K近邻算法

K近邻算法（K-Nearest Neighbor，KNN）是一种简单、懒惰（Lazy Learning）的有监督学习算法。简单是因为其不使用参数估计，只考虑特征变量之间的距离，基于距离来解决分类问题或回归问题。懒惰是因为其没有显式的学习过程或训练过程，只有接到预测任务时才会开始寻找近邻点，预测效率相对偏低。同时K近邻算法也是针对有响应变量的数据集，所以也是一种有监督学习方式。 K近邻算法既能够用来解决分类问题，也能够用来解决回归问题。

目录

CONTENTS

PART 01

K近邻算法的基本原理

K近邻算法的基本原理

K近邻算法的基本原理是，首先通过所有的特征变量构筑起一个特征空间，特征空间的维数就是特征变量的个数，然后针对某个测试样本d_i，按照参数 K 在特征空间内寻找与其最为近邻的 K 个训练样本观测值，最后依据这 K 个训练样本的响应变量值或实际分类情况，获得测试样本d_i的响应变量拟合值或预测分类情况。

其中对于分类问题，按照“多数票规则”来确定，也就是说，K个训练样本中包含样本数最多的那一类是什么，测试样本d_i的分类就是什么；针对回归问题，则按照 K 近邻估计量来确定，也就是将 K 个训练样本响应变量值的简单平均值作为测试样本d_i的响应变量拟合值。

由此可以看出，K近邻算法比较简单，也没有使用参数，所以当不了解数据分布或者没有任何先验知识时，K 近邻算法是一个不错的选择。

K近邻算法的基本原理

在使用K近邻算法时，需要注意以下事项：

1、**所有的特征变量均需为连续变量。**这是因为K近邻算法中的核心概念是“近邻”，那么怎么来衡量“近邻”呢，就需要定义“距离”，通常情况下是用欧氏距离。

欧氏距离是最常见的距离度量，衡量的是多维空间中两个样本点之间的绝对距离。假设有n个特征（即n维特征空间），训练样本x的特征变量向量为$(x_1, x_2, \ldots\ldots, x_n)$,训练样本y的特征变量向量为$(y_1, y_2, \ldots\ldots, y_n)$，则测试样本y与训练样本x之间的欧式距离为：

$$d(x,y) := \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

从上面的公式也可以看出，欧式距离是针对连续变量的，如果特征变量为分类变量，将欧式距离将无法计算。

K近邻算法的基本原理

2、在进行 K 近邻算法之前，需要对特征变量进行标准化。因为原始数据的量纲差距可能很大，如果不进行标准化，可能会因为量纲的问题引起距离测算的较大偏差，容易被那些取值范围大的变量所主导，可能会造成某个特征值对结果的影响过大，从而会大大降低模型的效果。

3、当 k 值较小时，K 近邻算法对于“噪声”比较敏感。因为 K 近邻算法在寻找 K 个近邻值时，并不依据响应变量的信息，只是通过特征变量在特征空间内寻找，所以即使一些特征变量对于响应变量的预测是毫无意义的（也就是所谓的“噪声”），也会被不加分别的考虑，从而在一定程度上形成了干扰，影响了预测效果。

4、K 近邻算法适用于样本示例全集容量较大，且远大于特征变量数的情形。因为对于高维数据来说，针对特定测试样本，可能很难找到 K 个训练样本近邻值，估计效率也会大大下降。

在 python 中，K 近邻算法针对分类问题的函数为 KNeighborsClassifie( )，而针对回归问题的函数 KNeighborsRegressor( )。

K值的选择

K值代表着在K近邻算法中选择多少个近邻值用于算法构建，也是K近邻算法中最为重要的参数。比较极端的情形有两种，一种是K取值为1的时候，也就说对于特定样本示例，只选择与自己最近的近邻值，最近近邻值的响应变量值为多少或者分为何类，特定样本示例也就相应的取值为多少或者分为何类，完全一致；另一种是K取值为整个样本容量的时候，也就是说对于每一个样本示例，都是用所有的样本示例作为其近邻值，不再区分。

K值的选择

不难看出，两种极端情形都不可取。更一般的，如果K取值过小，整体模型变得复杂，那么就仅使用较小的邻域中的训练样本进行预测，只有距离非常近的（相似的）样本才会起作用，会导致模型的方差过大，或者说容易过拟合，导致模型的泛化能力不足，比如在前述K取值为1的时候，可以通过数学证明其泛化错误率上界为贝叶斯最优分类器错误率的两倍；而K取值过大，整体的模型变得简单，那么就会使用过大邻域中的训练样本进行预测，距离非常远的（不太相似的）样本也会起作用，会导致模型的偏差过大，或者说容易欠拟合，导致模型没有充分利用最临近（相似）样本的信息，比如在前述K取值为整个样本容量的时候，K近邻算法对每一个测试样本的预测结果将会变成一样（或者说每个测试样本在整个样本全集中的位置与地位都是一样的）。

K值的选择

所以，从K取值为1开始，随着取值的不断增大，K近邻算法的预测效果会逐渐提升，然而当达到一定数值（也就是最优数值）后，随着取值进一步的增大，K近邻算法的预测效果就会逐渐下降。针对特定问题，需要找到最为合适的K，使得K近邻算法能够达到最优效果。

