南昌大学
NANCHANG UNIVERSITY

05

决策树

本课内容

1. 决策树原理

2. 决策树的构建

3. 决策树构建举例

本课内容

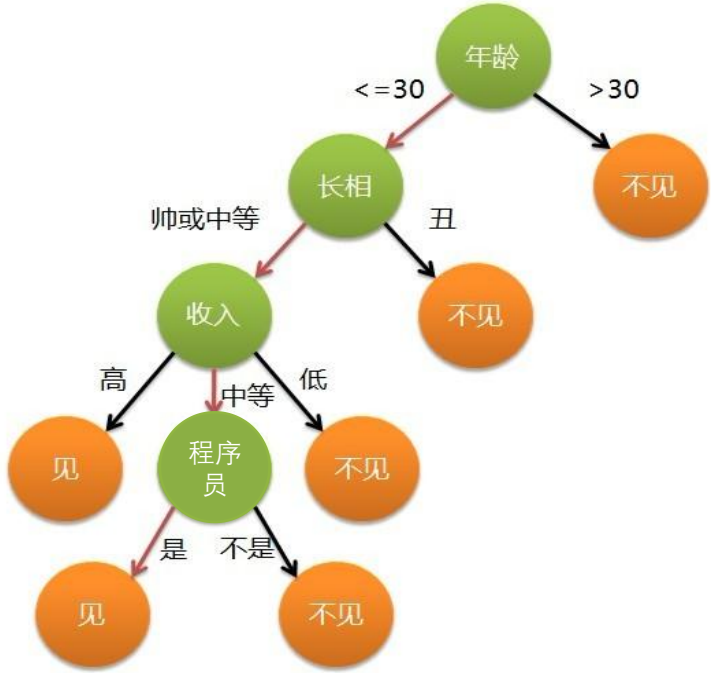
1. 决策树原理

2. 决策树的构建

3. 决策树构建举例

什么是决策树

- **决策树的基本思想**：模拟人类进行级联选择或决策的过程，按照属性的某个优先级依次对数据的全部属性进行判别，从而得到输入数据所对应的预测输出。
- 决策树是一个**树形结构**，其中每个内部节点代表一个特征（或属性），每个分支代表一个决策规则，每个叶子节点代表一个预测结果。决策树通过一系列的问题对数据进行分类或预测。
- **测试序列**：从根结点到某一叶子结点的路径。



什么是决策树

- 用决策树进行预测时：
  - 从根结点开始，对样本的相应特征进行测试，所以决策树模型可以被认为是**if-then规则的集合**。
  - 递归地对样本进行测试，直至样本被划分到某个叶子结点。
  - 最后，根据该叶子结点的分数$\boldsymbol{w}_m$对测试样本进行预测：

$$f(\boldsymbol{x}) = \sum_{m=1}^{M} \boldsymbol{w}_m \mathbb{I}(\boldsymbol{x} \in \mathbf{R}_m) = \sum_{m=1}^{M} \boldsymbol{w}_m \emptyset(\boldsymbol{x}, \boldsymbol{v}_m)$$

其中，$\mathbf{R}_m$表示第m个叶子结点所代表的特征空间；$\mathbb{I}(.)$ 为示性函数，括号中的条件满足则取1，否则取0。

什么是决策树

$$f(\boldsymbol{x}) = \sum_{m=1}^{M} \boldsymbol{w}_m \mathbb{I}(\boldsymbol{x} \in \mathbf{R}_m) = \sum_{m=1}^{M} \boldsymbol{w}_m \emptyset(\boldsymbol{x}, \boldsymbol{v}_m)$$

- $\boldsymbol{w}_m$是第m个叶子结点的预测值：
  - 对于回归任务，$\boldsymbol{w}_m$通常为第m个叶子结点所有样本的y的均值，这时预测结果的L2损失最小；
  - 对于分类任务，$\boldsymbol{w}_m$通常为第m个叶子结点所有样本的y的分布，分类结果可以取分布中概率最大的类别。
- $\boldsymbol{v}_m$表示为第m个叶子结点对应的参数，包括从根结点到第m个叶子结点的路径上，每个结点选择的特征及划分阈值。

