第七章 非参数学习机器与集成学习

苏智勇
可视计算研究组
南京理工大学
suzhiyong@njust.edu.cn
https://zhiyongsu.github.io

主要内容

7.1 引言

7.2 近邻法

7.3 决策树与随机森林

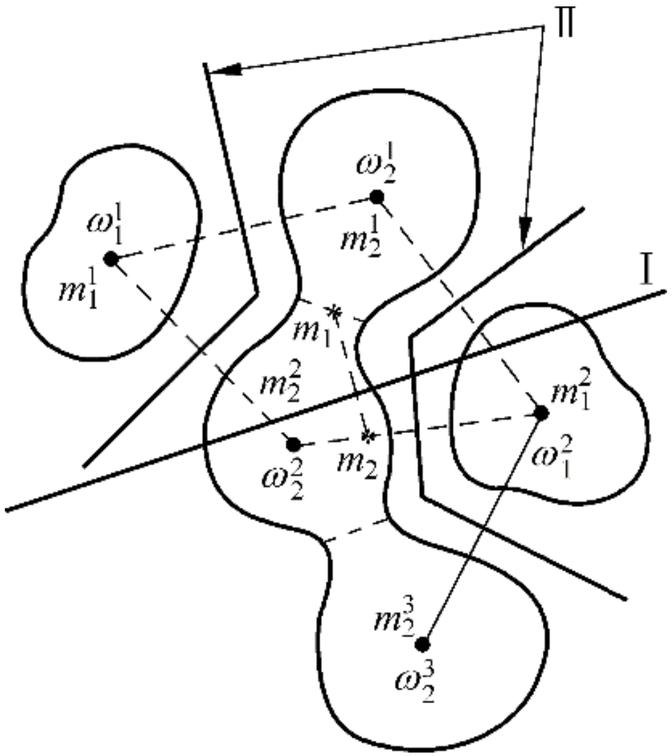
7.4 Boosting集成学习

7.1 引言

- **参数学习机器**
  - 先确定学习机器实现的函数集，然后选择函数集中的函数
- **非参数学习机器**
  - 通过对训练样本的学习直接构建分类机器
  - 无法用一个包含若干待定参数的函数来表示

7.2 近邻法

- 回顾
  - 最简单的分段线性分类器
    - 把各类划分为若干子类，以子类中心作为代表点，考查新样本到各代表点的距离并将它分到最近的代表点所代表的类
    - 极端情况，所有样本都作为代表点，**无需构造分类面**
  - 近邻法（Nearest-Neighbor method）



7.2.1 最近邻法

- 样本集$S_N = \{(\boldsymbol{x}_1, \theta_1), (\boldsymbol{x}_2, \theta_2), \cdots, (\boldsymbol{x}_N, \theta_N)\}$，$\boldsymbol{x}_i$为样本$i$的特征向量，$\theta_i$为样本$i$的类别标号，$\theta_i = \{1,2,\cdots,c\}$
  - **定义1**：样本$\boldsymbol{x}_i$与$\boldsymbol{x}_j$之间的距离$\delta(\boldsymbol{x}_i, \boldsymbol{x}_j)$：比如欧氏距离$\|\boldsymbol{x}_i - \boldsymbol{x}_j\|$。对未知样本$\boldsymbol{x}$，求$S_N$中与之距离最近的样本$\boldsymbol{x}'$（类别为$\theta'$）

$$\delta(\boldsymbol{x}, \boldsymbol{x}') = \min_{j=1,\ldots,N} \delta\left(\boldsymbol{x}, \boldsymbol{x}_j\right)$$

则将$\boldsymbol{x}$分到θ'类，即$\hat{\omega}(x) = \theta'$（或记作$\hat{\omega}_1(x)$）

7.2.1 最近邻法

- 样本集$S_N = \{(\boldsymbol{x}_1, \theta_1), (\boldsymbol{x}_2, \theta_2), \cdots, (\boldsymbol{x}_N, \theta_N)\}$，$\boldsymbol{x}_i$为样本$i$的特征向量，$\theta_i$为样本$i$的类别标号，$\theta_i = \{1,2,\cdots,c\}$

  - **定义2**：ω_i类别判别函数：

$$g_i(\boldsymbol{x}) = \min_{\boldsymbol{x}_j \in \omega_i} \delta(\boldsymbol{x}_i, \boldsymbol{x}_j), \quad i = 1,\cdots,c$$

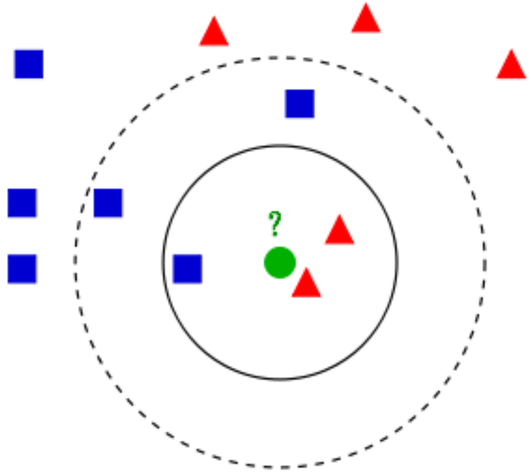
决策规则：

$$if \quad g_j(\boldsymbol{x}) = \min_{i=1,\ldots,c} g_i(\boldsymbol{x}), \quad \text{then} \quad \boldsymbol{x} \in \omega_j$$

简言之：以距离新样本最近的已知样本的类别作为新样本类别

7.2.2 k-近邻法

- 最近邻法（1-近邻法）的推广
  - 找出$\boldsymbol{x}$的k个近邻，看其中多数属于哪一类，则把$\boldsymbol{x}$分到哪一类
- **$k-$近邻的一般表示**
  - N个已知样本，分属c个类别ω_i，$i=1,\cdots,c$
  - k_i，$i=1,\cdots,c$，为$\boldsymbol{x}$的k个近邻中属于ω_i的样本数
  - 判别函数：$g_i(\boldsymbol{x})=k_i$，$i=1,\cdots,c$
  - 决策规则：$if \quad g_j(\boldsymbol{x})=\max\limits_{i=1,\ldots,c} k_i, \quad \text{then} \quad \boldsymbol{x}\in\omega_j$



7.2.2 k-近邻法

- **问题**
  - k的取值、距离的度量方式
  - 计算和存储成本很大：需要和每个已知样本比较和排序
  - 票数接近时风险较大，有噪声时风险加大
  - 有限样本下性能如何
- **改进**
  - 减少计算量和存储量
  - 引入拒绝机制
  - 根据实际问题修正投票方式：如根据距离远近进行加权

7.2.3 近邻法的快速算法

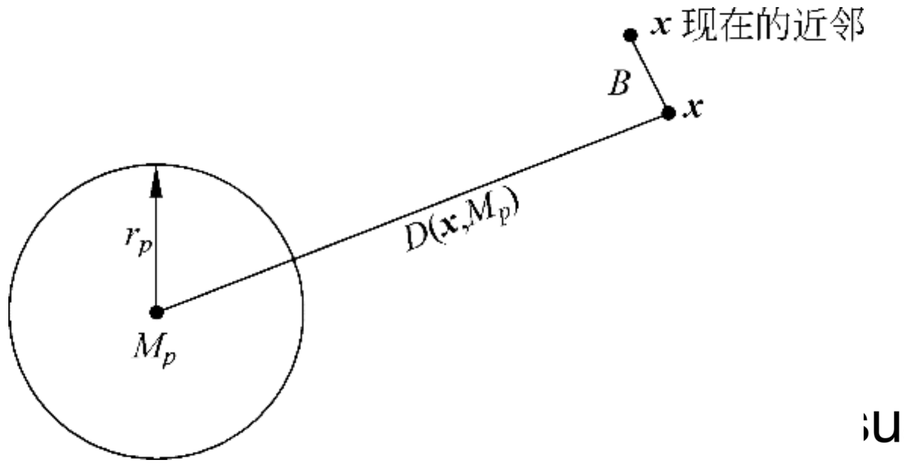
- **近邻法在计算上的问题**
  - 需存储所有训练样本
  - 新样本需与每个样本作比较
- **快速算法基本思想**
  - 把样本集分级分成多个子集（树状结构）
  - 每个子集（结点）可用较少几个量代表
  - 通过将新样本与各结点比较排除大量候选样本
  - 只有最后的结点（子集）中逐个样本比较，找出近邻

