第3章 k近邻法 (k-NN)

可分类 < 2分 / 多分

可回归

3.1 k近邻算法

算法 3.1

输入：训练数据集 $T=[(x_1,y_1),\cdots,(x_N,y_N)]$

$x_i \in X \subseteq R^n$，$y_i \in Y=\{c_1,\cdots,c_k\}$ (待)

实例特征向量 x (距离衡量 方法总结)

1. 根据给定的距离度量（→欧氏距离），在训练集中找到与 x 最近的 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

输出：实例 x 所属的类别 y.

3.2 k近邻模型

k近邻方法没有显示的模型形式

单元 (cell) 当 $k=1$ 时，叫最近邻模型

没有显示模型形式的，我们都要回到训练集中去找

★ 区别之前模型. 之前从数据集中 → 模型

断开

x_{N+1} → 模型 → y_{N+1}

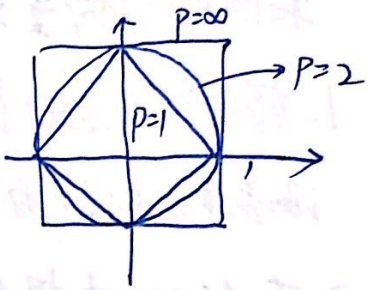
距离度量

欧氏距离 $\|x-y_i\|_2$

L_p 距离（更一般的表示方法）

$$L_p(x_0, x_i) = \sqrt[p]{|x^{(1)} - x_i^{(1)}|^p + |x^{(2)} - x_i^{(2)}|^p + \dots + |\quad|^p}$$

p 一般取 1（曼哈顿）, 2（欧氏）, ∞



$p=1$：$L_1(x_i, x_j) = \sum_{l=1}^{n} |x_i^{(l)} - x_j^{(l)}|$

$p=2$：$L_2(x_i, x_j) = \left(\sum_{l=1}^{n} |x_i^{(l)} - x_j^{(l)}|^2\right)^{\frac{1}{2}}$

$p=\infty$ $L_\infty(x_i, x_j) = \max_l |x_i^{(l)} - x_j^{(l)}|$ —各距离坐标最大值

k值的选择

采用交叉验证方法

① 将数据合为 { 训练集 / 验证集

② 从验证集拿出实例，给出特征向量 x

③ 在训练集中取不同 k 值来看 x 类别

④ 与验证集真实类别做比较，从而确定 k

分类决策规则

$$\frac{1}{k}\sum_{x_i\in N_k(x)} I(y_i \neq c_j)$$

多数表决规则等价于经验风险最小化

3.3 k近邻法的实现：kd树（二叉树） $O(\log N)$

↗不是k类，而是k维的意思

↖训练实例

① 时间复杂度 $O(\log N)$

1) 适用于训练实例数远大于空间维数.

2) 训练实例数接近空间维数，kd数效率越接近线性扫描

② 目的：快速搜索k个最近邻点

③ 每个结点对应于k维空间划分中的一个超矩形区域

算法3.2（构造平衡kd树）

输入：k维空间数据集 $T=\{x_1, x_2, \cdots, x_N\}$，$x_i=(x_i^{(1)}, \cdots x_i^{(k)})^T$

(1) 构造根节点，包含T中k维全部~

(2) 选择 $x^{(1)}$ 为坐标轴，以所有实例的中位数为切分点

切分由通过切分点并与坐标轴 $x^{(1)}$ 垂直的超平面实现

↳父节点，左子节点：$x^{(1)}$ 小于切分~，右~大于 $x^{(1)}$~

(3) 重复：对深度为 j 的结点，选择 $x^{(l)}$ 为切分坐标轴，

$l=j(\mathrm{mod}\ k)+1$，直到没有实例存在

↑

★这个深度 j.（根节点的 j 为 0）

算法3.3（用kd树的最近邻搜索）

① 选找到叶结点（小于往左，大于往右）

② 从此叶结点为"当前最近点"

③ 递归地向上回退，在每个结点上：

1) 重新找最近点

2) 画圆，半径内找相交 ／ 不相交 — 回退
                        ＼ 相交 — 移到另一子结点，继续递归

④ 回退到根节点，搜索结束。

最后的"当前最近点"即为 x 的最近邻点。