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

# 机器学习入门

## 决策树(分类树、回归树)

## Decision Trees

## (Classification Tree, Regression Tree)

河北师范大学软件学院

2017.03.29-04.08

本课件主要内容及有关例子，主要参考了

1. 周志华，《机器学习》
2. 李航，《统计学习方法》

特此感谢！

## 监督式学习

分类
- A. 懒惰学习--KNN法(K最近邻法)分类模型
- B. 概率学习 --贝叶斯分类器
- C. 分而治之--分类树

回归
- D. KNN法回归
- E. 分而治之--回归树
- F. 最小二乘回归

集成学习

## 非监督式学习

- 聚类 -- G. K-Means Clustering
- 特征提取 -- H. PCA

## 模型评价

# 主要内容

## PART1.决策树

基于树形结构的决策—决策树

包括：决策树构建方法；决策树的剪枝

1 非度量特征(*nonmetric features*)

2 决策树

3 过学习与决策树的剪枝

## PART2.(以决策树为个体模型的)集成学习

1 Bootstrap Aggregating(bagging)

2 Random Forest

## 样本的特征描述

(1) **度量型特征** (***metric features***)

(2) **非度量型特征** (***nonmetric features***)

如：　名义特征/标称数据 (***nominal features***)

　　序数特征 (***ordinal features***)

　　区间特征 (***interval features***)

## 非度量型特征描述的样本分类，处理方式：

**方式1**

$$\text{非度量型特征} \xrightarrow{\text{编码}} \text{度量型特征} \xrightarrow[\text{的样本分类}]{\text{基于度量型特征}} \text{分类结果}$$

编码可能会
- 造成信息损失
- 引入人为信息

**方式2**

基于非度量型特征的样本直接分类

**决策树**可直接面向非度量型、度量型特征描述的样本.

# 主要内容

## PART 1.决策树

1 非度量特征(*nonmetric features*)

2 决策树

- 2.1 决策树的引入
- 2.2 树的划分选择
- 2.3 决策树的构建(模型的学习)

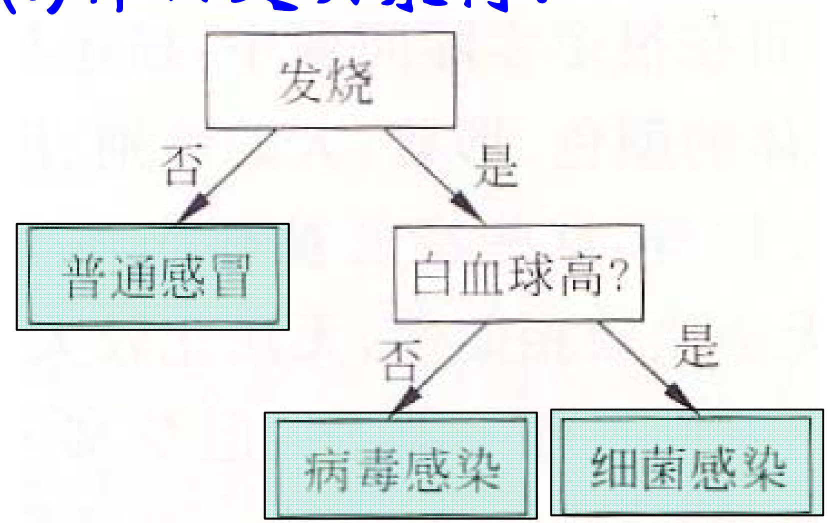
三个著名的决策树构建方法

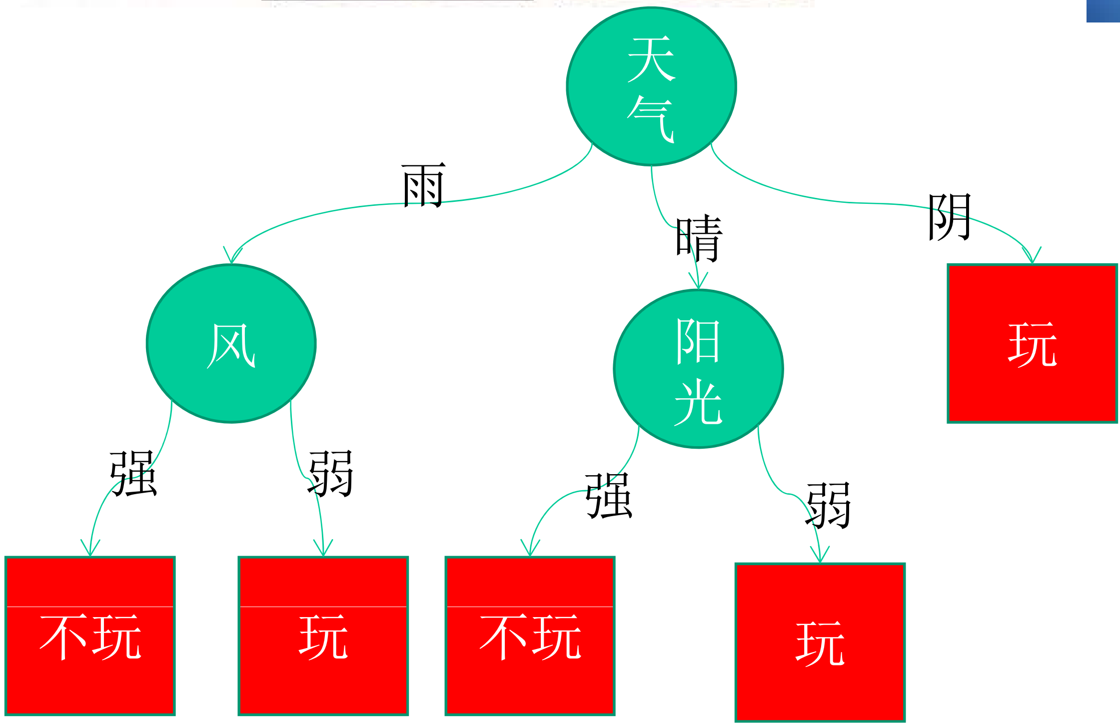
- ID3
- C4.5
- CART

3 过学习与决策树的剪枝

## PART2.以决策树为个体模型的集成学习

# (1)什么是决策树？




决策树是关于if-then规则的集合

规则互斥、完备

**决策树是一种以倒立树形结构描述的决策规则集合。**

决策树由根结点、内部结点、叶结点组成。

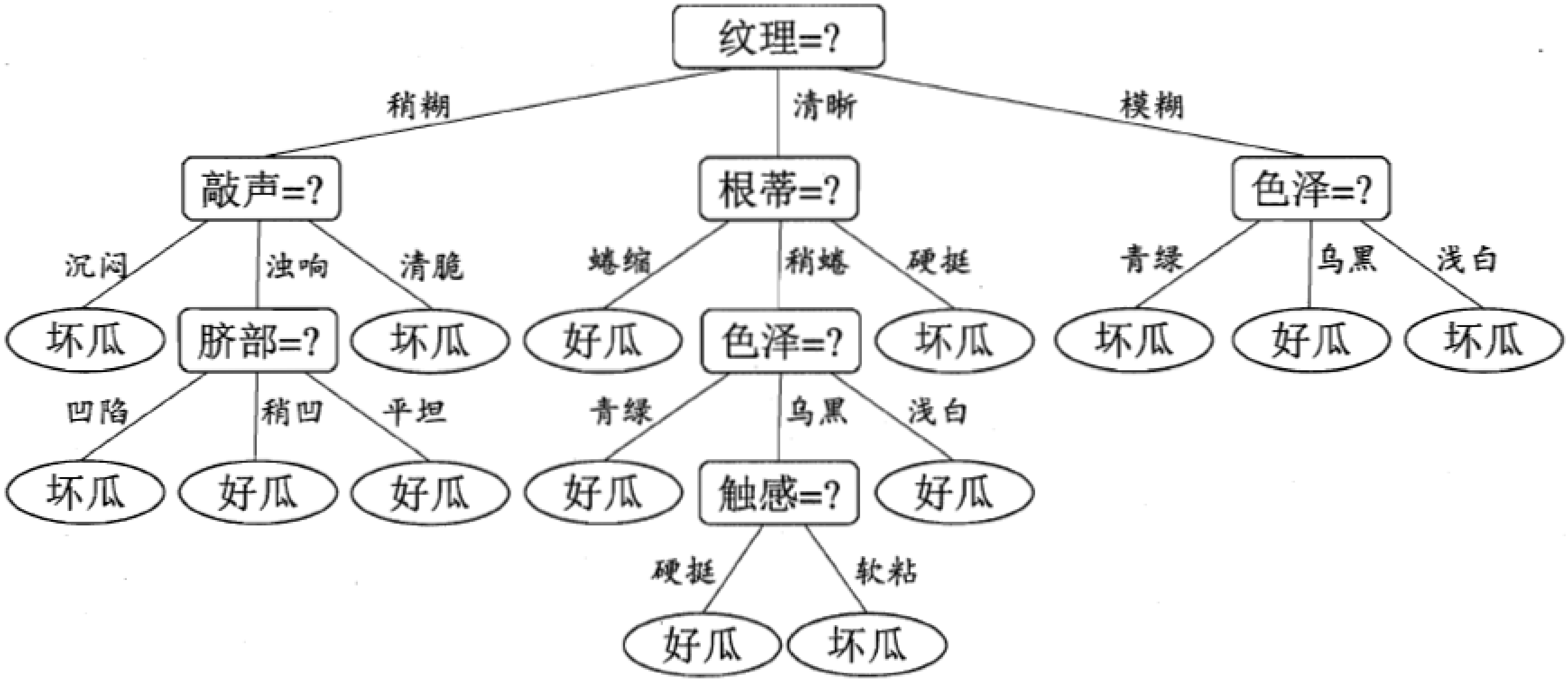
每个非叶结点代表一个测试(查询)，该结点的每个分枝表示该测试的一个结果；每个叶结点代表一个决策结果。