7.2.3 近邻法的快速算法

- **分枝定界算法**
  - 符号约定
    - $\mathcal{X}_p$：节点p对应的样本子集
    - N_p：$\mathcal{X}_p$中的样本数
    - M_p：子集$\mathcal{X}_p$中的样本均值（中心点）
    - $r_p = \max\limits_{\boldsymbol{x}_i \in \mathcal{X}_p} D(\boldsymbol{x}_i, M_p)$：$\mathcal{X}_p$中离中心点最远的距离
    - B：当前搜索到的最近邻距离

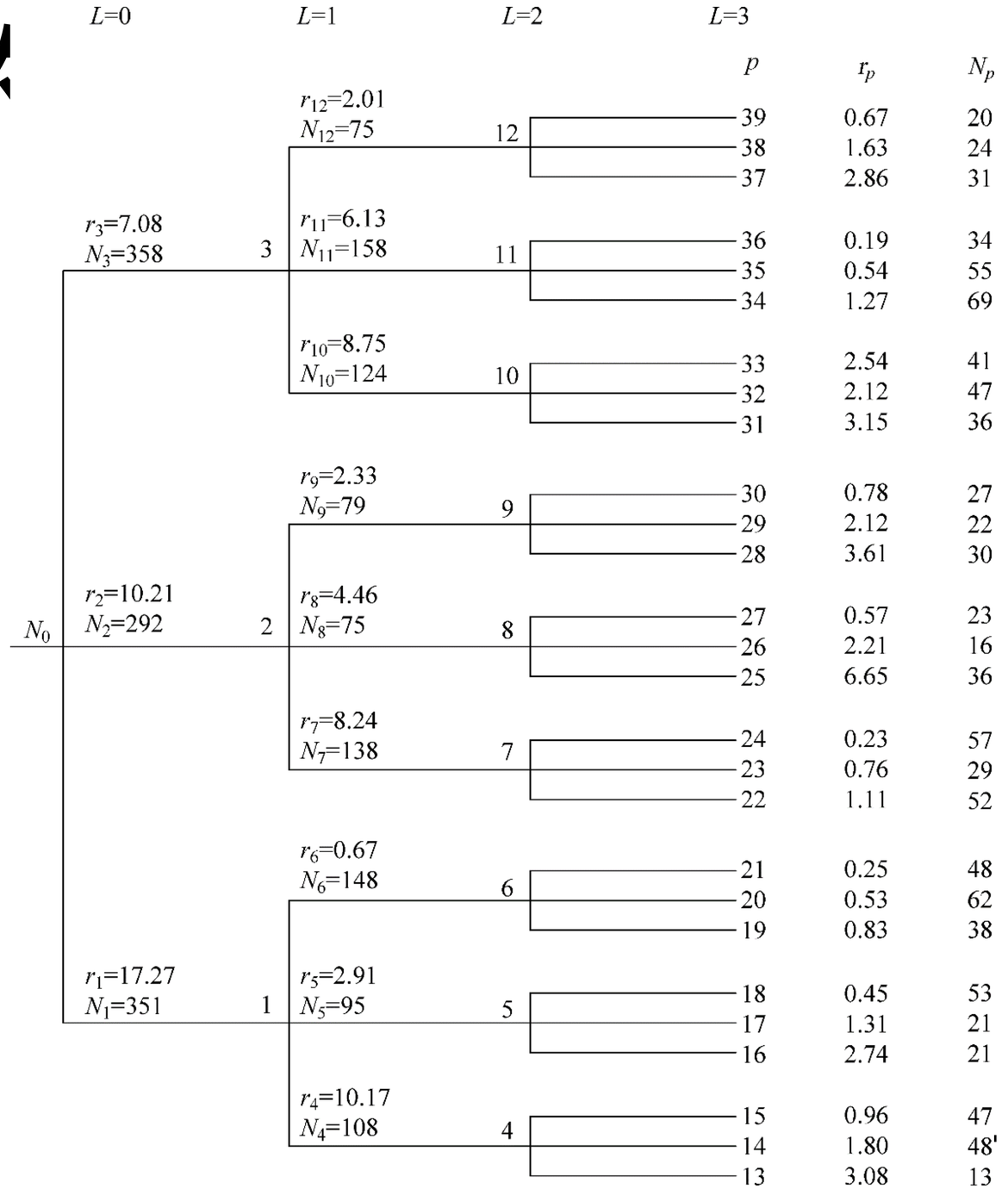
7.2.3 近邻法的快速算法

- **分枝定界算法**
  - 两条规则
    - 对新样本$\boldsymbol{x}$，节点p，如果存在$D(\boldsymbol{x}, M_p) > B + r_p$，则$\boldsymbol{x}_i \in \mathcal{X}_p$不可能是$x$的最近邻
    - 如果$D\left(\boldsymbol{x}, M_p\right) > B + D(\boldsymbol{x}_i, M_p)$，则$x_i \in \mathcal{X}_p$不可能是$\boldsymbol{x}$的最近邻



7.2.3 近邻法的

- **分枝定界算法**
  - 两大步
    - 样本集的分级分解，计算并存储$\mathcal{X}_p$的$M_p, r_p, D(\boldsymbol{x}_i, M_p)$
    - 用分枝定界算法搜索x的最近邻



7.2.3 近邻法的

- **树搜索算法（最近邻）**

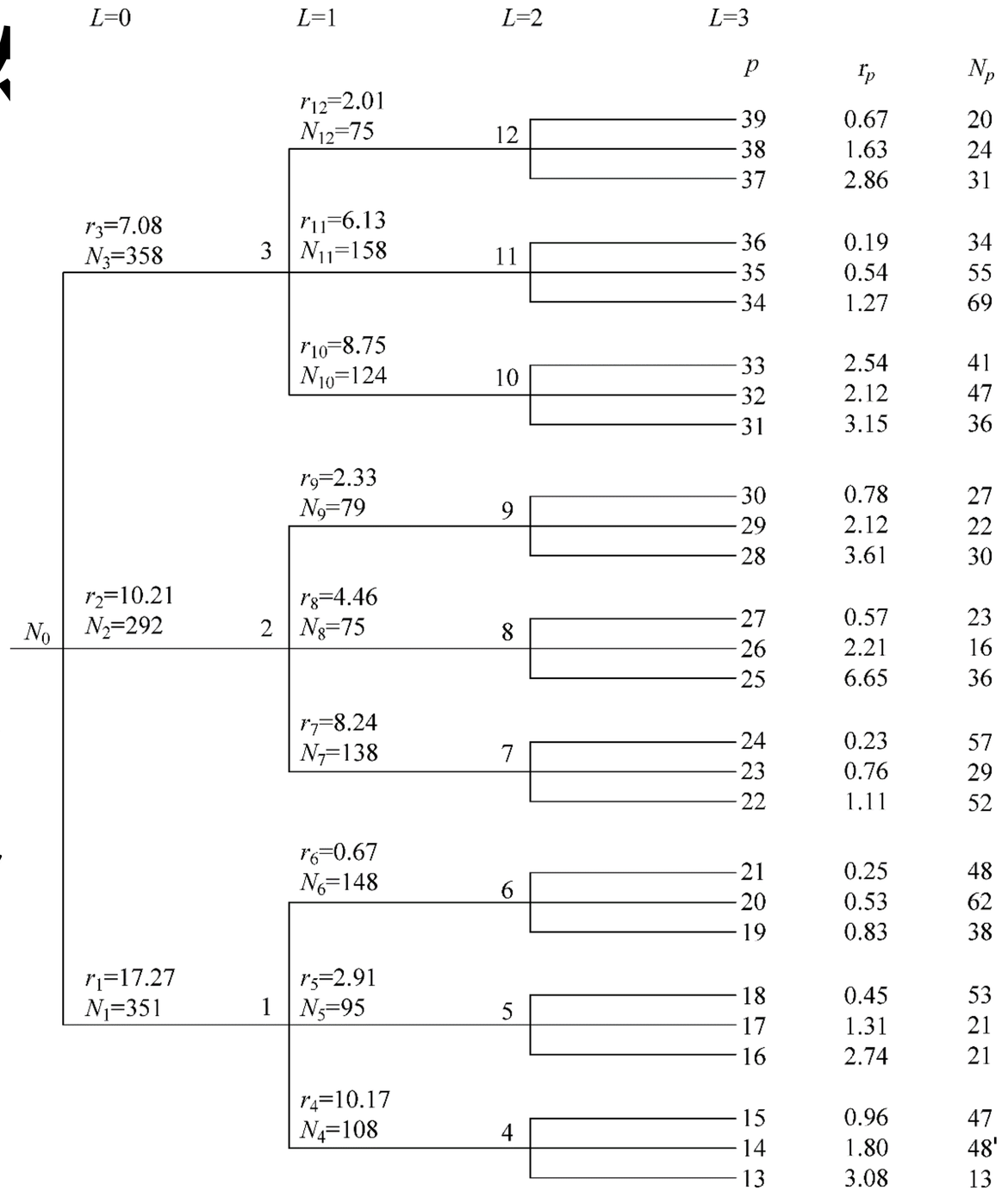
(1)置$B=\infty$，$L=1$，$p=0$（L是当前水平，p是当前节点）

