



首都师范大学

人工智能数学基础Python实践

第十章 数据的空间变换

张苗苗

信息工程学院

□在Python中实现以下操作：

- 常用核函数
- SVM分类算法

□所需的python库：

- `numpy`
- `scikit-learn`

练习1-- 常用核函数的实现



在python中实现以下常用的三种核函数，并计算
 $X = [1,2,3,4]$, $Y = [5,6,7,8]$ 分别使用不同核函数
计算的结果。

$$K(x, y) = x \cdot y$$

$$K(x, y) = [\gamma(x \cdot y) + c]^d$$

$$K(x, y) = e^{\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)}$$

练习1-- 常用核函数的实现



```
import numpy as np
def linear(X, Y): #线性核函数
    K= X.dot(Y.T)
    return K

def gaussian (X,Y, sigma): #高斯核函数， sigma为参数
    K= np.exp(-np.linalg.norm(X-Y)**2 / (2 * sigma**2))
    return K

def poly(X, Y, gamma,c,degree): #多项式核函数
    K = X.dot(Y.T)
    K= (gamma*K + c)**degree
    return K

#计算
X=np.array([[1,2,3,4]]) #4维行向量
Y=np.array([[5,6,7,8]])
print("使用线性核函数计算的结果为: ",linear(X,Y))
print("使用高斯核函数计算的结果为: ",gaussian(X,Y,1))
print("使用多项式核函数计算的结果为: ",poly(X,Y,1,0,1))
```

□ Scikit-learn库是一个通用型开源机器学习库，它几乎涵盖了所有的机器学习算法，并且搭建了高效的数据挖掘的框架。

□分类算法库

包括**SVC**， NuSVC， 和LinearSVC 3个类

对于SVC， NuSVC， 和LinearSVC 3个分类的类， SVC和 NuSVC差不多， 区别仅仅在于**对损失的度量方式不同**， 而LinearSVC是线性分类， 也就是不支持各种低维到高维的核函数， 仅仅支持线性核函数， 对线性不可分的数据不能使用

□回归算法库

□ 基本流程

- 导入模块:

```
from sklearn.svm import SVC
```

- SVC()实例化:

```
model = SVC()
```

- 训练集训练模型:

```
model = model.fit(x,y)
```

- 测试集预测标签:

```
label = predict(X)
```

□ **SVC重要参数: C, kernel, degree, gamma, coef0**

```
sklearn.svm.SVC(C=1.0, kernel='rbf', degree=3,  
gamma='auto', coef0=0.0, shrinking=True,  
probability=False, tol=0.001, cache_size=200,  
class_weight=None, verbose=False, max_iter=-1,  
decision_function_shape=None, random_state=None)
```

C: 惩罚系数，默认值为1.0，C越大，即对分错样本的惩罚程度越大，因此在训练样本中准确率越高，但是泛化能力降低，也就是对测试数据的分类准确率降低。相反，减小C的话，容许训练样本中有一些误分类错误样本，泛化能力强。对于训练样本带有噪声的情况，一般采用后者，把训练样本集中错误分类的样本作为噪声。

kernel: 算法中采用的核函数类型，默认是rbf,高斯径向基核函数。

degree: 只对kernel = 'poly' 有用，表示选择的多项式的最高次数，默认值是3。

gamma: 选择rbf,poly,和sigmoid函数作为kernel后自带的参数，gamma越大，支持向量越少，可能导致过拟合；gamma越小，支持向量越多，导致欠拟合。默认值是' auto' ，是使用特征位数的倒数。

coef(): 是kernel=' poly' 或 'sigmoid' 设置的核函数的常数值。

SVC中的方法:

- **decision_function(X)**: 数据集X到分类超平面的距离
- **fit(x,y)**: 在数据集 (x,y) 上使用SVM模型
- **predict(X)**: 预测数据值X的标签
- **score(X,y)**: 返回给定测试集和对应标签的平均准确率

SVC中的属性:

- **support_**: 以数组的形式返回支持向量的索引，即在所有的训练样本中，哪些样本成为支持向量
- **support_vectors_**: 返回支持向量，当前模型的所有支持向量
- **dual_coef_**: 支持向量在决策函数中的系数
- **coef**: 特征系数，只有线性核函数的时候可用
- **intercept_**: 核函数中的常数项（截距值），与coef_共同构成核函数的参数值

练习2-- 线性可分SVM实现



训练数据集：

正样点： (4,3) (3,3)

负样点： (1,1)

利用SVC对测试集 (4,5) (0,0) , (1,3) 进行预测
(P303, 10.6.7)

步骤：

- 1. 准备数据**
- 2. 训练算法**
- 3. 预测数据**
- 4. 查看相关参数**

练习2-- 线性可分SVM实现



x #训练数据

y

X # 测试数据

model = SVC (kernel='linear') #模型

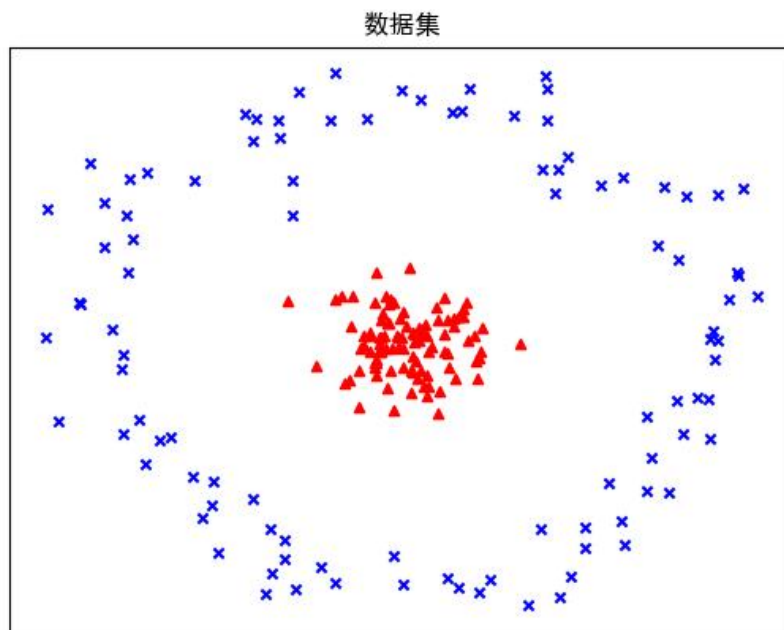
model.fit (x,y)

model.predict(X) # 预测

输出模型的参数

练习3-- 非线性SVM实现

随机生成如图所示的两类数据，构造三种核函数（线性、多项式、高斯）的算法拟合数据集，并画出拟合的分类超平面（P306, 10.7.2）



随机生成如图所示的两类数据，构造三种核函数（线性、多项式、高斯）的算法拟合数据集，并画出拟合的分类超平面（P310，综合实例）



首都师范大学

谢谢!