若为分类树，则决策结果为预测类别；
若为回归树，则决策结果为实数值。

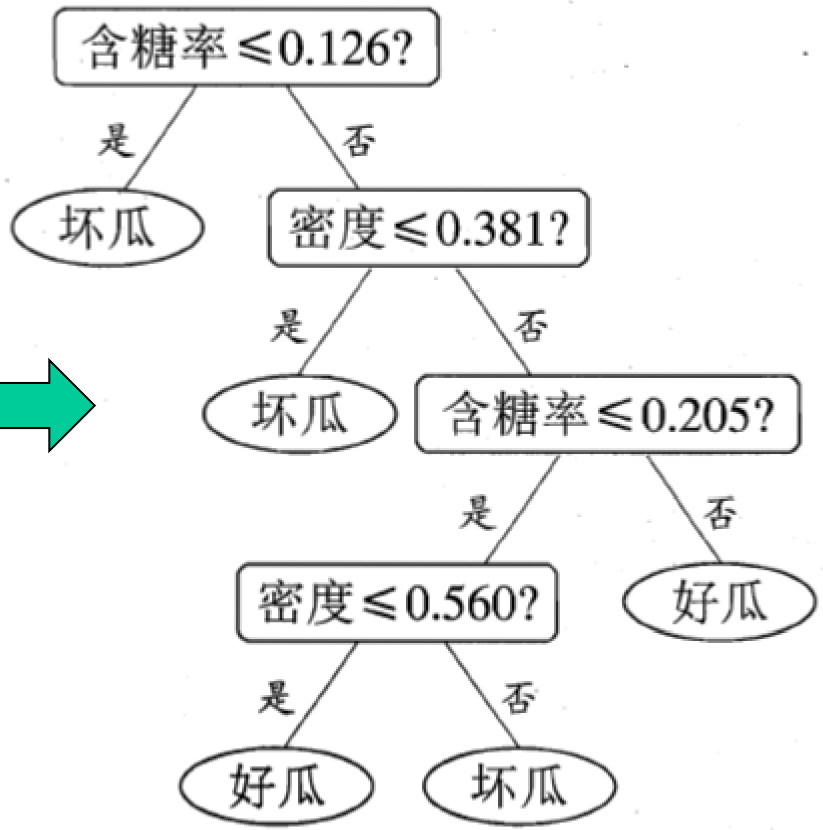
从根节点通向叶节点的一条路径对应一个决策规则。

决策树是应用最广的归纳推理方法之一，模型直观

| 编号 | 色泽 | 根蒂 | 敲声 | 纹理 | 脐部 | 触感 | 好瓜 |
|---|---|---|---|---|---|---|---|
| 1 | 青绿 | 蜷缩 | 浊响 | 清晰 | 凹陷 | 硬滑 | 是 |
| 2 | 乌黑 | 蜷缩 | 沉闷 | 清晰 | 凹陷 | 硬滑 | 是 |
| 3 | 乌黑 | 蜷缩 | 浊响 | 清晰 | 凹陷 | 硬滑 | 是 |
| 4 | 青绿 | 蜷缩 | 沉闷 | 清晰 | 凹陷 | 硬滑 | 是 |
| 5 | 浅白 | 蜷缩 | 浊响 | 清晰 | 凹陷 | 硬滑 | 是 |
| 6 | 青绿 | 稍蜷 | 浊响 | 清晰 | 稍凹 | 软粘 | 是 |
| 7 | 乌黑 | 稍蜷 | 浊响 | 稍糊 | 稍凹 | 软粘 | 是 |
| 8 | 乌黑 | 稍蜷 | 浊响 | 清晰 | 稍凹 | 硬滑 | 是 |
| 9 | 乌黑 | 稍蜷 | 沉闷 | 稍糊 | 稍凹 | 硬滑 | 否 |
| 10 | 青绿 | 硬挺 | 清脆 | 清晰 | 平坦 | 软粘 | 否 |
| 11 | 浅白 | 硬挺 | 清脆 | 模糊 | 平坦 | 硬滑 | 否 |
| 12 | 浅白 | 蜷缩 | 浊响 | 模糊 | 平坦 | 软粘 | 否 |
| 13 | 青绿 | 稍蜷 | 浊响 | 稍糊 | 凹陷 | 硬滑 | 否 |
| 14 | 浅白 | 稍蜷 | 沉闷 | 稍糊 | 凹陷 | 硬滑 | 否 |
| 15 | 乌黑 | 稍蜷 | 浊响 | 清晰 | 稍凹 | 软粘 | 否 |
| 16 | 浅白 | 蜷缩 | 浊响 | 模糊 | 平坦 | 硬滑 | 否 |
| 17 | 青绿 | 蜷缩 | 沉闷 | 稍糊 | 稍凹 | 硬滑 | 否 |


纹理=?
稍糊
清晰
模糊
敲声=?
根蒂=?
色泽=?
沉闷
浊响
清脆
蜷缩
稍蜷
硬挺
青绿
乌黑
浅白
坏瓜
脐部=?
坏瓜
好瓜
色泽=?
坏瓜
坏瓜
好瓜
坏瓜
凹陷
稍凹
平坦
青绿
乌黑
浅白
坏瓜
好瓜
好瓜
好瓜
触感=?
好瓜
硬挺
软粘
好瓜
坏瓜

| 编号 | 密度 | 含糖率 | 好瓜 |
|---|---|---|---|
| 1 | 0.697 | 0.460 | 是 |
| 2 | 0.774 | 0.376 | 是 |
| 3 | 0.634 | 0.264 | 是 |
| 4 | 0.608 | 0.318 | 是 |
| 5 | 0.556 | 0.215 | 是 |
| 6 | 0.403 | 0.237 | 是 |
| 7 | 0.481 | 0.149 | 是 |
| 8 | 0.437 | 0.211 | 是 |
| 9 | 0.666 | 0.091 | 否 |
| 10 | 0.243 | 0.267 | 否 |
| 11 | 0.245 | 0.057 | 否 |
| 12 | 0.343 | 0.099 | 否 |
| 13 | 0.639 | 0.161 | 否 |
| 14 | 0.657 | 0.198 | 否 |
| 15 | 0.360 | 0.370 | 否 |
| 16 | 0.593 | 0.042 | 否 |
| 17 | 0.719 | 0.103 | 否 |



## (2)决策树的优势

- 语义可表示性

----从根节点到叶节点的一条决策规则为合取式
----利用合取式和析取式获得某个类别的明确描述

- 决策速度快

只需一系列关于样本的简单查询，即可对样本的输出做出判断

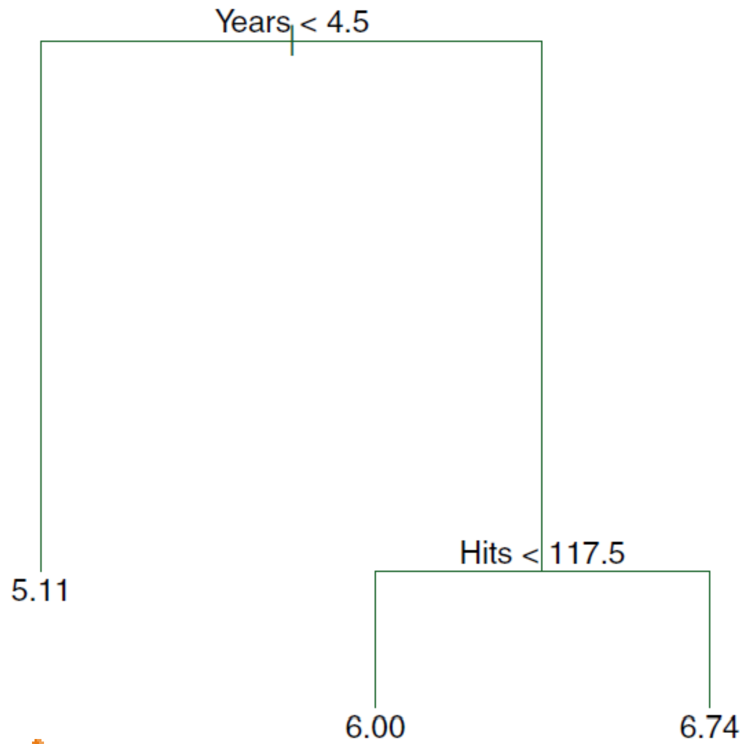
- 可以很自然的嵌入专家的先验知识

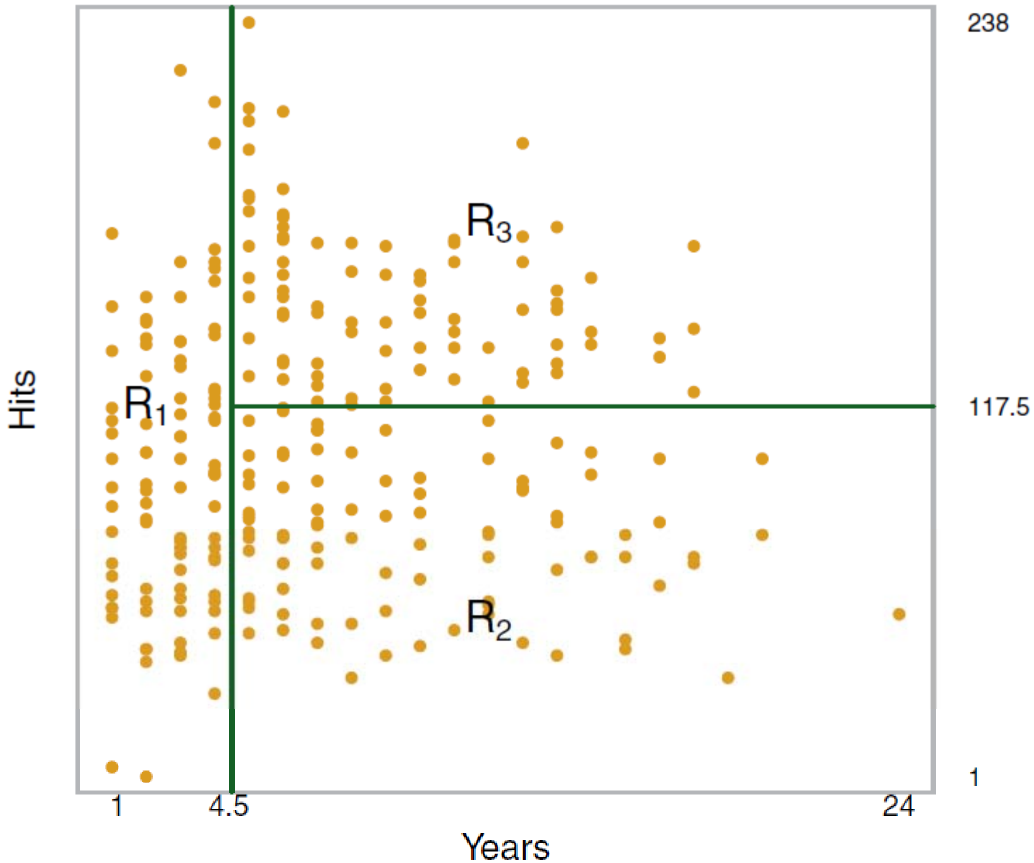
(3)决策树的叶结点与特征空间的划分及相应决策结果(决策域)

例：基于**回归树**，预测棒球运动员的**薪金**.

两种特征：

- Years—棒球运动员在大联盟中的效力时间
- Hits －棒球运动员在上一年度的击球打数



## (4)决策树模型的学习与使用

- 模型的监督式学习—决策树的构建

归纳：决策规则的生成。

基于一定数量训练样本，从数据中学习决策规则，自动构造。

特征空间的最终划分

- 模型的使用—利用生成的规则，对观测样本进行决策推理

## A. 模型的学习

- **特征选择**

  基于训练样本集，从中选择最优划分特征

- **决策树的生成(模型的局部选择)**

  递归生成决策树，拟合训练样本

- **决策树的剪枝(模型的全局选择)**

  简化模型，使其泛化能力更好

  许多分枝反映的是训练样本中的噪声和孤立点

  为避免过学习，应控制树的规模，检测和剪枝

  预剪枝(prepruning)、后剪枝(postpruning)

## 决策树的生成：

从上到下，分而治之(divide-and-conquer)，递归生长。

最初，所有训练样本都在根结点，所有样本根据每次选择出的特征，递归、逐渐划分；

选出来的特征称为一个划分(split)或查询(query)，查询的选择基于启发式或统计特征，满足如下条件之一时，划分操作停止：

- 所有落入某一结点的样本均属于同一类

  该结点成为叶结点，标记为该类别

- 没有特征能够进一步用于划分样本集

  该结点成为叶结点，类别标签为落入该结点的多数样本所属的类别

- 没有任何样本落入某一结点

  该结点成为叶结点，类别标签为落入父结点多数样本所属类别

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

# B. 模型的使用

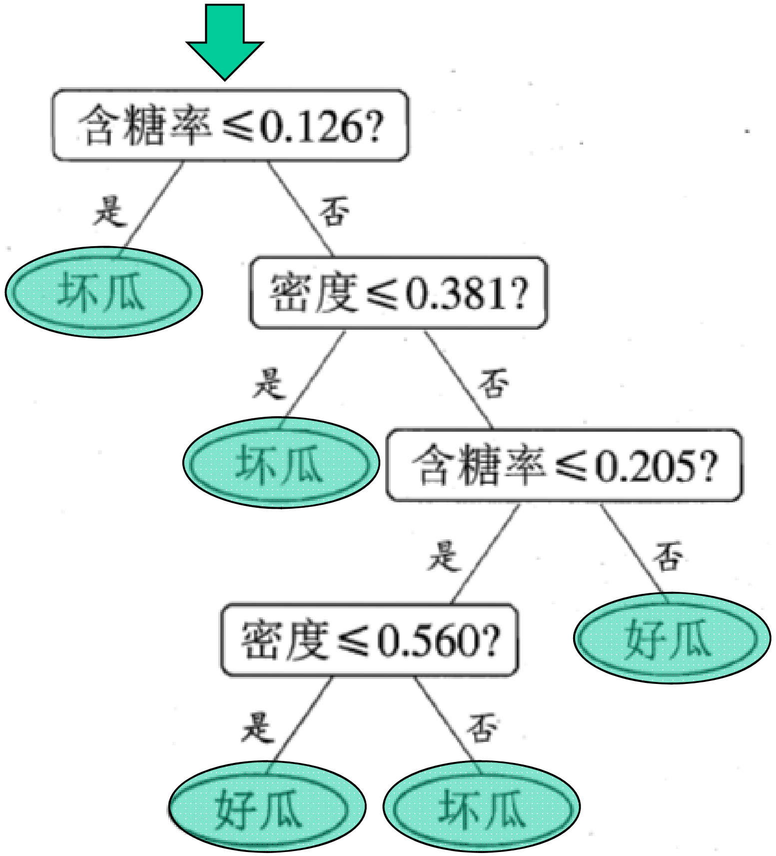
从根结点开始，对输入样本的特征取值提问

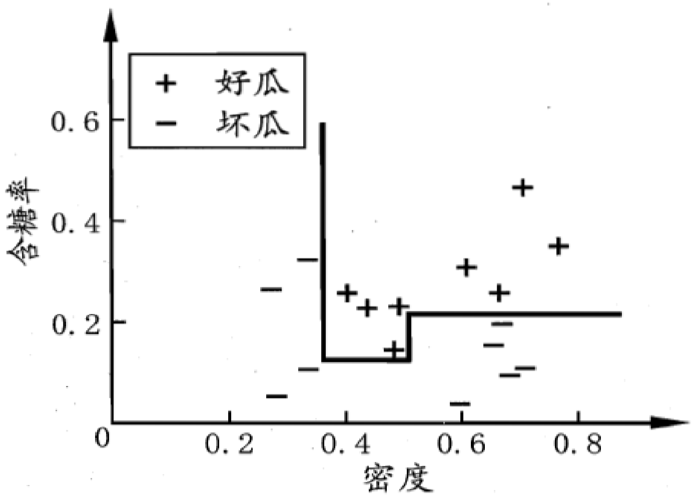
与根结点相连的不同分枝，对应于特征的不同取值

根据不同回答，转向相应的分枝

在新到达的结点处，做类似的分枝判断…

持续上述过程，直到叶子结点，输出该叶子结点对应的类别标记(或函数值)。



# 主要内容

## PART 1.决策树

1 非数值特征(*nonmetric features*)

2 决策树

2.1 决策树的引入

2.2 特征选择

2.3 决策树的构建(模型的学习)

三个著名的决策树构建方法

ID3

C4.5

CART

3 过学习与决策树的剪枝

## PART2.以决策树为个体模型的集成学习

## (1)有关概念

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

- 纯结点(数据集)、不纯结点(数据集)

若到达某结点的训练样本集只含一类样本，则该结点为纯(pure)结点，或为同质(homogenous) 结点

否则，为不纯(impure)、或异构(heterogeneous)结点。

- 结点的不纯度(impurity，杂度)

关于决策树结点不纯程度的度量.

