K近邻

1.模型

K近邻模型由3要素构成：距离度量，k指选择和分类决策规则

距离度量：常见的有闵可夫斯基距离，欧式距离，曼哈顿距离

k值：直接关系到模型的复杂度，k越小，预测结果越精确，模型越复杂，对近邻的实例点越敏感

k太大则模型太简单，会忽略训练实例中的大量有用信息

分类决策规则：多数表决规则。其实就是分类损失函数为0-1函数时的损失函数最小化原则

算法1（k邻近法）

输入：训练数据集

$$T=\{(x_1,y_1),(x_2,y_2),...,(x_N,y_N)\}$$

其中$x_i \in \chi \subseteq R^n$为样本的特征向量，$y_i \in \Upsilon = \{c_1,c_2,...,c_K\}$为实例的类别，i=1,2,...,N；样本特征向量x（新样本）；

输出：样本x所属的类y。

（1）根据给定的距离度量，在训练集T中找出与x最相邻的k个点，涵盖这k个点的邻域记作$N_k(x)$；

（2）在$N_k(x)$中根据分类决策规则（如多数表决）决定x的类别y：

$$y = arg\ \max_{c_j} \sum_{x_i \in N_k(x)} I(y_i = c_i), i=1,2,...,N; j=1,2,...,K$$

式中I为指示函数，即当$y_i = c_i$时I为1，否则为0。

2.kd树

可参照https://zhuanlan.zhihu.com/p/22557068理解

每次寻找

算法 3.2（构造平衡 *kd* 树）

输入：k 维空间数据集 $T=\{x_1,x_2,\cdots,x_N\}$，

其中 $x_i=(x_i^{(1)},x_i^{(2)},\cdots,x_i^{(k)})^{\mathrm{T}}$，$i=1,2,\cdots,N$；

输出：*kd* 树.

（1）开始：构造根结点，根结点对应于包含 T 的 k 维空间的超矩形区域.

选择 $x^{(1)}$ 为坐标轴，以 T 中所有实例的 $x^{(1)}$ 坐标的中位数为切分点，将根结点对应的超矩形区域切分为两个子区域．切分由通过切分点并与坐标轴 $x^{(1)}$ 垂直的超平面实现.

由根结点生成深度为 1 的左、右子结点：左子结点对应坐标 $x^{(1)}$ 小于切分点的子区域，右子结点对应于坐标 $x^{(1)}$ 大于切分点的子区域.

将落在切分超平面上的实例点保存在根结点.

（2）重复：对深度为 j 的结点，选择 $x^{(l)}$ 为切分的坐标轴，$l=j(\mathrm{mod}\,k)+1$，以该结点的区域中所有实例的 $x^{(l)}$ 坐标的中位数为切分点，将该结点对应的超矩形区域切分为两个子区域．切分由通过切分点并与坐标轴 $x^{(l)}$ 垂直的超平面实现.

由该结点生成深度为 $j+1$ 的左、右子结点：左子结点对应坐标 $x^{(l)}$ 小于切分点的子区域，右子结点对应坐标 $x^{(l)}$ 大于切分点的子区域.

将落在切分超平面上的实例点保存在该结点.

（3）直到两个子区域没有实例存在时停止．从而形成 *kd* 树的区域划分．■

例子：

例 3.2 给定一个二维空间的数据集：

$$T=\{(2,3)^{\mathrm{T}},(5,4)^{\mathrm{T}},(9,6)^{\mathrm{T}},(4,7)^{\mathrm{T}},(8,1)^{\mathrm{T}},(7,2)^{\mathrm{T}}\}$$

构造一个平衡 *kd* 树[3].

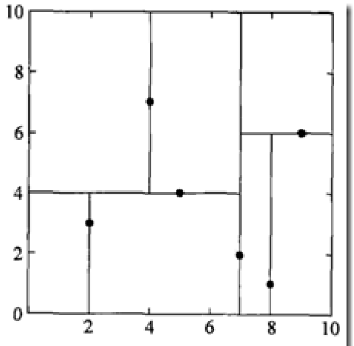

图 3.3 特征空间划分

搜索kd树：

对于搜索k近邻，就是k次搜索最近邻，因此只需要关注最近邻搜索

例子：

例 3.3 给定一个如图 3.5 所示的 *kd* 树，根结点为 *A*，其子结点为 *B*，*C* 等. 树上共存储 7 个实例点；另有一个输入目标实例点 *S*，求 *S* 的最近邻.

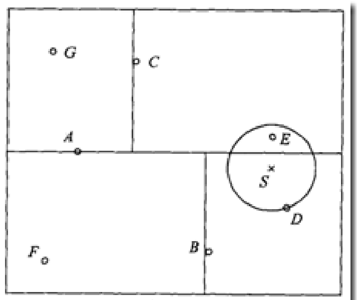

图 3.5 通过 *kd* 树搜索最近邻

假设要搜索的点为S，首先找到包含点S的那个叶节点（这里是D），以S为圆心，SD为半径画圆。如果在D节点的所有实例

点钟有距离S更近的点，那么该点一定在这个圆内。以D节点为近似最近邻，然后回溯到父节点B，在B的另一个子节点F内搜索最近邻。F的区域与圆不相交，不可能有最近邻点。继续向上回溯到节点A，在A的另一个子节点C内搜索最近邻。C区域与圆相交，切点E比D距离S更近，E成为新的最近邻。最后得到点E是S的最近邻。