(2)将当前节点所有**直接后继节点**放入目录表中，并对这些节点计算$D(\boldsymbol{x}, M_p)$

(3)对步骤(2)中每个节点p，根据规则1，如果有$D(\boldsymbol{x}, M_p) > B + r_p$，则从目录表中去掉

(4)如果步骤(3)目录表中已没有节点，则后退到前一水平，即置$L=L-1$。如果$L=0$则停止，否则转步骤(3)。如目录表中有节点存在，则转步骤(5)

(5)在目录表中选择最近节点p'，它使$D(\boldsymbol{x}, M_p)$最小化，并称该p'为当前执行节点，从目录表中去掉p'。如果当前水平L是最终水平，则转步骤(6)。否则置$L=L+1$，转步骤（2）



7.2.3 近邻法的快速算法

- **树搜索算法（最近邻）**

(6)对现在执行节点p'中的每个$\boldsymbol{x}_i$，利用规则(2)作如下检验。如果

$$D(\boldsymbol{x}, M_p) > D(\boldsymbol{x}_i, M_p) + B$$

则$\boldsymbol{x}_i$不是$\boldsymbol{x}$的最近邻。否则计算$D(\boldsymbol{x}, \boldsymbol{x}_i)$。若

$$D(\boldsymbol{x}, \boldsymbol{x}_i) < B$$

置$NN = i$和$B = D(\boldsymbol{x}, \boldsymbol{x}_i)$。

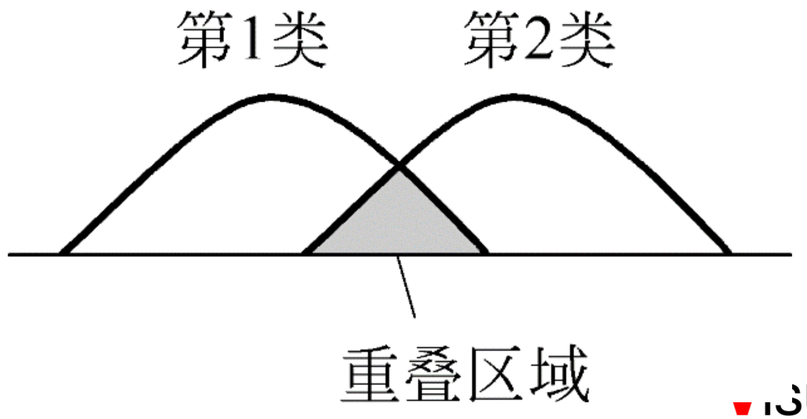
在当前执行节点中所有$\boldsymbol{x}_i$被检验之后，转步骤(3)。

当算法结束时，输出$\boldsymbol{x}$的最近邻$\boldsymbol{x}_{NN}$和x的距离$D(\boldsymbol{x}, \boldsymbol{x}_{NN}) = B$。

k-近邻只需修改步骤（6）

7.2.4 剪辑近邻法

- **动机**
  - 处在两类交界处或分布重合区的样本可能误导近邻法决策，应将它们从样本集中去掉
- **基本思路**
  - 考查样本是否为可能误导样本，若是则从样本集中去掉——剪辑
  - 考查方法是通过试分类，认为错分样本为误导样本



7.2.4 剪辑近邻法

- **基本做法**
  - 将已知样本集划分为测试集$\mathcal{X}^{NT}$和训练集$\mathcal{X}^{NR}$:

$$\mathcal{X}^{N} = \mathcal{X}^{NT} \bigcup \mathcal{X}^{NR}, \mathcal{X}^{NT} \cap \mathcal{X}^{NR} = \phi$$

  - **剪辑**：用训练集$\mathcal{X}^{NR}$中的样本对测试集$\mathcal{X}^{NT}$中的样本进行近邻法分类。剪掉$\mathcal{X}^{NT}$中被错分的样本，$\mathcal{X}^{NT}$中剩余样本构成剪辑样本集$\mathcal{X}^{NTE}$
  - **分类**：利用$\mathcal{X}^{NTE}$和近邻法对未知样本$\boldsymbol{x}$分类。

7.2.4 剪辑近邻法

- **多重剪辑方法MULTIEDIT**

(1)（划分）把样本集随机划分为s个子集，$\mathcal{X}_1, \ldots, \mathcal{X}_1, \ldots, \mathcal{X}_s, s \geq 3$

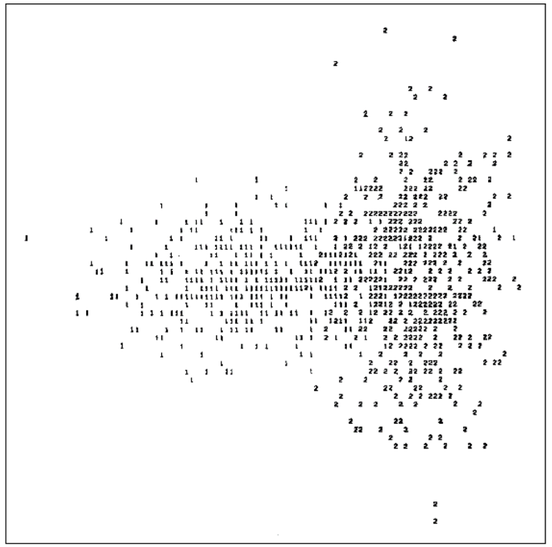
(2)（分类）用$\mathcal{X}_{(i+1)mod(s)}$对$\mathcal{X}_i$中的样本分类，$i = 1,..., s$

(3)（剪辑）去掉(2)中错分的样本

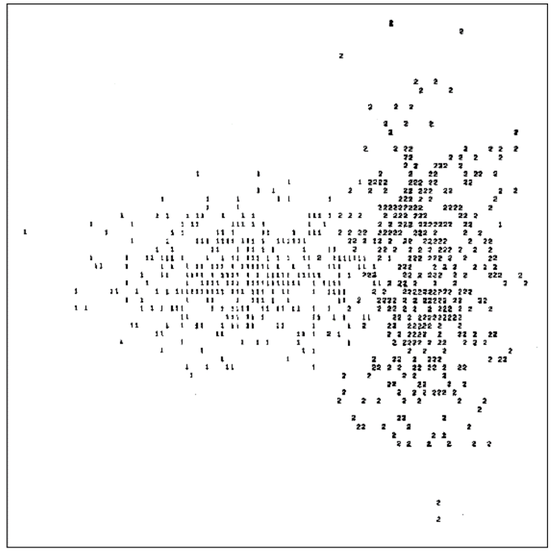
(4)（混合）将剩下的样本合在一起，形成新的样本集$\mathcal{X}^{NE}$

(5)（迭代）用新的样本集$\mathcal{X}^{NE}$替代原样本集，转步骤1。如果最近m次迭代都没有样本被剪掉，则停止，用最后的$\mathcal{X}^{NE}$作为分类的样本集

