天善智能
TIANSHAN SOFT

第9章 用决策树预测电信用户离网

《Python数据科学：全栈技术详解》

讲师：Ben

自我介绍

- 天善商业智能和大数据社区　　讲师 –Ben
- 天善社区 ID - Ben_Chang
- https://www.hellobi.com – 学习过程中有任何相关的问题都可以提到技术社区数据挖掘版块。

主要内容

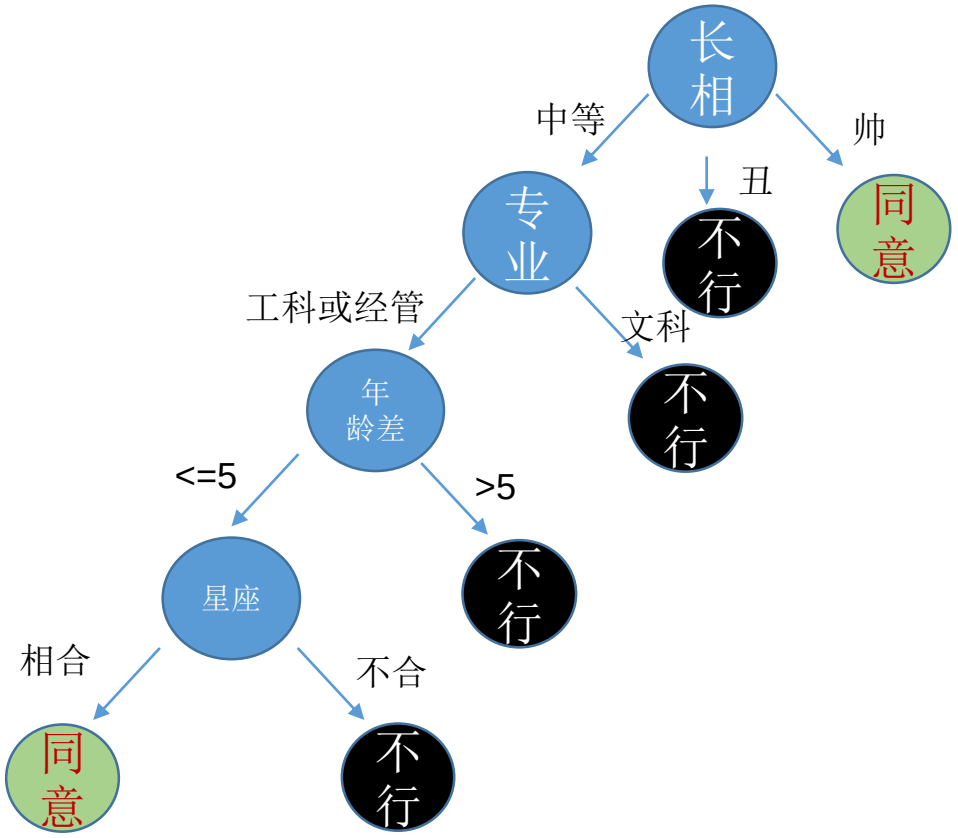
- 决策树建模思路
- Quinlan系列决策树(ID3、C4.5、C8.0)建模原理
- CART建模原理
- 模型修剪
- 模型评估
- 随机森林与组合算法

天善智能
TIANSHAN SOFT

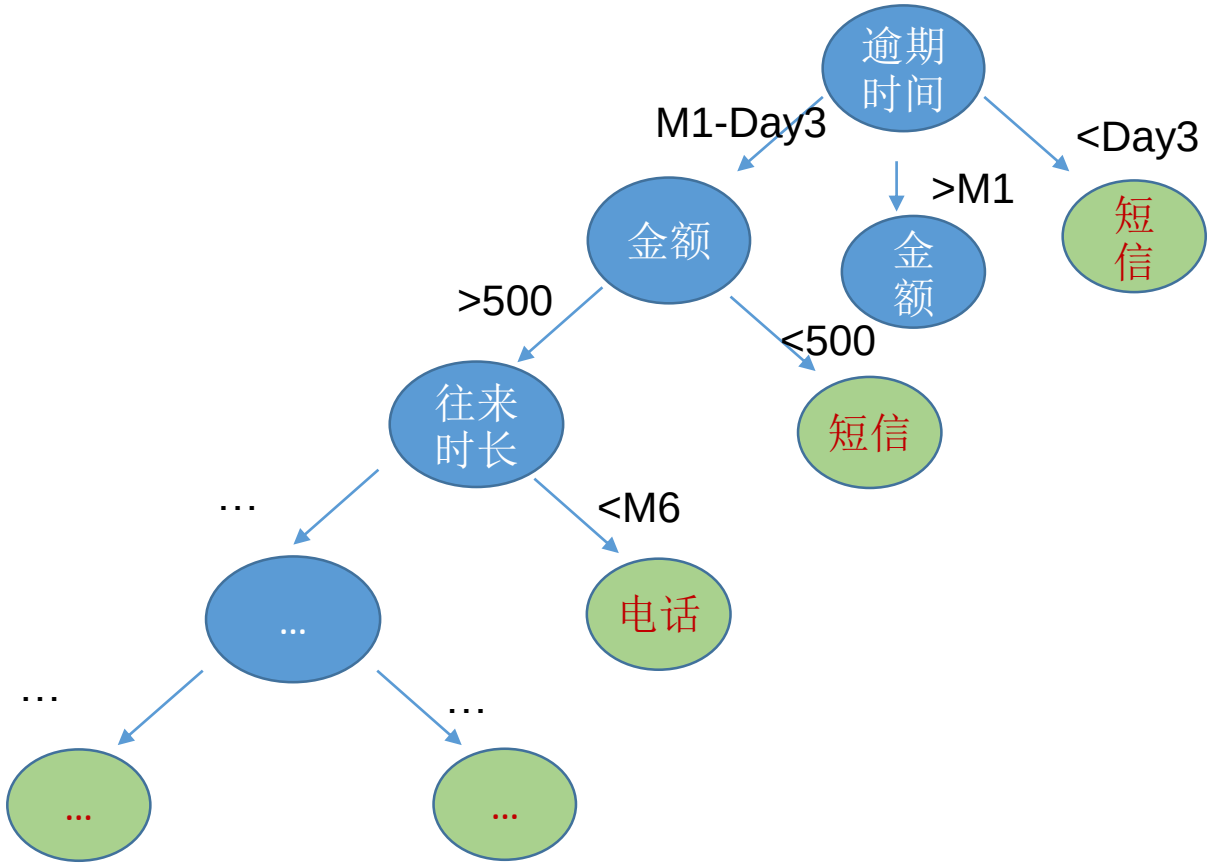
决策树建模思路

生活中的决策行为

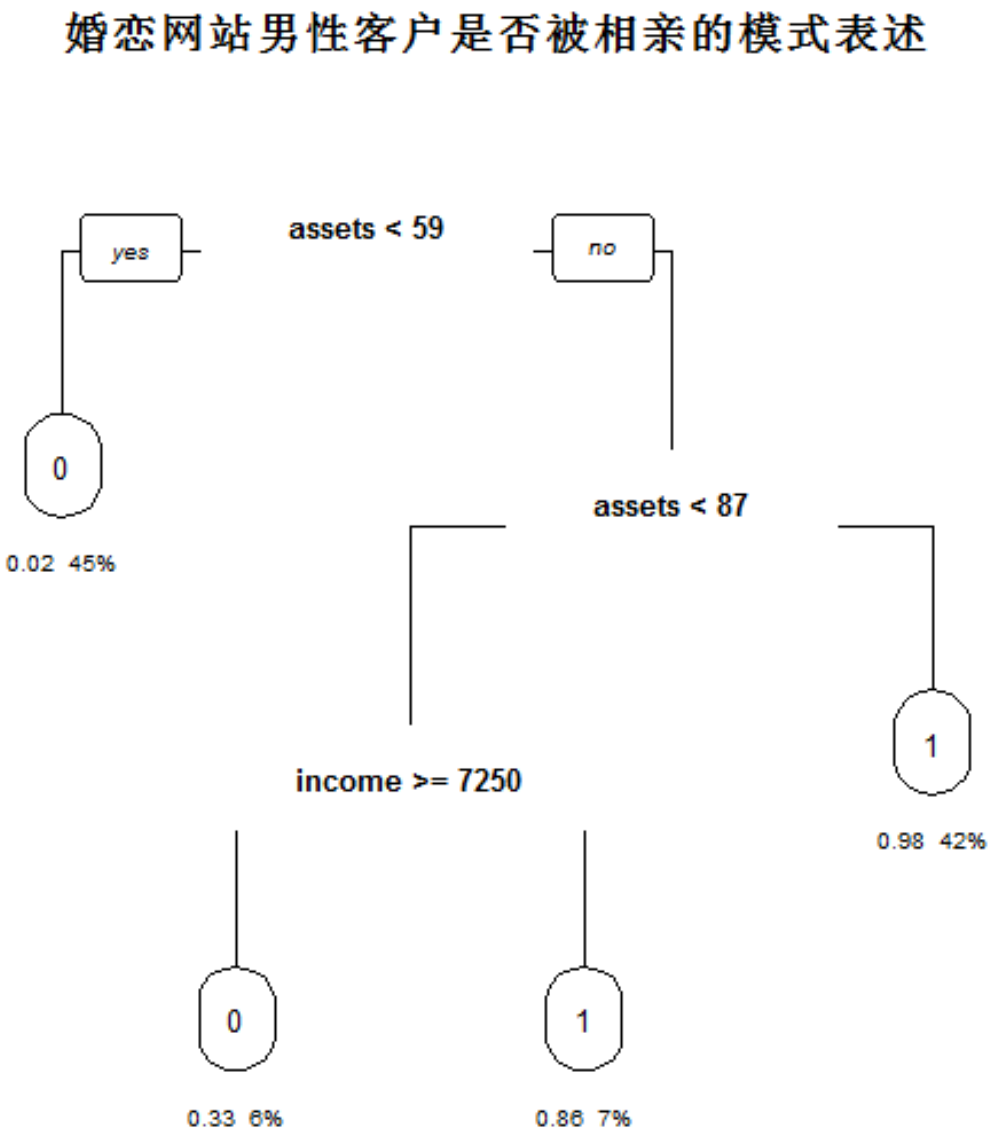
某大学BBS鹊桥板块女生相亲决策树

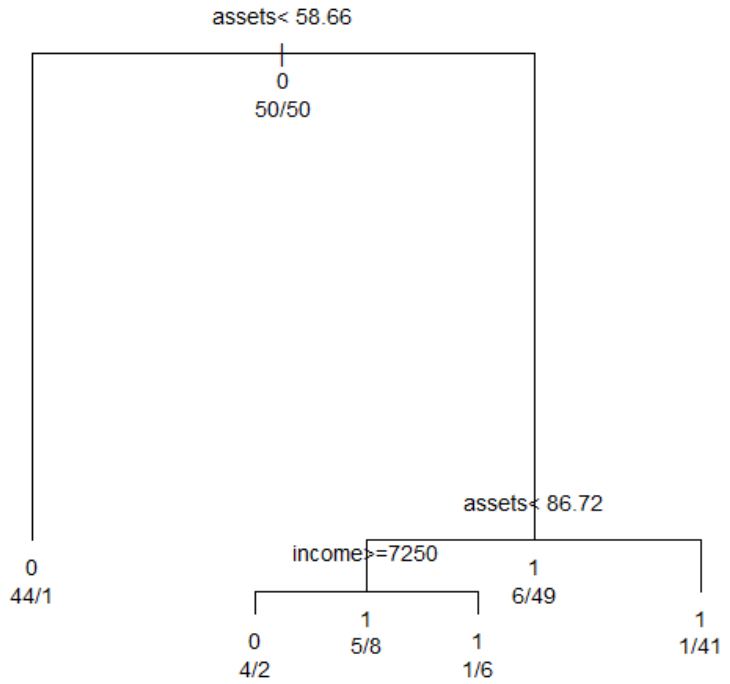


用决策树做催收策略模型



婚恋网站男性客户是否被相亲的模式表述



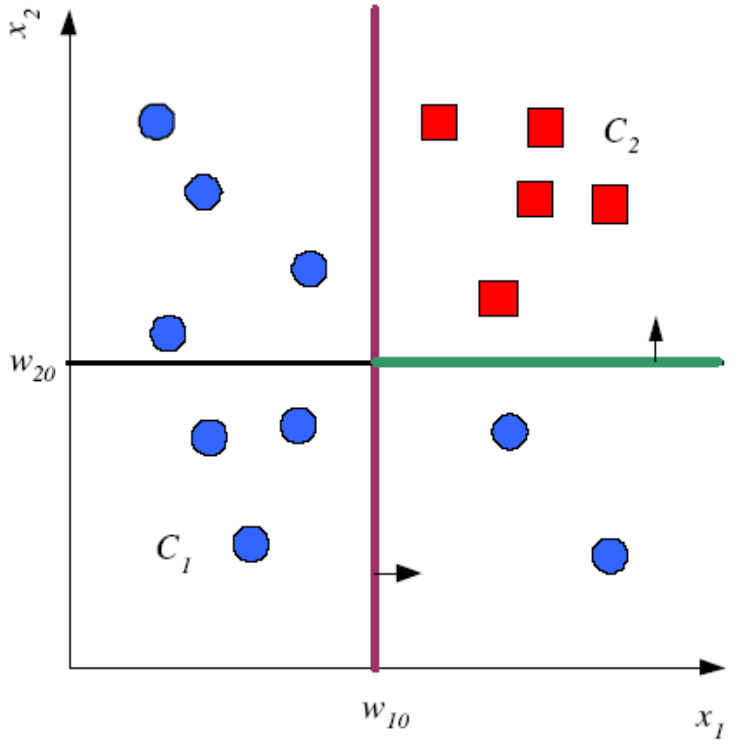


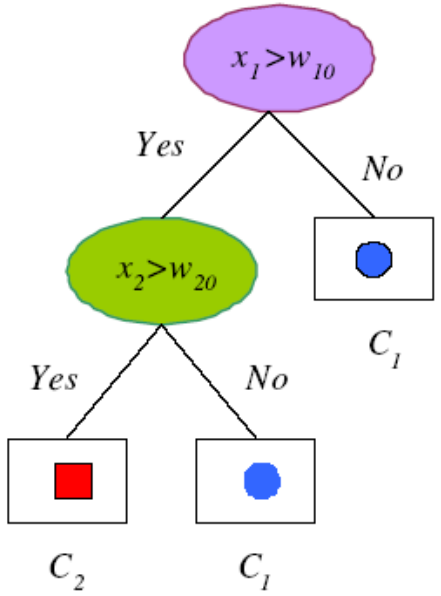
决策树算法概述

- 70年代后期至80年代初期，Quinlan开发了ID3算法（迭代的二分器）；后来Quinlan改进了ID3算法，称为C4.5算法，最近又发布了C8.0；
- 1984年，多位统计学家在著名的《Classification and regression tree》书里提出了CART算法；
- ID3和CART几乎同期出现，引起了研究决策树算法的旋风，至今已经有多种算法被提出。