在k值的具体选择方面，一是在大多数情况下，k值都比较小；二是为了避免产生相等占比的情况，k值一般取奇数；三是为了更加精确的找到合适的K值，建议设置一个K取值区间，然后通过交叉验证等方法分别计算其预测效果，从而找到最好的K值。

K近邻算法的变种

一、设定K个近邻样本的权重

前面我们介绍的K近邻算法，K个训练样本的地位是完全一样的，只要成为了K个中的一个，不论这些训练样本与测试样本d_i之间的距离如何，都会被不加区别的对待。但是在很多情况下，用户可能会希望给予距离测试样本d_i更近的训练样本以更大的权重，这时候就可以在KNeighborsClassifie或KNeighborsRegressor函数中加入weights参数。

weights：用于设置近邻样本的权重，可选择"uniform", “distance” 或自定义权重。

默认选项为"uniform"，也就是说所有最近邻样本权重都一样。

选项"distance"，意味着训练样本权重和特定测试样本 di 的距离成反比例，距离越近、权重越大。如果样本是呈簇状分布的，即不同种类的样本在特征空间中聚类效果较好，那么采取"uniform"是不错的，但是如果样本分布比较乱，不同类别的样本相互交织在一起，那么使用选项"distance"，即在预测类别或者做回归时，更近的近邻所占的影响因子更大，可能预测效果更优。

除此之外，用户还可自定义权重，进一步调优参数。

K近邻算法的变种

二、限定半径最近邻法

前面我们讲述的 K 近邻算法，针对某个测试样本d_i，是在特征空间内寻找与其最为近邻的 K 个训练样本观测值来完成计算，不论这些近邻值与特定测试样本d_i实际的距离如何，只考虑找到 K 个值；还有一种算法是限定半径最近邻法，使用距离半径的方式，也就说针对某个测试样本d_i，在特征空间内寻找距离半径（radius）以内的训练样本，只要是在距离半径以内的训练样本，不论其个数有多少，都将其作为近邻值来处理，在 python 中，限定半径最近邻法分类函数和回归函数分别为：

RadiusNeighborsClassifier 和 RadiusNeighborsRegressor。

半径的设置通过参数 radius 来实现，参数默认值是 1.0，在具体数值的选择方面，总体原则是应尽量保证每类训练样本与其他类别样本之间的距离更远，所以半径的选择与样本分布紧密相关，用户可以通过交叉验证法来选择一个较小的半径。

此外，在 RadiusNeighborsClassifier 和 RadiusNeighborsRegressor 中也可以加入 weights 参数，用于调节距离在半径以内的近邻样本的权重，其含义与 K 近邻算法相同。

PART 02

数据准备

数据准备

本节我们以“数据4.1”和“数据8.1”为例进行讲解，关于数据详情可参阅“第四章 线性回归算法”及“第八章　朴素贝叶斯算法”中的相关介绍。

针对“数据4.1”，我们讲解回归问题的K近邻算法，以V1营业利润水平作为响应变量，以V2固定资产投资、V3平均职工人数、V4研究开发支出作为特征变量。

针对“数据8.1”，我们讲解分类问题的K近邻算法，以转型情况（V1）为响应变量，以存款规模（V2）、EVA（V3）、中间业务收入（V4）、员工人数（V5）为特征变量

载入分析所需要的模块和函数

在进行分析之前，我们首先载入分析所需要的模块和函数，读取数据集并进行观察。

示例

参阅教材内容

PART 03

回归问题K近邻算法示例

变量设置及数据处理

示例

参阅教材内容

构建K近邻回归算法模型

示例

参阅教材内容

如何选择最优的K值

示例

参阅教材内容

最优模型拟合效果图形展示

示例

参阅教材内容

PART 04

分类问题K近邻算法示例

变量设置及数据处理

示例

参阅教材内容

构建K近邻分类算法模型

示例

参阅教材内容

如何选择最优的K值

示例

参阅教材内容

最优模型拟合效果图形展示

示例

参阅教材内容

绘制K近邻分类算法ROC曲线

示例

参阅教材内容

运用两个特征变量绘制K近邻算法决策边界图

示例

参阅教材内容

普通KNN算法、带权重KNN、指定半径KNN三种算法对比

一、基于验证集法（将样本分割为训练样本和测试样本）进行对比

二、基于10折交叉验证法进行对比

示例

参阅教材内容

PART 05

习　题

习题

1、使用使用“数据5.1”数据文件（详情已在第5章中介绍），把响应变量设定为“V1征信违约记录”，将其他变量作为特征变量，具体包括“V2资产负债率”、“V6主营业务收入”、“V7利息保障倍数”、“V8银行负债”、“V9其他渠道负债”，构建K近邻分类算法模型。

（1）载入分析所需要的库和模块

（2）变量设置及数据处理

（3）以K能取到的最小值、最大值、中间值分别构建K近邻分类算法模型

（4）选择最优的K值，利用最优K值构建K近邻分类算法模型

（5）图形化展示最优模型拟合效果

（6）绘制K近邻分类算法ROC曲线

感谢聆听

THANKS