如：熵不纯度、Gini不纯度、误差不纯度等

设到达某结点的训练样本集$\boldsymbol{D}$含$K$个不同类别，$\boldsymbol{D}=\boldsymbol{D}_1\cup\cdots\cup\boldsymbol{D}_K$

类别集合$\boldsymbol{Y}=\{\omega_1,\ldots,\omega_K\}$　　$K=|\boldsymbol{Y}|$

样本容量$N=|\boldsymbol{D}|=\sum_{j=1}^{|\boldsymbol{Y}|}|\boldsymbol{D}_j|=\sum_{j=1}^{K}N_j$

第$j$类出现的概率 $P_j=\frac{|\boldsymbol{D}_j|}{|\boldsymbol{D}|}=\frac{N_j}{N}$

- **熵不纯度(entropy impurity)**

$$I_{Entropy}(\boldsymbol{D})=-\sum_{i=1}^{|\boldsymbol{Y}|}P_i\log_2 P_i$$

约定：　$0\log 0=0$

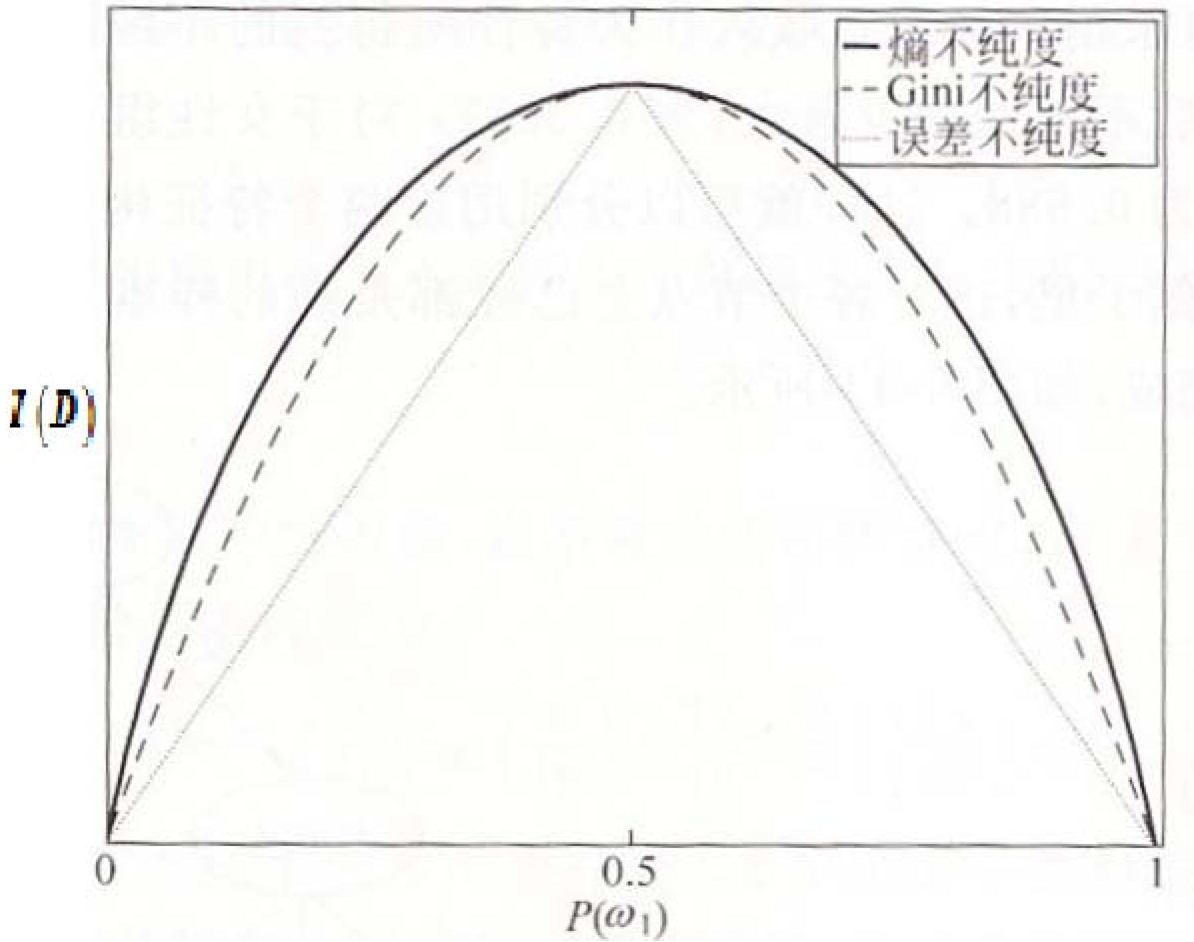
- Gini不纯度(Gini impurity)/方差不纯度

$$\boldsymbol{I}_{Gini}\left(\boldsymbol{D}\right)=\sum_{j=1}^{K}\sum_{\substack{i=1\\ i\neq j}}^{K}P_iP_j=1-\sum_{j=1}^{K}P_j^2$$

- 误差不纯度

$$\boldsymbol{I}_{Error}\left(\boldsymbol{D}\right)=1-\max_{j\in\{1,\dots,k\}}P_j$$

# 两类别分类，关于同一数据集的三种不纯度度量与某类概率关系



## (2)基于“不纯度”的选择规则

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

决策树的结点生成，伴随着**特征选择**。

一般而言，随着结点划分的不断进行，希望决策树分枝结点所含样本尽量来自相同类别，即：节点"纯度"不断增加.

设到达**某结点**的**数据集**$D$内，属于第$j$个类别的样本构成集合$D_j, j=1,\ldots.,K$

$$D = D_1 \cup D_2 \cdots \cup D_K$$

数据集$D$内样本关于特征$a$的取值为$m$个$\left\{a^{(1)}, a^{(2)}, \ldots, a^{(m)}\right\}$, 其中对应$a$=$a^{(i)}$的样本构成子集$D^{(i)}$，并且在子集$D^{(i)}$内，属于第$j$个类别的样本集合$D_j^{(i)}$，则有：

$$D^{(i)} = D_1^{(i)} \cup D_2^{(i)} \cdots \cup D_K^{(i)}$$

若基于特征$a$的取值情况，对数据集$D$所在**结点**划分，得$m$个分枝结点，并且第$i$个节点包含的样本集为$D^{(i)}$

$$D = D^{(1)} \cup D^{(2)} \cup \cdots \cup D^{(m)}$$

## A. 信息增益(Information Gain) --绝对增益

特征$a$对训练集$D$的**信息增益**$Gain(D,a)$:

--基于特征$a$对某结点数据集$D$划分，导致的不纯度减少量

$$\boxed{Gain(D,a)=I_{Entropy}(D)-\sum_{i=1}^{m}\frac{\left|D^{(i)}\right|}{|D|}I_{Entropy}(D_i)}$$

样本集$D$所在结点不纯度：$I_{Entropy}(D)=-\sum_{j=1}^{K}P_j\log_2 P_j=-\sum_{j=1}^{K}\frac{\left|D_j\right|}{|D|}\log_2\frac{\left|D_j\right|}{|D|}$

第$i$个子结点的不纯度：$I_{Entropy}(D_i)=-\sum_{j=1}^{K}\frac{\left|D_j^{(i)}\right|}{\left|D^{(i)}\right|}\log_2\frac{\left|D_j^{(i)}\right|}{\left|D^{(i)}\right|}$

例：ID3决策树内每个非叶结点的特征选择，采用最大“绝对信息增益”准则，选特征

$$a^{*}=\arg\max_{a\in A} Gain(D,a)$$

但上述准则，对那些具有较多离散取值的特征，更为偏好，为减少这种不利影响，引入“相对信息增益”。

## B. 信息增益率(Information Gain Ratio)—相对增益

$$Gain\_ratio(D,a)=\frac{Gain(D,a)}{IV(a)}$$

特征$a$对训练集$D$的**绝对信息增益**$Gain(D,a)$

$$Gain(D,a)=I_{Entropy}(D)-\sum_{i=1}^{m}\frac{|D^{(i)}|}{|D|}I_{Entropy}(D_i)$$

$$=-\sum_{j=1}^{K}\frac{|D_j|}{|D|}\log_2\frac{|D_j|}{|D|}-\sum_{i=1}^{m}\frac{|D^{(i)}|}{|D|}\left[-\sum_{j=1}^{K}\frac{|D_j^{(i)}|}{|D^{(i)}|}\log_2\frac{|D_j^{(i)}|}{|D^{(i)}|}\right]$$

特征$a$在训练集$D$的属性"**固有值**"(Intrinsic Value, IV)

$$IV(a)=-\sum_{i=1}^{m}\frac{|D^{(i)}|}{|D|}\log_2\frac{|D^{(i)}|}{|D|}$$

C4.5决策树基于候选特征，估计“增益率”平均值，确定增益率高出平均水平、并具有最大增益率的特征：

$$a^* = \arg\max_{a \in A^*} Gain\_ratio(D, a)$$

## C. 基于“基尼指数”的信息增益

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

$$Gain_{Gini}(D,a)=I_{Gini}(D)-\sum_{i=1}^{m}\frac{|D^{(i)}|}{|D|}I_{Gini}\left(D^{(i)}\right)$$

$$=\left(1-\sum_{j=1}^{K}\left(\frac{|D_j|}{|D|}\right)^2\right)-\sum_{i=1}^{m}\frac{|D^{(i)}|}{|D|}\left[1-\sum_{j=1}^{K}\left(\frac{|D_j^{(i)}|}{|D^{(i)}|}\right)^2\right]$$

特征$a$关于训练集$D$的(**划分后**)**基尼指数**(*Gini Index*)

$$\boldsymbol{Gini\_index}(D,a)=\sum_{i=1}^{m}\frac{|D^{(i)}|}{|D|}I_{Gini}\left(D^{(i)}\right)=\sum_{i=1}^{m}\frac{|D^{(i)}|}{|D|}\left[1-\sum_{j=1}^{K}\left(\frac{|D_j^{(i)}|}{|D^{(i)}|}\right)^2\right]$$

CART决策树基于最小“划分后基尼指数” 原则，进行结点特征选择。

$$a^{*} = \arg\min_{a \in A} Gini\_index(D, a)$$

# 主要内容

## PART 1.决策树

1 非数值特征(*nonmetric features*)

2 决策树

2.1 决策树的引入

2.2 树的划分选择

2.3 决策树的构建(模型的学习)

三个著名的决策树构建方法

ID3
C4.5
CART

3 过学习与决策树的剪枝

## PART2.以决策树为个体模型的集成学习

## 决策树算法的研究历史

- 第一个决策树算法称为**CLS (Concept Learning System) [E. B. Hunt, J. Marin, and P. T. Stone's book "*Experiments in Induction*" *published by Academic Press in 1966]***

- 真正引发决策树研究热潮的算法是**ID3 [J. R. Quinlan**'s **paper in a book "*Expert Systems in the Micro Electronic Age*"*edited by D. Michie, published by Edinburgh University Press in 1979]***

  其增量版本还有：ID4，ID5等.