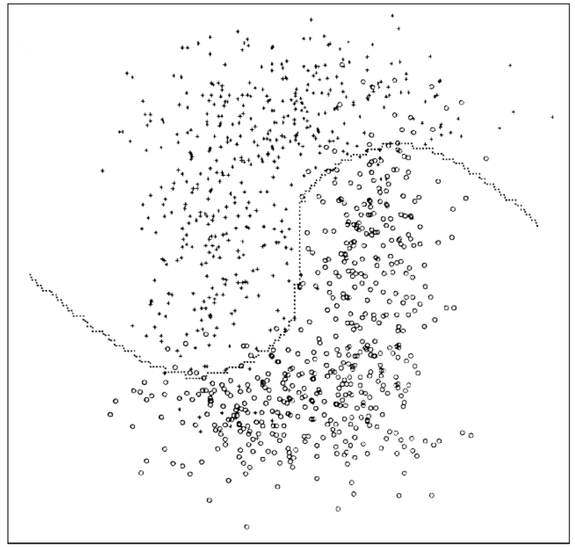
7.2.4 剪辑近邻法



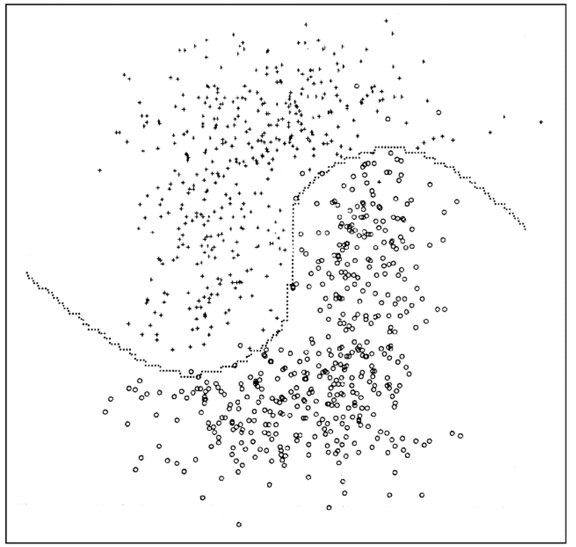
(a) 原始样本集



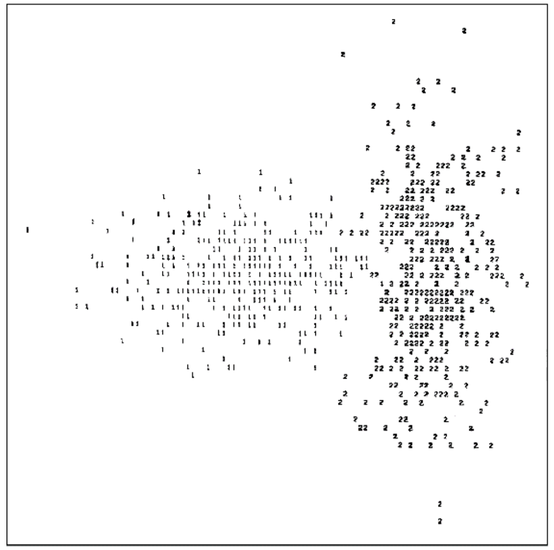
(b) 经过一次剪辑的样本集



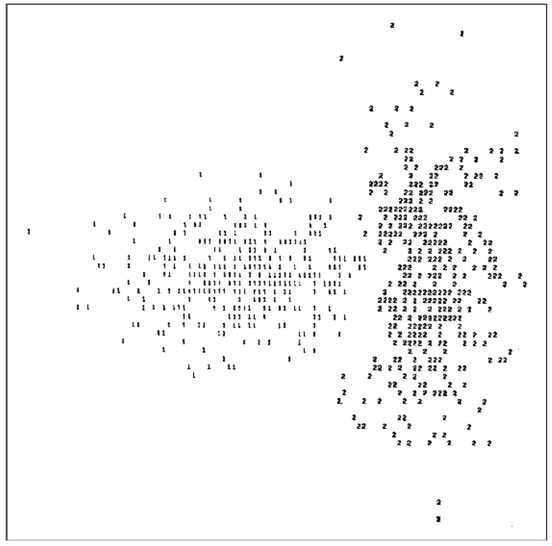
(a) 原始数据和贝叶斯分类面



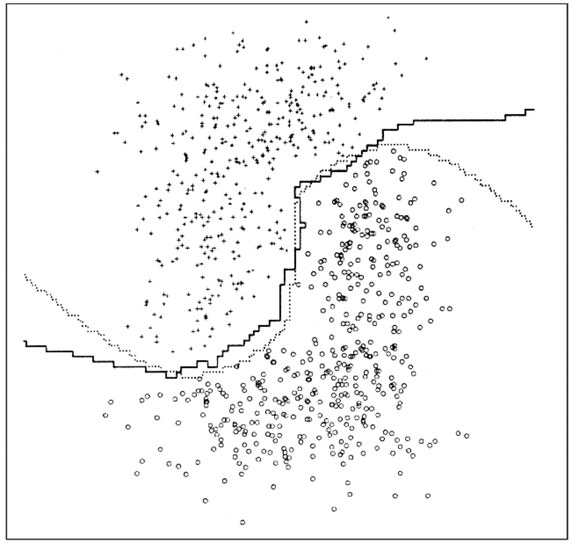
(b) 经过一次剪辑的数据与贝叶斯分类面



(c) 经过三次剪辑的样本集



(d) 算法终止时的样本集



(c) 最终的剪辑结果、多重剪辑近邻法的分类面与贝叶斯分类面

7.2.5 压缩近邻法

- **CONDENSE算法**

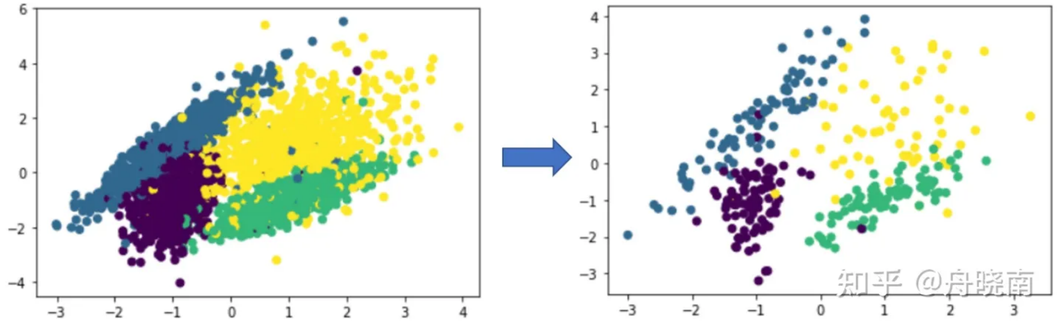
  -**动机**：同一类型的样本大量集中在类簇的中心，而这些集中在中心的样本对分类没有起到太大的作用，因此可以舍弃掉这些样本。

  -将样本集$\mathcal{X}^N$分为$\mathcal{X}_S, \mathcal{X}_G$两个子集，开始时$\mathcal{X}_S$中只有一个样本，其余均在$\mathcal{X}_G$中

  -考查$\mathcal{X}_G$中的每一个样本x，若用$\mathcal{X}_S$中的样本能够对它正确分类，则该样本保留在$\mathcal{X}_G$，否则移到$\mathcal{X}_S$中

