



分类算法：支持向量机(SVM)算法深度解析

支持向量机基础

1. 1.1 SVM的基本概念

支持向量机(Support Vector Machine, SVM)是一种监督学习模型，主要用于分类和回归分析。其核心思想是找到一个超平面，使得两类数据在该超平面上的间隔最大化。SVM通过使用核技巧(kernel trick)处理非线性可分问题，能够有效地在高维空间中进行分类。

1.1 示例代码

```
from sklearn import svm
from sklearn.datasets import make_blobs

# 生成数据
X, y = make_blobs(n_samples=100, centers=2, random_state=6)

# 创建SVM分类器
clf = svm.SVC(kernel='linear')

# 训练模型
clf.fit(X, y)

# 预测新数据点
new_data = [[1, 2], [3, 4]]
predictions = clf.predict(new_data)
print(predictions)
```

2. 1.2 最大间隔原理

SVM的目标是找到一个决策边界，即超平面，它能够将不同类别的数据点尽可能远地分开。这个决策边界的选择基于最大间隔原则，即最大化支持向量（最靠近决策边界的训练数据点）到决策边界的距离。

2.1 图解

最大间隔原理
最大间隔原理

3.1.3 线性可分SVM

当数据线性可分时，SVM通过找到一个线性超平面来分类数据。这个超平面由支持向量确定，支持向量是距离超平面最近的数据点。线性SVM的目标是最大化这个间隔，同时确保所有数据点都在超平面的正确一侧。

3.1 示例代码

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm

# 生成线性可分数据
X = np.array([[3, 4], [1, 4], [1, 0],
              [4, 1], [3, 2], [5, 2]])
y = [0, 0, 0, 1, 1, 1]

# 创建线性SVM分类器
clf = svm.SVC(kernel='linear')

# 训练模型
clf.fit(X, y)

# 绘制决策边界
w = clf.coef_[0]
a = -w[0] / w[1]
xx = np.linspace(0, 5)
yy = a * xx - (clf.intercept_[0]) / w[1]

b = clf.support_vectors_[0]
yy_down = a * xx + (b[1] - a * b[0])
b = clf.support_vectors_[-1]
yy_up = a * xx + (b[1] - a * b[0])

plt.plot(xx, yy, 'k-')
plt.plot(xx, yy_down, 'k--')
plt.plot(xx, yy_up, 'k--')

plt.scatter(clf.support_vectors_[0], clf.support_vectors_[0],
            s=80, facecolors='none')
plt.scatter(X[:, 0], X[:, 1], c=y, cmap=plt.cm.Paired)

plt.axis('tight')
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/246000201032010201>