- 最流行的决策树算法**C4.5 [J. R. Quinlan**'s **book "*C4.5: Programs for Machine Learning*"*published by Morgan Kaufmann in 1993]*** 以ID3为蓝本，可处理连续特征的算法.

  C5.0 是C4.5的修订版，面向大数据集分类，在执行效率、内存使用用方面做了改进.

- 通用的决策树算法***CART*** **(*Classification and Regression Tree*) [L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone's book "Classification and Regression Trees" published by Wadsworth in 1984]**

- 基于决策树的较强学习算法还有一种称为**随机森林** (Random Forests)的集成算法**[L. Breiman's MLJ'01 paper "Random Forests"]**

- 其他强调伸缩性的决策树算法如： SLIQ、SPRINT、RainForest等.

➔ **ID3，C4.5， CART，Random Forests**

# ID3=>C4.5=>C5.0

- **John Ross Quinlan**
  - ID3　1975年
  - C4.5 1993年
  - C5.0 1998年
  - 2011年获得KDD创新奖

- KDD—Conference on Knowledge Discovery and Data mining
- http://www.rulequest.com/Personal/
- http://rulequest.com/download.html
- http://www.rulequest.com/

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

# ID3决策树

交互式对分法的第3版
Interactive Dichotomizer-3

## (1) ID3 算法基本思想

基于奥克姆剃刀准则(Occam ‘s Razor-- We should always accept the simplest answer that correctly fits our data.)

➔ A good decision tree is the simplest decision tree.

The simplest decision tree that covers all examples should be the least likely to include unnecessary constraints

节点的评价----熵不纯度
新节点的生成----基于目前还没有使用的属性“最大信息增益”

算法基本点：

- 若当前结点只含同一类样本,则为纯结点，则停止分裂；
- 若当前属性表中再无可用属性，则根据多数表决确定该结点的类标号，停止分裂；
- 其它：选择最佳分裂的属性(最大信息增益足够大)

根据所选属性取值(特征取值数目决定了该节点分裂为后继子节点的数目)，逐一进行分裂；递归构造决策树。

- ID3决策树仅仅适用于离散、或者非数值型样本集。
  不处理缺失信息、不涉及剪枝、。
- 每个结点的分枝数目与该结点所用的特征取值数目一致。
- 基于“最大绝对信息增益”准则，确定当前结点分裂所使用的特征。
- 算法直到所有叶结点的不纯度最小(如：到达该结点的训练样本来自同一类别)、或者不再有可用的特征时停止
- ID3算法的标准版，仅涉及树的生成，无剪枝步骤

## (2)ID3算法

**输入**：训练样本集$D$, 特征集$A$，阈值$\varepsilon$

**输出**：决策树$T$

**步骤：**

**STEP1.** 若$D$中所有样本属于同一类$\omega_k$,则$T$为单结点树，并将$\omega_k$作为该结点的类别标记，返回$T$;

**STEP2.** 若$A$为空集，则$T$为单结点树，并将$D$中训练样本数目最多的类别$\omega_k$作为该结点的类别标记，返回$T$;

**STEP3.** 若$A$不是空集，计算$A$中各特征 $a \in A$对样本集$D$的信息增益$\{g(D,a), a \in A\}$，并选择具有**最大信息增益**的特征$a_g$:

若特征$a_g$的**信息增益**$g(D,a_g) < \varepsilon$，则执行**3.1**；否则执行**3.2.**

**ID3算法**只有决策树的生成部分，未涉及裁剪，易产生**过拟合**。

# ID3算法(续)

**步骤：**

**STEP3.** 若特征$a_g$的**信息增益**$g\left(D,a_g\right)<\varepsilon$，则执行**3.1**；否则执行**3.2.**

**3.1** 置$T$为单结点树，将$D$中具有最多训练样本数目的类别$\omega_k$作为该结点的类别标记，并且返回$T$；

**3.2** 对特征$a_g$的每一可能值$a_g^{(i)}$，按照$a_g=a_g^{(i)}$，并将$D$划分为若干非空子集$D^{(i)}$，将$D^{(i)}$中具有最多训练样本数目的类别作为标记，构建子结点，由结点及其子结点构成树$T$，返回$T$；

**STEP4.** 对第$i$个子结点，以$D^{(i)}$为训练集，以$A-\left\{a_g\right\}$为特征集，**递归调用STEP1-STEP3**得到子树$T_i$, 返回$T_i$。

# C4.5决策树

## Classifier 4.5

# (1)C4.5算法是对ID3的扩展

决策树学习的实际问题:

决策树增长的深度的确定;
连续数值特征的处理;
用于筛选特征的度量指标的确定;
特征不完整的训练数据的处理;
....

针对上述问题，ID3扩展为C4.5

C4.5 的特别之处:

- 连续数值特征的处理
- 缺失值的处理

C4.5是ID3算法的后继和改进

可以处理实值数据

每个划分的分枝因子等于查询属性的取值个数

采用信息增益率作为选择查询的依据

首先让树充分生长，然后利用分枝的统计显著性来实现剪枝

C4.5为目前最为流行的决策树算法

## (2)C4.5(Classifier 4.5)算法描述

**输入**：训练样本集$D$, 特征集$A$，阈值$\varepsilon$

**输出**：决策树$T$

**步骤**：

**STEP1.** 若$D$中所有样本属于同一类$\omega_k$,则$T$为单结点树，并将$\omega_k$作为该结点的类别标记，返回$T$;

**STEP2.** 若$A$为空集，则$T$为单结点树，并将$D$中具有最多训练样本的类别$\omega_k$作为该结点的类别标记，返回$T$;

**STEP3.** 若$A$不是空集，计算$A$中各特征 $a\in A$对样本集$D$的信息增益比$\{g_R(D,a)\}$，并选择具有**最大信息增益比**的特征$a_g$:

若特征$a_g$的**信息增益比**$g_R(D,a_g)<\varepsilon$，则执行**3.1**；否则执行**3.2.**

# C4.5算法(续)

**步骤：**

**STEP3.** 若特征$a_g$的**信息增益**比$g_R\left(D,a_g\right)<\varepsilon$，则执行**3.1**；否则执行**3.2.**

**3.1** 置$T$为单结点树，将$D$中具有最多训练样本的类别$\omega_k$作为该结点的类别标记，并且返回$T$；

**3.2** 对特征$a_g$的每一可能值$a_g^{(i)}$，按照$a_g=a_g^{(i)}$，并将$D$划分为若干非空子集$D^{(i)}$，将$D^{(i)}$中具有最多训练样本的类别作为标记，构建子结点，由结点及其子结点构成树$T$，返回$T$；

**STEP4.** 对第$i$个子结点，以$D^{(i)}$为训练集，以$A-\left\{a_g\right\}$为特征集，**递归调用STEP1-STEP3**得到子树$T_i$，返回$T_i$。

## (3)C4.5 算法关于连续数值特征的处理方式—二分法

设训练样本集$\boldsymbol{D}$关于特征集$\boldsymbol{A}$中的**某连续特征**$a$出现了$\boldsymbol{n}$个不同取值,

这些取值按照升序排列有: $\{a^1, a^2, \ldots, a^n\}$

基于**划分点**$t$，可将数据集$\boldsymbol{D}$分成两个子集:

$$\boldsymbol{D}_t^- = \{\boldsymbol{x} \mid \boldsymbol{x} \in \boldsymbol{D}, \text{并且} \boldsymbol{x}(a) \le t\}$$

$$\boldsymbol{D}_t^+ = \{\boldsymbol{x} \mid \boldsymbol{x} \in \boldsymbol{D}, \text{并且} \boldsymbol{x}(a) > t\}$$

关于**连续特征**$a$,划分点$t$**的候选取值集合** $\boldsymbol{T}_a = \left\{ \frac{a^i + a^{i+1}}{2} \mid 1 \le \boldsymbol{i} \le \boldsymbol{n} - 1 \right\}$

其中$\frac{a^i + a^{i+1}}{2}$为区间$[a^i, a^{i+1})$的中位点.

样本集$D$基于**划分点**$t$划分后的绝对信息增益：

$$Gain(D,a,t) = I_{Entropy}(D) - \sum_{\lambda\in\{-,+\}} \frac{|D_t^\lambda|}{|D|} I_{Entropy}(D_t^\lambda)$$

对于**连续特征**$a$，应选择使$Gain(D,a,t)$取最大值的最优划分点$t^*$：

$$t^* = \arg\max_{t\in T_a} Gain(D,a,t)$$

$$Gain(D,a) = Gain(D,a,t^*)$$

其中 $T_a = \left\{ \frac{a^i + a^{i+1}}{2} \mid 1 \le i \le n-1 \right\}$

注意：连续特征$a$可在决策树中被使用多次.

# (4)C4.5算法关于特征缺失值的处理方式

两个核心问题：

- 如何在特征取值缺失情况下，进行划分特征的选择？
- 给定划分特征，若训练样本集关于该特征取值存在部分缺失，如何进行样本集的有效划分？

# 例：存在特征取值缺失的样本集

| 编号 | 色泽 | 根蒂 | 敲声 | 纹理 | 脐部 | 触感 | 好瓜 |
|---|---|---|---|---|---|---|---|
| 1 | – | 蜷缩 | 浊响 | 清晰 | 凹陷 | 硬滑 | 是 |
| 2 | 乌黑 | 蜷缩 | 沉闷 | 清晰 | 凹陷 | – | 是 |
| 3 | 乌黑 | 蜷缩 | – | 清晰 | 凹陷 | 硬滑 | 是 |
| 4 | 青绿 | 蜷缩 | 沉闷 | 清晰 | 凹陷 | 硬滑 | 是 |
| 5 | – | 蜷缩 | 浊响 | 清晰 | 凹陷 | 硬滑 | 是 |
| 6 | 青绿 | 稍蜷 | 浊响 | 清晰 | – | 软粘 | 是 |
| 7 | 乌黑 | 稍蜷 | 浊响 | 稍糊 | 稍凹 | 软粘 | 是 |
| 8 | 乌黑 | 稍蜷 | 浊响 | – | 稍凹 | 硬滑 | 是 |
| 9 | 乌黑 | – | 沉闷 | 稍糊 | 稍凹 | 硬滑 | 否 |
| 10 | 青绿 | 硬挺 | 清脆 | – | 平坦 | 软粘 | 否 |
| 11 | 浅白 | 硬挺 | 清脆 | 模糊 | 平坦 | – | 否 |
| 12 | 浅白 | 蜷缩 | – | 模糊 | 平坦 | 软粘 | 否 |
| 13 | – | 稍蜷 | 浊响 | 稍糊 | 凹陷 | 硬滑 | 否 |
| 14 | 浅白 | 稍蜷 | 沉闷 | 稍糊 | 凹陷 | 硬滑 | 否 |
| 15 | 乌黑 | 稍蜷 | 浊响 | 清晰 | – | 软粘 | 否 |
| 16 | 浅白 | 蜷缩 | 浊响 | 模糊 | 平坦 | 硬滑 | 否 |
| 17 | 青绿 | – | 沉闷 | 稍糊 | 稍凹 | 硬滑 | 否 |

设训练样本集$\boldsymbol{D}=\{(\boldsymbol{x}_i,\boldsymbol{y}_i),\boldsymbol{i}=1,\ldots,\boldsymbol{m}\}$关于特征集$\boldsymbol{A}$中的**某特征**$a$出现了取值的部分缺失，类别标号$\boldsymbol{y}_i\in Y.$

其中，不存在缺失值的样本子集为$\widetilde{\boldsymbol{D}}.$

设$\widetilde{\boldsymbol{D}}$关于**特征**$a$取值共$\boldsymbol{V}$个，构成集合$\{a^1,a^2,\ldots,a^V\}$

$$\begin{cases}\widetilde{\boldsymbol{D}}\text{中，关于特征}a\text{取值为}a^v\text{的样本构成子集}\widetilde{\boldsymbol{D}}^v\\ \widetilde{\boldsymbol{D}}\text{中，来自第}\boldsymbol{k}\text{类的样本构成子集}\widetilde{\boldsymbol{D}}_k\end{cases}$$

显然：
$$\begin{cases}\widetilde{\boldsymbol{D}}=\widetilde{\boldsymbol{D}}^1\cup\widetilde{\boldsymbol{D}}^2\cup\cdots\cup\widetilde{\boldsymbol{D}}^V\\ \widetilde{\boldsymbol{D}}=\widetilde{\boldsymbol{D}}_1\cup\widetilde{\boldsymbol{D}}_2\cup\cdots\cup\widetilde{\boldsymbol{D}}_{|Y|}\end{cases}$$

