智能软件开发方向基础

第四章 K-近邻模型
K-Nearest Neighbour

张朝晖

2023.3~6

河北师范大学软件学院
Software College of Hebei Normal University

序号	内容
1	概述
2	机器学习的基本概念
3	模型的选择与性能评价
4	数据的获取、探索与准备
5	近邻模型------分类、回归
6	决策树模型------分类、回归
7	集成学习------分类、回归
8	(朴素)贝叶斯模型------分类
9	聚类
10	特征降维及低维可视化(PCA, t-SNE)
11	总复习

K近邻模型的关键问题

- 分类问题的描述、分类模型的性能评价
- 回归问题的描述、回归模型的性能评价
- 距离度量
- 样本的规范化预处理

1. K近邻分类
   - K近邻分类的算法描述
   - K近邻分类系统决策性能的影响因素
   - 如何面向分类，进行K值优选
   - 决策规则
2. K近邻回归
   - K近邻回归的算法描述
   - K近邻预测系统性能的影响因素
   - 如何面向回归问题，进行K值优选
   - 预测规则

河北师范大学软件学院
Software College of Hebei Normal University

PART1. K-近邻分类

PART2. K-近邻回归

河北师范大学软件学院
Software College of Hebei Normal University

PART1. K近邻分类

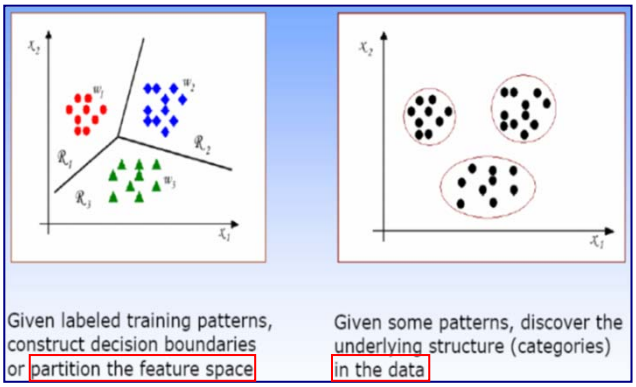
K-Nearest Neighbor Classification

关键词：

分类问题的定义
KNN分类模型
距离度量与样本的规范化预处理
超参数
交叉验证
性能评价

河北师范大学软件学院
Software College of Hebei Normal University

分类(Classification)与聚类(Clustering)的区别



河北师范大学软件学院
Software College of Hebei Normal University

主要内容

1. 分类问题的一般描述
2. K近邻分类算法的描述
3. K近邻分类的三个基本要素
   - 距离度量
   - 超参数K值的确定
   - 决策规则
4. K近邻算法的实现—kd树(k-dimensional Tree)
   - Kd树的构建
   - 待决策样本的K近邻搜索分支定界搜索
5. K近邻分类模型的评价(思考)

1. 分类问题的一般描述

给定带有类别标记的训练样本集$\{(\boldsymbol{x}_i, y_i), i=1,\cdots,N\}$.

其中：

$\boldsymbol{x}_i$----第i个观测样本的特征向量，$\boldsymbol{x}_i=[x_{i1},...,x_{id}]^T \in R^d$

y_i----第i个观测样本的类别标号$\begin{cases}\boldsymbol{C}=2, & y_i \in Y=\{1,2\}\\ \boldsymbol{C}>2, & y_i \in Y=\{1,2,...,C\}\end{cases}$

要求：

基于上述样本集，设计分类模型--**分类模型的监督式学习**；

对特征空间的任意观测$\boldsymbol{x}$进行类别决策--**模型的使用**

主要内容

1. 分类问题的一般描述
2. K近邻分类算法的描述
3. K近邻分类的三个基本要素
   - 距离度量方式
   - 超参数K值
   - 决策规则
4. K近邻算法的实现—kd树(k-dimensional Tree)
5. KNN分类模型的评价

2.1K近邻分类模型的引入

近朱者赤，近墨者黑

懒惰学习

无显式的训练过程，直接进行基于训练样本的空间划分

非参数法分类模型

准备工作轻松

训练样本集、近邻数**K**、距离度量(及预处理)

2.2 K近邻分类算法的描述

输入：

(1)**训练样本集**$\boldsymbol{D}=\{(\boldsymbol{x}_i, y_i), i=1,\cdots,N\} \subset R^d \times Y$，并且有：

$\boldsymbol{x}_i$----第i个训练样本的特征向量，$\boldsymbol{x}_i \in R^d$

y_i----第i个训练样本的类别标号$\begin{cases}\boldsymbol{C}=2, & y_i \in Y=\{1,2\}\\ \boldsymbol{C}>2, & y_i \in Y=\{1,2,...,C\}\end{cases}$

(2)**观测样本**$\boldsymbol{x}$

输出： 观测样本$\boldsymbol{x}$所属的类别y.

