

SVM小实验

211250143 王铭嵩

一、SVM算法原理

对于支持向量机，一个点（样例）对应的Margin是其到分界超平面的垂直距离。具有最小Margin的点称为支持向量。SVM的核心要义是最大化所有训练样本的最小间隔Margin。

二、代码实现

实验使用sklearn的svm库来实现支持向量机。

- 数据获取：

依据作业要求实现

```
def get_data():  
    np.random.seed(0)  
    X = np.r_[np.random.randn(20, 2) - [2, 2], np.random.randn(20, 2) + [2, 2]]  
    Y = [-1] * 20 + [1] * 20  
    return X, Y
```

- 创建分类器并处理数据：

```
# 创建SVM分类器  
clf = svm.SVC(kernel='linear')  
clf.fit(X, Y)
```

- 获取支持向量并输出：

```
# 获取支持向量并输出  
support_vectors = clf.support_vectors_  
print(support_vectors)
```

- 获取超平面参数并输出：

```
# 获取超平面参数并输出
w = clf.coef_
b = clf.intercept_
print(f"w={w},b={b}")
```

- 利用 `matplotlib.pyplot` 包来可视化结果。其中支持向量被突出显示；超平面和间隔以直线方式打印：

```
# 可视化数据和超平面
plt.scatter(X[:, 0], X[:, 1], c=Y, cmap=plt.cm.Paired)
plt.scatter(support_vectors[:, 0], support_vectors[:, 1], s=100, facecolors='none', edgecolors='k')
ax = plt.gca()
xlim = ax.get_xlim()
ylim = ax.get_ylim()

# 创建网格来绘制超平面
xx, yy = np.meshgrid(np.linspace(xlim[0], xlim[1], num=50), np.linspace(ylim[0], ylim[1], num=50))
xy = np.c_[xx.ravel(), yy.ravel()]
Z = clf.decision_function(xy).reshape(xx.shape)

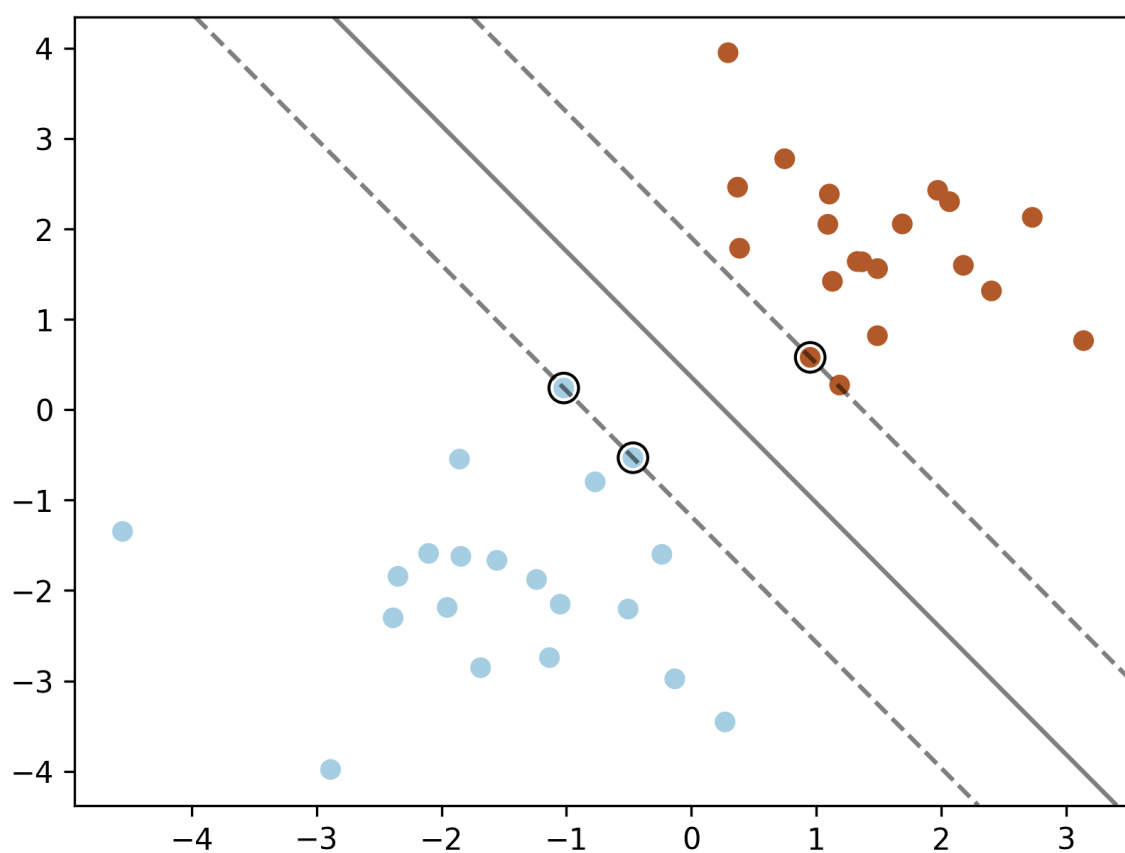
# 绘制超平面和间隔
ax.contour(*args: xx, yy, Z, colors='k', levels=[-1, 0, 1], alpha=0.5, linestyle=['--', '-', '--'])
plt.savefig(*args: 'result.png', dpi=300)
plt.show()
```

三、实验结果

支持向量及超平面参数输出：

```
[[ -1.02126202  0.2408932 ]
 [ -0.46722079 -0.53064123]
 [  0.95144703  0.57998206]]
w= [[0.90230696  0.64821811]], b= [-0.23452129]
```

可视化：



支持向量为被突出表示的点；灰线为超平面。