K 近邻法

判别模型

张峻伟
2017年10月18日

概念：

给定一个训练数据集，对新的输入实例，在训练数据集中找到与该实例最邻近的k个实例。

来源：

解决一个测试对象同时与多个训练对象匹配，导致一个训练对象被分到了多个类的问题，究竟属于哪一个类的问题。

主要内容：

K近邻算法、模型及三个基本要素、实现方法

KNN模型及三个基本要素

- 距离度量

L_p 距离定义为:

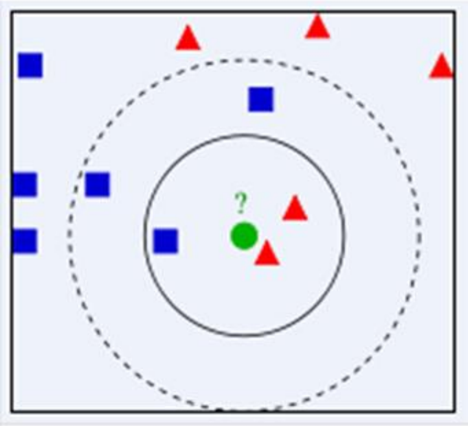
$$L_p(x_i, x_j) = \left(\sum_{l=1}^{n} |x_i^{(l)} - x_j^{(l)}|^p \right)^{\frac{1}{p}}$$

其中$x_i \in \mathbf{R}^n$, $x_j \in \mathbf{R}^n$, 其中L_∞定义为：

$$L_\infty(x_i, x_j) = \max_l |x_i^{(l)} - x_j^{(l)}|$$

- K值的选择

与实例相似的实例个数为k，当k较小时，**近似误差减小，估计误差增大**，易受噪声污染和过拟合；**一般采用小的K值，再采用交叉验证法**。**近似误差类似训练误差**，指与最优结果的相似程度大小；**估计误差**指与最优误差之间的相近程度大小



- 分类决策规则

多数表决法

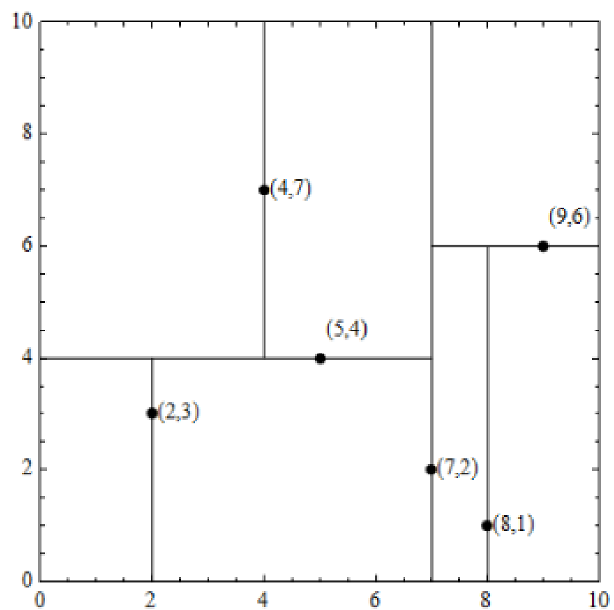
KNN的实现：kd树

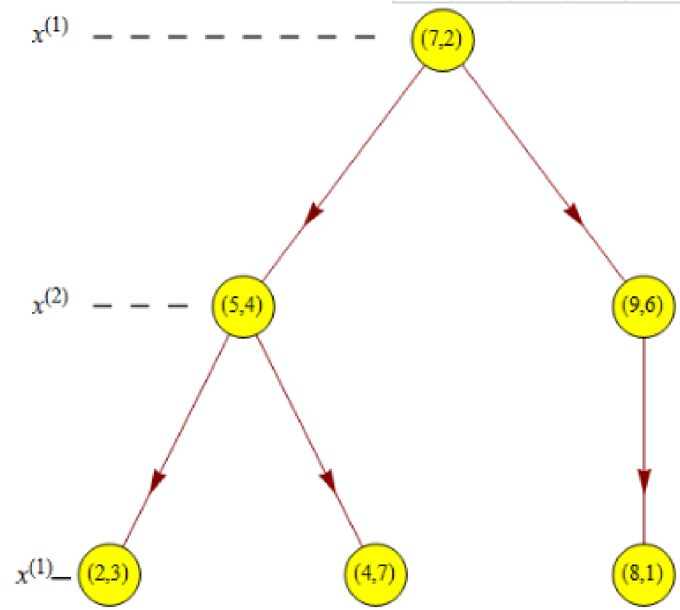
● 构造kd树

输入：K维空间数据集T，其中 $x_i=(x_i^{(1)},x_i^{(2)},\dots,x_i^{(k)}),i=1,2,\dots,N$
输出：kd树

例. 给定一个二维空间数据集：$T=\{(2,3),(5,4),(9,6),(4,7),(8,1),(7,2)\}$，构造一个平衡kd树。

解：根结点对应包含数据集T的矩形，选择$x^{(1)}$轴，6个数据点的$x^{(1)}$坐标中位数是6，这里选最接近的(7,2)点，以平面$x^{(1)}=7$将空间分为左、右两个子矩形（子结点）；接着左矩形以$x^{(2)}=4$分为两个子矩形（左矩形中{(2,3),(5,4),(4,7)}点的$x^{(2)}$坐标中位数正好为4），右矩形以$x^{(2)}=6$分为两个子矩形，如此递归，最后得到如下图所示的特征空间划分和kd树。





● 搜索kd树

- 在kd树中找出包含目标点x的叶结点：从**根结点**出发，递归的向下访问kd树。若目标点**当前维的坐标值小于切分点的坐标值**，则移动到左子结点，否则移动到右子结点。直到子结点为叶结点为止；
- 以此叶结点为“当前最近点”；
- 递归的**向上回退**，在每个结点进行以下操作：
  （a） 如果该结点保存的实例点比当前最近点距目标点更近，则以该实例点为“当前最近点”；
  （b） 当前最近点一定存在于该结点一个子结点对应的区域。检查该**子结点的父结点的另一个子结点**对应的区域是否有更近的点。具体的，检查另一个子结点对应的区域是否与以目标点为球心、以目标点与“当前最近点”间的距离为半径的超球体相交。如果相交，可能在另一个子结点对应的区域内存在距离目标更近的点，**移动到另一个子结点**。接着，递归的进行最近邻搜索。如果不相交，向上回退。
- 当**回退到根结点时，搜索结束**。最后的“当前最近点”即为 x的最近邻点

KNN的补充

1、最近邻算法扩展——**距离加权最近邻算法**

对K各近邻的贡献加权，根据他们相对查询点的距离，将较大的权值赋给较近的近邻。例如，在上表逼近离散目标函数的算法中，我们可以根据每个近邻与*xq*的距离平方的倒数加权这个近邻的“选举权”。提高了噪点的鲁棒性

2、不同维度权值之间的**归一化问题**（x-min）/（max-min）

3、 近邻间的距离会被大量的**不相关属性所支配**，例如：每个实例由20个属性描述，但在这些属性中仅有2个与它的分类是有关。在这种情况下，这两个相关属性的值一致的实例可能在这个20维的实例空间中相距很远。结果，依赖这20个属性的相似性度量会误导*k*-近邻算法的分类。近邻间的距离会被大量的不相关属性所支配。这种由于存在很多不相关属性所导致的难题，有时被称为维度灾难。**解决方法：** 当计算两个实例间的距离时对每个属性加权