什么是决策树

- 同其他机器学习模型一样，决策树模型的目标函数包含两部分：**训练集上的损失函数之和**以及**正则项**。

$$J(\boldsymbol{\theta}, \lambda) = \sum_{i=1}^{N} L(f(\mathbf{x}_i; \boldsymbol{\theta}), y_i) + \lambda R(\boldsymbol{\theta})$$

- 决策树模型的损失函数与结点的不纯净度有关。不同的决策树算法中，不纯净度的度量稍有不同（**ID3：信息增益；C4.5：信息增益率；CART：GINI系数/均方误差**）。
- 正则项可取**L1正则（叶子结点的数目）**或**L2正则（叶子结点分数的平方和**，与叶子节点的数目和噪声的影响有关）。

本课内容

1. 决策树原理

2. 决策树的构建

3. 决策树构建举例

建树

- 决策树的训练就是决策树的建树过程，对应特征空间的划分。

- 选择最优决策树的问题是个NP（Non-deterministic Polynomial）完全问题，因此一般采用启发式方法近似求解。

- 决策树的学习算法通常会递归地选择最优特征及划分阈值，并根据该特征对训练数据进行划分，使得划分后的每个数据子集越纯净越好。

建树

- 决策树生成算法：

1、构建根结点：将所有训练数据放在根结点处，并将该结点加入叶子结点列表。

2、若叶子结点列表为空，则算法结束；否则从叶子结点列表中挑选1个叶子结点。

①若该叶子结点的样本集合已足够纯净，则计算该叶子结点对应的预测分数，并将其从叶子结点列表中删除（即作为最终叶子结点不再划分）；

②否则，采用每个特征的每个可能划分方式对该节点的样本集合进行划分并计算划分后的纯净度；从所有划分中，选择一个最优划分（划分后不纯净度下降最多），将训练数据划分为若干子集，每个子集为当前结点的子结点，并将这些子结点加入叶子结点列表。

建树

- 流行的决策树算法有：**ID3、C4.5和CART。**
- 不同决策树方法区别：**选择属性/阈值($x_j=t$)进行节点分裂的准则不同**
  - ID3：信息增益最大（对标签 y 提供信息最多的特征），倾向于选择取值多的特征进行分裂
  - C4.5：ID3的改进，信息增益率最大
  - CART：二分递归划分——将当前样本集合划分为两个子集，使得生成的每个非叶子结点都有两个分支（二叉树）。

    分类：GINI指数最小

    回归：均方误差最小

信息熵

- “信息熵”是度量样本集合纯度最常用的一种指标，假定当前样本集合$\mathcal{D}$中第c类样本出现的概率为$p_c(c = 1, 2, \ldots, \mathrm{C})$，则$\mathcal{D}$的信息熵定义为