什么是决策树？

- 决策树是以树型结构组织的规则集合
  - 易于理解是如何预测
  - 易于构建和可视化
  - 简约表示和执行能力
  - 是一个有效的数据挖掘技术





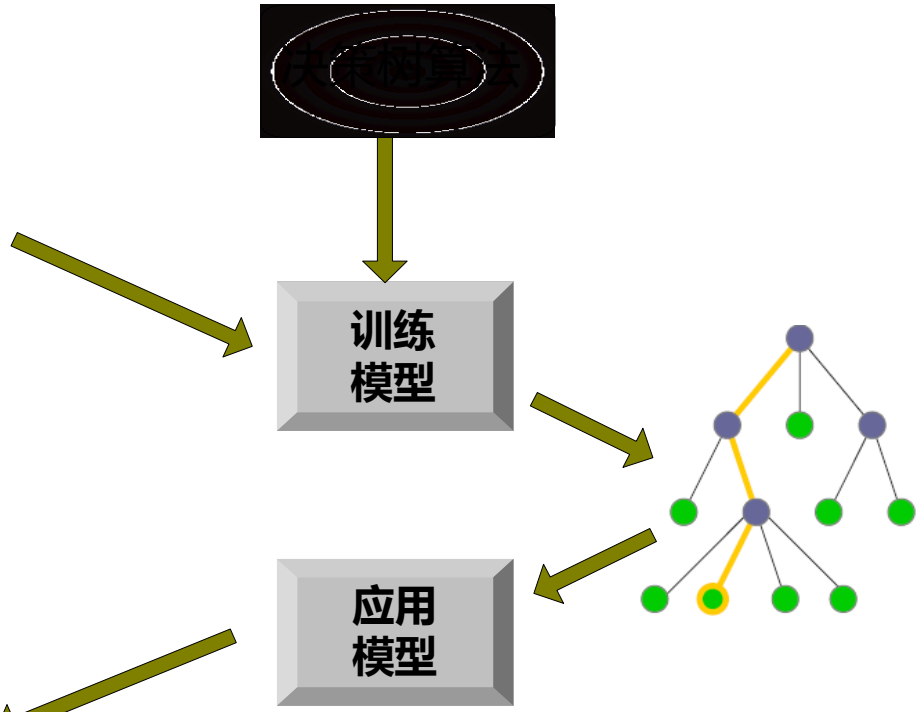
训练和应用决策树

训练集

客户ID	曾经逾期	开卡时长	交易趋势	无交易月	曾破产?	违约标志
张三	有	3个月	下降40%	3	无	1
李四	无	6个月	上升30%	1	无	0
…	…	…	…		…	…

预测集

客户ID	曾经逾期	开卡时长	交易趋势	无交易月	曾破产?	违约违约
A	无	4	下降10%	0	无	?
B	无	6	下降70%	3	有	?



训练集从何来？

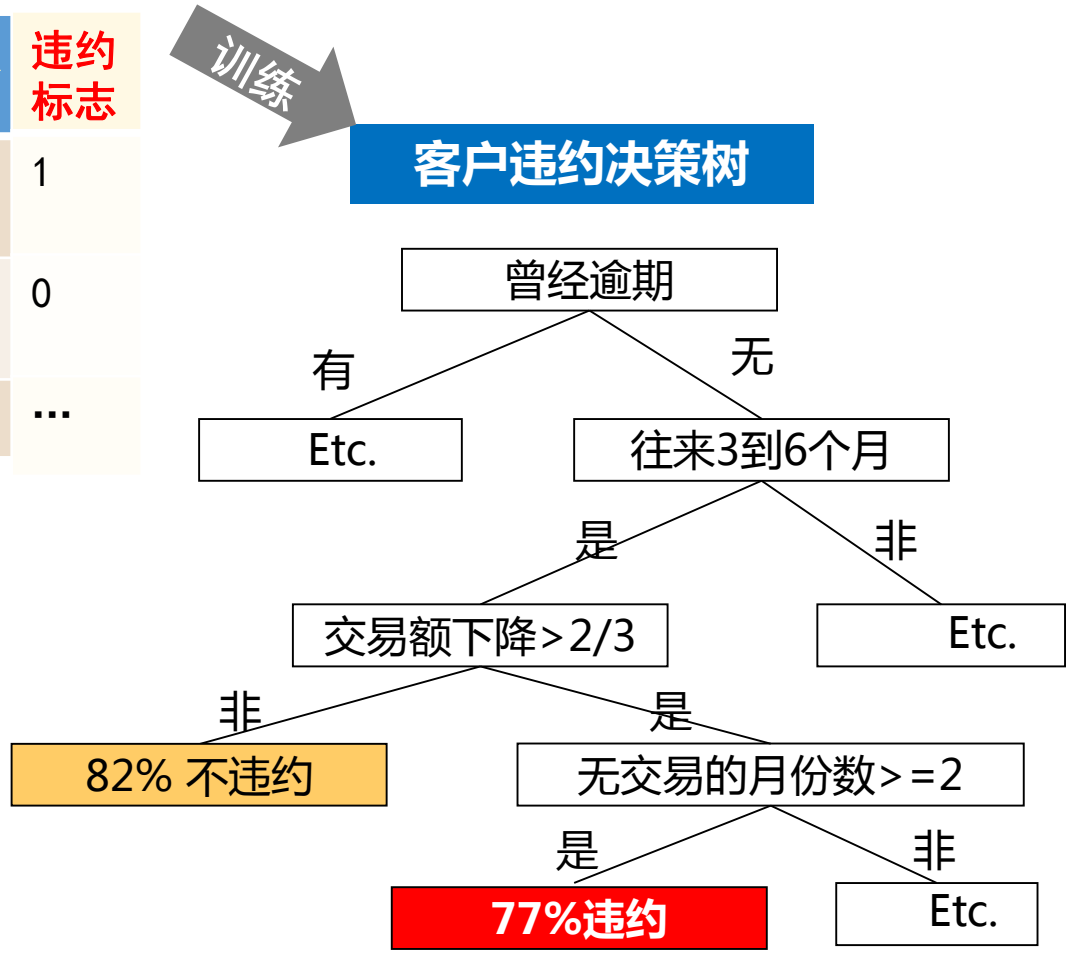
从历史数据中产生训练集

通过观察**3月-8月**的客户行为及其**9月-11月**的违约情况

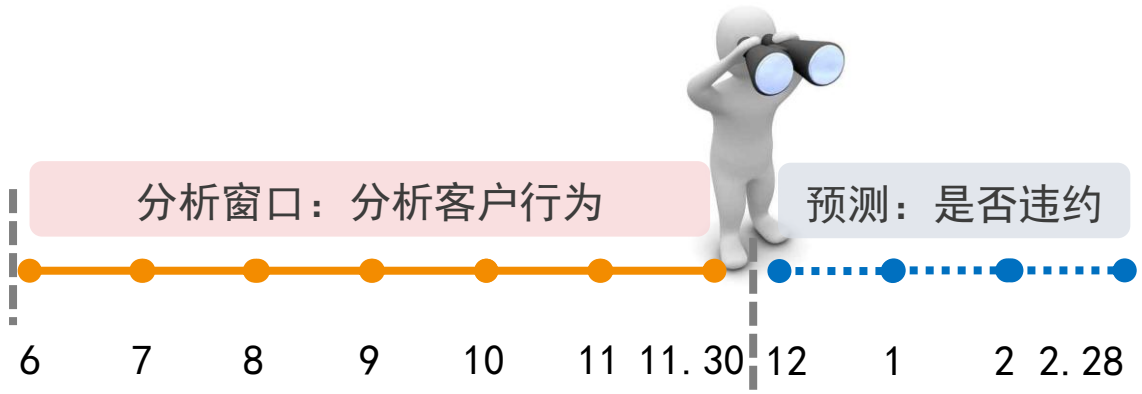
客户ID	曾经逾期	开卡时长	交易趋势	无交易月	曾破产?	违约标志
张三	有	3个月	下降40%	3	无	1
李四	无	6个月	上升30%	1	无	0
…	…	…	…		…	…

训练决策树

客户ID	曾经逾期	开卡时长	交易趋势	无交易月	曾破产?	违约标志
张三	有	3个月	下降40%	3	无	1
李四	无	6个月	上升30%	1	无	0
…	…	…	…		…	…



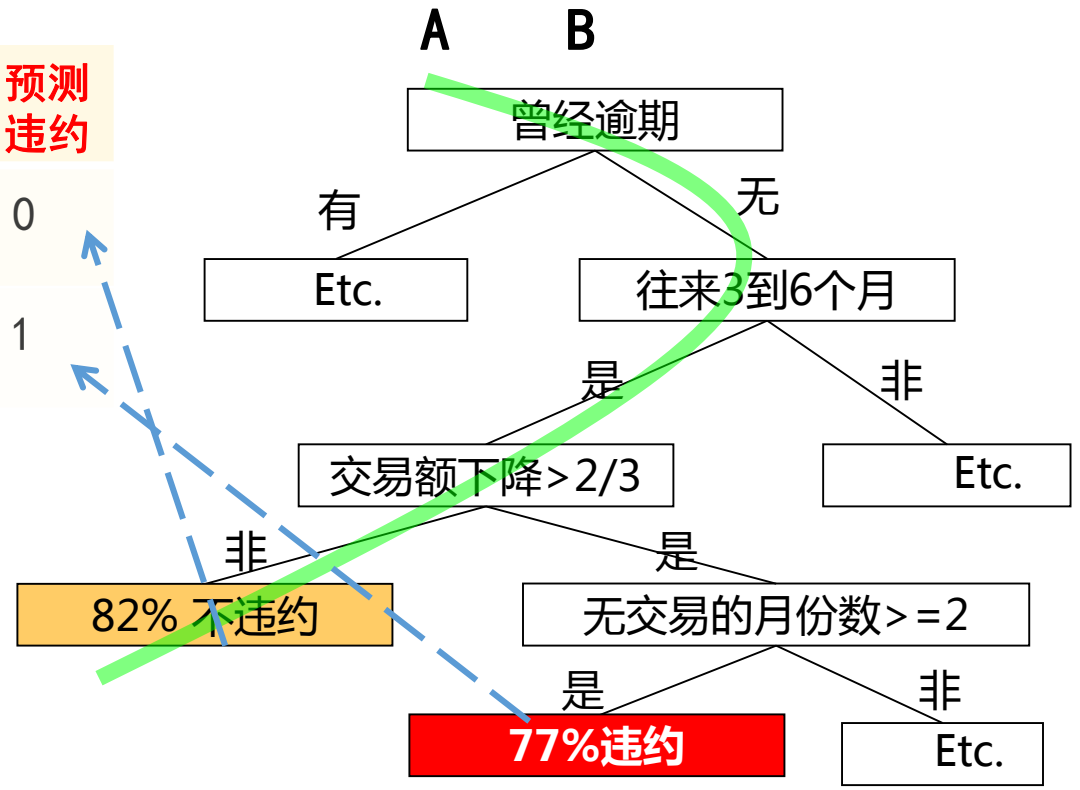
产生预测集



客户ID	曾经逾期	开卡时长	交易趋势	无交易月	曾破产?	预测违约
A	无	4	下降10%	0	无	?
B	无	6	下降70%	3	有	?

