k-近邻 k Nearest Neighbors

概述

k-近邻（简记kNN, k-NearestNeighbor）算法是一种基本分类与回归方法，分类针对的是具有离散标签的数据，回归针对的是具有连续标签的数据。

k近邻方法背后的原理是从训练样本中找到与新点在距离上最近的k个点，然后用这k个点预测标签。距离使用任何度量，欧氏距离（standard Euclidean distance）是最常见的选择。kNN是非泛化的机器学习方法，因为它只是简单地”记住”了其所有的训练数据。

k 近邻算法实际上利用训练数据集对特征向量空间进行划分，属于非参数方法。

k值的选择、距离的度量以及**决策规则** 是k近邻算法的三个基本要素。

算法流程

假设有一个带有标签的样本数据集，输入没有标签的新数据后，将新数据的每个特征与样本集中数据对应的特征进行比较，从而的出新数据的分类或预测值：

1. 计算新数据与样本数据集中每条数据的距离；
2. 对求得的所有距离进行排序（从小到大，越小表示越相似）；
3. 取前k个样本数据对应的标签，根据决策规则计算预测值。

算法特点

优点：对异常值不敏感，无数据输入假定

缺点：模型大(跟训练数据一样大)，计算复杂度高，高维情况下不可靠(再多的训练数据在高维下都是稀疏的)

适用数据范围：数值型和标称型

补充

算法流程中计算复杂度主要在 (1) (2)，实践上一般会采用空间索引的技术降低复杂度，例如 KD-Tree, Ball-Tree

用scikit-learn实现kNN分类器

数据说明

数据存放在csv文件 datingTestSet.csv ，无header，每一行是一个约会网站用户的属性以及在用户A心目中的分类。
前三列分别为每年获得的飞行里程数、玩电子游戏所耗时间百分比、每周消费的冰淇淋数，第四列为某用户A对这些用户的评价，有3类标签：1表示不喜欢; 2表示有点喜欢; 3表示非常喜欢

模型说明

class sklearn.neighbors.KNeighborsClassifier(n_neighbors=5, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski', metric_params=None, n_jobs=1, **kwargs)

数据标准化

kNN依赖距离的度量，为了消除量纲的影响，应当对数值型自变量进行标准化

使用sklearn.preprocessing.StandardScaler()创建scaler，并用训练数据训练scaler。

在预测阶段，也要用同一个scaler对测试的输入标准化。

距离的度量

metric : string or callable, default 'minkowski'

p : integer, optional (default = 2)

minkowski距离: 向量差的p范数

决策规则

weights : str or callable, optional (default = 'uniform')
'uniform' : 多数投票

'distance' : 按距离加权投票(距离越近权值越大)

[callable] : 自定义

k值的选择

对不同的k，训练集做KFold-CrossValidation，找到平均score最高的k

使用sklearn.model_selection.cross_val_score做KFold-CrossValidation

代码

详看 kNNC-dating.py