  -依次类推，直到没有样本再需要搬移为止。最后用$\mathcal{X}_S$中的样本作为代表样本，对未来样本进行近邻法分类

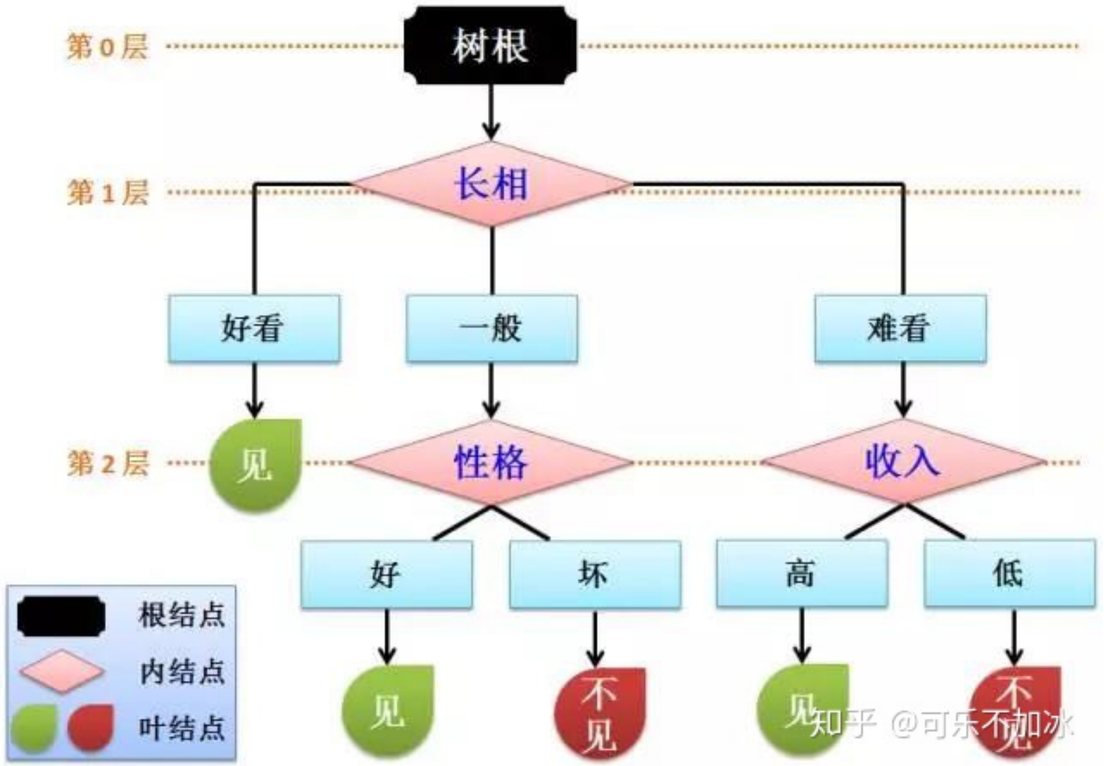
- 可与剪辑法配合使用：去噪

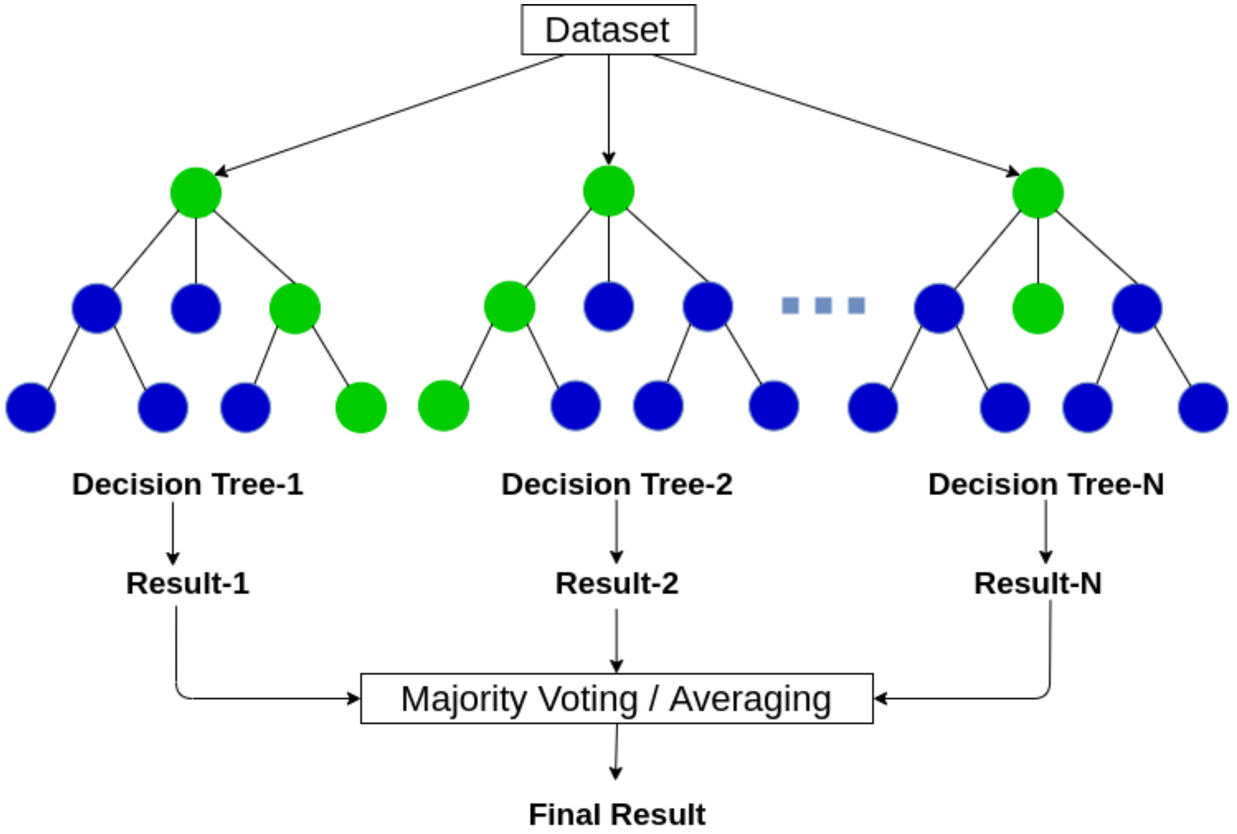


7.2.5 压缩近邻法

- **总结**
  - 近邻法快速算法：样本数不变
  - 剪辑近邻法：去掉两类边界附近的样本
  - 压缩近邻法：去掉类中心附近的样本

7.3 决策树与随机森林


第0层
树根
第1层
长相
好看
一般
难看
第2层
见
性格
收入
好
坏
高
低
见
不见
见
不见
根结点
内结点
叶结点
知乎 @可乐不加冰


Dataset
Decision Tree-1
Decision Tree-2
Decision Tree-N
Result-1
Result-2
Result-N
Majority Voting / Averaging
Final Result