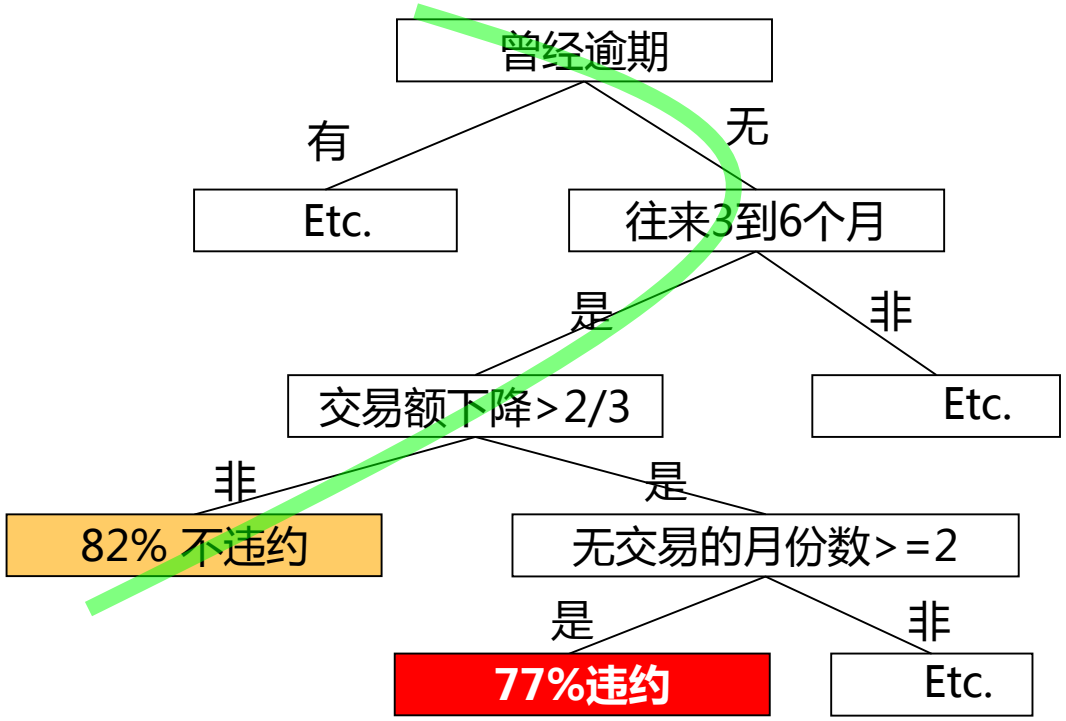
预测性数据挖掘示例——应用示例

客户ID	曾经逾期	开卡时长	交易趋势	无交易月	曾破产?	预测违约
A	无	4	下降10%	0	无	0
B	无	6	下降70%	3	有	1



决策树的路径

- 决策树的一条路径解释了预测
  - 是对数据的探索
  - 对数据轮廓的描述
  - 能进行预测与分类
  - 了解哪些变量最重要
  - 能发现意料之外的模式



天善智能
TIANSHAN SOFT

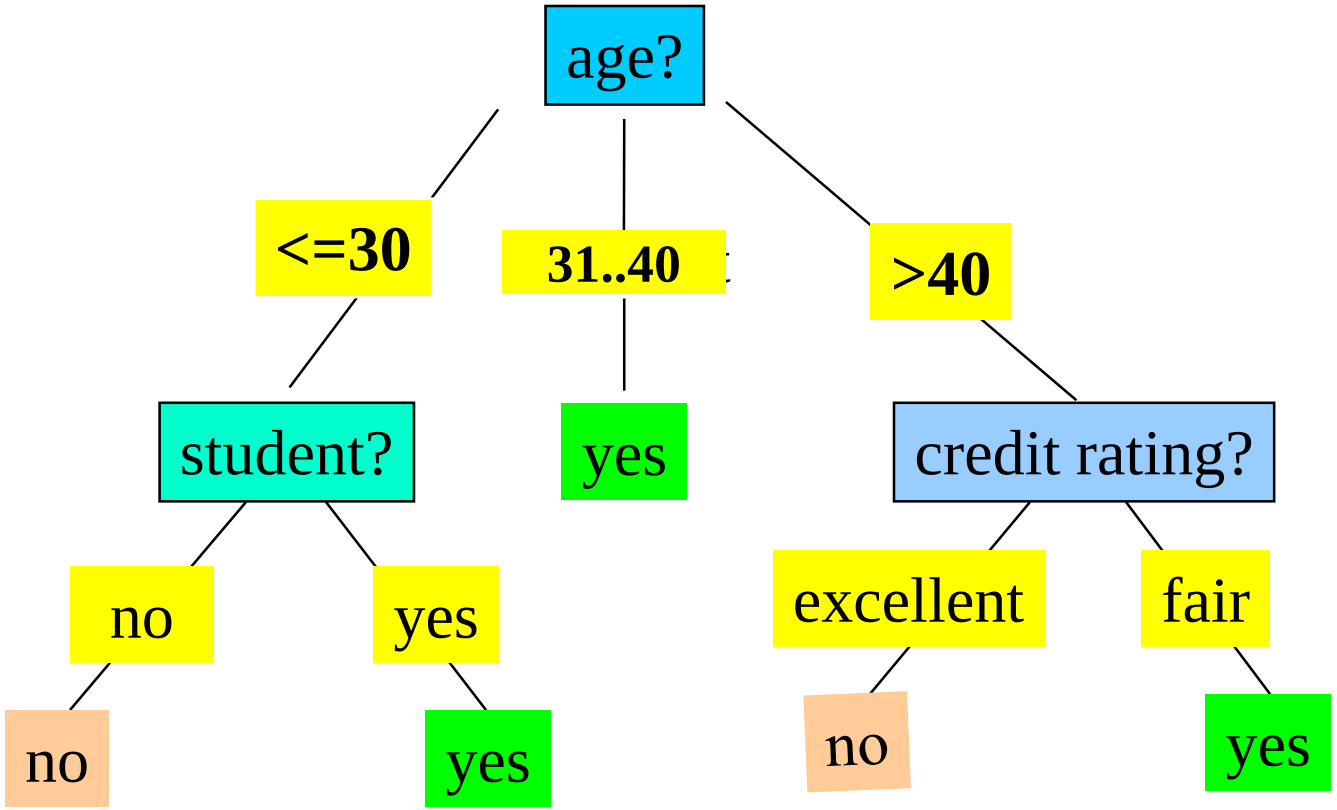
决策树建模基本原理

例子：训练数据

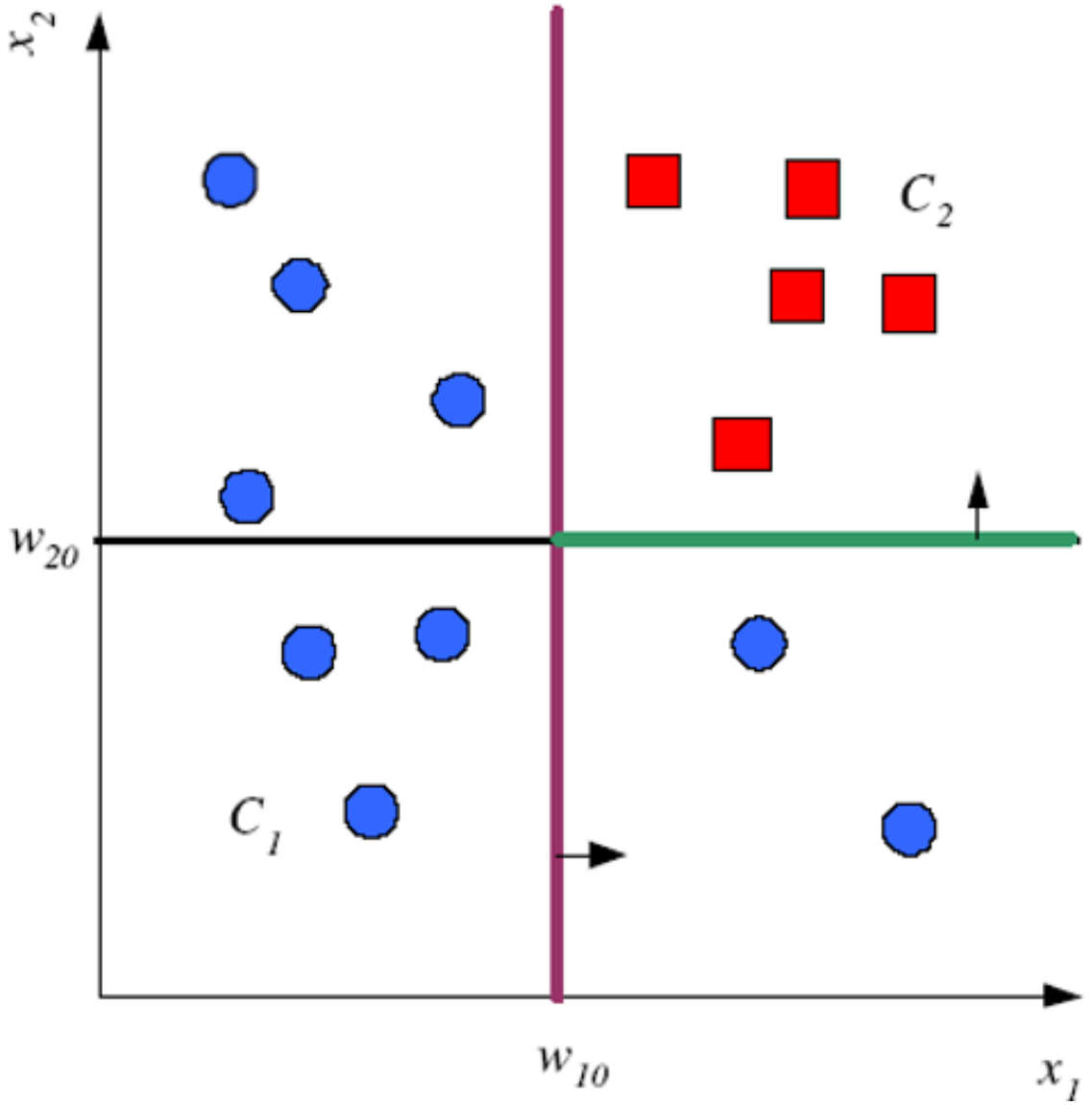
AllElectronics公司客户购买电脑数据

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

例子：期待输出的结果


age?
<=30
31..40
>40
student?
yes
credit rating?
no
yes
excellent
fair
no
yes
no
yes

寻找最纯净的分组方法



算法的核心问题

- 该按什么样的次序来选择变量（属性）？
- 最佳分离点（连续的情形）在哪儿？

拆分规则

Age:

表 - age * buys_computer				
		buys_computer		
		no	yes	合计
age				
31…40	频数	0	4	4
<=30	频数	3	2	5
>40	频数	2	3	5
合计	频数	5	9	14

Δentropy（ID3、C4.5、C50）	ΔGini（CART）	Logworth（CHAID）
0.246	0.102	1.30

Income:

表 - income * buys_computer				
		buys_computer		
		no	yes	合计
income				
high	频数	2	2	4
low	频数	1	3	4
medium	频数	2	4	6
合计	频数	5	9	14

0.029	0.016	0.74

天善智能
TIANSHAN SOFT

Quinlan系列决策树建模原理
ID3、C4.5、C5.0

ID3决策树原理介绍

➢ID3的建模步骤：

一、建树

- 选择最有解释力度的变量
- 对于每个变量选择最优分割点

二、剪树

- 前向剪枝：控制生成树的规模
- 后项剪枝：删除没有意义的分组

ID3算法

- 信息增益计算

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

$$Gain(A) = Info(D) - Info_A(D)$$

ID3输入为分类变量：信息增益计算

$$Info(D) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.940 \text{ 位}$$

$$Info_{age}(D) = \frac{5}{14}\times\left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right) + \frac{4}{14}\times\left(-\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4}\right)$$
$$+ \frac{5}{14}\times\left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right)$$
$$= 0.694 \text{ 位}$$

$$Gain(age) = Info(D) - Info_{age}(D) = 0.940 - 0.694 = 0.246 \text{ 位}$$

Age属性的信息增益最高，故首先选择这个变量

ID3输入为分类变量：信息增益计算

Training data tuples from the *AllElectronics* customer database.