$$H(\mathcal{D}) = -\sum_{\mathrm{c}=1}^{\mathrm{C}} p_c \log_2 p_c$$

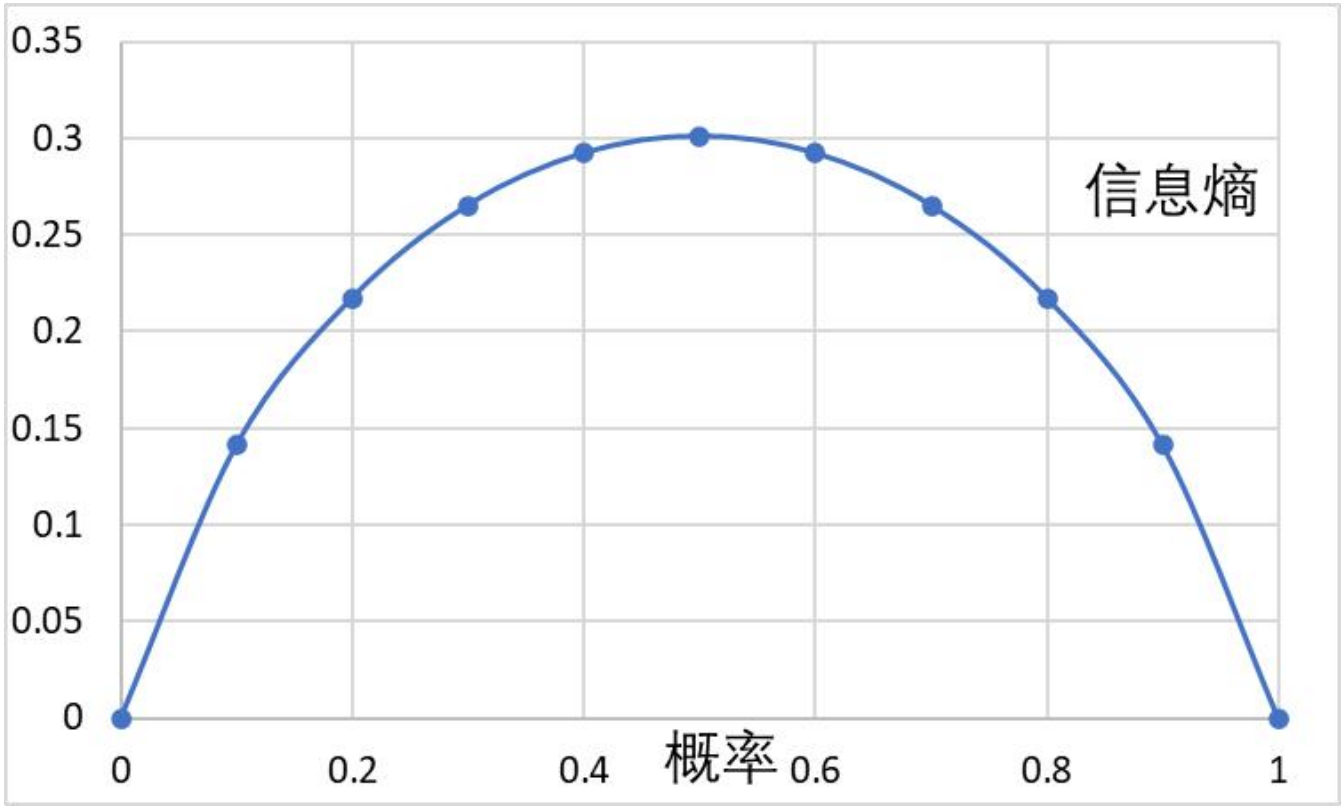
  - 计算信息熵时约定：若$p_c = 0$，则$p_c \log_2 p_c = 0$。
  - $H(\mathcal{D})$的值越小，则$\mathcal{D}$的纯度越高。
  - $H(\mathcal{D})$的最小值为0，最大值为$\log_2 C$

信息熵

- 假设类别数为2：

$$H(\mathcal{D}) = -p\log p - (1-p)\log(1-p)$$

0	0.00
0.1	0.14
0.2	0.22
0.3	0.27
0.4	0.29
0.5	0.30
0.6	0.29
0.7	0.27
0.8	0.22
0.9	0.14
1	0.00



ID3

- ID3决策树学习算法[Quinlan, 1986]以信息增益为准则来选择划分属性
  - 令当前节点的样本集合为$\mathcal{D}$
  - 用样本的比例估计概率分布：$p(Y=c)=\hat{\pi}_c=\frac{1}{|\mathcal{D}|}\sum_{i\in\mathcal{D}}\mathbb{I}(y_i=c)$
  - 分裂之前的熵：$H(\mathcal{D})=-\sum_{c=1}^{\mathrm{C}}p(Y=c)\log_2 p(Y=c)$
  - 分裂成V个子集之后的熵：$H_X(\mathcal{D})=\sum_{v=1}^{V}\frac{|\mathcal{D}_v|}{|\mathcal{D}|}H(\mathcal{D}_v)$
  - 信息增益：$gain_X(\mathcal{D})=H(\mathcal{D})-H_X(\mathcal{D})$