对于$\forall x \in D$,引入样本权重$\omega_x$， $\sum_{x \in D} \omega_x = 1$

$D$内关于**特征**$a$，无缺失值样本所占比例 $\rho = \frac{\sum_{x \in \tilde{D}} \omega_x}{\sum_{x \in D} \omega_x}$

$\tilde{D}$内第$k$类的样本所占比例 $\tilde{p}_k = \frac{\sum_{x \in \tilde{D}_k} \omega_x}{\sum_{x \in \tilde{D}} \omega_x}$

$\tilde{D}$内关于**特征**$a$取值为$a^v$的样本所占比例 $\tilde{r}_v = \frac{\sum_{x \in \tilde{D}^v} \omega_x}{\sum_{x \in \tilde{D}} \omega_x}$

**特征取值存在部分缺失时的信息增益：**

$$Gain(D,a) = \rho Gain(\tilde{D},a) = \rho\left[I_{Entropy}(\tilde{D}) - \sum_{v=1}^{V} \tilde{r}_v I_{Entropy}(\tilde{D}^v)\right]$$

其中$I_{Entropy}(\tilde{D}) = -\sum_{k=1}^{|Y|} \tilde{p}_k \log_2 \tilde{p}_k$

# CART 决策树

## Classification And Regression Tree

## 分类与回归树

## (1)CART树的引入

核心思想相同

主要区别

- CART既可用于分类，也可用于对连续变量的回归

- 每个结点只能有两个子结点，决策树为二叉树，不易产生数据碎片，精确度往往也会高于多叉树，所以在CART算法中，采用了二元划分----递归二叉树

- 不纯性度量
  - 分类目标：Gini指标
  - 连续目标：最小平方残差、最小绝对残差

用独立的验证集对训练集生长的树进行后剪枝

## (2)回归树

## CART树--最小二乘回归树的生成算法

**基本思想：**

一个回归树对应输入空间(或特征空间)的一个划分，以及在该划分单元上的输出值。

在训练样本集$D$所在的输入空间，递归地将每个区域划分为两个子区域，并根据落入每个子区域的训练样本输出值，决定该子区域的输出，构建二叉树。

## CART树--最小二乘回归树生成算法

**输入**：训练样本集$D=\{(x_i,y_i),i=1,...,N\}$，$x_i\in R^d$

**输出**：回归树$f(x)$

**步骤**：

**STEP1.** 从输入向量$x$中选择**最优切分变量$j$**以及**切分点$s$**,求解：

$$\min_{j,s}\left[\min_{c_1}\boxed{\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2}+\min_{c_2}\boxed{\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2}\right]$$

遍历输入向量$x$的每个变量$j$：对固定的切分变量$j$，选择使上述目标函数值最小的$(j,s)$对。

# CART树--最小二乘回归树生成算法(续)

**步骤：**

**STEP2.** 用上述$(j,s)$对，确定划分区域$R_1(j,s)$，$R_2(j,s)$，并确定相应输出值。

$$R_1(j,s)=\{x|\,x^{(j)}\le s\},\ R_2(j,s)=\{x|\,x^{(j)}>s\}$$

$$\hat{c}_m=\frac{1}{N_m}\sum_{x_i\in R_m(j,s)} y_i,\quad x\in R_m,\qquad m=1,2$$

**STEP3.** 继续对两个子区域调用**STEP1**、**STEP2**，直到满足停止条件。

**STEP4.** 将输入空间划分为M个区域：$R_1,\ldots,R_M$；生成决策树。

$$f(x)=\sum_{m=1}^{M}\hat{c}_m I(x\in R_m)$$

## (3)**分类树**

**CART树--递归二叉分类树**的**生成算法**

**基本思想：**

一个分类树对应输入空间(或特征空间)的一个划分，以及在该划分单元上的类别输出值。

根据训练样本集$\boldsymbol{D}$，从根结点开始，将输入空间进行划分，递归构建二叉分类树。

借助**基尼指数**进行特征选择，同时决定该特征的**最优二值切分点**

# CART树--递归二叉分类树生成算法



**输入**：训练样本集$D=\{(x_i,y_i),i=1,\dots,N\}$，

其中：$x_i\in R^d$，$y_i\in\{1,2,\dots,K\}$

**输出**：CART决策树

**步骤**：

**STEP1.** 设到达当前节点的数据集为$D$。遍历输入向量$x$**现有变量**的每个特征$a$，根据$D$中训练样本关于该变量的所有可能取值，确定所有可能的切分点$s$；对于固定的切分变量$j$，选择使**基尼指数**最小的切分点。最终得到**基尼指数最小的**$(j,s)$对。

$$D=D_1\bigcup D_2,$$

$$D_1=\{(x_i,y_i)\in D\mid x_{ij}\le s\},D_2=\{(x_i,y_i)\in D\mid x_{ij}>s\}$$

在特征$a$的切分点$s$处，集合$D$的基尼指数：

$$\mathrm{Gini}(D,a_s)=\frac{|D_1|}{|D|}\mathrm{Gini}(D_1)+\frac{|D_2|}{|D|}\mathrm{Gini}(D_2)$$

## CART树--递归二叉分类树生成算法(续)

**步骤：**

**STEP1.**(续) $\mathrm{Gini}(D_m)=1-\sum_{k=1}^{K}\left(\frac{|C_{mk}|}{|D_m|}\right)^2 \quad m=1,2$

$C_{mk}$为$D_m$数据集中第$k$类训练样本子集

**STEP2.** 用上述最优特征及最优切分点$(j,s)$对，确定划分区域$R_1(j,s)$，$R_2(j,s)$，将当前结点划分为两个子结点，并将训练集$D_1,D_2$按照特征分配到两个子结点中。

**STEP3.** 继续对两个子结点递归调用**STEP1**、**STEP2**，直到满足停止条件。

**STEP4.** 将输入空间划分为M个区域：$R_1,...,R_M$；生成决策树。

# 主要内容

## PART 1.决策树

1 非数值特征(*nonmetric features*)

2 决策树

2.1 决策树的引入

2.2 树的划分选择

2.3 决策树的构建(模型的学习)

3 过学习与决策树的剪枝

## PART2.以决策树为个体模型的集成学习

# 1.模型的过拟合(overfitting)和欠拟合(underfitting)

- 分类模型的误差：

  训练误差，是在训练样本上误分类样本比例

  泛化误差，是模型关于未知样本分类的期望误差

  训练误差越低，模型的学习能力越好；

  泛化误差越低，模型的推广能力越强

- 好的分类模型应具有低训练误差和低泛化误差。

- 具有较低训练误差的模型，其泛化误差可能高于具有较高训练误差的模型，这种情况称为模型过拟合(过学习)

- 决策树规模很小时，训练和检验误差都很大，  这种情况为模型的欠拟合(欠学习)，原因是模型尚未学习到数据的真实结构。

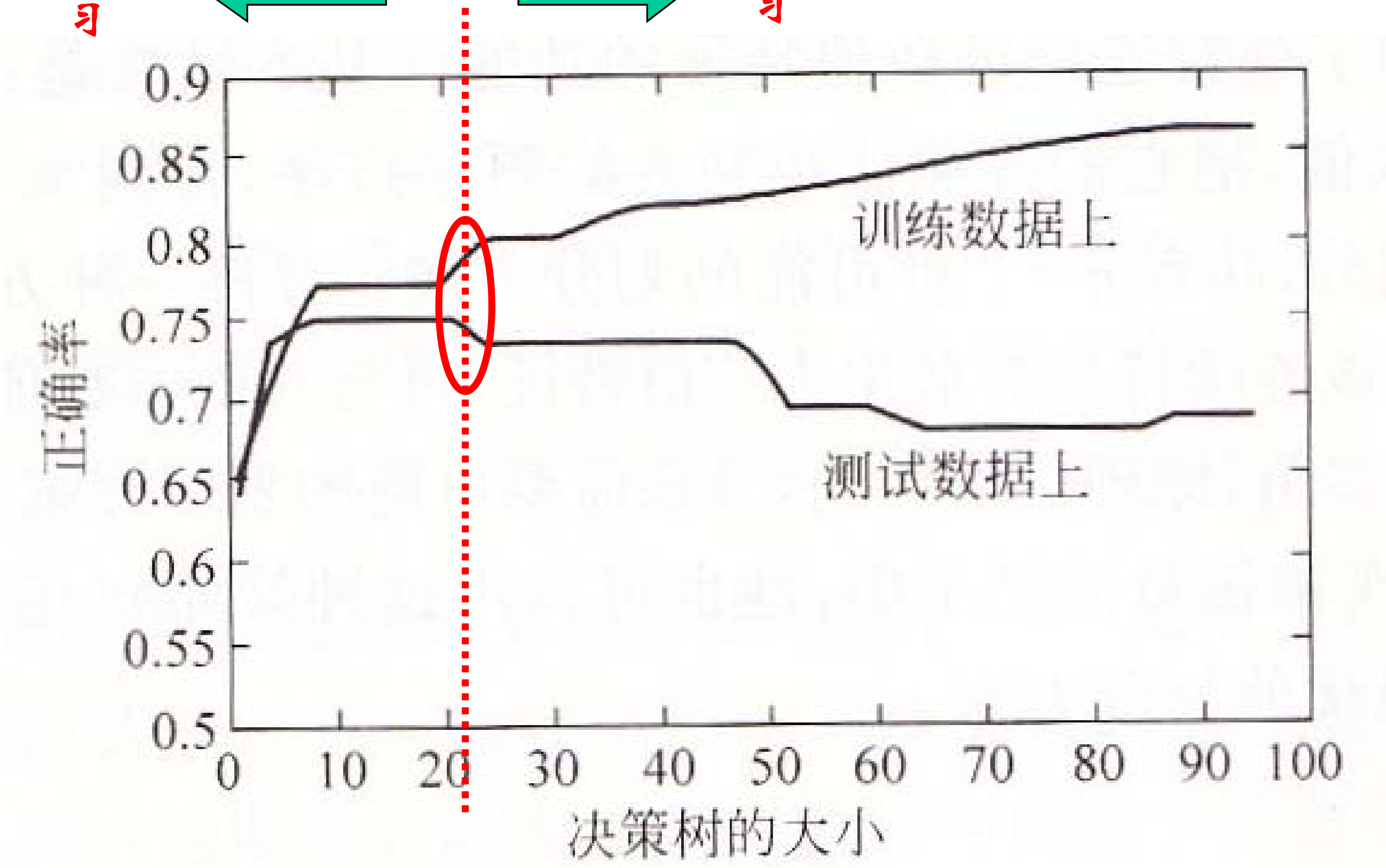
- 随着决策树节点数的增加，模型的训练误差和检验误差都会随之下降。当树的规模变得太大时，即使训练误差还在继续降低，但是检验误差开始增大，导致模型过拟合(过学习),其原因在于过分关注采样偶然性或噪声等因素影响。