RID	age	income	student	credit_rating	Class: buys_computer
1	<=30	high	no	fair	no
2	<=30	high	no	excellent	no
3	31 . . . 40	high	no	fair	yes
4	>40	medium	no	fair	yes
5	>40	low	yes	fair	yes
6	>40	low	yes	excellent	no
7	31 . . . 40	low	yes	excellent	yes
8	<=30	medium	no	fair	no
9	<=30	low	yes	fair	yes
10	>40	medium	yes	fair	yes
11	<=30	medium	yes	excellent	yes
12	31 . . . 40	medium	no	excellent	yes
13	31 . . . 40	high	yes	fair	yes
14	>40	medium	no	excellent	no

$$I(s_1, s_2) = I(9, 5) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.940$$

Entropy

ID3输入为分类变量：信息增益计算

下一步，需要计算每个属性的期望信息需求。从属性 *age* 开始。需要对 *age* 的每个类考察 *yes* 和 *no* 元组的分布。对于 *age* 的类 “*youth*”，有 2 个 *yes* 元组，3 个 *no* 元组。对于类 “*middle_aged*”，有 4 个 *yes* 元组，0 个 *no* 元组。对于类 “*senior*”，有 3 个 *yes* 元组，2 个 *no* 元组。使用（8.2）式，如果元组根据 *age* 划分，则对 D 中的元组进行分类所需要的期望信息为：

For *age* = “*<=30*”:

$s_{11} = 2 \quad s_{21} = 3 \quad I(s_{11}, s_{21}) = 0.971$

For *age* = “*31 . . . 40*”:

$s_{12} = 4 \quad s_{22} = 0 \quad I(s_{12}, s_{22}) = 0$

For age = “*>40*”:

$s_{13} = 3 \quad s_{23} = 2 \quad I(s_{13}, s_{23}) = 0.971$

$$Info_{age}(D) = \frac{5}{14} \times \left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} \right) + \frac{4}{14} \times \left(-\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4} \right) + \frac{5}{14} \times \left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} \right)$$

$$= 0.694 \text{ 位}$$

ID3输入为分类变量：构造第一层

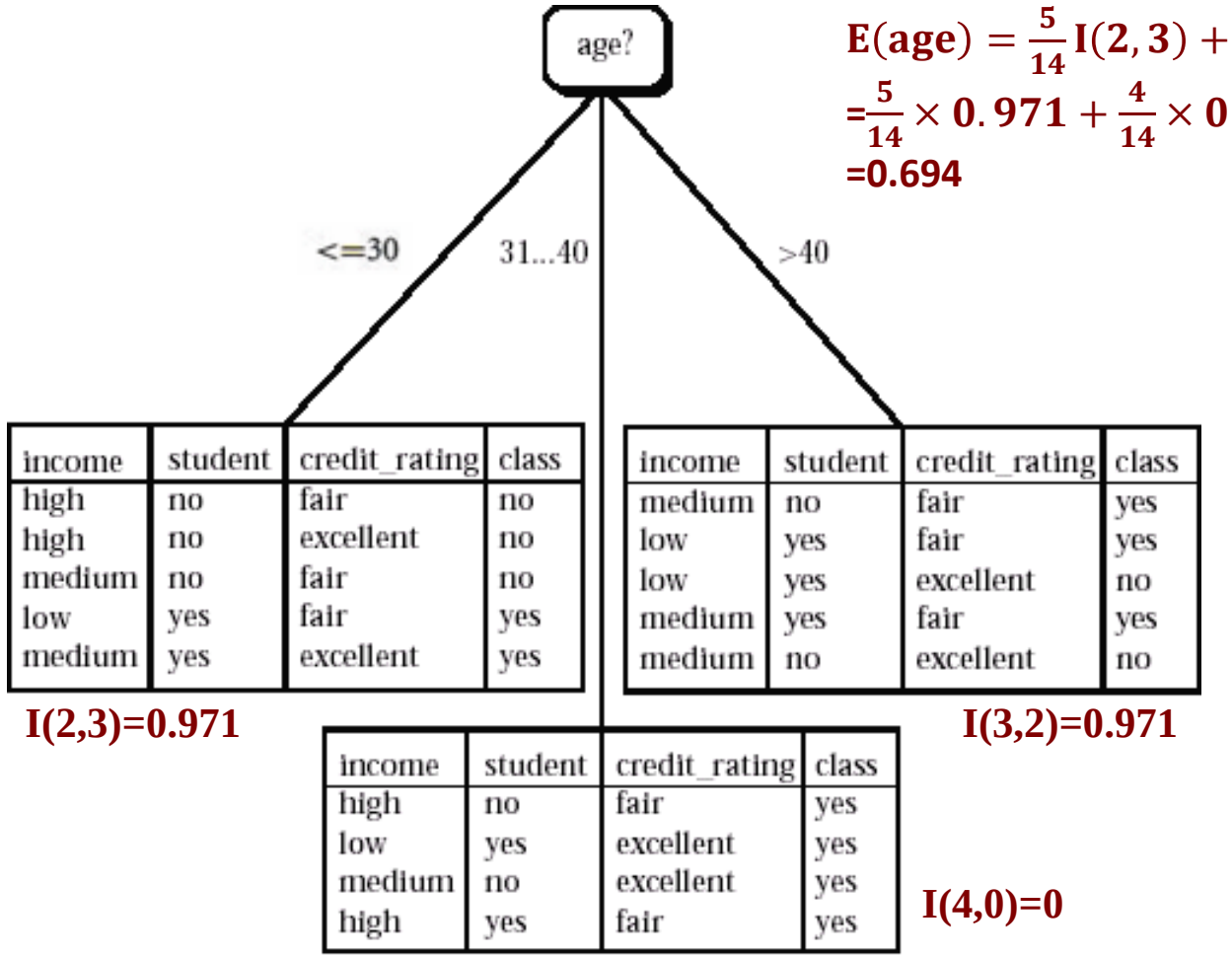
因此，这种划分的信息增益

Information Gain

$$Gain(age) = Info(D) - Info_{age}(D) = 0.940 - 0.694 = 0.246 \text{ 位}$$

类似地，可以计算 $Gain(income) = 0.029$ 位，$Gain(student) = 0.151$ 位，$Gain(credit\_rating) = 0.048$ 位。由于 *age* 在属性中具有最高的信息增益，所以它被选作分裂属性。结点 *N* 用 *age* 标记，并且每个属性值生长出一个分枝。然后元组据此划分，如图 8.5 所示。注意，落在分区 *age* = "*middle_aged*" 的元组都属于相同的类。由于它们都属于类 "*yes*"，所以要在该分枝的端点创建一个树叶，并用 "*yes*" 标记。算法返回的最终决策树如图 8.2所示。

ID3输入为分类变量：构造第一层



income	student	credit_rating	class
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

I(2,3)=0.971

income	student	credit_rating	class
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

I(4,0)=0

income	student	credit_rating	class
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

I(3,2)=0.971

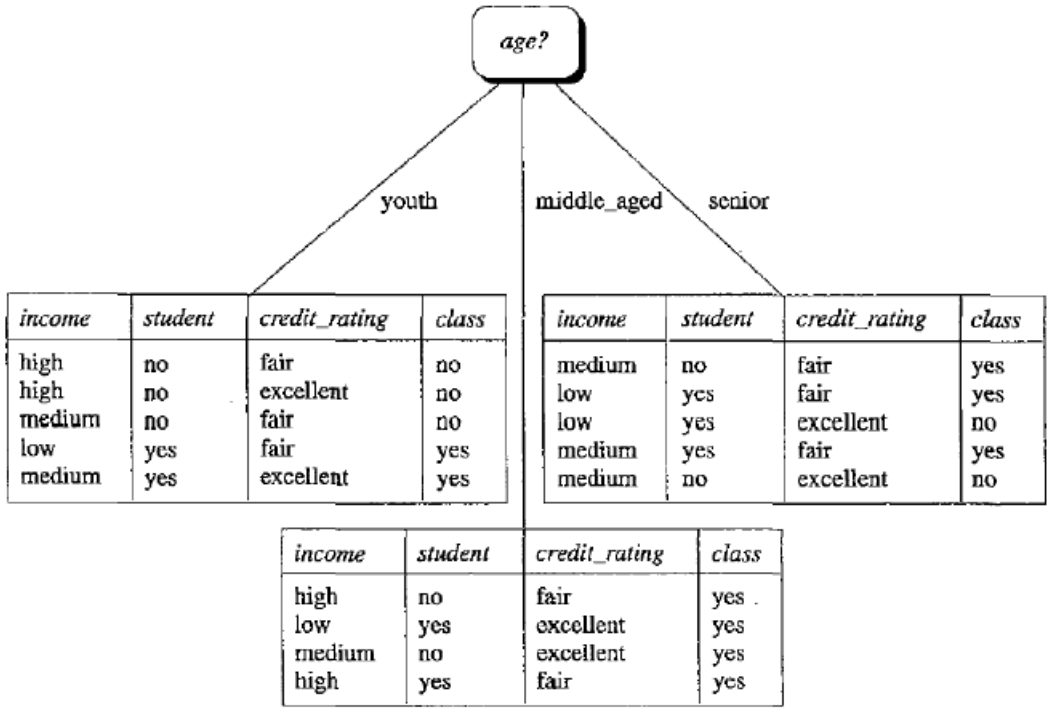
$$E(age) = \frac{5}{14}I(2,3) + \frac{4}{14}I(4,0) + \frac{5}{14}I(3,2)$$

$$= \frac{5}{14} \times 0.971 + \frac{4}{14} \times 0 + \frac{5}{14} \times 0.971$$

=0.694

ID3输入为分类变量：**构造第二层——**继续在子树重复挑选变量的步骤

- 已经可以肉眼观察：左侧student，右侧credit_rating，下方直接输出叶子yes

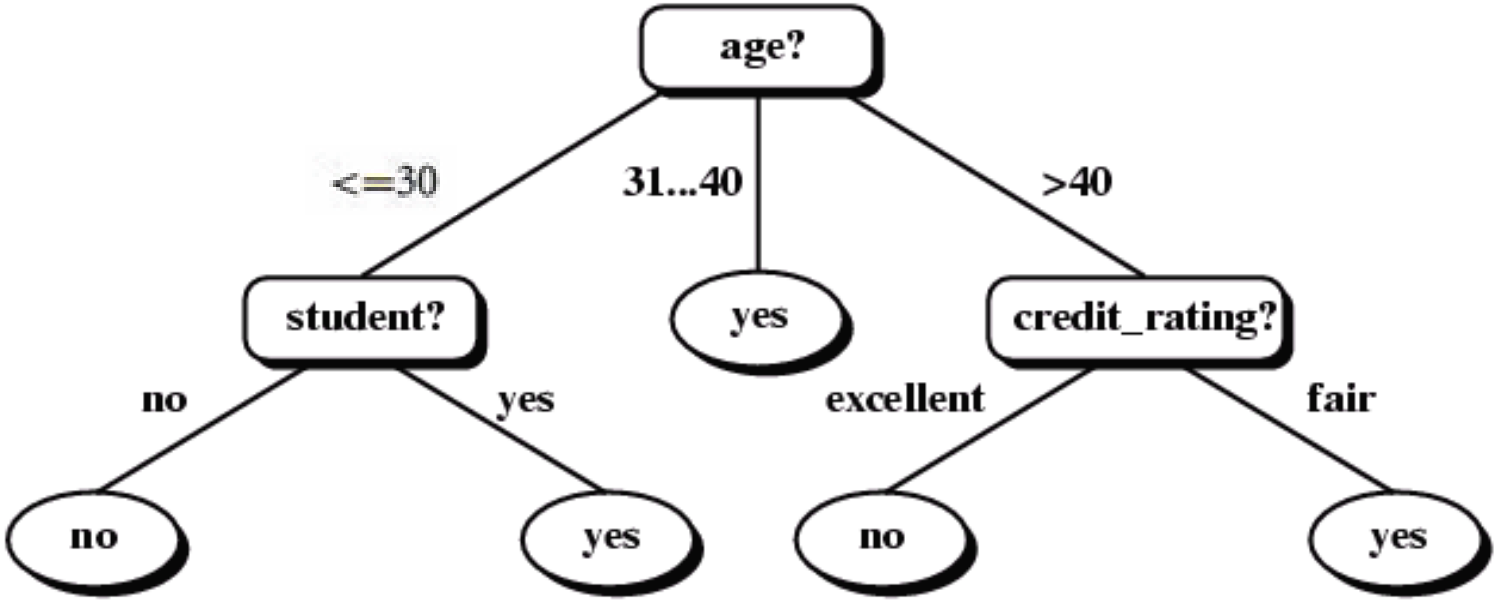


income	*student*	*credit_rating*	*class*
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

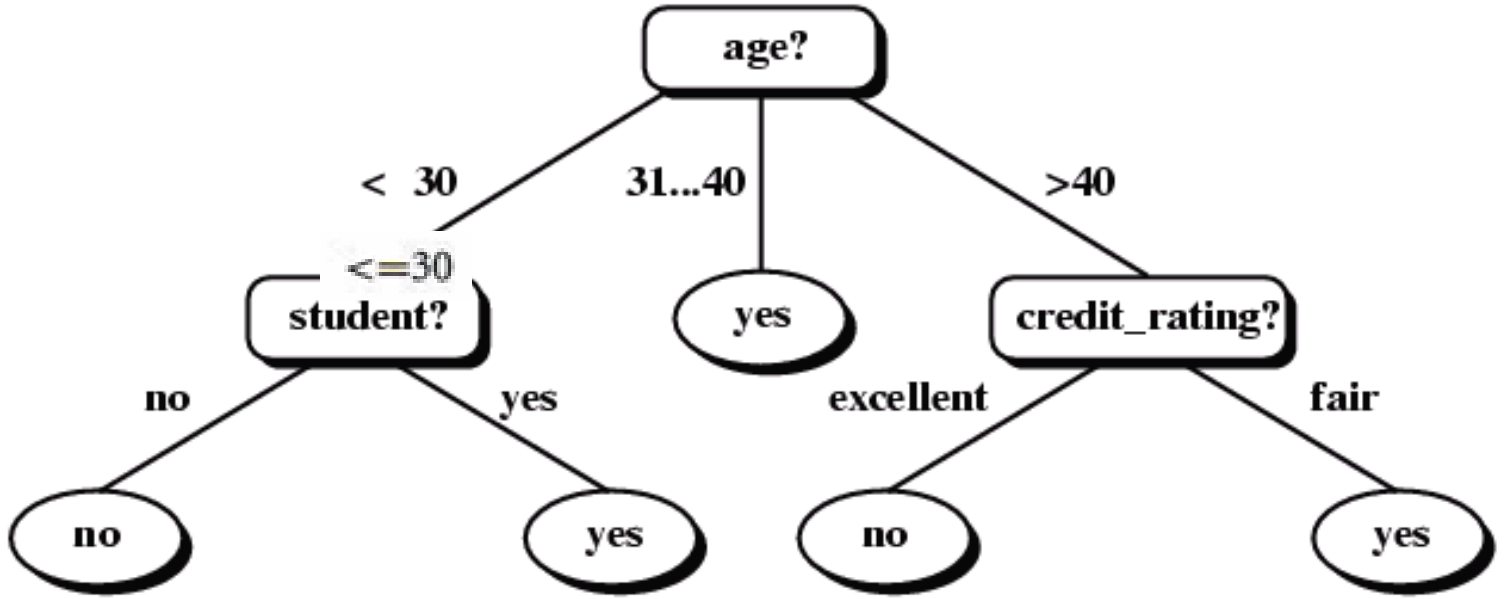
income	*student*	*credit_rating*	*class*
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

income	*student*	*credit_rating*	*class*
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