信息增益

日志密度L	好友密度F	是否使用真实头像H	账号是否真实R
s	s	no	no
s	l	yes	yes
l	m	yes	yes
m	m	yes	yes
l	m	yes	yes
m	l	no	yes
m	s	no	no
l	m	no	yes
m	s	no	yes
s	s	yes	no

$$H(\mathcal{D}) = -0.7\log_2 0.7 - 0.3\log_2 0.3 = 0.879$$

日志密度L:

$$H_L(\mathcal{D}) = 0.3 \times \left(-\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3}\right) +$$

$$0.4 \times \left(-\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4}\right) +$$

$$0.3 \times \left(-\frac{3}{3}\log_2\frac{3}{3} - \frac{0}{3}\log_2\frac{0}{3}\right) +$$

$$= 0.603$$

$$gain_L(\mathcal{D}) = H(\mathcal{D}) - H_L(\mathcal{D}) = 0.276$$

$$gain_F(\mathcal{D}) = 0.553$$

$$gain_H(\mathcal{D}) = 0.033$$

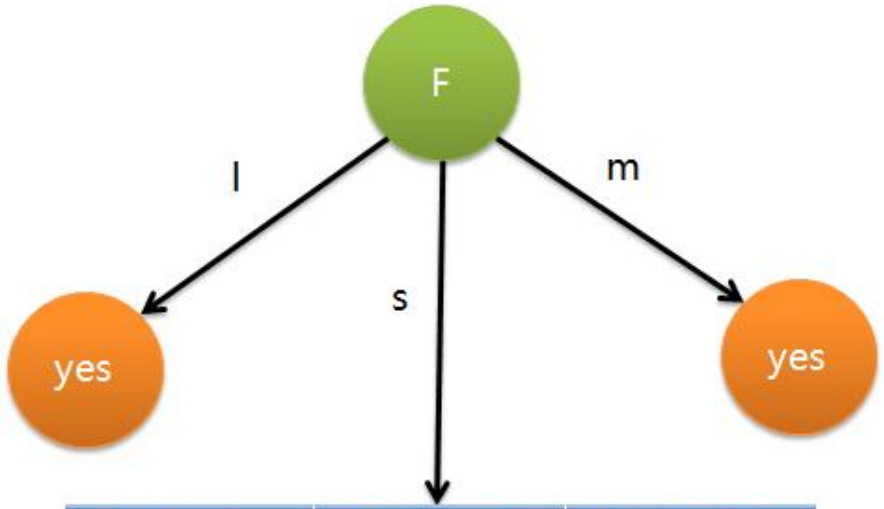
信息增益

日志密度L	好友密度F	是否使用真实头像H	账号是否真实R
s	s	no	no
s	l	yes	yes
l	m	yes	yes
m	m	yes	yes
l	m	yes	yes
m	l	no	yes
m	s	no	no
l	m	no	yes
m	s	no	yes
s	s	yes	no

$gain_L(\mathcal{D}) = 0.276$ $gain_F(\mathcal{D}) = 0.553$

$gain_H(\mathcal{D}) = 0.033$

因为F具有最大的信息增益（或最小熵），所以第一次分裂选择F为分裂属性，分裂后的结果如下图表示：



日志密度	是否使用真实头像	账号是否真实
s	no	no
m	no	no
m	no	yes
s	yes	no

C4.5

- ID3选择信息增益最大的特征进行分裂，倾向于选择取值多的特征
  - 如ID号取值有很多，会分裂出很多子节点，但对分类没有意义
- C4.5：选择信息增益率最大的特征进行分裂
  - 分裂信息：$split\_info_X(\mathcal{D}) = -\sum_{v=1}^{V} \frac{|\mathcal{D}_v|}{|\mathcal{D}|} log_2 \frac{|\mathcal{D}_v|}{|\mathcal{D}|}$
  - 信息增益率：$gain\_ratio_X(\mathcal{D}) = \frac{gain_X(\mathcal{D})}{split\_info_X(\mathcal{D})}$
  - 即：增加的信息不要以分割太细为代价。