2.2 K近邻分类算法的描述 - 续

STEP0. 训练集$\boldsymbol{D}$的输入部分预处理，并记录预处理的使用参数

STEP1. 指定**距离度量**，并**选择K值**

STEP2. 训练集$\boldsymbol{D}$内找到预处理的样本$\boldsymbol{x}$的前**K个近邻，记为**$N_K(\boldsymbol{x})$

$N_K(\boldsymbol{x})=N_{K,1}(\boldsymbol{x}) \cup \cdots \cup N_{K,C}(\boldsymbol{x})$

$N_{K,i}(\boldsymbol{x})$----$\boldsymbol{x}$的前K个近邻中来自第i类的训练样本

STEP3. 结合指定的**分类规则**，对$\boldsymbol{x}$的类别y 进行预测

$$\hat{y}=\underset{i\in Y=\{1,...,C\}}{\operatorname{argmax}} \sum_{x_j \in N_k(x)} I(y_j=i)$$

注：指示函数$I(y_j=i)=\begin{cases}1, if\ y_j=i\\ 0, if\ y_j \neq i\end{cases}$

在给定训练集的前提下：

各样本是否预处理、
不同的距离度量方式、
不同K值、
以及不同的决策规则，

会导致不同的分类结果

主要内容

1. 分类问题的一般描述
2. K近邻分类算法的描述
3. K近邻分类的三个基本要素
   3.1 距离度量与预处理
      (1)典型的距离度量方式
      (2)样本的规范化预处理
   3.2 超参数K值的确定
   3.3 决策规则
4. K近邻算法的实现—kd树(k-dimensional Tree)
5. K近邻分类模型的评价

(1)典型的距离度量方式

对于$\forall \boldsymbol{x}_i, \boldsymbol{x}_j \in \boldsymbol{X} \subseteq \boldsymbol{R}^d \quad \boldsymbol{x}_i = [x_{i1} \quad \cdots \quad x_{id}]^T \quad \boldsymbol{x}_j = [x_{j1} \quad \cdots \quad x_{jd}]^T$

A. L_p距离 $d_p(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_p = \left(\sum_{k=1}^{d} |x_{ik} - x_{jk}|^p\right)^{\frac{1}{p}} \quad p > 0$

B. 绝对值距离$(manhattan\ \ distance$，**L_1距离**$)$

$$d_1(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_1 = \sum_{k=1}^{d} |x_{ik} - x_{jk}|$$

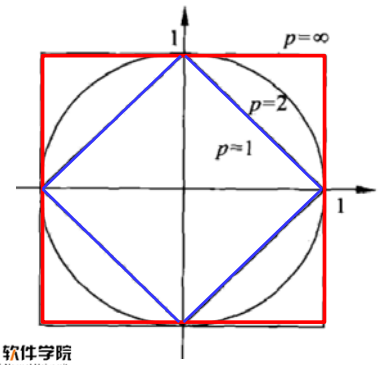
C. 欧式距离(L_2距离) $d_2(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 = \left(\sum_{k=1}^{d} |x_{ik} - x_{jk}|^2\right)^{\frac{1}{2}}$

D. 切氏距离(L_∞距离，拉格朗日距离)

$$d_\infty(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_\infty = \max_{1 \le k \le d} |x_{ik} - x_{jk}|$$

例：平面中与原点的Lp距离为1的点的集合

- P=1
- P=2
- P为无穷大



(2)用于距离度量的样本的"标准化预处理"

A."标准化预处理" 的必要性： $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_p = \left(\sum_{k=1}^{d} |x_{ik} - x_{jk}|^p\right)^{\frac{1}{p}}$

距离计算时，通常对各个特征变量均等对待。

若**大数量级、大动态范围的特征**以及**小数量级、小动态范围特征**同时参与距离估算，常面临如下风险：

起主导作用的前者可能淹没后者特征变化；

起主导作用的特征所含的类鉴别信息不一定明显

不同特征的量纲不同、固定量纲下采用不同度量单位，也会导致不同数量级的特征取值。有必要**去量纲化**。

因此，需要对所有样本的特征描述部分，进行标准化预处理。

(2)用于距离度量的样本的"标准化预处理"

B."标准化预处理" 应避免信息泄露

"标准化预处理" 所用参数取值，应来自训练样本集，而不能使用其他数据集.