ID3输入为分类变量：构造第二层——结果



ID3生成分类规则



IF *age* = "<=30" AND *student* = "*no*" THEN *buys_computer* = "*no*"

IF *age* = "<=30" AND *student* = "*yes*" THEN *buys_computer* = "*yes*"

IF *age* = "31 . . . 40" THEN *buys_computer* = "*yes*"

IF *age* = ">40" AND *credit_rating* = "*excellent*" THEN *buys_computer* = "*no*"

IF *age* = ">40" AND *credit_rating* = "*fair*" THEN *buys_computer* = "*yes*"

ID3的缺点

- 倾向于选择水平数量较多的变量
- 输入变量必须是分类变量 (连续变量必须离散化)

C4.5

- 增加了连续变量二分法；
- 信息增益的方法倾向于首先选择因子数较多的变量
- 信息增益的改进：增益率

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

$$GrianRate(A) = \frac{Grain(A)}{SplitInfo_A(D)}$$

单个分类或等级变量:决策树遍历搜索

对于分类变量，假设该输入变量有4个水平，则依次遍历所有的组合形式，计算熵增益率最大的那个组合方式

A1 v.s. A2 v.s. A3 v.s. A4

A1、A2 v.s. A3 v.s. A4

A1、A2、A3 v.s. A4

A1、A3 v.s.A2、 A4

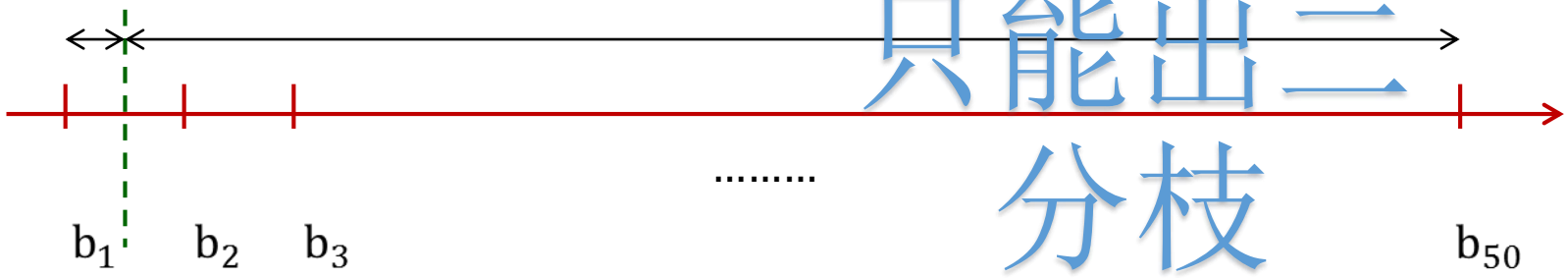
……

分类变量可以出多分枝

说明：C4.5决策树不能处理等级变量，要么作为分类变量，要么作为连续变量。这需要分析人员提前设置好。设为因子类型即为分类变量，否则为连续变量。

单个连续或等级变量:决策树分割搜索

对于**连续**变量，先等宽方式分为50组，依次取阀值分割成两组，计算熵增益率最大的那个分割方式



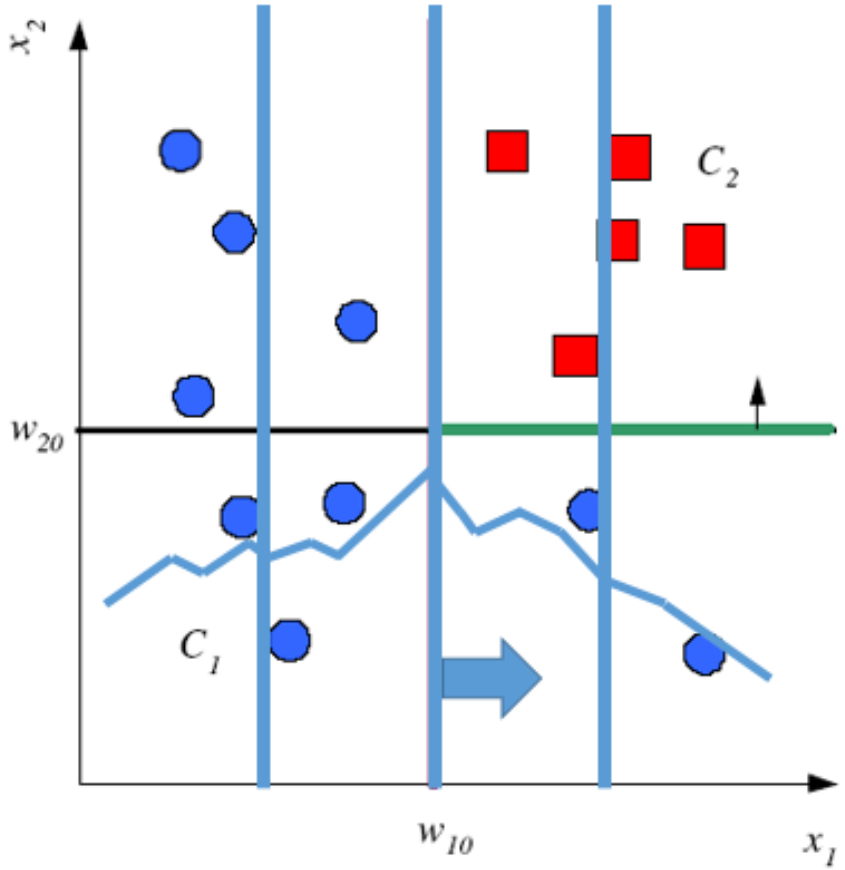
	t_1	t_2
b_1		
$b_2 - b_{50}$		

	t_1	t_2
$b_1 - b_2$		
$b_3 - b_{50}$		

说明：C4.5决策树不能处理等级变量，要么作为分类变量，要么作为连续变量。这需要分析人员提前设置好。设为因子类型即为分类变量，否则为连续变量。

比较多个变量的优先级

假设都是连续变量，先各自做分割，并计算每个分割的**熵增益率**。

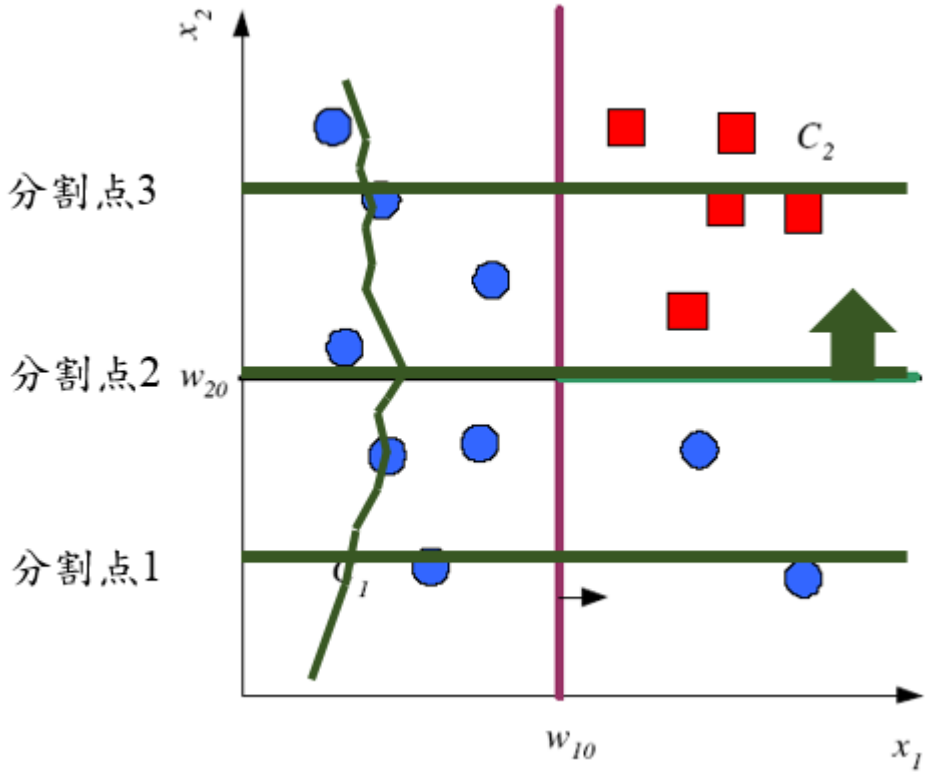


分割点1　分割点2　分割点3

最大
熵增益率(x_1)的点为“分割点2（w_{10}）”
0.35

比较多个变量的优先级

假设都是连续变量，先各自做分割，并计算每个分割的**熵增益率**。



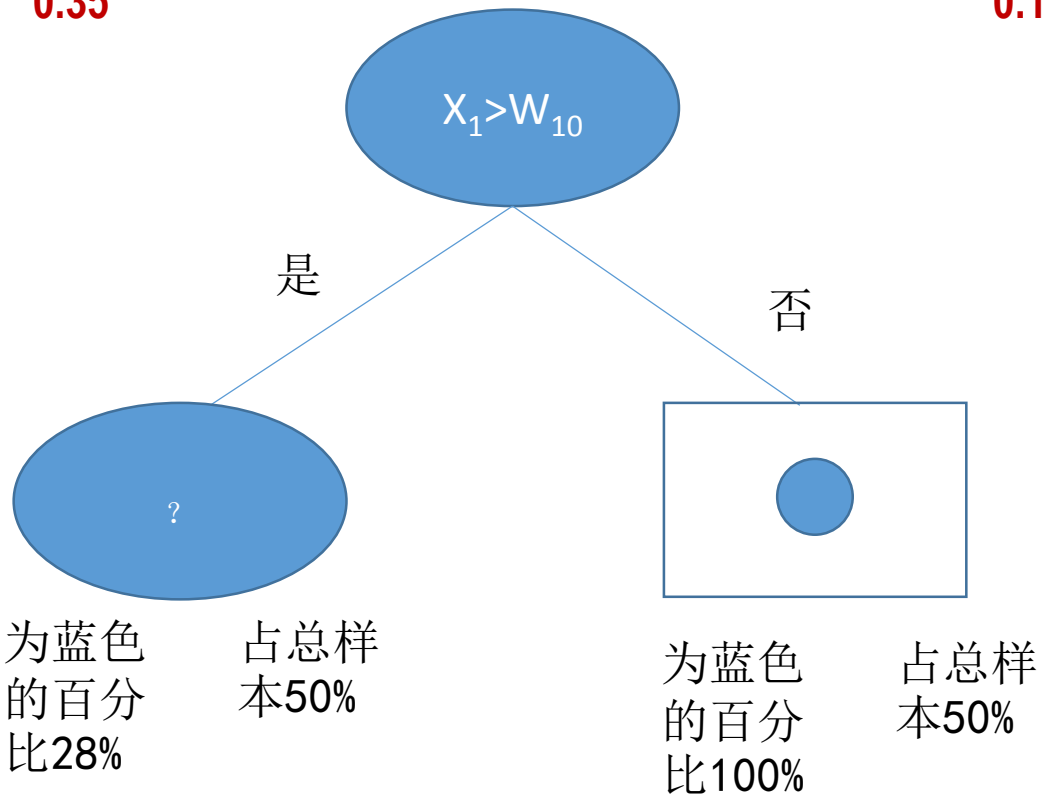
最大
熵增益率(x_2)的点为“分割点2（w_{20}）”
0.14

比较多个变量的优先级

比较每个变量所能达到的最大**熵增益率**，取最大的那个作为本次分割选择的变量，该变量对应最大**熵增益率**的分割点作为分割依据。

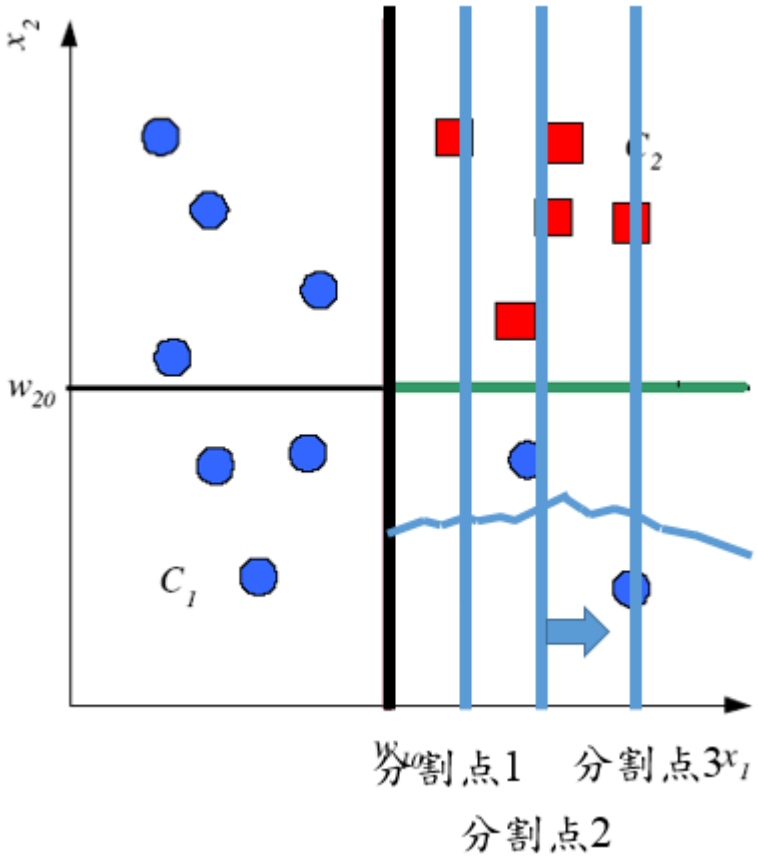
最大熵增益率(x_1)的点为“分割点2（w_{10}）” V.S. **最大熵增益率(x_2)的点为“分割点2（w_{20}）”**

