第9章　判定树

刘家锋

哈尔滨工业大学

第9章　判定树

1. 9.1 判定树的概念
2. 9.2 判定树的学习
3. 9.3 剪枝
4. 9.4 连续特征判定树

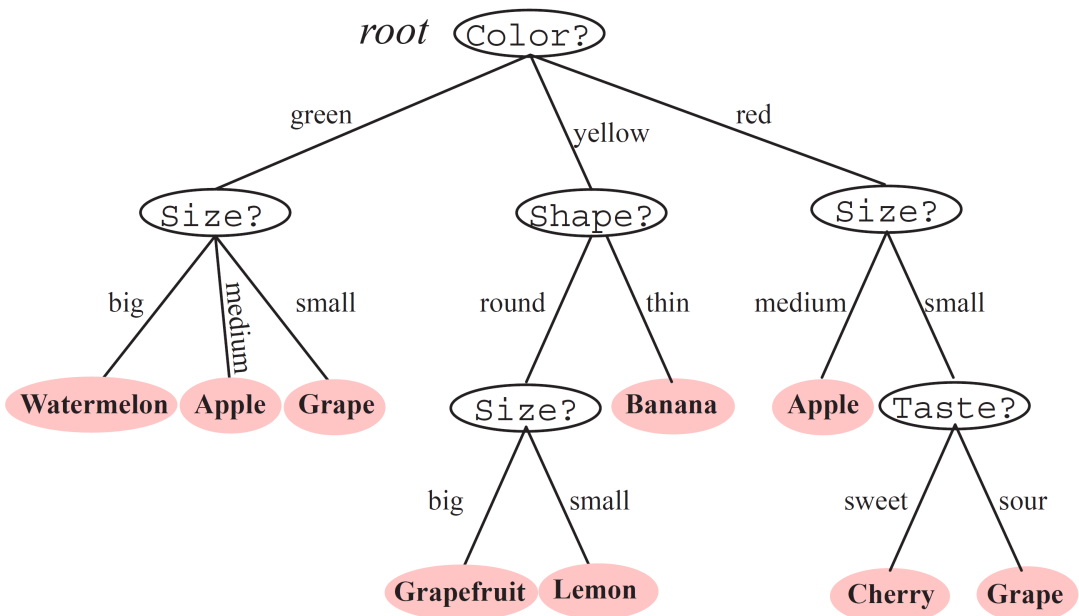
9.1 判定树的概念

判定树

- **判定树**
  - 判定树是一种常用的模式分类方法，也被称为决策树；
  - 判定树适用于离散特征分类问题，也可用于连续特征分类；
- **判定树的结构**
  - 叶节点：对应一个类别；
  - 中间节点：对应某个特征，节点下的分支对应该特征的某个取值；

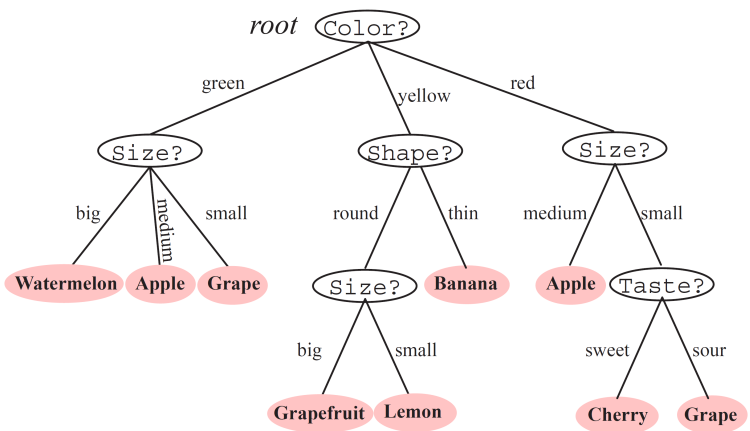
判定树分类

- **判定树的识别过程**
  - 从根节点开始，测试节点的特征，根据待识样本的特征，决定下行分支；
  - 在叶节点得到样本的类别；



判定树与规则

- **从判定树到规则**
  - 一颗判定树对应一个规则集；
  - 每条从根节点到叶节点的分支路径对应一条规则；



(颜色 = 绿) ∧ (大小 = 大) ⟶ 西瓜

[(颜色 = 绿) ∧ (大小 = 中)]∨

[(颜色 = 红) ∧ (大小 = 中)] ⟶ 苹果

[(颜色 = 绿) ∧ (大小 = 小)] ∨ [(颜色 = 红)∧

(大小 = 小) ∧ (味道 = 酸)] ⟶ 葡萄

(颜色 = 黄) ∧ (形状 = 圆) ∧ (大小 = 大) ⟶ 柚子

. . . . . .

9.2 判定树的学习

判定树的学习问题

- **学习问题**
  - 给定训练样本集合：$D = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$
  - 给定特征集合：$A = \{a_1, \cdots, a_d\}$
  - 构造判定树，既能够分类训练集D中的样本，也能够分类未来的测试样本$\mathbf{x}$；
- **判定树的构造**
  - 采用“分而治之”的策略，在每个节点选择一个特征，将训练集分成若干子集，每个子集对应一个特征取值；
  - 直到子集中只包含属于某一类的样本为止，将其作为叶节点，标注为该类别；

基本算法

Algorithm 1 判定树学习算法

Input: 训练集$D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$，特征集$A = \{a_1, \cdots, a_d\}$;
Output: 以node为根节点的一棵决策树

```
function TreeGenerate(D, A)
    生成节点node;
    if 满足递归停止条件 then 标记节点；return
    end if
    从A中选择最优划分特征a_*;
    for a_*的每一个值a_*^v do
        为node生成一个分支，D_v表示D中a_*取值a_*^v的样本子集;
        if D_v = Φ then
            分支标记叶节点为D中样本最多类；return ;
        else
            以TreeGenerate(D_v, A \ {a_*})为分支节点;
        end if
    end for
end function
```

