K近邻算法

概述

算法采用测量不同特征值之间的距离的方法进行分类。优点是该算法精度高、对异常值不敏感，无数据输入的各种假定；缺点是计算复杂度和空间复杂度都比较高。算法适用的数据类型主要是数值型和标称型。

工作原理

存在一个样本数据集合，也称训练样本集，样本集中的数据都存在标签。在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，然后算法提取样本集中特征最相似数据的分类标签。一般来说我们选择样本数据集中前k个最相似的数据。从k个最相似数据中出现次数最多的分类，作为新数据的分类。

算法基本流程  
a. 计算已知类别数据集中的点与当前点之间的距离

b. 按照距离递增次序排序

c. 选取与当前点距离最小的k个点

d. 确定前k个点所在类别的出现频率

e. 返回前k个点出现频率最高的类别作为当前点的预测分类

python实现kNN核心代码

示例：使用kNN改进约会网站配对效果

分析并处理数据集

样本数据中主要包含三个不同的特征，每年获得的飞行常客里程数，玩视频游戏所耗时间的百分比，每周消费的冰激凌公升数以及标签信息，我们通下面的函数将文本信息转换为python可以分析处理的训练样本矩阵和类标签向量

Python file2matrix()

归一化数值

我们计算的距离的时候，要求每个特征对结果的影响的贡献权值接近，然而飞行里程数远大于另外两个特征，导致其对结果的贡献权值更大，我们利用下述公式处理三个特征值：

NewValue=(oleValue-min)/(max-min)

这样我们将数据转换到了0和1之间，代码如下：

Python代码-归一化特征值

测试

我们将数据经过上述出来后，计算测试向量的数量，将数据分为测试集合训练集两部分，将其输入到上面的kNN核心分类器中，计算错误率并返回结果

Python 代码-测试

然后我们可以写一个主函数，允许用户自己输入三个特征值来进行判断，本文略去

小结

优点：

1.简单好用，容易理解，精度高，理论成熟，既可以用来做分类也可以用来做回归；

2.可用于数值型数据和离散型数据；

3.训练时间复杂度为O(n)；无数据输入假定；

4.对异常值不敏感

缺点：

1.计算复杂性高；空间复杂性高；

2.样本不平衡问题（即有些类别的样本数量很多，而其它样本的数量很少；

3.一般数值很大的时候不用这个，计算量太大。但是单个样本又不能太少 否则容易发生误分。

4.最大的缺点是无法给出数据的内在含义。

所以笔者认为在数据明确而且相对适中的时候，kNN算法进行分类效果很好