0.35 **0.14**



比较多个变量的优先级

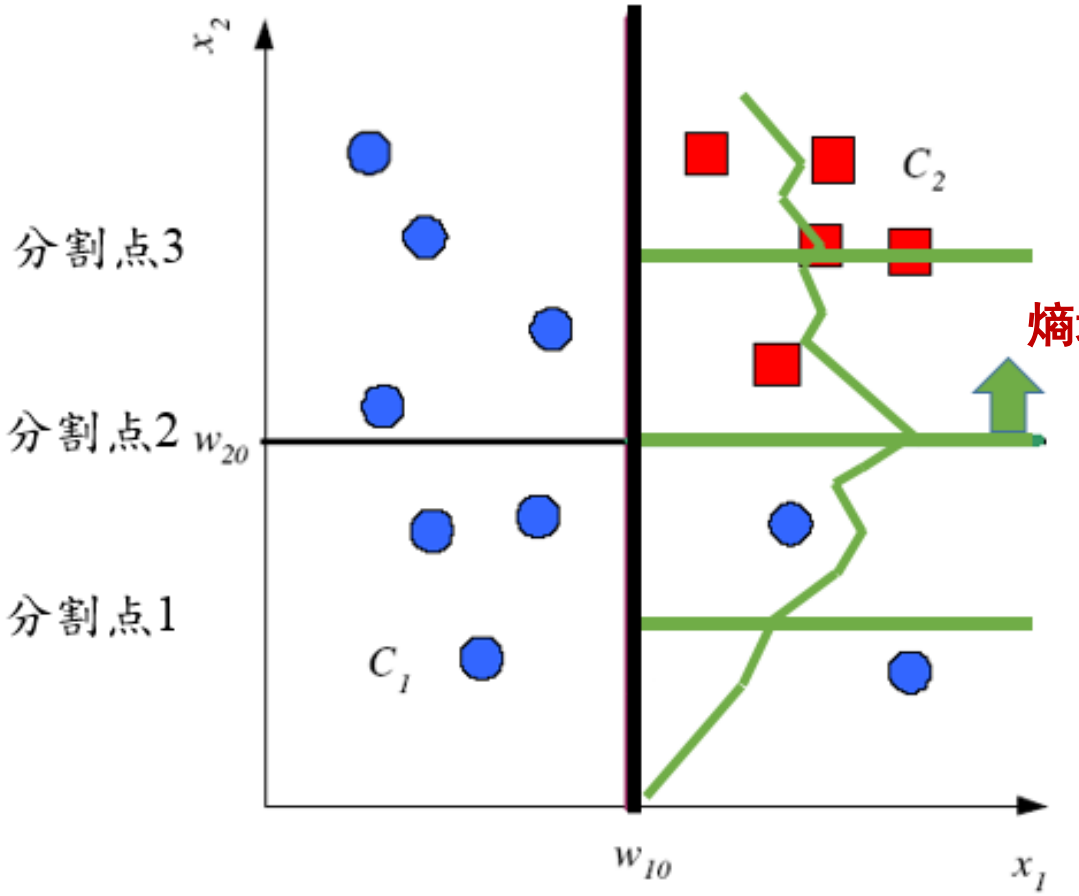
在$X_1>W_{10}$的组中，对X_1再进行遍历，计算**熵增益率**。



最大
熵增益率(x_1)的点为“分割点2”
0.04

比较多个变量的优先级

在$X_1>W_{10}$的组中，对X_2再进行遍历，计算**熵增益率**。



最大
熵增益率(x_2)的点为“分割点2（w_{20}）”
0.42

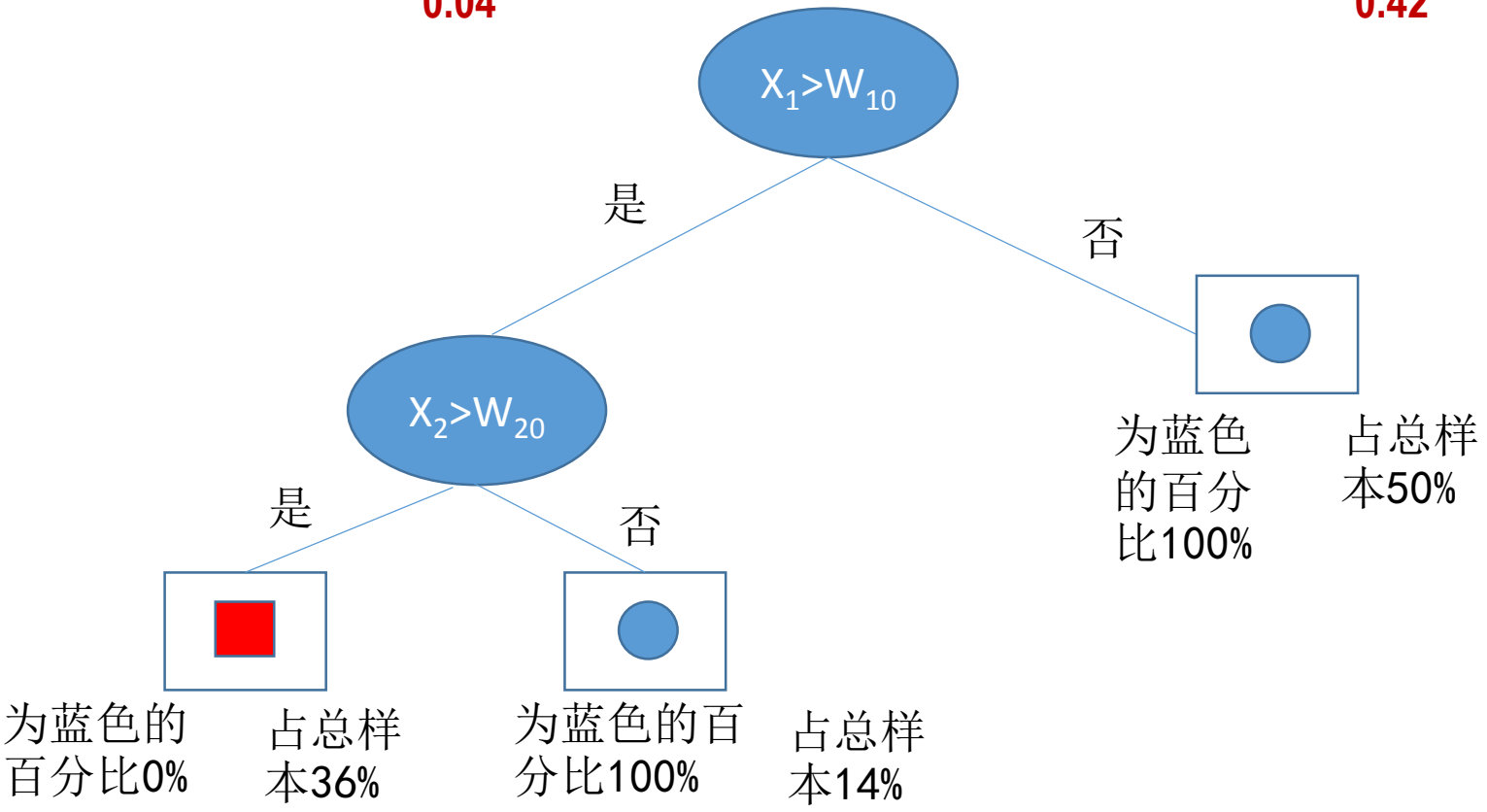
比较多个变量的优先级

在$X_1>W_{10}$的组中，比较每个变量所能达到的最大**熵增益率**，取最大的那个作为本次分割选择的变量，该变量对应最大**熵增益率**的分割点作为分割依据。

最大熵增益率(x_1)的点为“分割点2”
0.04

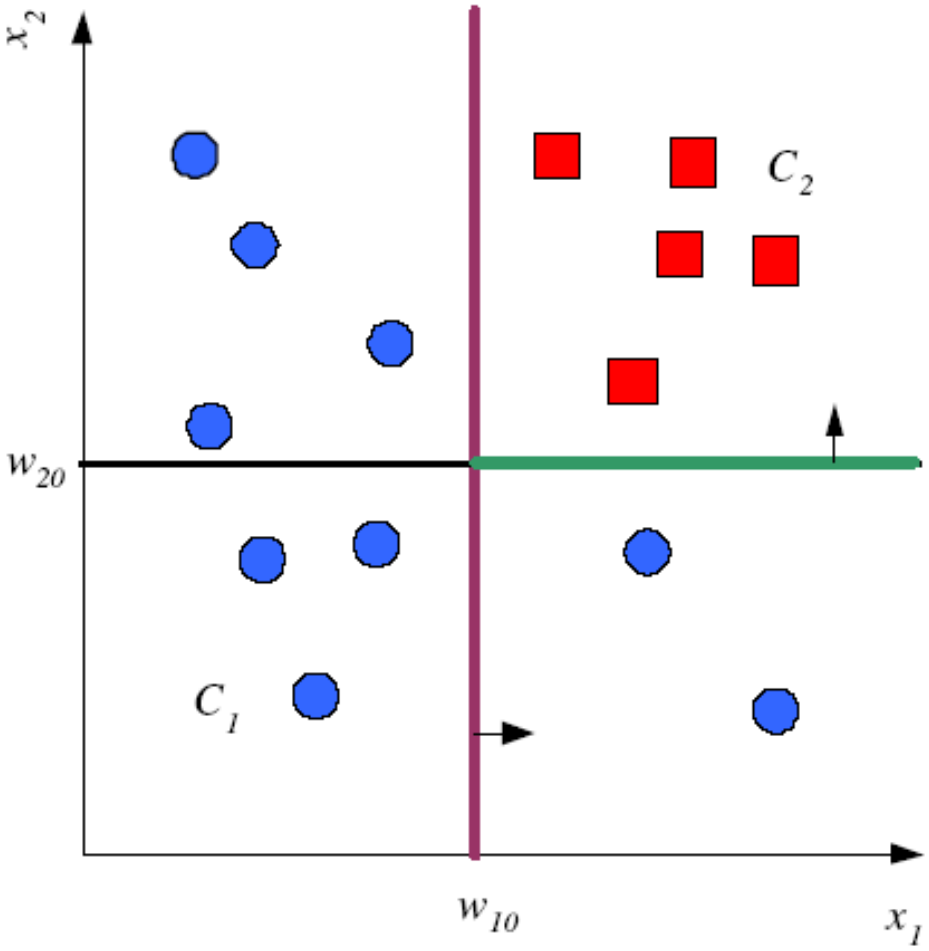
V.S.