- 若训练数据缺乏具有代表性的样本，并且样本规模较小，模型也会产生过拟合。

欠学习

过学习






# ID3决策树的过拟合现象

## 2.决策树的剪枝(pruning)

目的：控制决策树规模，防止模型的过拟合

### 策略1：先剪枝(pre-pruning，预剪枝)

实质——控制决策树的生长

在完全拟合整个训练集之前就停止决策树的生长。

决策树生长过程中，对每个结点在划分前先进行估计。

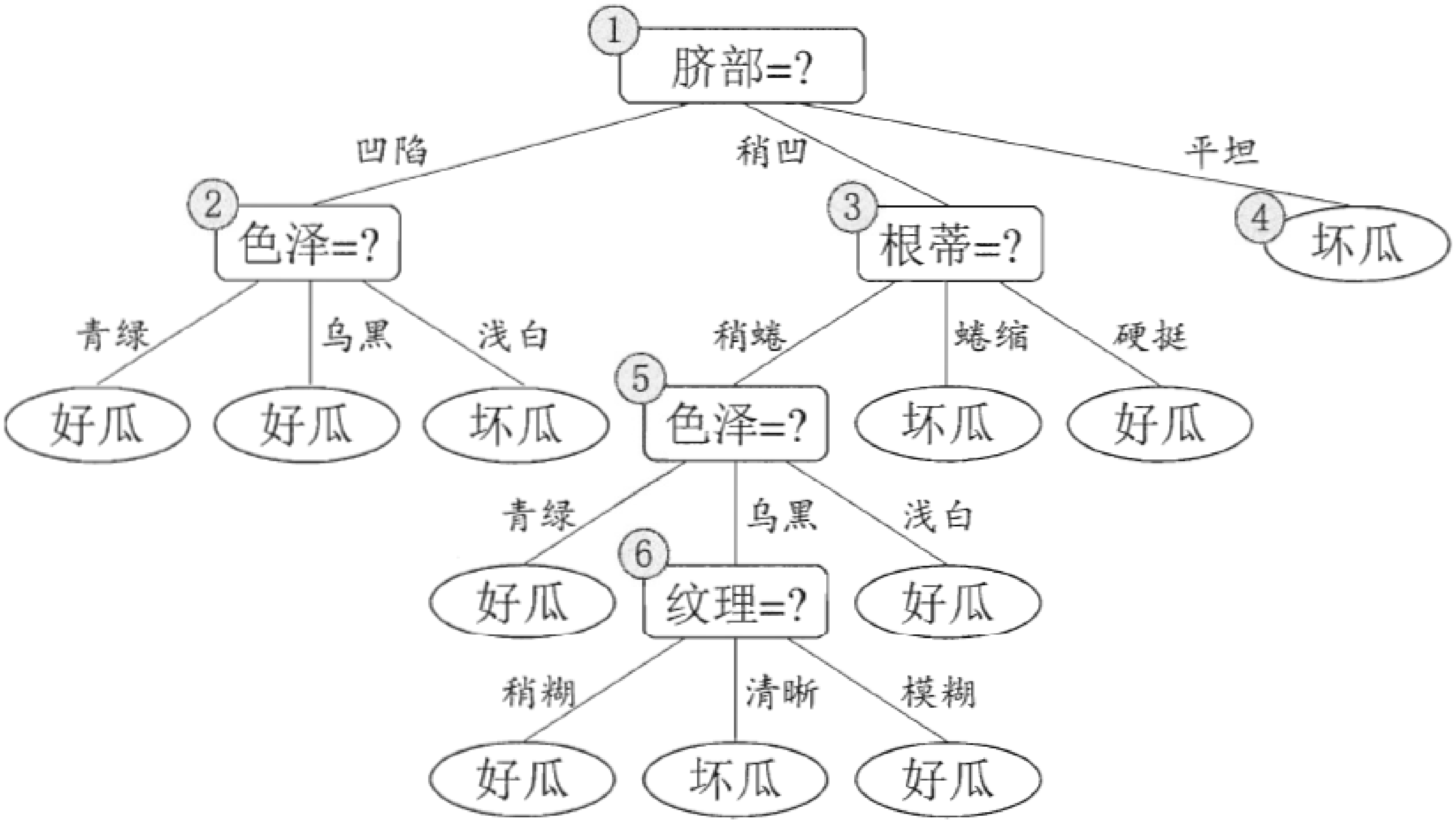
若当前结点的划分不能导致决策树泛化性能的提升，则停止划分，并将该结点标记为叶结点。

用于决策树生长的训练集

| 编号 | 色泽 | 根蒂 | 敲声 | 纹理 | 脐部 | 触感 | 好瓜 |
|---|---|---|---|---|---|---|---|
| 1 | 青绿 | 蜷缩 | 浊响 | 清晰 | 凹陷 | 硬滑 | 是 |
| 2 | 乌黑 | 蜷缩 | 沉闷 | 清晰 | 凹陷 | 硬滑 | 是 |
| 3 | 乌黑 | 蜷缩 | 浊响 | 清晰 | 凹陷 | 硬滑 | 是 |
| 6 | 青绿 | 稍蜷 | 浊响 | 清晰 | 稍凹 | 软粘 | 是 |
| 7 | 乌黑 | 稍蜷 | 浊响 | 稍糊 | 稍凹 | 软粘 | 是 |
| 10 | 青绿 | 硬挺 | 清脆 | 清晰 | 平坦 | 软粘 | 否 |
| 14 | 浅白 | 稍蜷 | 沉闷 | 稍糊 | 凹陷 | 硬滑 | 否 |
| 15 | 乌黑 | 稍蜷 | 浊响 | 清晰 | 稍凹 | 软粘 | 否 |
| 16 | 浅白 | 蜷缩 | 浊响 | 模糊 | 平坦 | 硬滑 | 否 |
| 17 | 青绿 | 蜷缩 | 沉闷 | 稍糊 | 稍凹 | 硬滑 | 否 |

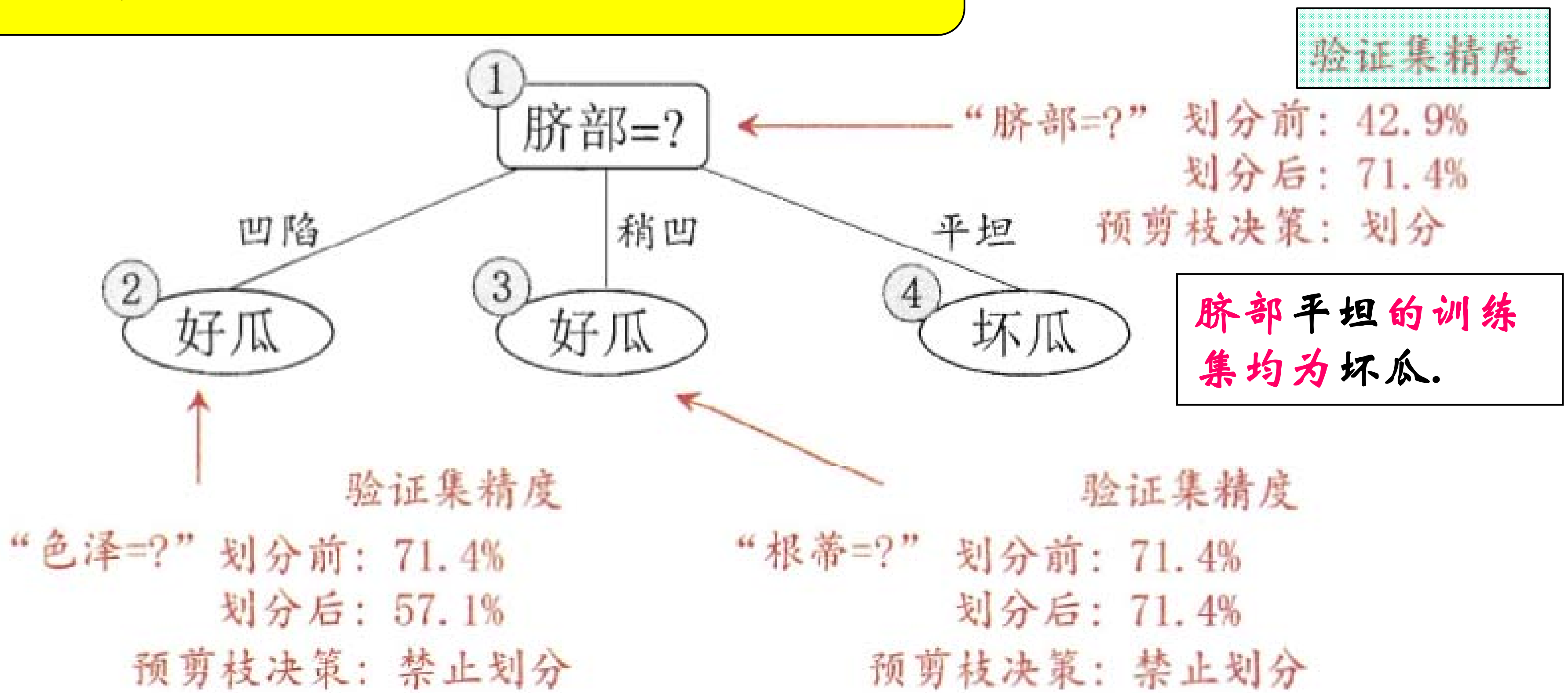
用于决策树预剪枝的验证集

| 编号 | 色泽 | 根蒂 | 敲声 | 纹理 | 脐部 | 触感 | 好瓜 |
|---|---|---|---|---|---|---|---|
| 4 | 青绿 | 蜷缩 | 沉闷 | 清晰 | 凹陷 | 硬滑 | 是 |
| 5 | 浅白 | 蜷缩 | 浊响 | 清晰 | 凹陷 | 硬滑 | 是 |
| 8 | 乌黑 | 稍蜷 | 浊响 | 清晰 | 稍凹 | 硬滑 | 是 |
| 9 | 乌黑 | 稍蜷 | 沉闷 | 稍糊 | 稍凹 | 硬滑 | 否 |
| 11 | 浅白 | 硬挺 | 清脆 | 模糊 | 平坦 | 硬滑 | 否 |
| 12 | 浅白 | 蜷缩 | 浊响 | 模糊 | 平坦 | 软粘 | 否 |
| 13 | 青绿 | 稍蜷 | 浊响 | 稍糊 | 凹陷 | 硬滑 | 否 |




基于训练集完全生长的决策树

最终这个预剪枝得到的决策树**只存在一层结点划分**，这样的决策树称为“**决策树桩**”




脐部平坦的训练集均为坏瓜.

脐部凹陷的训练集内多数为好瓜.

脐部稍凹的训练集内一半为好瓜.

## 策略2：后剪枝(*post-pruning*)

**实质：** 决策树生长后处理，合并分枝

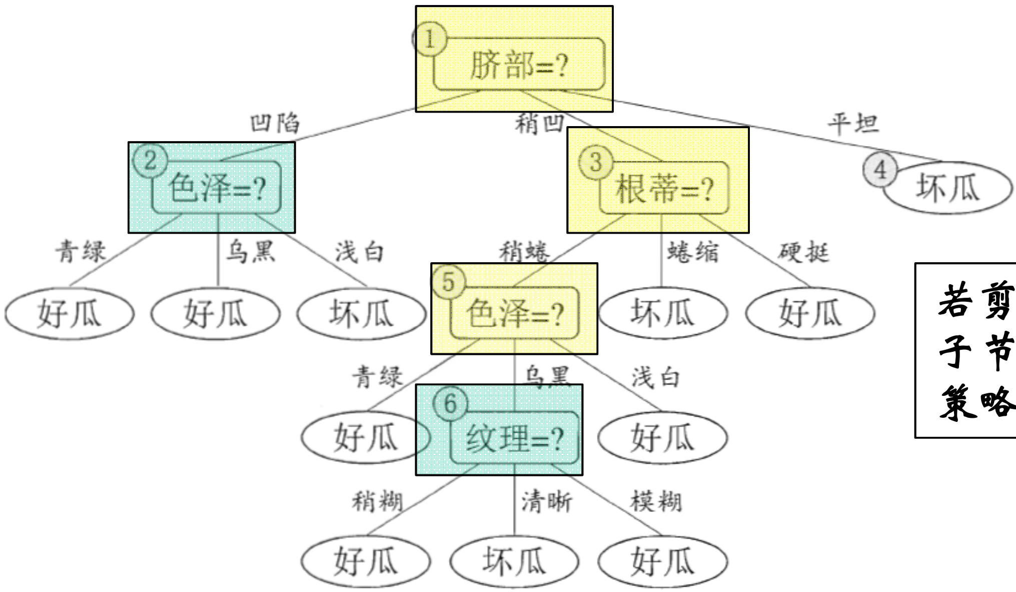
**初始阶段**--决策树按照最大规模生长。

**剪枝阶段**--修剪完全增长的决策树。

自底向上，对非叶子结点进行考察。

若将该结点的子树替换为叶子结点能带来决策树泛化性能的提升，则砍掉该子树，以该结点作为叶子结点。

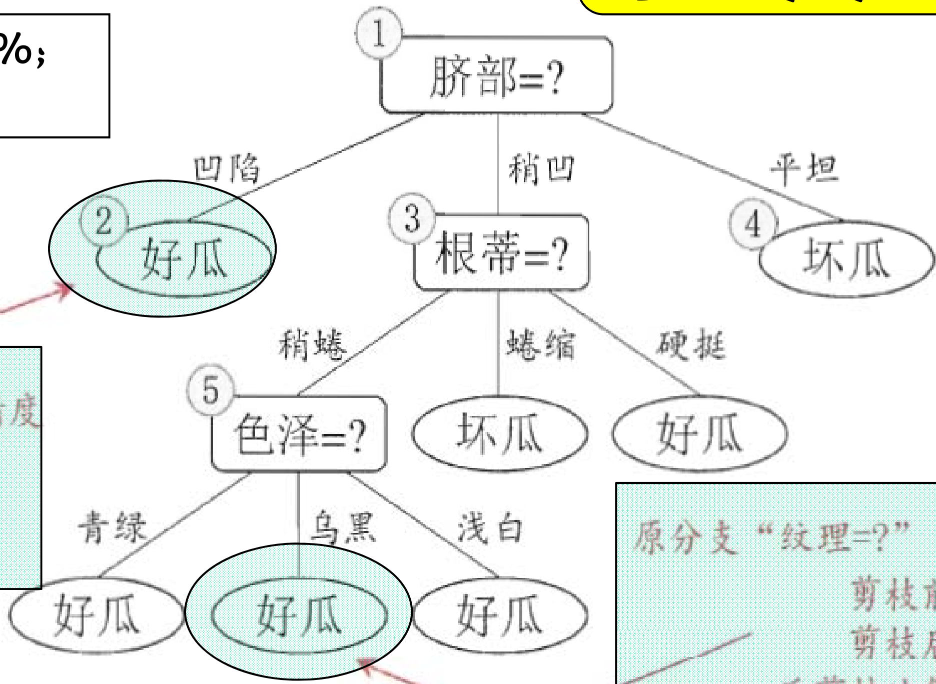
**阶段1. 基于训练集的完整决策树生长。**




若剪去结点⑤的子树，以“好瓜”作为叶子节点类别，验证集精度57.1%.按照简单策略，应剪.但此处较为保守，未剪.