C.样本特征的"标准化预处理"的方式：

特征的平移、尺度调整

sklearn.preprocessing. StandardScaler

方式1--0均值,1方差的标准化预处理(推荐使用)

首先，**利用训练集**估计各特征的两种统计量

$$\begin{cases}\mu_k = \dfrac{1}{N}\sum_{i=1}^{N} x_{ik} \\ \sigma_k = \sqrt{\dfrac{1}{N}\sum_{i=1}^{N}(x_{ik}-\mu_k)^2}\end{cases} \quad k=1,...,d$$

训练集 $\boldsymbol{D}=\{(\boldsymbol{x}_i, y_i), i=1,\cdots,N\}$

然后，任意观测样本 $\boldsymbol{x}\in\boldsymbol{R}^d$ 的预处理：

处理前 $\boldsymbol{x}=[x_1 \quad \cdots \quad x_d]^T$

处理后 $\boldsymbol{x}'=[x_1' \quad \cdots \quad x_d']^T$ 其中 $x_k' = \dfrac{x_k-\mu_k}{\sigma_k} \quad k=1,...,d$

方式2--将原始样本特征取值进行线性映射

sklearn.preprocessing.MinMaxScaler

首先，利用训练集确定 $\begin{cases}x_{k\_min} = \min\limits_{i=1,...,N}\{x_{ik}\} \\ x_{k\_max} = \max\limits_{i=1,...,N}\{x_{ik}\}\end{cases} \quad k=1,...,d$

然后，任意观测样本 $\boldsymbol{x}\in\boldsymbol{R}^d$ 的预处理：

处理前 $\boldsymbol{x}=[x_1 \quad \cdots \quad x_d]^T$

处理后 $\boldsymbol{x}'=[x_1' \quad \cdots \quad x_d']^T$

若线性映射至 $[0,1]$ 则 $x_k' = \dfrac{x_k - x_{k\_min}}{x_{k\_max}-x_{k\_min}} \quad k=1,...,d$

若线性映射至 $[-1,1]$ 则 $x_k' = \dfrac{2(x_k - x_{k\_min})}{x_{k\_max}-x_{k\_min}} - 1 \quad k=1,...,d$

河北师范大学软件学院 Software College of Hebei Normal University

主要内容

1. 分类问题的一般描述
2. K近邻分类算法的描述
3. K近邻分类的三个基本要素
   - 3.1 距离度量
     - 典型的距离度量方式、样本的规范化预处理
   - 3.2 超参数K值的确定
     - m-fold cross validation + 分类模型的评价指标
   - 3.3 决策规则
4. K近邻算法的实现—kd树(k-dimensional Tree)
5. K近邻分类模型的评价

河北师范大学软件学院 Software College of Hebei Normal University

A. 对超参数K值进行选择的意义

任意观测x的类别预测：

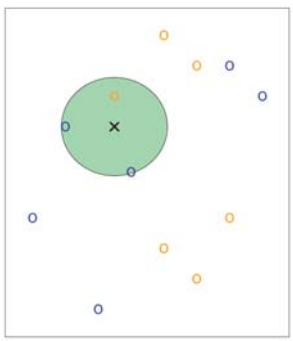
(1)若使用**较小的K值**，则利用x较小邻域训练样本进行类别预测，只有更接近x的训练样本(更相似)才对预测结果有作用，预测结果对近邻的训练样本类别更为敏感。

