

```

In [1]: # 导入操作系统库
import os
# 更改工作目录
os.chdir(r"D:\softwares\applied statistics\pythoncodelearning\chap3\sourcecode")
# 导入绘图库
import matplotlib.pyplot as plt
# 导入支持向量机模型
from sklearn import svm
# 导入决策边界可视化工具
from sklearn.inspection import DecisionBoundaryDisplay
# 导入数据集生成工具
from sklearn.datasets import make_blobs
# 导入绘图库中的字体管理包
from matplotlib import font_manager
# 实现中文字符正常显示
font = font_manager.FontProperties(fname=r"C:\Windows\Fonts\SimKai.ttf")
# 使用seaborn风格绘图
plt.style.use("seaborn-v0_8")
# 生成样本
n_samples_1 = 1000
n_samples_2 = 100
centers = [[0.0, 0.0], [2.0, 2.0]]
clusters_std = [1.5, 0.5]
# 分类数据
X, y = make_blobs(
    n_samples=[n_samples_1, n_samples_2], # 分别的样本量
    centers=centers, # 聚类中心
    cluster_std=clusters_std, # 标准差
    random_state=0,
    shuffle=False,
)
# 线性SVM模型
clf = svm.SVC(kernel="linear", C=1.0)
# 模型拟合
clf.fit(X, y)
# 加权的SVM模型
wclf = svm.SVC(kernel="linear", class_weight={1: 10})
# 模型拟合
wclf.fit(X, y)
# 开始绘图
fig, ax = plt.subplots(figsize=(6,6), tight_layout=True)
# 绘制散点
ax.scatter(X[:, 0], X[:, 1], c=y, cmap=plt.cm.Paired, edgecolors="k")
# 绘制SVM的决策边界
disp = DecisionBoundaryDisplay.from_estimator(
    clf,
    X,
    plot_method="contour",
    colors="k",
    levels=[0],
    alpha=0.5,
    linestyle=["--"],
    ax=ax
)
# 绘制加权的SVM的决策边界
wdisp = DecisionBoundaryDisplay.from_estimator(
    wclf,
    X,

```

```
plot_method="contour",
colors="r",
levels=[0],
alpha=0.5,
linestyles=["-"],
ax=ax
)
# 添加图例
ax.legend(
    [disp.surface_.collections[0], wdisp.surface_.collections[0]],
    ["non weighted", "weighted"],
    loc="upper right",
)
plt.show()
fig.savefig("../codeimage/code4.pdf")
```