7.3.1 非数值特征的量化

- **名义特征**：正交编码

  -例如颜色、形状、性别、职业、字符串中的字符等

- **序数特征**：等同于名义特征处理或转化为数值特征

  -例如序号、分级，不能看作是欧氏空间中的数值

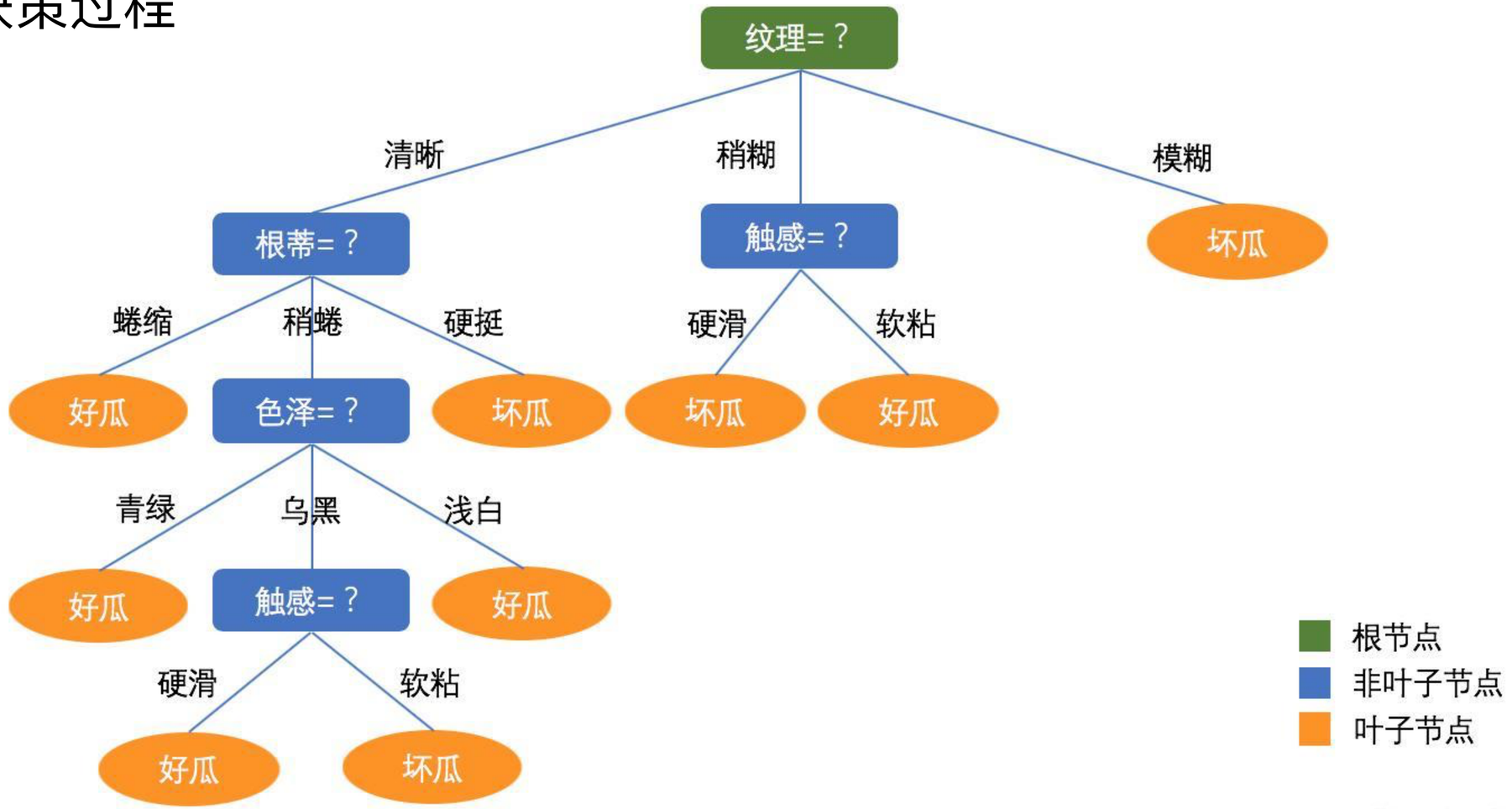
- **区间特征**：通过设定阈值变成二值特征或序数特征

  -与研究目标之间的关系呈现出明显的非线性。取值是实数，可以比较大小，但是没有一个“自然的”零，比值没有意义

  -例如年龄、温度、考试成绩等

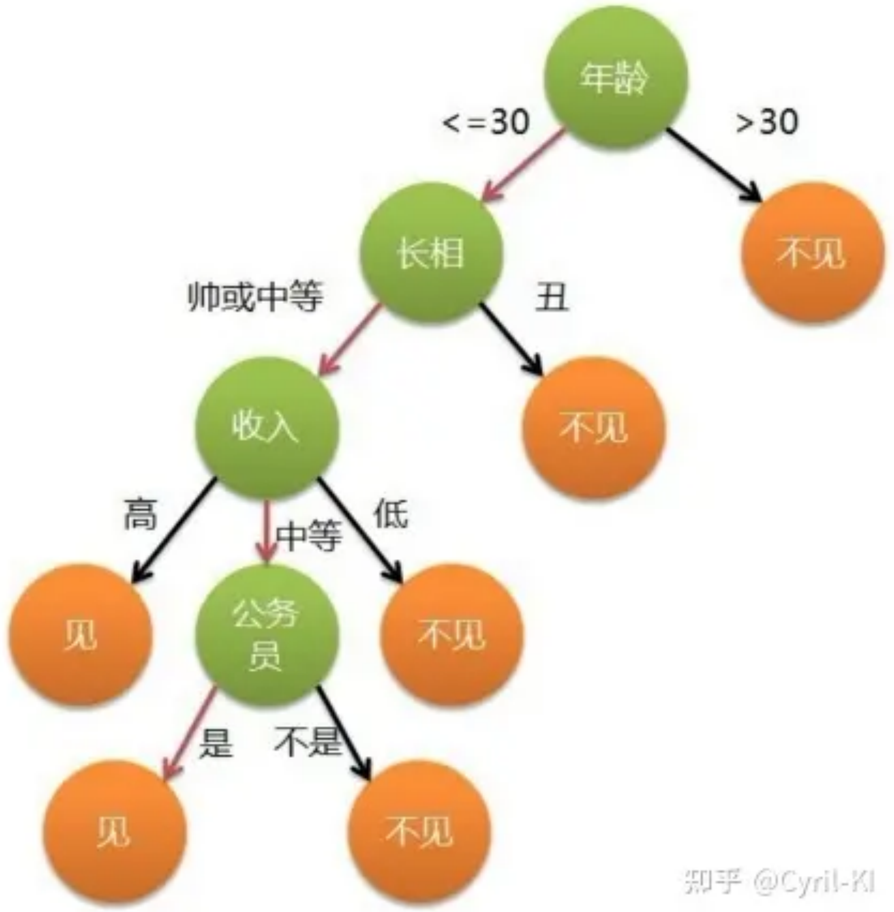
7.3.2 决策树

- 树状决策过程



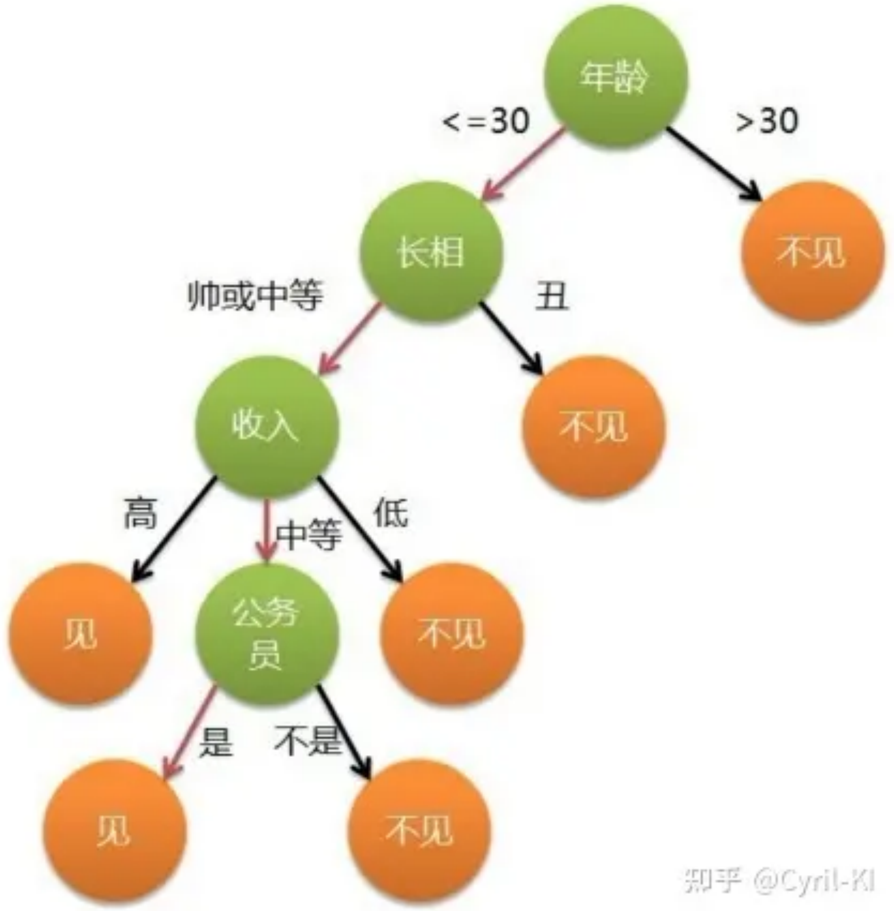
7.3.2 决策树

- **决策树**
  - **组成**
    - **节点**：一个特征和相应的决策规则
    - **根**：最顶端的节点
    - **叶节点**：只包含单纯一类的样本，不需要继续划分



7.3.2 决策树

- 决策树
  - 构造
    - 特征选择
    - 决策树的生成
    - 决策树的修剪



7.3.2 决策树

- **ID3（interactive dichotomizer-3）方法：交互式二分法**
  - **基本思想**：通过选择有辨别力的特征对数据进行分类，直到每个叶节点上只包含单一类型的数据为止
  - 香农熵（Shannon Entropy）：设离散型随机变量X的取值有$x_1, x_2, \cdots, x_k$，其发生概率分别为$p_1, p_2, \cdots, p_k$，其信息熵为：

$$I = -\left(P_1 \log_2 P_1 + P_2 \log_2 P_2 + \ldots + P_k \log_2 P_k\right) = -\sum_{i=1}^{k} P_i \log_2 P_i$$