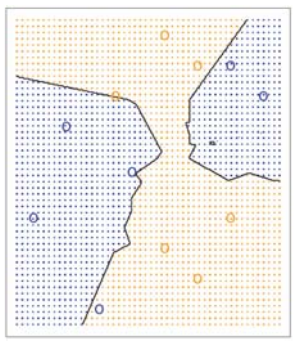
若数据分布复杂、或噪声影响严重，易导致高的"预测错误率"。模型复杂。

最小**K=1**，为**最近邻分类**。

(2)若使用**较大的K值**，则需要利用x较大邻域的训练样本进行类别预测，使得远离x的训练样本(更相异)对预测结果也有作用，使预测发生错误。模型更简单。

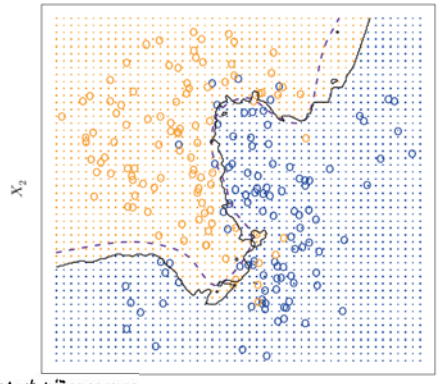
最大**K=N**，每个位置的预测结果为具有最大训练样本数目的类别。



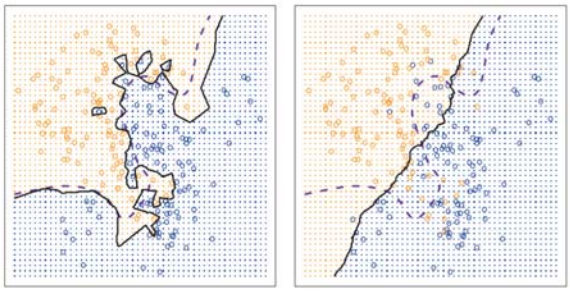


图：K-近邻规则、两类问题，基于训练样本的特征空间网格划分.K=3.
左图，训练样本； 右图，二维特征空间的决策结果

河北师范大学软件学院 Software College of Hebei Normal University



河北师范大学软件学院 Software College of Hebei Normal University



超参数空间：离散/连续
数据集的使用：交叉验证
超参数空间的基点的评价：错误率

B. 基于 m-fold cross validation 的K值选择

以单轮m-fold cross validation为例：

STEP1.训练集(分层)随机打乱，均分成m等份，每一份样本数目$\frac{N}{m}$.

STEP2.对于每个备选**K值**，

2-1. for i = 1,...,m do

拿出第i份作为**验证集**，其余m-1份构成**估计集**

利用**估计集**，对**验证集**的每个样本进行类别预测，得**验证集**预测错误率$Err_i(K)$

2-2. 估计对应于该备选**K值**的

$$\begin{cases} \text{平均错误率} & \mu_{Err(K)}=\frac{1}{m}\sum_{i=1}^{m}Err_i(K) \\ \text{标准差} & \sigma_{Err(K)}=\sqrt{\frac{1}{m}\sum_{i=1}^{m}\left[Err_i(K)-\mu_{Err(K)}\right]^2} \end{cases}$$

表示为$\mu_{Err(K)}\pm\sigma_{Err(K)}$

STEP3.取最小$\mu_{Err(K)}$对应的**K值**为最终选择结果；

若同时有多个**K值**有最小$\mu_{Err(K)}$，则取较小$\sigma_{Err(K)}$对应的**K值**。

主要内容

1. 分类问题的一般描述
2. K近邻分类算法的描述
3. K近邻分类的三个基本要素
   3.1 距离度量
   典型的距离度量方式、样本的规范化预处理
   3.2 超参数K值的确定
   m-fold cross validation + 分类模型的评价指标
   3.3 决策规则
   胜者为王(多数表决)、加权投票
4. K近邻算法的实现—kd树(k-dimensional Tree)
5. K近邻分类模型的评价

方式1. 多数表决(胜者为王)--传统的K近邻决策方式

观测x的**K个近邻**$N_K(x)=N_{K,1}(x)\cup\cdots\cup N_{K,C}(x)$

来自第j类的近邻数：

$$k_j=\sum_{x_i\in N_K(x)}I(y_i=j)=\left|N_{K,j}(x)\right| \qquad j=1,...,C$$

$$K=\sum_{j=1}^{C}k_j$$

决策规则：若$k_l=\max_{j=1,...,C}k_j$,则将x决策为第l类.

样本x与第j类的相似度：$Sim(x,j)=\frac{\left|N_{K,j}(x)\right|}{K}=\frac{k_j}{K}$ $\quad j=1,...,C$

样本x关于第j类的后验概率 $P(y=j\mid x)=\frac{k_j}{K}$ $\quad j=1,...,C$

方式1. 多数表决(胜者为王)--续

特殊情况的处理

x的**K个近邻**$N_K(x)$中，多个类别训练样本数目同时最大

即：存在多个$l\in\{1,...,C\}$，同时满足$k_l=\max_{j=1,...,C}k_j$,

则可采取如下方式之一，进行x的最终类别预测：

A. 随机选择其中一个类别

B. 将x分到其最近邻的那个类别

C. 针对这些类别k_l个近邻，分别计算相应均值向量，将x分到与其最近的均值向量的那个类别.

...

距离加权法，可以扩充至K=N

方式2. 基于距离的加权投票

观测x的**K个近邻**$N_K(x)=N_{K,1}(x)\cup\cdots\cup N_{K,C}(x)$

由近及远，$N_K(x)$中**K个训练样本**类别标号$a_1,...,a_K$，

K个训练样本关于x的距离为$\delta_{a_1}\le\delta_{a_2}\le\cdots\le\delta_{a_K}$

加权投票：统计第j类的训练样本的**决策权重**

$$W_j=\sum_{i=1}^{K}w(\delta_{a_i})I(a_i=j) \qquad j=1,...,C$$

类别决策：若$W_l=\max_{j=1,...,C}W_j$,则将x决策为第l类.