C4.5

- 存在的问题:
  - 信息增益率准则倾向于选择可取值数目较少的属性
- C4.5 [Quinlan, 1993]使用了一个启发式：先从候选划分属性中找出信息增益高于平均水平的属性，再从中选取信息增益率最高的。
- 避免过拟合：引入剪枝策略（从决策树上裁剪掉一些子树或者叶子结点，并将其根节点或父节点作为新的叶结点，从而简化分类树模型）。
- 决策树剪枝策略有**预剪枝**和**后剪枝**两种。

预剪枝

- 一种控制决策树复杂度的方式是预剪枝，即在构造决策树的同时进行剪枝。

- 在构建决策树时，当树达到某些条件（达到**最大树深度**、叶子结点**数目达到最大值**、叶子结点的**纯净度达到一定精度**、结点中的**样本数量小于某个阈值**、最优划分带来的**增益小于某个阈值**），停止树的增长，也被称为提前终止。

后剪枝

- 后剪枝：给定一个完全树，自底向上进行剪枝，直到根节点。
- 设树T的叶子结点个数为|T|，叶子结点的索引为t=1, 2, …, |T|，定义树的分数为

$$C_\alpha(T) = \sum_{t=1}^{|T|} |D_t| H(D_t) + \alpha |T|$$

  - 形式同机器学习模型的目标函数：$J(\boldsymbol{\theta}, \lambda) = \sum_{i=1}^{N} L(f(\mathbf{x}_i; \boldsymbol{\theta}), y_i) + \lambda R(\boldsymbol{\theta})$
  - 设某个叶子节点退回到父节点之前和之后的整棵树分别为T和T'，对应的分数分别为$C_\alpha(T)$和$C_\alpha(T')$，若$C_\alpha(T') < C_\alpha(T)$，则进行剪枝，父节点变成新的叶子节点。
  - 当α从0开始增大，树的一些分支被剪掉，得到不同α对应的树，可采用交叉验证得到最佳的α。
- 后剪枝可以**保证剪枝操作不会降低决策树模型的泛化性能**，因此通常采用后剪枝策略。

CART

- 对于决策树模型，还可采用基尼指数作为划分标准来选择最优属性。
- 对于任意给定的一个C分类问题，假设样本点属于第c类的概率为p_c，则关于这个概率分布p的基尼指数定义为：

$$\mathrm{Gini}(p) = \sum_{c=1}^{C} p_c(1-p_c) = 1 - \sum_{c=1}^{C} p_c^2$$

- 对于任意给定的样本集合$\mathcal{D}$，其基尼指数可定义为

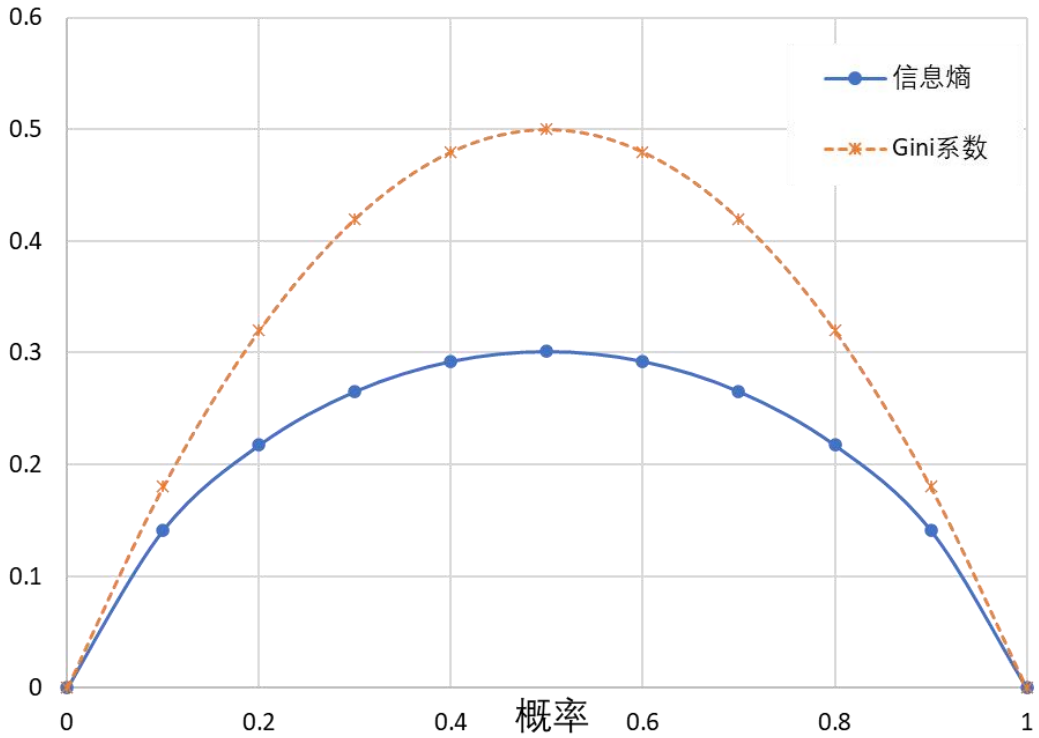
$$\mathrm{Gini}(\mathcal{D}) = 1 - \sum_{c=1}^{C} \left(\frac{|\mathcal{D}_c|}{|\mathcal{D}|}\right)^2$$