**阶段2. 基于验证集的完整决策树后剪枝。**

对结点⑥减去子树，验证集精度57.1%;
若不剪，验证集精度42.9%




原分支“色泽=?” 验证集精度
剪枝前：57.1%
剪枝后：71.4%
后剪枝决策：剪枝

原分支“纹理=?” 验证集精度
剪枝前：42.9%
剪枝后：57.1%
后剪枝决策：剪枝

# 策略2：后剪枝(*post-pruning*)

## 剪枝规则

<u>例：最小代价与复杂性的折中</u>:

平衡“错误率的增加”与“模型复杂度的降低”

## 决策树的剪枝

设决策树$T$的叶结点数目为$|T|$，叶结点序号$t=1,\ldots,|T|$;训练样本集到达叶结点$t$的样本数为$N_t$,其中第$k$类的样本数为$N_{tk}$, $k=1,\ldots,K$;

叶结点$t$的经验熵$H_t(T)$，控制参数$\alpha \geq 0$

$$H_t(T) = -\sum_{k=1}^{K} \frac{N_{tk}}{N_t} \log \frac{N_{tk}}{N_t}$$

**决策树T关于训练样本的拟合误差：**

$$C(T) = \sum_{t=1}^{|T|} N_t H_t(T) = -\sum_{t=1}^{|T|} N_t \sum_{k=1}^{K} \frac{N_{tk}}{N_t} \log \frac{N_{tk}}{N_t} = -\sum_{t=1}^{|T|} \sum_{k=1}^{K} N_{tk} \log \frac{N_{tk}}{N_t}$$

″**模型关于训练样本的拟合误差**″+″**模型的复杂度**″=″**模型的损失函数**″

$$C_\alpha(T) = C(T) + \alpha|T|$$

决策树的后剪枝，就是在给定$\alpha$的前提下，选择具有最小$C_\alpha(T)$的子树。

**决策树$T$的后剪枝算法**

**输入**：生成算法产生的整棵树$T$，参数$\alpha$

**输出**：对树$T$修剪，得到的子树$T_\alpha$

**步骤**：

$STEP1.$计算每个结点(不只是叶结点)的经验熵。

$STEP2.$递归地从树的叶结点向上回溯。

设**一组叶结点**回溯到其父结点**之前**、**之后**的整体树分别为$T_B$和$T_A$;对应的损失函数值分别为

$$C_\alpha(T_B)=C(T_B)+\alpha|T_B| \quad C_\alpha(T_A)=C(T_A)+\alpha|T_A|$$

若$C_\alpha(T_A)\le C_\alpha(T_B)$，则剪枝，将**叶结点**的父结点作为新的叶结点

$STEP3.$返回$STEP2$，直到不能继续为止，得到损失函数最小的子树$T_\alpha$

## (3)关于“剪枝”的讨论:

- **预剪枝**可能过早终止决策树的生长,
  存在欠拟合风险.

- **后剪枝技术**倾向于产生更好的结果
  根据完全生长的决策树作出剪枝决策,需要更多时间开销.
  欠拟合的风险小,泛化性能优于预剪枝决策树.

- “先剪枝”与“后剪枝”结合

# PART1.决策树

1.1 非数值特征(*nonmetric features*)

1.2 决策树

1.3 过学习与决策树的剪枝

# PART2.以决策树为个体模型的集成学习

1 Bootstrap Aggregating(bagging)

个体模型是决策树，也可以是是其它分类模型

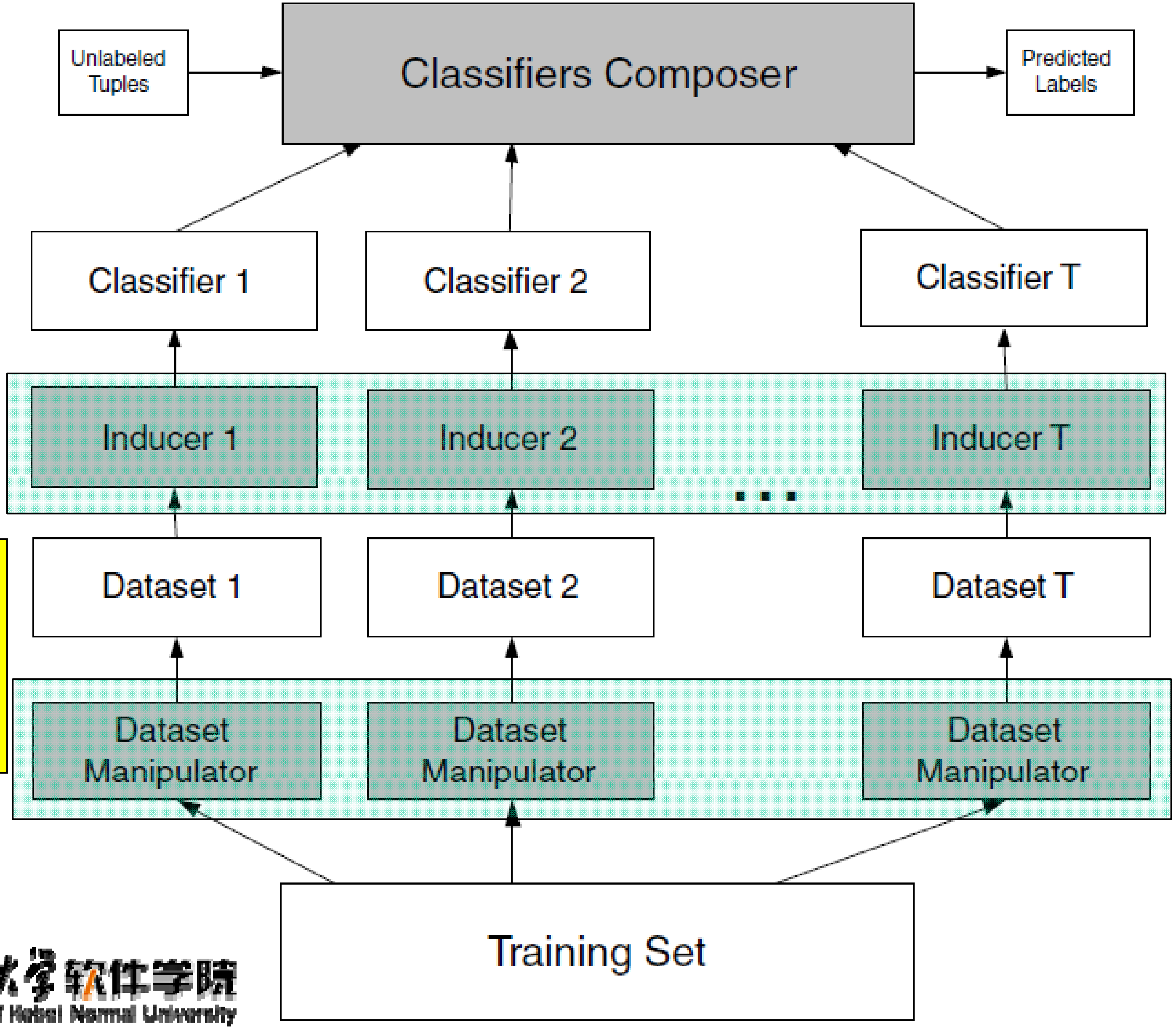
2 Random Forest(RF)

个体模型是决策树

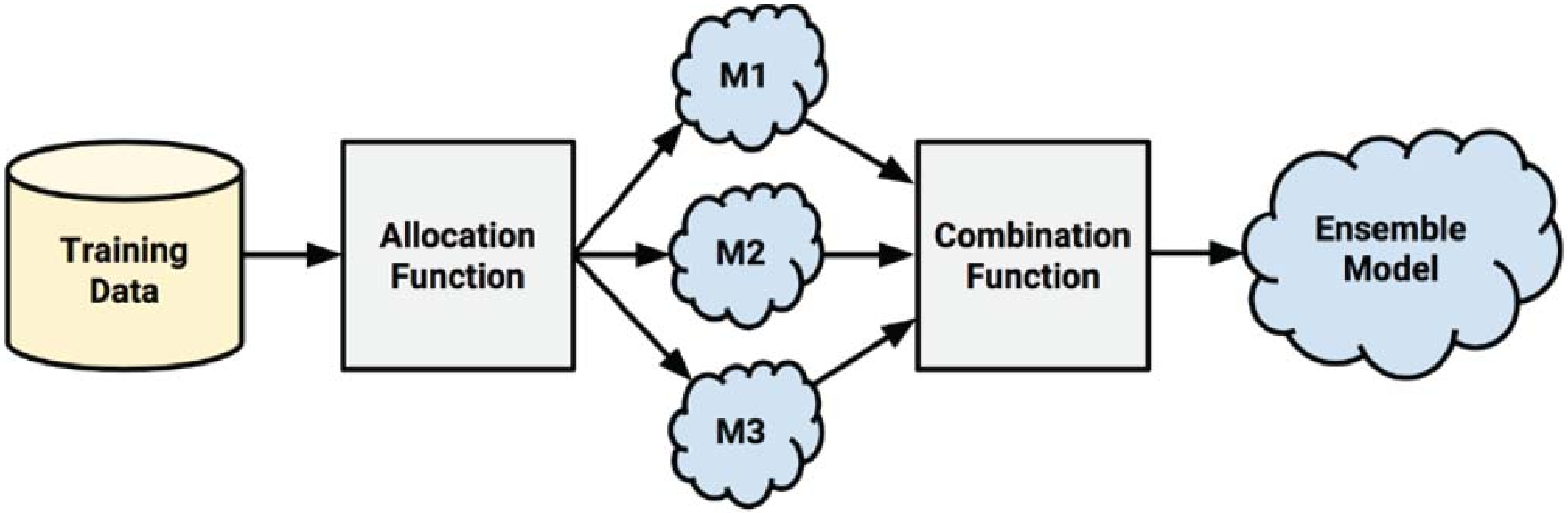
分类—简单投票；回归—简单平均

## 例：面向分类任务的独立个体模型并行集成

个体模型的学习可以并行完成.