如何确定K个近邻的投票权重？

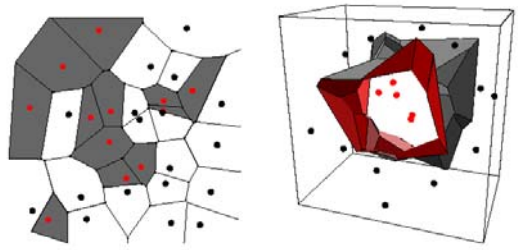
例：x的前**K个近邻**$N_K(x)$的投票权重的估计方式

A. $w(\delta_{a_i})=\dfrac{1}{\delta_{a_i}^{\ 2}}$

B. $w(\delta_{a_i})=\exp\left[-\delta_{a_i}^{\ 2}\right]$

C. $w(\delta_{a_i})=\dfrac{1}{\delta_{a_i}}, \quad \delta_{a_i}\neq 0$

各近邻的归一化权重： $w(\delta_{a_i}) \Leftarrow \dfrac{w(\delta_{a_i})}{\sum_{j=1}^{K} w(\delta_{a_j})}, i=1,...,K$



图： 最近邻规则、两类问题，基于训练样本点的特征空间 的Voronoi网格划分.

左图，二维特征空间； 右图，三维特征空间

主要内容

1. 分类问题的一般描述
2. K近邻分类算法的描述
3. K近邻分类的三个基本要素
   - 3.1 距离度量
   - 3.2 超参数K值的确定
   - 3.3 决策规则

4. K近邻算法的实现—kd树(k-dimensional Tree, k维树)
   - 对k维特征空间的训练集进行**有效存储**，以便针对新的输入样本对其近邻进行快速检索的树形结构。
   - kd树的构建、基于树的搜索

基于**训练样本集**$D=\{(x_i,y_i),i=1,\cdots,N\}$， $x_i\in R^k$
进行K近邻分类，时间复杂度为$O(Nd)$

K近邻法省略了“监督式学习”阶段
其对任意观测样本x的类别决策，均依赖于**训练集的“记忆”**

K近邻法也称为：
Non – Generalizing Machine Learning Method

如何“有效记住”训练集？
如何在“记住”的训练集内快速有效搜索x的K近邻?

注意区分：k近邻的“k”、kd树的“k”

训练样本集$D=\{(x_i,y_i),i=1,\cdots,N\}$，并且$x_i\in R^k$
[1]**kd树的构建----训练集的“有效记忆”**

构建递归二叉树，将k维特征空间的训练集按层划分，形成若干子集， 每个子集对应特征空间一个子区域

[2]**kd树的搜索--**在kd树中搜索x**的近邻**

对于任意观测样本$x\in R^d$， 以此为查询对象，在kd树内快速搜索其近邻。

kd树算法涉及两个部分：构建kd树、搜索kd树

算法1----kd树的构建 CreateKDTree
(平衡)递归二叉树
输入：
(1)**样本集**$D=\{(x_i,y_i),i=1,\cdots,N\}$
其所在空间超**矩形体***Range*
其中：
$x_i=[x_i^{(1)},\ldots,x_i^{(k)}]^T$ ----第i个训练样本的特征向量， $x_i\in R^k$

kd树是一个以划分方式存储k维空间数量有限的观测点集的数据结构--二叉树结构

(2)结点分裂的**终止条件**m_0:
到达该结点的训练样本数 $\le m_0$ 例： $m_0=1$

输出： kd树结构

例：Scikit-learn默认值=30

可以选择使用方差最大的特征进行比较和划分，以减少划分后区域内数据点的方差（数据点更密集）

STEP1. 开始：令$j=0$,**基于整个训练集构造根节点。**

(1) **确定切分特征***split*.

$l=j \bmod k+1=l$, 选择以$x^{(l)}=x^{(l)}$特征为切分特征*split*坐标轴.

(2) **确定切分点**.

将$\boldsymbol{D}$内所有样本按照切分特征*split*取值升序排列， 以*split*取值的中位数为**切分点***s*.

采用垂直于*split*坐标轴、并经过切分点的超平面为**切分面**.

(3)记录该结点所使用的切分特征及切分点，并在该结点处保存经过切分面的训练样本点

(4)**切分整个训练集所在的超矩形区域Range**，得左、右子区域.