- 分裂成V个子集之后的基尼指数：$Gini_X(\mathcal{D}) = \sum_{v=1}^{V} \frac{|\mathcal{D}_v|}{|\mathcal{D}|} Gini(\mathcal{D}_v)$

信息熵 vs 基尼指数

- 假设类别数为2：

$$H(\mathcal{D}) = -p\log p - (1-p)\log(1-p) \qquad \mathrm{Gini}(\mathcal{D}) = p(1-p) + (1-p)p$$



CART

- ID3算法和C4.5算法构建的决策树模型都属于分类树；**CART算法既可构造分类树，也可构造回归树。**
- CART每次将数据集$\mathcal{D}$只划分为左右两个集合$\mathcal{D}_L$和$\mathcal{D}_R$，使得生成的每个非叶子结点都有两个分支（**二叉树**）。因此，增加节点时不仅要选择最佳特征，还要选择该特征的最佳划分阈值。
- 对候选划分$\theta=(X, t)$，选择特征X，分裂阈值设为t，定义数据集$\mathcal{D}$关于特征X和划分阈值t的不纯净度的度量为（$H(\cdot)$为某种不纯净度的度量）：

$$H(\mathcal{D}|\theta) = \frac{N_L}{N} H(\mathcal{D}_L) + \frac{N_R}{N} H(\mathcal{D}_R)$$

CART

- 回归：$H(\cdot)$为集合中样本响应y与响应均值$\bar{y} = \sum_{i=1}^{N} y_i$的残差平方和，也就是样本集合内样本的方差：$var(\mathcal{D}) = \sum_{i=1}^{N} (y_i - \bar{y})^2$
  - 集合内样本方差越小，表示该集合样本的响应变化越小，即数据越纯净。同时，$var(\mathcal{D})$越小，表示对该特征区域用响应均值$\bar{y}$预测时，预测残差平方最小。
- 分类：$H(\cdot)$为基尼指数。

CART

- 由于CART算法构造的决策树必是一颗二叉树，因此，对于特征为多于两个取值的情况（多元属性），需要进行二元划分。例如，对于年龄特征A有三个取值{青年，中年，老年}，可以得到三对不同取值形式，即：
  - 年龄=青年，年龄≠青年；
  - 年龄=中年，年龄≠中年；
  - 年龄=老年，年龄≠老年。
- 对于各取值形式（$i = 1, 2, 3$）划分得到的子数据集$\mathcal{D}_l$、$\mathcal{D}_r$，基尼指数为：

$$Gini_{A,i}(\mathcal{D}) = \frac{|\mathcal{D}_l|}{|\mathcal{D}|}\mathrm{Gini}(\mathcal{D}_l) + \frac{|\mathcal{D}_r|}{|\mathcal{D}|}\mathrm{Gini}(\mathcal{D}_r)$$