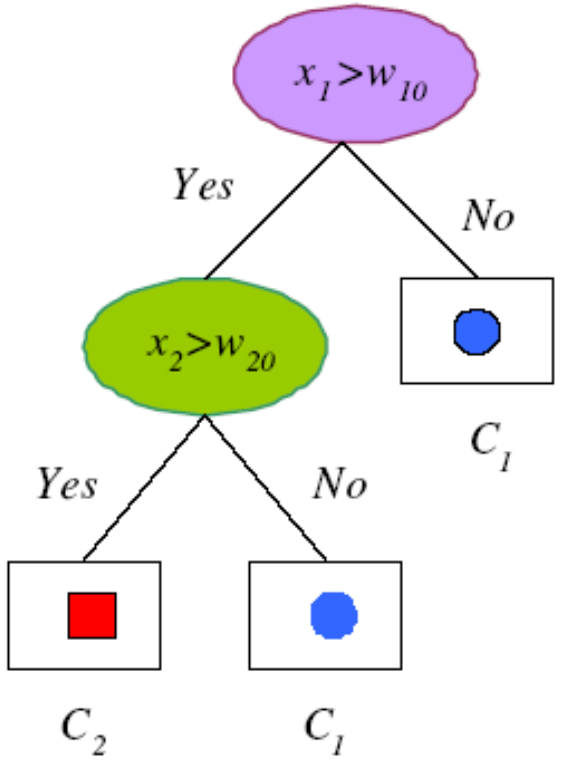
最大熵增益率(x_2)的点为“分割点2（w_{20}）”
0.42



比较多个变量的优先级

最终结果。





R中的C5.0算法(Python目前没有实现)

建树：

- 沿用C4.5的方法

剪枝：

- 前剪枝：
  - minCases叶子的最小样本量
  - winnow事先是否进行变量相关性筛选
- 后剪枝：
  - CF越小，要求模型的置信度越高
  - noGlobalPruning事后拆分规则有用性检验

主要创新处：

- 纳入了Boost的方法，可以做组合模型

参考书：
Quinlan R (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers

天善智能
TIANSHAN SOFT

CART决策树建模原理

CART决策树原理介绍

➢CART的建模步骤：

一、建树

- 选择最有解释力度的变量
- 对于每个变量选择最优分割点

二、剪树

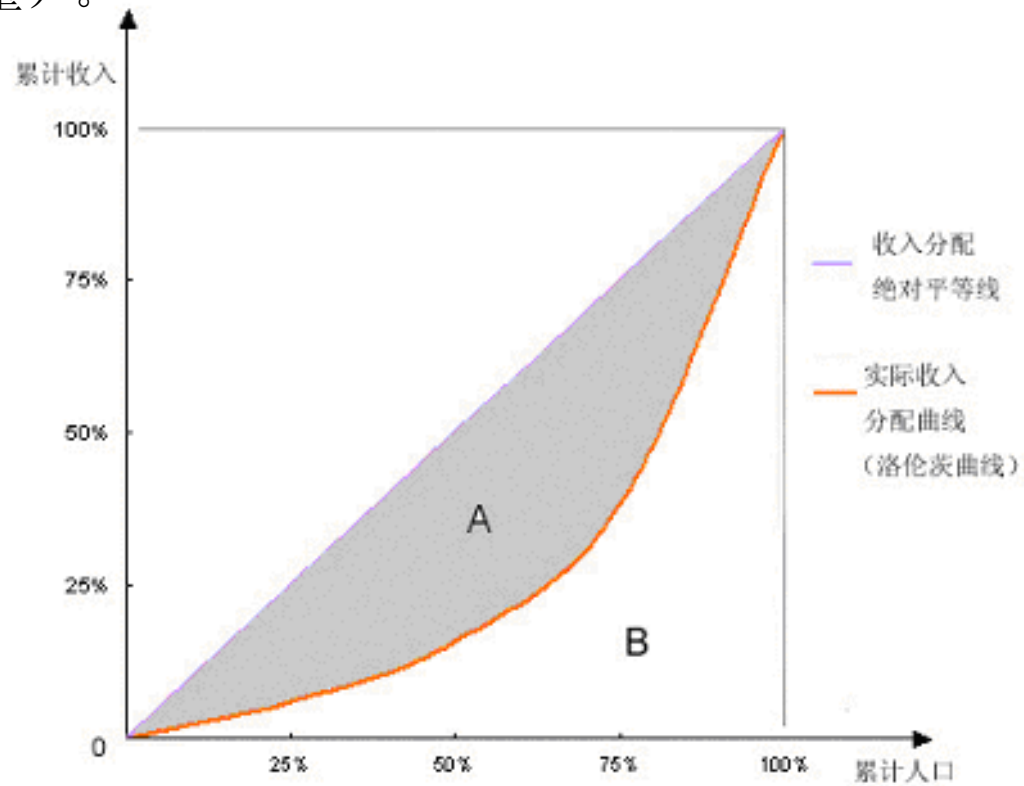
- 前向剪枝：控制生成树的规模
- 后项剪枝：删除没有意义的分组

基尼系数

赫希曼根据洛伦茨曲线提出的判断分配平等程度的指标。设实际收入分配曲线和收入分配绝对平等曲线之间的面积为A，实际收入分配曲线右下方的面积为B。并以A除以（A+B）的商表示不平等程度。这个数值被称为基尼系数或称洛伦茨系数。如果A为零，基尼系数为零，表示收入分配完全平等；如果B为零则系数为1，收入分配绝对不平等。收入分配越是趋向平等，洛伦茨曲线的弧度越小，基尼系数也越小，反之，收入分配越是趋向不平等，洛伦茨曲线的弧度越大，那么基尼系数也越大。另外，可以参看帕累托指数(是指对收入分布不均衡的程度的度量）。

假定一定数量的人口按收入由低到高顺序排队，分为人数相等的n组，从第1组到第i组人口累计收入占全部人口总收入的比重为wi，则说明：该公式是利用定积分的定义将对洛伦茨曲线的积分(面积B)分成n个等高梯形的面积之和得到的。

$$G = 1 - \frac{1}{n}\left(2\sum_{i=1}^{n-1} W_i + 1\right)$$



决策树建树原理

第1步：从众多输入变量中选择当前最佳分组变量

分类树：对于数值型输入变量。将数据按升序排列；然后，从小到大依次以相邻数值的中间值作为组限，将样本分成两组，并计算两组样本输出变量值的差异性，也称异质性。理想的分组应该尽量使两组输出变量值的异质性总和达到最小，即“纯度”最大，也就是使两组输出变量值的异质性随着分组而快速下降，“纯度”快速增加。
CART采用Gini系数测度输出变量的异质性 其数学定义为：

$$G(t)=1-\sum_{j=1}^{k} p^2(j \mid t)$$

CART采用Gini系数的减少量
测度异质性下降的程度：

$$\Delta G(t) = G(t) - \frac{N_r}{N} G(t_r) - \frac{N_l}{N} G(t_l)$$

对于分类型输入变量。由于CART只能建立二叉树，对于多分类型输入变量，首先需将多类别合并成两个类别，形成超类；然后，计算两超类下输出变量值的异质性。

计算基尼系数

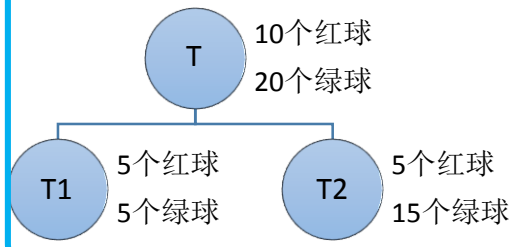
✓GINI系数计算示例：

✓$gini(T) = 1 - \sum p_j^2 = 1 - \sum(\frac{n_j}{s})^2$

✓$gini_{split}(T) = \frac{S_1}{S_1+S_2} gini(T_1) + \frac{S_2}{S_1+S_2} gini(T_2)$

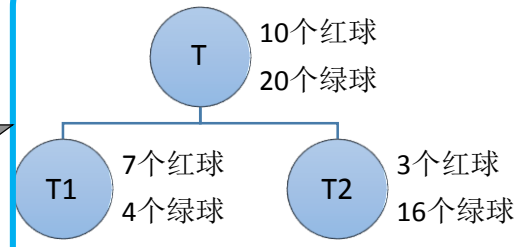
- Pj为类别j在样本T中出现的频率
- Nj为样本T中类别j的个数
- S为T中样本的个数
- S1，S2与T1，T2关系类似

划分1



- $\text{gini}(T) = 1 - \left(\frac{10}{10+20}\right)^2 - \left(\frac{20}{10+20}\right)^2 \approx 0.444$
- $\text{gini}(T_1) = 1 - \left(\frac{5}{5+5}\right)^2 - \left(\frac{5}{5+5}\right)^2 = 0.5$
- $\text{gini}(T_2) = 1 - \left(\frac{5}{5+15}\right)^2 - \left(\frac{15}{5+15}\right)^2 = 0.375$
- $\text{gini}_{s1}(T) = \frac{5+5}{5+5+5+15} \times 0.5 + \frac{5+15}{5+5+5+15} \times 0.375 \approx 0.417$

划分2



- $\text{gini}(T) = 1 - \left(\frac{10}{10+20}\right)^2 - \left(\frac{20}{10+20}\right)^2 \approx 0.444$
- $\text{gini}(T_1) = 1 - \left(\frac{4}{4+7}\right)^2 - \left(\frac{7}{4+7}\right)^2 \approx 0.463$
- $\text{gini}(T_2) = 1 - \left(\frac{3}{3+16}\right)^2 - \left(\frac{16}{3+16}\right)^2 \approx 0.266$
- $\text{gini}_{s2}(T) = \frac{4+7}{4+7+3+16} \times 0.463 + \frac{3+16}{4+7+3+16} \times 0.266 \approx 0.338$

CART算法

2、从分组变量的众多取值中，找到最佳分割点

最佳分割点的确定方法与最佳分组变量的确定是同时进行的。

计算示例

AllElectronics公司客户购买电脑数据

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

CART变量重要性选择

计算每个候选变量对被解释变量的重要性指标，CART使用的指标为基尼系数。

Age：

表 - age * buys_computer				
		buys_computer		
		no	yes	合计
age				
31...40	频数	0	4	4
<=30	频数	3	2	5
>40	频数	2	3	5
合计	频数	5	9	14

Δentropy（ID3）	**ΔGini（CART）**	Logworth（CHAID）
0.246	**0.102**	1.30

Income：

表 - income * buys_computer				
		buys_computer		
		no	yes	合计
income				
high	频数	2	2	4
low	频数	1	3	4
medium	频数	2	4	6
合计	频数	5	9	14

0.029	**0.016**	0.74

单个分类或等级变量:决策树遍历搜索

对于分类变量，假设该输入变量有4个水平，则依次组合成两组，计算基尼系数变化最大的那个组合方式

	t_1	t_2
a_1		
a_2, a_3, a_4		

	t_1	t_2
a_2		
a_1, a_3, a_4		

	t_1	t_2
a_3		
a_1, a_2, a_4		

	t_1	t_2
a_4		
a_1, a_2, a_3		

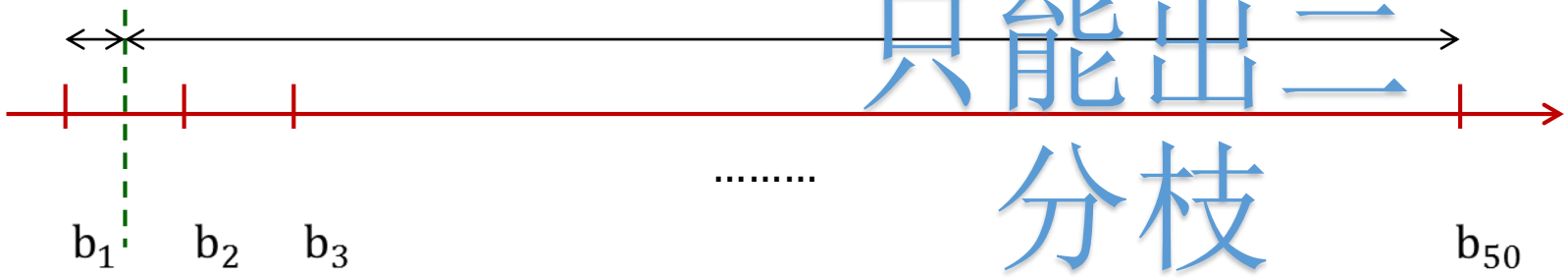
分类变量只能出二分枝

……

说明：CART决策树不能处理等级变量，要么作为分类变量，要么作为连续变量。这需要分析人员提前设置好。设为因子类型即为分类变量，否则为连续变量。

单个连续或等级变量:决策树分割搜索

对于**连续**变量，先等宽方式分为50组，依次取阀值分割成两组，计算基尼系数变化最大的那个分割方式。



	t_1	t_2
b_1		
$b_2 - b_{50}$		

	t_1	t_2
$b_1 - b_2$		
$b_3 - b_{50}$		

说明：CART决策树不能处理等级变量，要么作为分类变量，要么作为连续变量。这需要分析人员提前设置好。设为因子类型即为分类变量，否则为连续变量。

比较多个变量的优先级

其处理方法和C50完全一致，不再赘述。

决策树方法总结

	C50	CART	CHAID
输出变量	只能是分类型	可以是分类型也可以是数值型	可以是分类型 也可以是数值型
	只能建分类树	既可建分类树又可建回归树	能够建立多叉树
树	可建多叉树	只能建二叉树	能够建立多叉树
确定最佳分组变量和分割点	以信息熵为基础 通过计算信息增益率确定	以Gini系数和方差作为选择依据	从统计显著性检验角度确定
决定决策树的标准	主要根据叶子中样本量进行前剪枝；依据测试样本进行后剪枝		从统计显著性检验角度确定

天善智能
TIANSHAN SOFT

模型修剪——以CART为例

剪枝

在决策树创建时，由于数据中的噪音和离群点，许多分枝反映的是训练数据中的异常。剪枝方法处理这种过分拟合的数据会影响模型的稳定性。通常使用统计度量剪掉最不可靠的分枝。剪枝后的树更小、更简单、更容易理解。

剪枝策略：

一、预修剪

其目标是控制决策树充分生长，可以事先指定一些控制参数，包括：

（1）决策树最大深度。如果决策树的层数已经达到指定深度，则停止生长 。

（2）树中父节点和子节点的最少样本量或比例。对于父节点，如果节点的样本量低于最小样本量或比例，则不再分组；对于子节点，如果分组后生成的子节点的样本量低于最小样本量或比例，则不必进行分组 。