信息熵用以描述信源的**不确定度（不纯度）**。对于决策而言，不确定度越小越好。

7.3.2 决策树

- **ID3（interactive dichotomizer-3）方法：交互式二分法**
  - 汽车顾客的例子，在不考虑任何特征时：

$$I(16,4) = -\left(\frac{4}{16}\log_2\frac{4}{16} + \frac{12}{16}\log_2\frac{12}{16}\right) = 0.8113$$

  - 如采用年龄作为根节点，则把所有样本分为两组：

$$I_{age} = \frac{6}{16}I(6,1) + \frac{10}{16}I(10,3) = 0.7946$$

  - 不纯度减少量（信息增益Information Gain）：

$$\Delta I_{age}(16) = I(16,4) - I_{age} = 0.0167$$

表 6-1　顾客数据

顾客编号	年　龄	性　别	月 收 入	是否购买
1	21	男	4000	否
2	33	女	5000	否
3	30	女	3800	否
4	38	女	2000	否
5	25	男	7000	否
6	32	女	2500	否
7	20	女	2000	否
8	26	女	9000	是
9	32	男	5000	是
10	24	男	7000	否
11	40	女	4800	否
12	28	男	2800	否
13	35	女	4500	否
14	33	男	2800	是
15	37	男	4000	是
16	31	女	2500	否

表 6-2　经过初步整理后的顾客数据

顾客编号	年　龄	性　别	月 收 入	是否购买
1	<30	男	中	否
2	≥30	女	中	否
3	≥30	女	中	否
4	≥30	女	低	否
5	<30	男	高	否
6	≥30	女	低	否
7	<30	女	低	否
8	<30	女	高	是
9	≥30	男	中	是
10	<30	男	高	否
11	≥30	女	中	否
12	<30	男	低	否
13	≥30	女	中	否
14	≥30	男	低	是
15	≥30	男	中	是
16	≥30	女	低	否

7.3.2 决策树

- **ID3（interactive dichotomizer-3）方法**
  - 一般地，不纯度减少量计算公式：

$$\Delta I(N) = I(N) - \left(P_1 I\left(N_1\right) + P_2 I\left(N_2\right) + \ldots + P_m I\left(N_m\right) \right), P_m = \frac{N_m}{N}$$

  - 上例中：

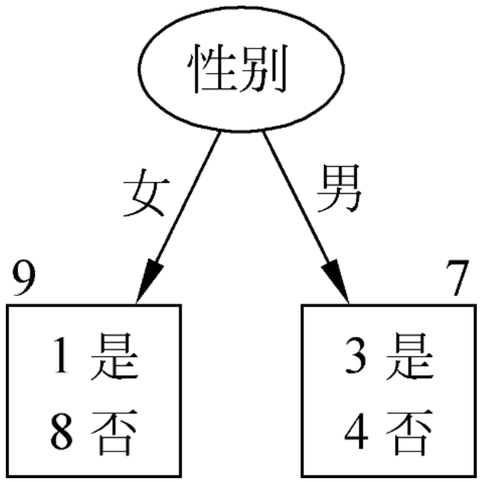
$$\Delta I_{gender}(16) = I(16,4) - I_{gender} = 0.0972$$

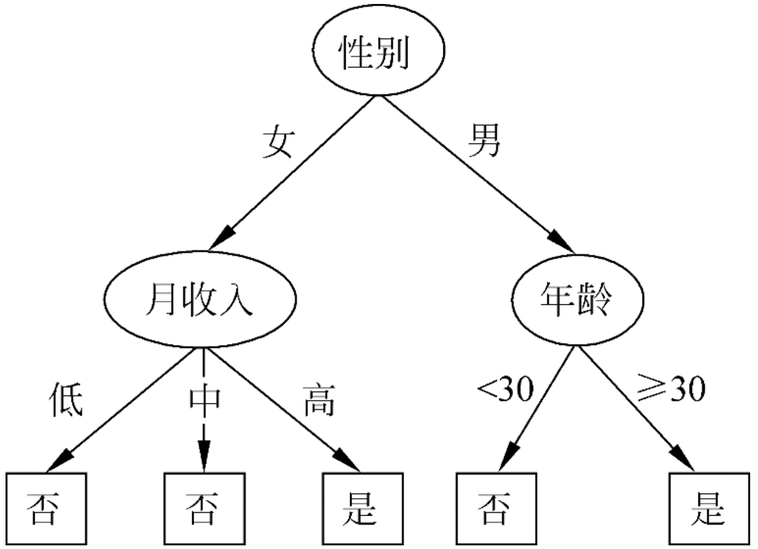
$$\Delta I_{income}\left(16\right) = I\left(16,4\right) - I_{income} = 0.0177$$

选取性别作第一个特征能够带来不纯度最大的减小。

7.3.2 决策树

- **ID3（interactive dichotomizer-3）方法**
  - 下一层节点构建
    - 分别考察月收入、年龄特征所得到的不纯度减少量，选择不纯度减少最大的特征作为决策树节点



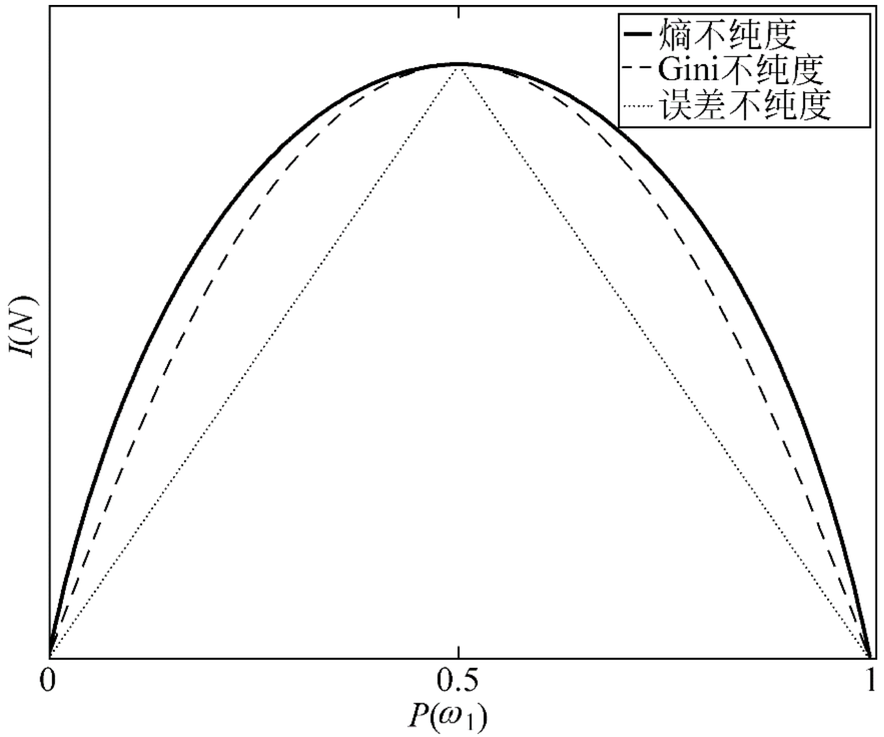


7.3.2 决策树

- **ID3（interactive dichotomizer-3）流程**
  - 计算当前节点包含的所有样本的信息熵（熵不纯度）；
  - 比较不同特征的信息增益，选择具有最大信息增益的特征赋予当前节点；该特征的取值个数决定了该节点下的分支数目；
  - 如果后继节点只包含一类样本，则停止该枝的生长；
  - 如果后继节点包含不同类样本，则重复以上步骤。