- 最后取其中最小基尼指数所对应的二元划分作为候选分支节点的判别条件。

CART

- 对于连续型特征，通常采用二分法将其离散化。
- 假设特征x_j在$\mathcal{D}$中出现了M个不同的取值，将这些值从小到大进行排序，记作$a_1, a_2, \cdots, a_M$，则共有$M-1$个候选划分点，依次为：$\frac{a_1+a_2}{2}, \frac{a_2+a_3}{2}, \cdots, \frac{a_{M-1}+a_M}{2}$。最后取其中最小基尼指数所对应的划分点作为候选分支节点的判别条件。
- 对于大数据集，训练数据中特征的取值数目M可能有很多，要考虑所有$M-1$个候选划分点的开销太大，此时可考虑将特征分成多个区间（等间隔划分或根据百分位数划分）。

本课内容

1. 决策树原理

2. 决策树的构建

3. 决策树构建举例

CART算法构造分类决策树举例

- 有一拖欠贷款人员训练样本数据集，使用CART算法基于该表数据构造分类决策树模型。

编号	房产状况	婚姻情况	年收入/千元	拖欠贷款
1	是	单身	125	否
2	否	已婚	100	否
3	否	单身	70	否
4	是	已婚	120	否
5	否	离异	95	是
6	否	已婚	60	否
7	是	离异	220	否
8	否	单身	85	是
9	否	已婚	75	否
10	否	单身	90	是

CART算法构造分类决策树举例

编号	房产状况	婚姻情况	年收入/千元	拖欠贷款
1	是	单身	125	否
2	否	已婚	100	否
3	否	单身	70	否
4	是	已婚	120	否
5	否	离异	95	是
6	否	已婚	60	否
7	是	离异	220	否
8	否	单身	85	是
9	否	已婚	75	否
10	否	单身	90	是

- 对于房产状况特征，根据是否有房划分数据集

  $\mathcal{D}$(有)={1,4,7}；$\mathcal{D}$(无)={2,3,5,6,8,9,10}

- $\mathcal{D}$(有)和$\mathcal{D}$(无)的基尼指数为：

  $Gini(\mathcal{D}(有))=1-(3/3)^2-(0/3)^2=0$

  $Gini(\mathcal{D}(无))=1-(4/7)^2-(3/7)^2=0.4849$

- 房产状况特征对$\mathcal{D}$进行子集划分时所得的基尼指数为：

  $Gini(\mathcal{D},房产状况)$

  $=3/10\times Gini(\mathcal{D}(有))+7/10\times Gini(\mathcal{D}(无))$

  $=0.343$

CART算法构造分类决策树举例

- 对婚姻情况特征划分，因为婚姻状况有三种，需对其构造二元划分：

“婚姻情况=已婚”和“婚姻情况≠已婚”

“婚姻情况=单身”和“婚姻情况≠单身”

“婚姻情况=离异”和“婚姻情况≠离异”

编号	房产状况	婚姻情况	年收入/千元	拖欠贷款
1	是	单身	125	否
2	否	已婚	100	否
3	否	单身	70	否
4	是	已婚	120	否
5	否	离异	95	是
6	否	已婚	60	否
7	是	离异	220	否
8	否	单身	85	是
9	否	已婚	75	否
10	否	单身	90	是

CART算法构造分类决策树举例

- 每种取值形式所对应的基尼指数分别为：

$$\text{Gini}(\mathcal{D},\text{婚姻})=4/10\times\text{Gini}(\mathcal{D}(\text{已婚}))+6/10\times\text{Gini}(\mathcal{D}(\neg\text{已婚}))=0.3$$

$$\text{Gini}(\mathcal{D},\text{婚姻})=4/10\times\text{Gini}(\mathcal{D}(\text{单身}))+6/10\times\text{Gini}(\mathcal{D}(\neg\text{单身}))=0.3667$$

$$\text{Gini}(\mathcal{D},\text{婚姻})=2/10\times\text{Gini}(\mathcal{D}(\text{离异}))+8/10\times\text{Gini}(\mathcal{D}(\neg\text{离异}))=0.4$$

- 对比上述计算结果，取分组：

“婚姻情况=已婚”和“婚姻情况≠已婚”