左子区域 $\boldsymbol{LeftRange}=\{x|x\in \boldsymbol{Range}$并且$x[split]\le s\}$

右子区域 $\boldsymbol{RightRange}=\{x|x\in \boldsymbol{Range}$并且$x[split]> s\}$.

(5)同时，落入左右子区域的训练样本分别构成左右子集。

左子集 $\boldsymbol{LeftD}=\{(x,y)|(x,y)\in \boldsymbol{D},$ 并且$x[split]\le s\}$

右子集 $\boldsymbol{RightD}=\{(x,y)|(x,y)\in \boldsymbol{D},$ 并且$x[split]> s\}$

(6)由根结点生成深度为1的左右子结点.

左子结点，其对应数据集以及超矩形区域为 ***LeftD, LeftRange***

右子结点，其对应数据集以及超矩形区域为 ***RightD, RightRange***

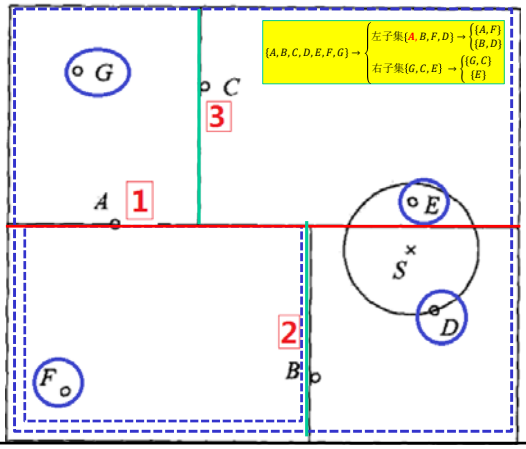
STEP2. 根结点的左子树生成.

若$|\boldsymbol{LeftD}|>\mathbf{m}_0$,对于深度为$\mathbf{j=1}$的左子结点，基于***LeftD***以及***LeftRange***，调用**CreateKDTree**，递归生成左子树.

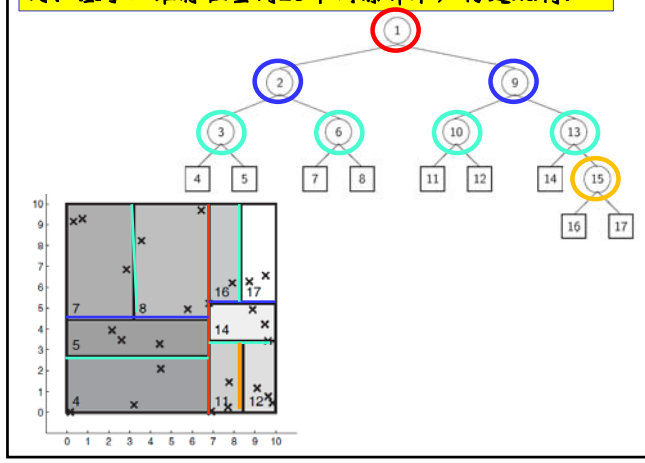
STEP3. 根结点的右子树生成.

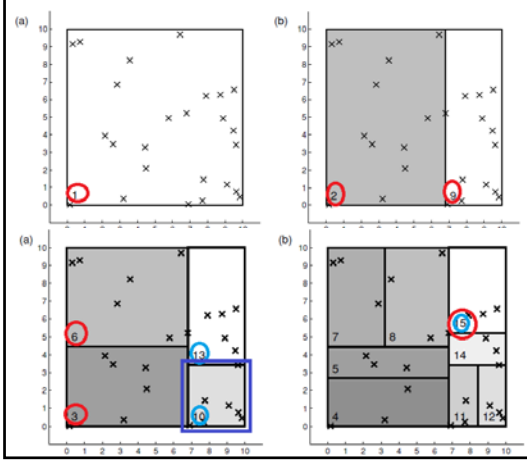
若$|\boldsymbol{RightD}|>\mathbf{m}_0$,对于深度为$\mathbf{j=1}$的右子结点，基于***RightD***以及***RightRange***，调用**CreateKDTree**，递归生成右子树.

例:基于训练样本A,B,C,D,E,F,G构建二维特征空间的kd树.



例：基于二维特征空间25个训练样本，构建kd树.