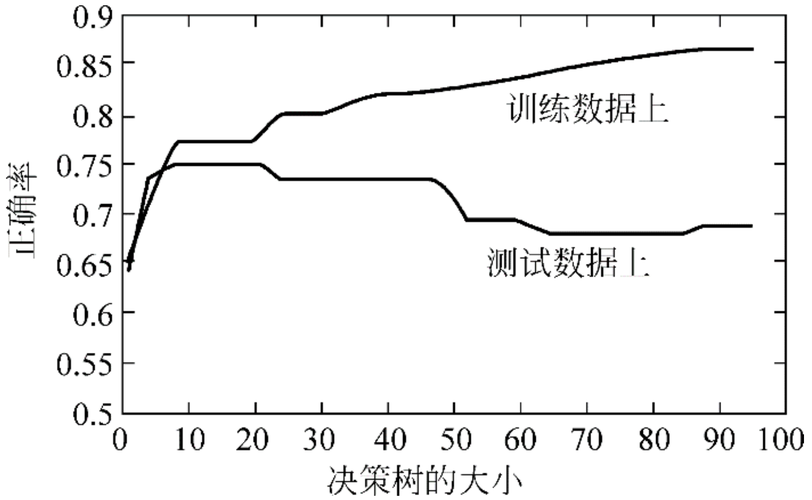
7.3.2 决策树

- **其他决策树算法**
  - 其他不纯度度量：
    - Gini不纯度（方差不纯度）：$I(N) = \sum_{m \neq n} P(\omega_m)P(\omega_n) = 1 - \sum_{j=1}^{k} P^2(\omega_j)$
    - 误差不纯度：$I(N) = 1 - \max_j P(\omega_j)$
  - C4.5算法：
    - 以信息增益率代替信息增益：$\Delta I_R(N) = \dfrac{\Delta I(N)}{I(N)}$
    - 能够处理连续数值特征
  - CART算法：
    - 每个节点上都采用二分法，最后构成二叉树



7.3.3 过学习与决策树的剪枝

- **决策树的过学习问题**
  - 算法在训练数据上表现好，在测试数据或新样本上表现较差
  - 为正确分类样本，不停地对节点划分，容易导致树分枝过多，造成决策树庞大，从而易出现过拟合。决策树越复杂，过拟合的程度会越高。



7.3.3 过学习与决策树的剪枝

- **剪枝**
  - **先剪枝：边构造边剪枝**

    在构造决策树的过程中，先对每个结点在划分前进行估计：如果当前结点的划分不能带来决策树模型泛化性能的提升，则不对当前结点进行划分并且将当前结点标记为叶结点。

    - **数据划分法**

      训练集+测试集；利用训练集对决策树生长，直至在测试集上的分类错误率达到最小时停止生长。

    - **阈值法**

      设定信息增益阈值，小于阈值时结点停止向下生长。

    - **信息增益的统计显著性分析**

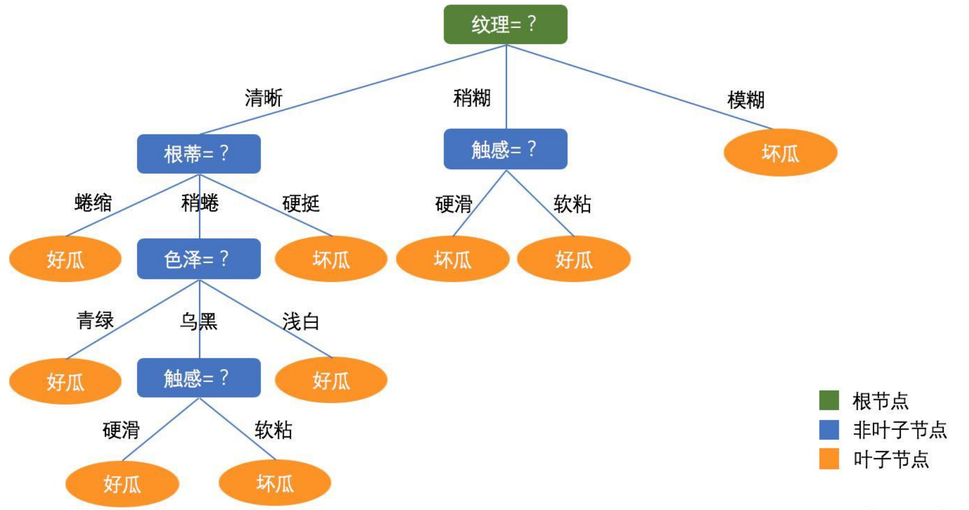
      统计已有结点的信息增益分布，如果继续生长得到的信息增益与该分布相比不显著（卡方检验），则停止生长。

7.3.3 过学习与决策树的剪枝

- **剪枝**
  - **后剪枝：构造完再剪枝**

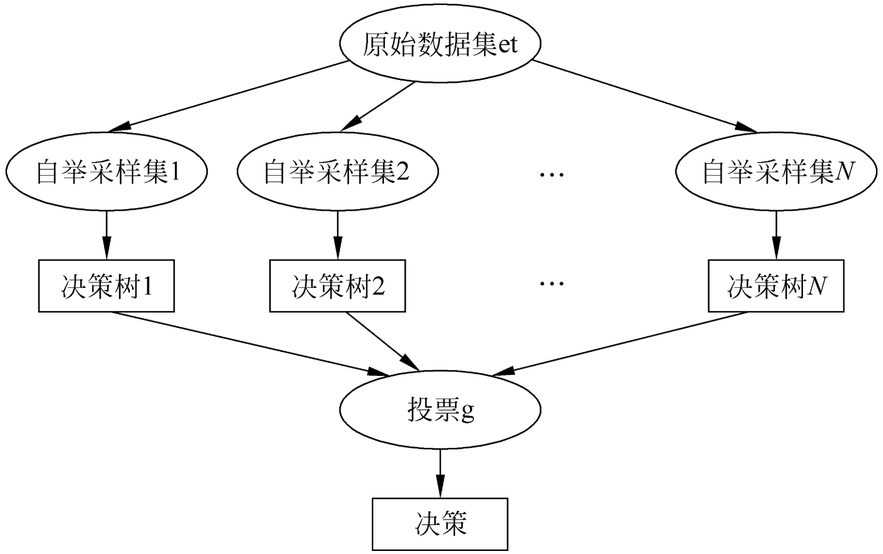
    先把整颗决策树构造完毕，然后自底向上的对非叶结点进行考察，若将该结点对应的子树换为叶结点能够带来泛化性能的提升（分支合并），则把该子树替换为叶结点。

    - 减少分类错误修剪法：错误率
    - 最小代价与复杂性的折衷：错误率、复杂度
    - 最小描述长度（MDL）准则：决策树编码



7.3.4 过学习与随机森林

- 对样本数据进行自举（bootstrap）**重采样**，得到多个样本集。所谓自举重采样，就是每次从原来的N个训练样本中有放回地随机抽取N个样本（包括可能的重复样本）：样本的随机性
- 用每个重采样样本集作为训练样本构造一个**决策树**。在构造决策树的过程中，每次从所有候选特征中随机地抽取m个特征，作为当前节点下决策的备选特征，从这些特征中选择最好地划分训练样本的特征
- 最后，对决策树的输出进行**投票**，以得票最多的类作为随机森林的决策



7.4 Boosting集成学习

- **AdaBoost算法**
  - 设给定N个训练样本$\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$，用$f_m(\boldsymbol{x}) \in \{-1,1\}(m = 1,...,M)$表示$M$个弱分类器在样本$\boldsymbol{x}$上的输出，算法过程如下：

    1. 初始化训练样本$\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$的权重，$w_i = \frac{1}{N}, i = 1,...,N$

    2. 对$m = 1 \to M$，重复以下过程

        ① 利用$\{w_i\}$加权后的训练样本构造分类器$f_m(\boldsymbol{x}) \in \{-1,1\}$

  - 注意：加权指的是指对分类器算法**目标函数**中各个样本所对应的项进行加权。例如，对于最小平方误差判别，加权后的最小平方误差（MSE）准则函数为：$\sum_{i=1}^{N} w_i \left(\boldsymbol{\alpha}^T \boldsymbol{x}_i - y_i\right)^2$

7.4 Boosting集成学习

- **AdaBoost算法**

② 计算样本用$\{w_i\}$加权后的分类错误率e_m，计算各分类器得分：

$$c_m = \log\left(\frac{1-e_m}{e_m}\right)$$

③ 令$w_i = w_i \exp\left[c_m 1_{(y_i \neq f_m(\boldsymbol{x}_i))}\right], i = 1,2,\cdots,N$, 并归一化使$\sum_{i=1}^{N} w_i = 1$

（$1_{(y_i \neq f_m(\boldsymbol{x}_i))}$表示当$y_i \neq f_m(\boldsymbol{x}_i)$时取1，否则取0）

3. 对于待分类样本$\boldsymbol{x}$，分类器的输出为$\operatorname{sgn}\left[\sum_{m=1}^{M} c_m f_m(\boldsymbol{x})\right]$