（3）树节点中输出变量的最小异质性减少量。如果分组产生的输出变量异质性变化量小于一个指定值，则不必进行分组。

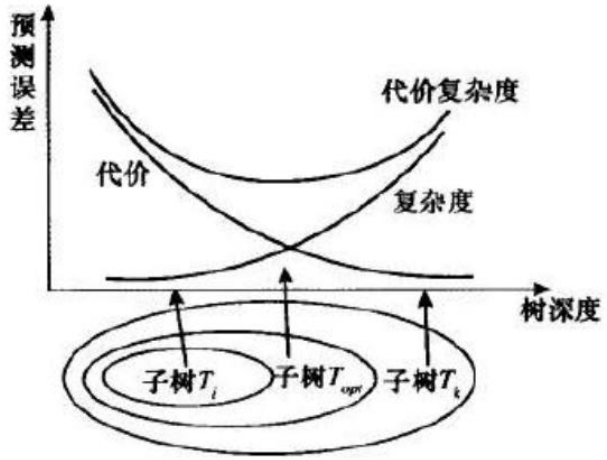
剪枝

二、后剪枝

后修剪技术允许决策树充分生长，然后在此基础上根据一定的规则，剪去决策树中那些不具有一般代表性的叶节点或子树，是一个边修剪边检验的过程。在修剪过程中，应不断计算当前决策子树对测试样本集的预测精度或误差，并判断应继续修剪还是停止修剪。

•CART采用的后修剪技术称为最小代价复杂性修剪法（Minimal Cost Complexity Pruning,MCCP）

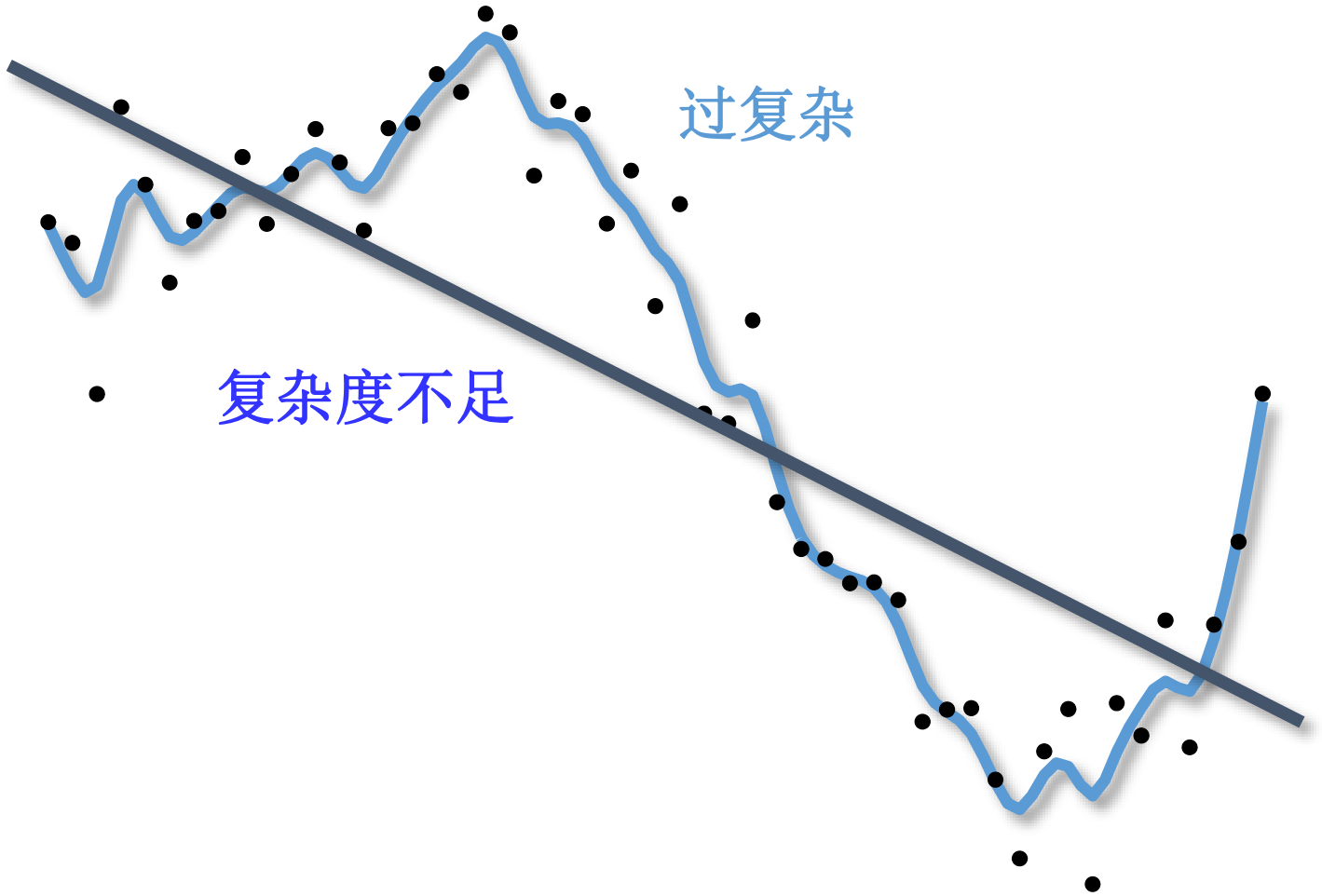
•MCCP有这样的基本考虑：首先，考虑的决策树虽然对训练样本有很好的预测精度，但在测试样本和未来新样本上不会仍有令人满意的预测效果；其次，理解和应用一棵复杂的决策树是一个复杂过程。因此，决策树修剪的目标是得到一棵“恰当”的树，它首先要具有一定的预测精度，同时决策树的复杂程度应是恰当的。



代价复杂度修剪

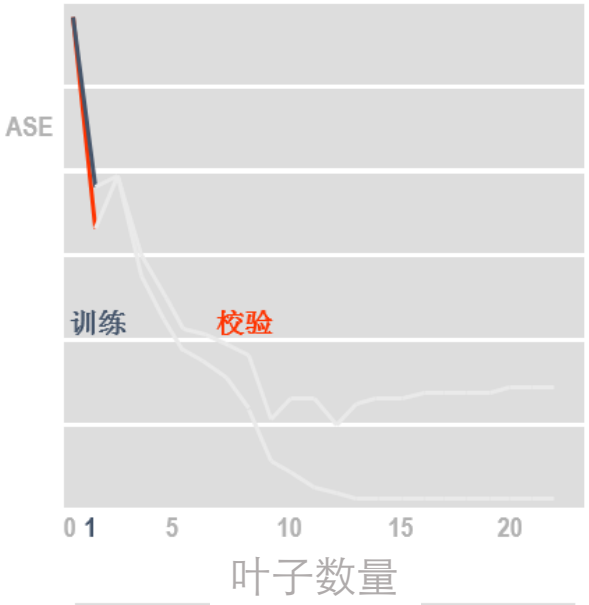
CART选择最终子树标准是：
选择交叉验证中错误最小的

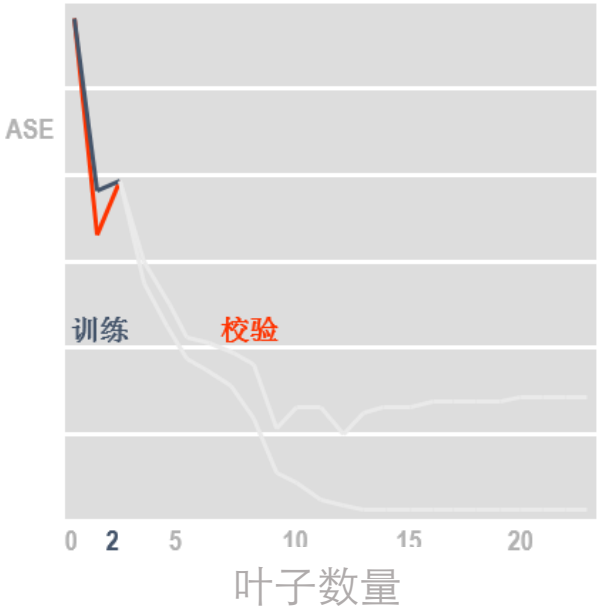
模型复杂度

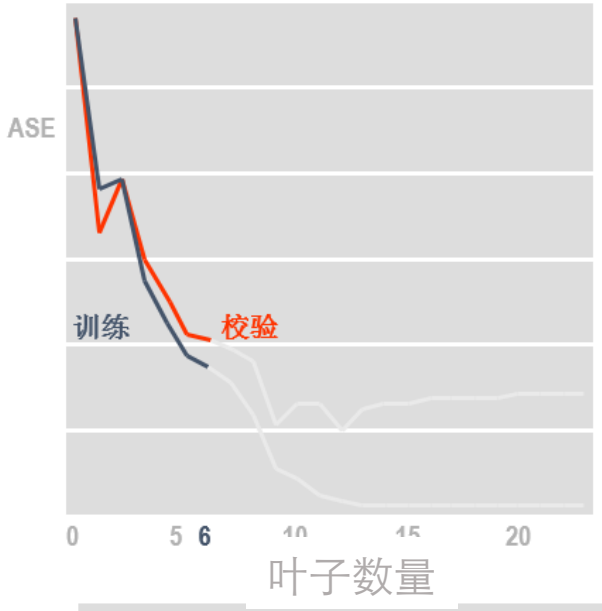


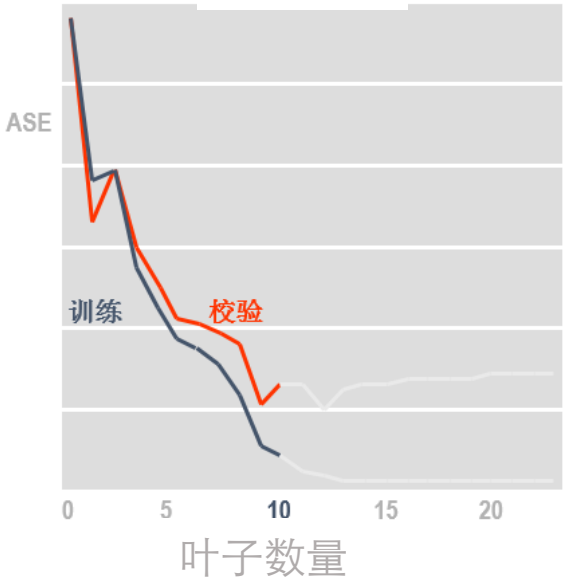
...

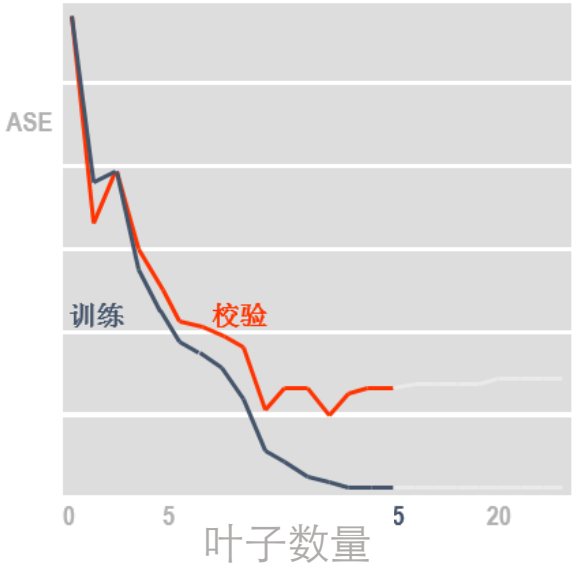
模型复杂度

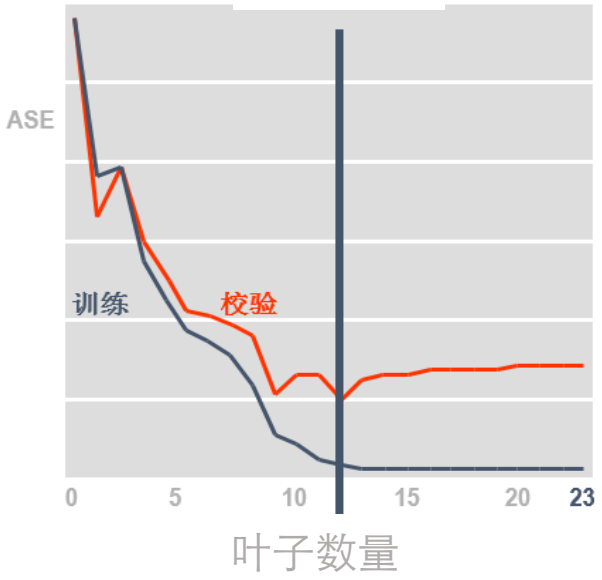












调整模型的超参数，使得模型由简单至复杂。根据模型在校验数据集上的表现，确定最合适的超参数。

CART的决策树修剪方法—总结

- 输入变量（自变量）：为分类型变量或连续型变量
- 输出变量（目标变量）：为分类型变量（或连续型：回归分析）
- 连续变量处理：N等分离散化
- 树分枝类型：二分枝
- 分割指标：gini增益（分割后的目标变量取值变异较小，纯度高）
- 先剪枝：决策树最大深度、最小样本分割数、叶节点包含的最小样本数、复杂度系数最小值
- 后剪枝：使用最小代价复杂度剪枝法

更多商业智能BI和大数据精品视频尽在 www.hellobi.com

BI、商业智能
数据挖掘 大数据
数据分析师
R语言 Python
机器学习
深度学习
人工智能
Hive Hadoop
Tableau
BIEE ETL
数据科学家
PowerBI

天善智能
TIANSHAN SOFT