。

任务 1：自定义数据实现 SVM 分类

- (1) 仔细阅读下面程序，运行程序，理解程序实现的功能；
- (2) 解释 `svm = SVC(kernel='rbf', gamma='auto', C=1, random_state=1)` 各个参数的功能；
- (3) 画出当 `kernel='linear', 'poly', 'rbf', 'sigmoid'` 时的图形, 并比较；
- (4) 画出当 `kernel='rbf', gamma='auto', gamma=10, gamma=50, gamma=100` 时的图形, 并比较。 (3-4 步注意用核函数+姓名做图名)

```
# 【任务 1 附件】：自定义数据实现 SVM 分类
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap
from sklearn import datasets
from sklearn.svm import SVC

# 以下为创建数据 X 和标签 y
np.random.seed(1)
# 创建 100 个二维数组，即 100 个 2 个特征的样本
X = np.random.randn(100, 2)
# np.logical_xor(bool1, bool2)，异或逻辑运算，如果 bool1 和 bool2 的结果相同则为 False，
# 否则为 True
# ++和一为一三象限，+-和-+为二四象限，如此做则 100 个样本必定线性不可分
```

```

y = np.logical_xor(X[:, 0] > 0, X[:, 1] > 0)
# 对 X 数组人为分类, 二四象限为 True, 即为 1 类; 一三象限为 False, 即为-1 类
y = np.where(y, 1, -1)

# 构建决策边界, 通用程序, 后面可直接使用
def plot_decision_regions(X, y, classifier=None):
    marker_list = ['o', 'x', 's']
    color_list = ['r', 'b', 'g']
    cmap = ListedColormap(color_list[:len(np.unique(y))])

    x1_min, x1_max = X[:, 0].min() - 1, X[:, 0].max() + 1
    x2_min, x2_max = X[:, 1].min() - 1, X[:, 1].max() + 1
    t1 = np.linspace(x1_min, x1_max, 666)
    t2 = np.linspace(x2_min, x2_max, 666)

    x1, x2 = np.meshgrid(t1, t2)
    y_hat = classifier.predict(np.array([x1.ravel(), x2.ravel()]).T)
    y_hat = y_hat.reshape(x1.shape)
    plt.contourf(x1, x2, y_hat, alpha=0.2, cmap=cmap)
    plt.xlim(x1_min, x1_max)
    plt.ylim(x2_min, x2_max)

    for ind, clas in enumerate(np.unique(y)):
        plt.scatter(X[y == clas, 0], X[y == clas, 1], alpha=0.8, s=50,
                    c=color_list[ind], marker=marker_list[ind], label=clas)

# 以下为关键语句
svm = SVC(kernel='rbf', gamma='auto', C=1, random_state=1)
svm.fit(X, y)
plot_decision_regions(X, y, classifier=svm)
SVC(C=1, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
    max_iter=-1, probability=False, random_state=1, shrinking=True,
    tol=0.001, verbose=False)

plt.title('kernel = rbf, gamma=auto')
plt.legend()
plt.show()

```

任务 2: 鸢尾花数据实现 SVM 分类

(1) 把上面程序中的数据换成鸢尾花数据和标签, 1-2 班用前两列数据, 3-4 班用后两列数据, 标签相同;

(2) 画出当 `kernel='linear'`, `'poly'`, `'rbf'`, `'sigmoid'` 时的图形, 并比较;

(3) 画出当 `kernel='rbf'`, `gamma='auto'`, `gamma=10`, `gamma=50`, `gamma=100` 时的图形, 并比较。 (2-3 步注意用核函数+姓名做图名)