4.2 搜索kd树

例--**在给定的kd树内搜索x的最近邻**

输入： (1)已经构造的**kd树**

(2)任意观测样本$\boldsymbol{x}=[x^{(1)},\ldots,x^{(k)}]^T$

输出： $\boldsymbol{x}$的近邻

STEP1. 二叉树搜索,找到含样本x的叶结点

从根结点出发,沿树自上向下，
将样本x的**切分特征**$split$取值与当前结点切分点s比较

$$\begin{cases} \text{若} x[split] \le s, \text{则该结点的左子结点胜出} \\ \text{若} x[split] > s, \text{则该结点的右子结点胜出} \end{cases}$$

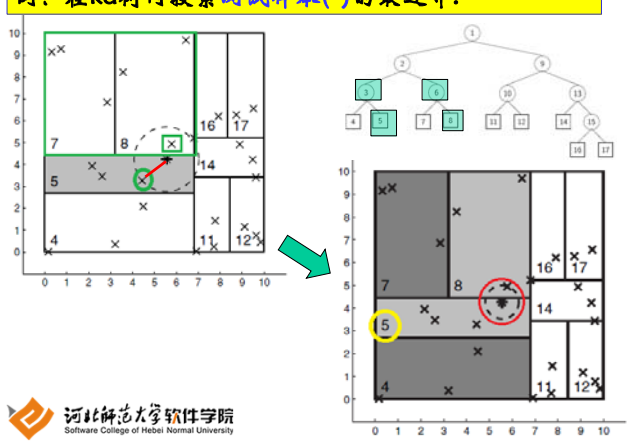
逐级进入生成结点，直到找到kd树中含样本x的叶结点

STEP2. 在该叶结点对应的训练样本子集内搜索，找到初始近邻，将其初始化为样本x的"**当前最近邻**x_0"
得到二者的距离$d_0 = d(x, x_0)$

STEP3. 从该叶结点开始，递归向上回溯.
对每个结点重复操作如下：
(1)记**当前结点**的父结点为q
(2)以样本x为中心，以当前最近距离d_0为半径，构建**超球**.
则x的**真正最近邻**x_0^*一定在该超球内.
(3)检查**超球**是否与**当前结点**的**兄弟节点**对应的**超矩形体**相交.
若是相交：
该**兄弟节点**所对应超矩形区域内， 可能存在关于x更近的样本. 移动至该**兄弟节点**，递归进行近邻搜索，若存在更近的近邻，则更新近邻及超球.
若不相交：
则向上回退,更新当前结点、父结点

STEP4. 若回退至根结点，则搜索结束，输出最终的近邻x_0^*

例：在kd树内搜索测试样本(*)的最近邻.



PART1. K-近邻分类
PART2. K-近邻回归

PART2. K近邻回归

K-Nearest Neighbors Regression, KNN回归

主要内容

1. 回归问题的一般描述
2. K近邻回归算法的描述
3. K近邻回归的预测规则
4. K近邻回归模型的评价(思考)

高维空间K近邻回归模型的性能远不如低维空间

1. 回归问题的一般描述

给定训练样本集$\{(\boldsymbol{x}_i, y_i), i=1,\cdots,N\}$.

其中：

$\boldsymbol{x}_i$----第i个观测样本的输入　　$\boldsymbol{x}_i \in R^d$

y_i----第i个观测样本的输出　　$y_i \in R$

要求：

基于上述样本集，构建预测模型$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$--**模型的回归学习**；

对特征空间的任意观测$\boldsymbol{x}$的输出y进行预测--**模型的使用**

主要内容

1. 回归问题的一般描述
2. K近邻回归算法的描述
3. K近邻回归的预测规则

2.1 K近邻回归模型的引入

非参数法回归.

无需明确预测模型的参数化形式，直接由训练样本数据预测任意观测$\boldsymbol{x}$所导致的输出$\hat{\boldsymbol{f}}(\boldsymbol{x})$.

懒惰学习

准备工作轻松

训练样本集、近邻数**K**、欧式距离

2.2 K近邻回归算法的描述

输入:(1)**训练样本集**$\boldsymbol{D}_{\text{train}} = \{(\boldsymbol{x}_i, y_i), i=1,\cdots,N\}$，并且有：

$\boldsymbol{x}_i \in R^d$----第i个训练样本的特征向量

$y_i \in R$----第i个训练样本的目标输出

(2)**观测样本**$\boldsymbol{x}$

输出: 观测样本$\boldsymbol{x}$应导致的输出y.