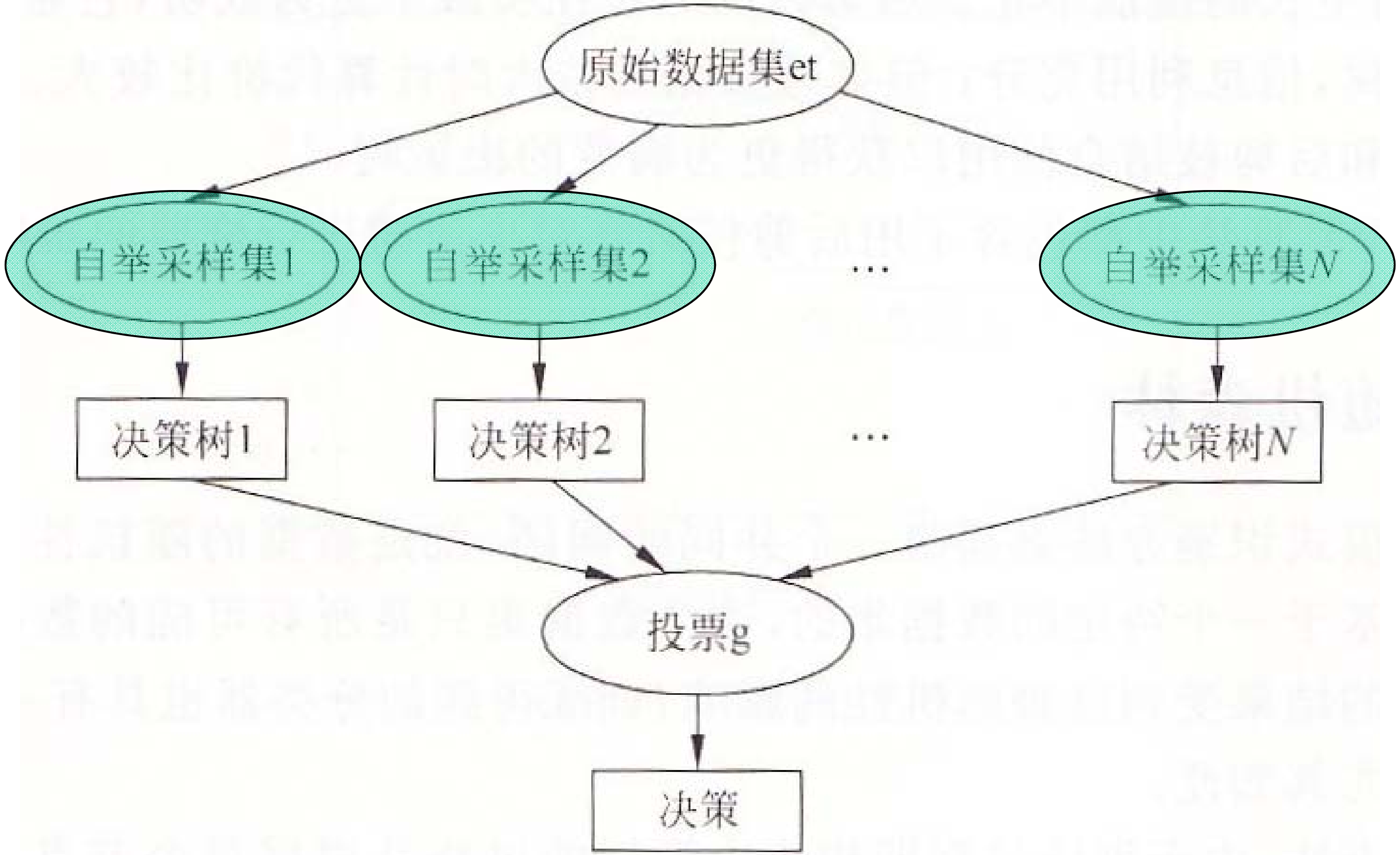



由原始的训练集生成的用于个体模型学习的训练集可以是部分重叠的、也可以是互斥的


Training Data
Allocation Function
M1
M2
M3
Combination Function
Ensemble Model

## 1. Bagging

- 基于样本集的自举采样集，构建多棵决策树；
- 多棵决策树投票决策



## *Bagging*（*Bootstrap AGGregatING*）算法

**输入：** 训练样本集$D=\{(x_i,y_i), i=1,\dots,m\}$；

监督式基学习器算法$\ell$；基学习器的数目$N$

**模型的学习阶段：**

初始化**基学习模型的集合**$E$为空集.

**Do t = 1,…,N**

由数据集$D$自举重采样得容量为$m$的数据集$D_t$;

基于数据集$D_t$，调用基学习器算法$\ell$，得个体模型$h_t(x)$;

更新$E$: $E \leftarrow E \cup \{h_t(x)\}$

**End**

**模型的使用阶段：**

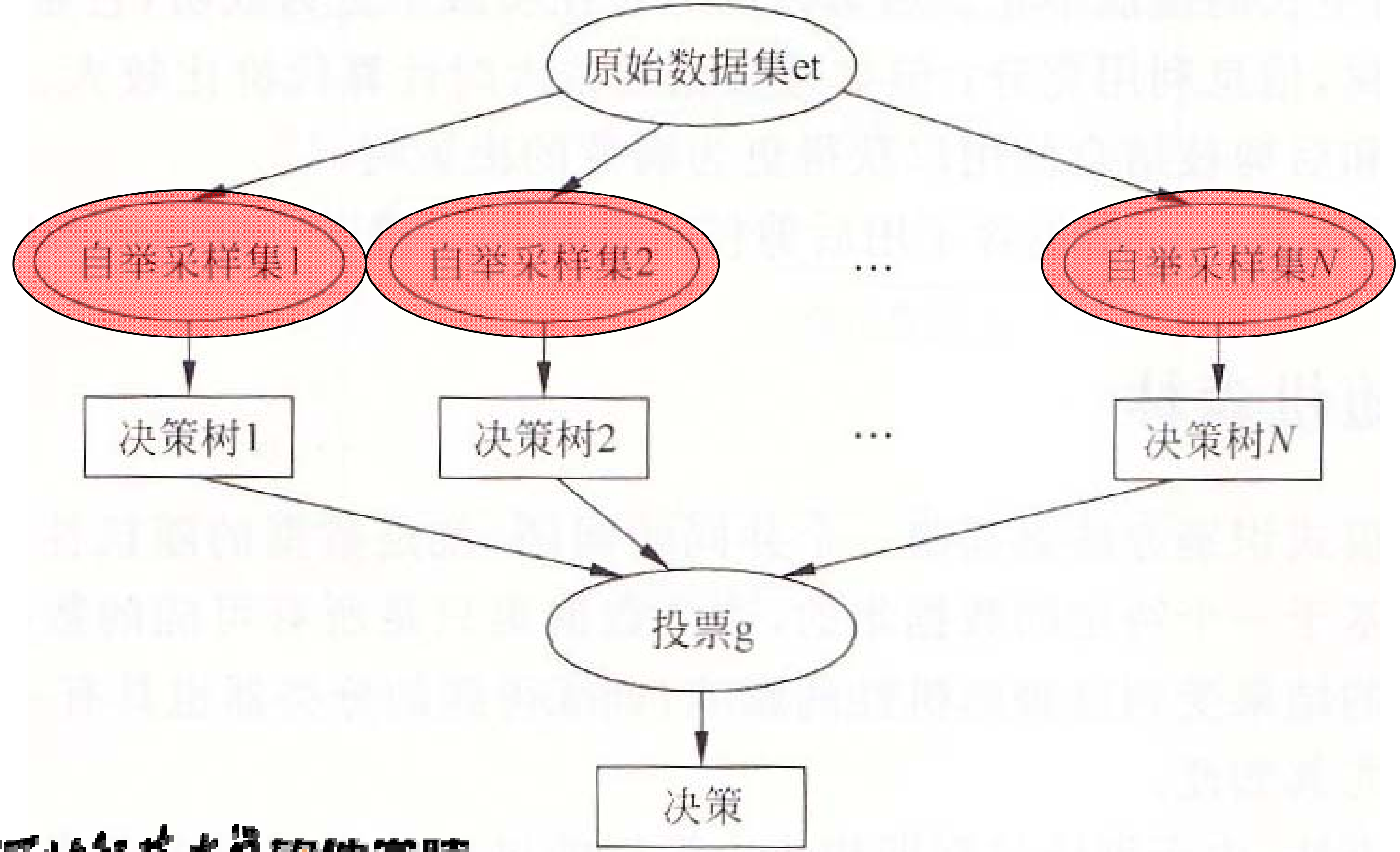
$$\text{对于任意观测}x\text{，集成预测}\begin{cases}\text{若为实值函数回归，则}\hat{y}=\dfrac{1}{N}\sum_{t=1}^{N}\hat{h}_t(x)\\ \text{若为分类，则}\hat{y}=\underset{j\in\{1,2,\dots,|Y|\}}{\arg\max}\sum_{t=1}^{N}I\left(\hat{h}_t(x)=j\right)\end{cases}$$

**输出：**$\hat{y}$

注意：$\hat{h}_t(x)$为个体模型$h_t(\cdot)$在$x$处产生的预测输出.

## 2. 随机森林(bagging + 扰动版的基学习器)

- 基于样本集自举重采样+特征子集随机抽取，构建多棵决策树，组成决策树的“森林”；
- 多棵决策树投票决策




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

## 2. 基本步骤

关键：样本采样、特征采样，各树彼此独立；投票无偏。

(1) 模型学习--构造N棵决策树(不剪枝)。

每棵树的构建，需要：

A--对样本数据进行"自举法(bootstrapping,或自助法)"重采样，得到1个样本集

有放回地随机抽取

出发点：使用相同特征空间的不同数据点

B—为该样本集，生成备选特征。

从特征集内随机抽取p个特征，形成该决策树学习所需要的特征子集。

(2) 模型的使用--决策：输入未知样本，得到多个决策树的输出：分类--投票，胜者为王；回归—平均，得输出。

## *Random Forest*算法

**输入：** 训练样本集$\boldsymbol{D}=\{(\boldsymbol{x}_i,\boldsymbol{y}_i),\boldsymbol{i}=1,\ldots,\boldsymbol{m}\}$，其中$\boldsymbol{x}_i=[x_{i1},\ldots,x_{id}]^T$；

监督式基学习器算法$\ell$；基学习器的数目$\boldsymbol{N}$

特征子集的特征容量$\boldsymbol{p}$

注意：$\hat{h}_t(\boldsymbol{x})$为个体模型$h_t(\cdot)$在$\boldsymbol{x}$处产生的预测输出.

**模型的学习阶段：**

初始化**基学习模型的集合**$\boldsymbol{E}$为空集.

**Do t = 1,…,N**

由原始的$\boldsymbol{d}$个特征随机抽取得$\boldsymbol{p}$个特征组成特征子集$\boldsymbol{F}_t$；

基于特征子集$\boldsymbol{F}_t$，由数据集$\boldsymbol{D}$自举重采样得容量为$\boldsymbol{m}$的数据集$\boldsymbol{D}_t$；

基于数据集$\boldsymbol{D}_t$，调用基学习器算法$\ell$，得个体模型$h_t(\boldsymbol{x})$；

更新$\boldsymbol{E}$: $\boldsymbol{E}\leftarrow\boldsymbol{E}\cup\{h_t(\boldsymbol{x})\}$

**End**

**模型的使用阶段：**

对于任意观测$\boldsymbol{x}$，集成预测
$$\begin{cases}\text{若为实值函数回归，则}\hat{\boldsymbol{y}}=\dfrac{1}{\boldsymbol{N}}\sum\limits_{t=1}^{N}\hat{h}_t(\boldsymbol{x})\\ \text{若为分类，则}\hat{\boldsymbol{y}}=\underset{j\in\{1,2,\ldots,|Y|\}}{\mathbf{argmax}}\sum\limits_{t=1}^{N}I\left(\hat{h}_t(\boldsymbol{x})=\boldsymbol{j}\right)\end{cases}$$

**输出：** $\hat{y}$

https://www.stat.berkeley.edu/~breiman/
http://statistics.berkeley.edu/memory/leo-breiman
https://cosx.org/2012/02/what-is-the-stat-dept-25-years-from-now/

**小故事: 老当益壮的李奥·布瑞曼**

李奥·布瑞曼 (Leo Breiman, 1928–2005) 是二十世纪伟大的统计学家. 他在二十世纪末公开宣称, 统计学界把统计搞成了抽象数学, 这偏离了初衷, 统计学本该是关于预测、解释和处理数据的学问. 他自称与机器学习走得更近, 因为这一行是在处理有挑战的数据问题. 事实上, 布瑞曼是一位卓越的机器学习学家, 他不仅是CART 决策树的作者, 还对集成学习有三大贡献: Bagging、随机森林以及关于 Boosting 的理论探讨. 有趣的是, 这些都是在他 1993 年从加州大学伯克利分校统计系退休后完成的.

布瑞曼早年在加州理工学院获物理学士学位，然后打算到哥伦比亚大学念哲学，但哲学系主任告诉他，自己最优秀的两个博士生没找到工作，于是布瑞曼改学数学，先后在哥伦比亚大学和加州大学伯克利分校获得数学硕士、博士学位. 他先是研究概率论，但在加州大学洛杉矶分校(UCLA)做了 7 年教授后他厌倦了概率论，于是主动辞职. 为了向概率论告别，辞职后他把自己关在家里半年写了本关于概率论的书，然后他到工业界做了 13 年咨询，再回到加州大学伯克利分校统计系做教授. 布瑞曼的经历极为丰富，他曾在 UCLA 学术假期间主动到联合国教科文组织工作，被安排到非洲利比里亚统计失学儿童数. 他是一位业余雕塑家，甚至还与人合伙在墨西哥开过制冰厂. 他自认为一生最重要的研究成果——随机森林，是 70 多岁时做出来的.

# 思考题

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

1. 什么是决策树?

   决策树模型的叶子结点与特征空间对应关系?
2. 如何利用到达决策树某结点处的训练集度量该结点的不纯度?(三种典型的结点不纯度度量方式)
3. ID3,C4.5,CART三种典型决策树的算法实现步骤?
4. 三种决策树模型中，非叶子结点所用的特征是采用何种规则进行选择的?给出具体的选择方式.
5. 哪种决策树模型还可用于实值函数回归?若用于回归，如何生成预测结果?
6. 以决策树作为个体模型，采用BAGGING以Random Forest还可实现个体模型的并行集成.给定已知类别标记的训练集，请对两种集成模型的实现步骤进行详细描述.