- 故有：$\text{Gini}(\mathcal{D},\text{婚姻})=0.3$

CART算法构造分类决策树举例

- 对于年收入特征，具体做法如下：

- 首先依据“年收入”特征取值对样本进行升序排序，从小到大依次用“年收入”特征相邻取值的均值作为划分阈值，将训练样本集划分为两个子集。结果如下表所示：

年收入	60		70		75		85		90		95		100		120		125		220
中间值	65		72．5		80		87．7		92．5		97．5		110		122．5		172．5		
	≤	>	≤	>	≤	>	≤	>	≤	>	≤	>	≤	>	≤	>	≤	>	
是	0	3	0	3	0	3	1	2	2	1	3	0	3	0	3	0	3	0	
否	1	6	2	5	3	4	3	4	3	4	3	4	4	3	5	2	6	1	
Gini	0.4		0.375		0.343		0.417		0.4		0.3		0.343		0.375		0.4		

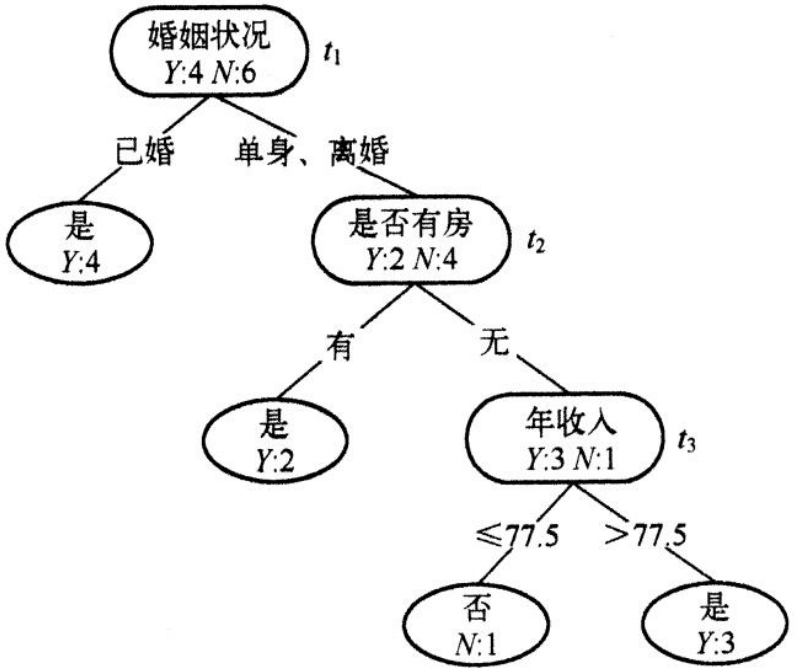
- 使用年收入特征对$\mathcal{D}$进行划分的最小基尼指数为：Gini($\mathcal{D}$, R=97.5)=0.3

CART算法构造分类决策树举例

- 婚姻情况和年收入特征所对应基尼指数并列最小，均为0.3。
- 不妨选取婚姻状况作为第一个划分点，将集合$\mathcal{D}$划分为$\mathcal{D}$(已婚)={2,4,6,9}和$\mathcal{D}$(¬已婚)={1,3,5,7,8,10}，得到如图所示的初始决策树。

CART算法构造分类决策树举例

- $\mathcal{D}$(已婚)中所有人均不欠贷款，故无需再划分；
- $\mathcal{D}$(¬已婚)递归调用上述过程继续划分，最后得到完整决策树，其中*Y*和*N*分别表示两类不同取值样本数目。



决策树模型的优点

- 容易解释
- 对特征预处理要求少
  - 理论上能处理离散值和连续值混合的输入
  - 对特征的单调变换不敏感 (只与数据的排序有关)
  - 能自动进行特征选择
  - 可处理缺失数据
- 可扩展到大数据规模

决策树模型的缺点

- 正确率不高：建树过程过于贪心
  - 可作为Boosting的弱学习器（深度不太深）
- 模型不稳定（方差大）：输入数据小的变化会带来树结构的变化
  - Bagging：随机森林
- 当特征数目相对样本数目太多时，容易过拟合