STEP0. 输入特征的预处理

STEP1. **选择K**

STEP2. 在训练集$\boldsymbol{D}_{\text{train}}$内找到样本$\boldsymbol{x}$的**K个近邻，记为**$N_K(\boldsymbol{x})$

STEP3. 结合指定的**预测规则**，对$\boldsymbol{x}$应导致输出y进行预测

$$\hat{y} = \frac{1}{K}\sum_{x_i \in N_K(\boldsymbol{x})} y_i$$

训练集固定的前提下，
不同的距离度量方式、不同的K值、
不同的预测规则，

会导致不同的预测结果

主要内容

1. 回归问题的一般描述
2. K近邻回归算法的描述
3. K近邻回归的预测规则

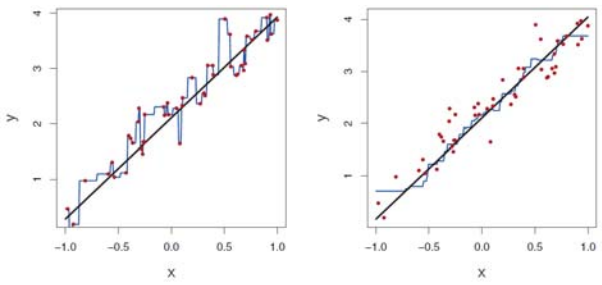
方式1. 等权平均 – –传统的K近邻决策方式

对于给定观测$\boldsymbol{x}$，利用距离度量，在训练样本集$\boldsymbol{D}_{train}=\{(x_i,y_i),i=1,\cdots,N\}$内确定$\boldsymbol{x}$的前**K个近邻**$N_K(x)$

则$\boldsymbol{x}$应的输出$\boldsymbol{y}$预测为

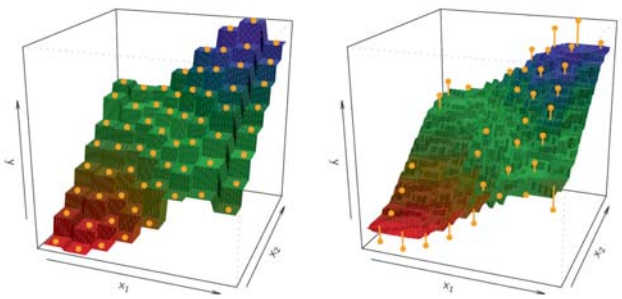
$$\hat{y}=\frac{1}{K}\sum_{x_i\in N_K(x)}y_i$$

例1. 一维特征空间的K近邻回归。
100个训练样本(左 K=1， 右K=9)



The blue curve corresponding to K = 9 represents a smoother fit.

例：二维特征空间的基于KNN回归，基于64个训练样本实现(orange dots).
Left: 近邻数K = 1，导致较为粗糙的函数拟合结果
Right: 近邻数K = 9，导致更为平滑的函数拟合结果



方式2. 基于距离的加权平均

对于给定的观测$\boldsymbol{x}$，利用距离度量，在训练样本集$\boldsymbol{D}=\{(x_i,y_i),i=1,\cdots,N\}$内确定$\boldsymbol{x}$的前**K个近邻**$N_K(x)$

对于$\forall x_j\in N_K(x)$，该训练样本关于$\boldsymbol{x}$的距离为$d(x,x_j)$
引入基于距离的预测权值，如：

$$w(d)=\frac{1}{d^2}\quad 或\quad w(d)=\exp\left[-d^2\right]$$

则$\boldsymbol{x}$的输出$\boldsymbol{y}$预测为

$$\hat{y}=\frac{\sum_{x_i\in N_K(x)}w\left(d(x,x_j)\right)y_i}{\sum_{x_i\in N_K(x)}w\left(d(x,x_j)\right)}$$

主要内容

1. 回归问题的一般描述
2. K近邻回归算法的描述
3. K近邻回归的预测规则

思考：如何采用m-fold CV法，对K近邻回归模型进行近邻数K值的优选？

关于KD树的说明(参见Scikit-learn)

Regarding the Nearest Neighbors algorithms, if two neighbors, have identical distances but different labels, the results will depend on the ordering of the training data.

Though the KD tree approach is very fast for low-dimensional (d<20) neighbors searches, it becomes inefficient as grows very large: this is one manifestation of the so-called "curse of dimensionality".

For small data sets (less than 30 or so), brute force algorithms can be more efficient than a tree-based approach.

K近邻小结

- 分类问题的描述、分类模型的性能评价
- 回归问题的描述、回归模型的性能评价
- 距离度量
- 样本的规范化预处理

1. K近邻分类
   - K近邻分类的算法描述
   - K近邻分类系统决策性能的影响因素
   - 如何面向分类，进行K值优选
   - 决策规则
2. K近邻回归
   - K近邻回归的算法描述
   - K近邻预测系统性能的影响因素
   - 如何面向回归问题，进行K值优选
   - 预测规则