判定树的构造

- **判定树学习算法**
  - 判定树的构造是一个递归的过程；
  - 递归的终止条件：
    1. 输入样本集都属于同一类别，节点标记为该类别；
    2. 输入属性集为空，或样本的所有属性相同，标记为样本最多的类别；
  - 判定树的构造，关键是如何选择“最优”的特征a^*；

最优特征

- **划分特征选择的原则**
  - 判定树根节点对应的样本集合包含所有类别的样本，而叶节点只包含一个类别的样本；
  - 判定树的构建过程可以看作是一个使得样本集合越来越“纯净”的过程；
  - 如果能够定义一个度量集合“纯度”的指标，在每个节点上应该选择使得“纯度”增加最快的属性来划分样本集合；

Information Entropy

- **信息熵**
  - 信息熵是度量样本集合纯度最常用的一个指标；
  - 样本集合D中第k类样本所占比例为p_k，D的信息熵定义为：

$$\mathrm{Ent}(D) = -\sum_{k=1}^{c} p_k \log_2 p_k$$

  - $\mathrm{Ent}(D)$的值越小，表示纯度越高，当所有样本属于一个类别时，取得最小值$\mathrm{Ent}(D) = 0$；（约定$p = 0$，$p \log_2 p = 0$）
  - $\mathrm{Ent}(D)$的最大值为$\log_2 c$；

ID3算法

- **信息增益**
  - 离散特征a有V个可能的取值：$\{a^1, \cdots, a^V\}$；
  - D^v表示D中特征$a = a^v$的样本集合，以特征a对数据集D进行划分的信息增益为：

$$\mathsf{Gain}(D, a) = \mathsf{Ent}(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} \mathsf{Ent}(D^v)$$

  - 第1项$\mathsf{Ent}(D)$是划分前数据集的信息熵；
  - 第2项是使用特征a划分之后的信息熵，$|D^v|/|D|$是每个分支的权重，样本越多权重越大；
  - ID3算法依据信息增益选择最优的划分特征：

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

例9.1

包含14个样本的数据集

编号	天气	温度	湿度	风力	打网球
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

例9.1

计算数据集D的信息熵，两个类别的占比：$p_1 = 9/14,\ p_2 = 5/14$;

$$\mathsf{Ent}(D) = -\sum_{k=1}^{2} p_k \log_2 p_k = -\left(\frac{9}{14}\log_2\frac{9}{14} + \frac{5}{14}\log_2\frac{5}{14}\right) = 0.9403$$

以特征“天气”划分为3个子集：

$$\begin{aligned} D^1(\mathsf{Sunny}) &= \{1,2,8,9,11\}, & p_1 &= 2/5,\ p_2 = 3/5 \\ D^2(\mathsf{Overcast}) &= \{3,7,12,13\}, & p_1 &= 4/4,\ p_2 = 0/4 \\ D^3(\mathsf{Rain}) &= \{4,5,6,10,14\}, & p_1 &= 3/5,\ p_2 = 2/5 \end{aligned}$$

计算子集的信息熵：

$$\begin{aligned} \mathsf{Ent}(D^1) &= -\left(\frac{2}{5}\log_2\frac{2}{5} + \frac{3}{5}\log_2\frac{3}{5}\right) = 0.971 \\ \mathsf{Ent}(D^2) &= -\left(\frac{4}{4}\log_2\frac{4}{4} + \frac{0}{4}\log_2\frac{0}{4}\right) = 0 \\ \mathsf{Ent}(D^3) &= -\left(\frac{3}{5}\log_2\frac{3}{5} + \frac{2}{5}\log_2\frac{2}{5}\right) = 0.971 \end{aligned}$$

例9.1

计算特征“天气”的信息增益：

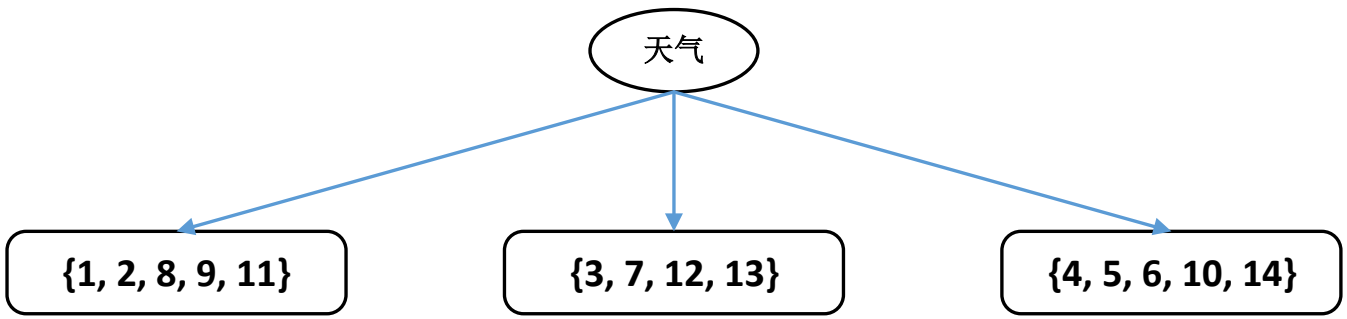
$$\begin{aligned}\mathsf{Gain}(D, \text{天气}) &= \mathsf{Ent}(D) - \sum_{v=1}^{3} \frac{|D^v|}{|D|} \mathsf{Ent}(D^v) \\ &= 0.9403 - \left(\frac{5}{14} \times 0.971 + \frac{4}{14} \times 0 + \frac{5}{14} \times 0.971 \right) \\ &= 0.2467\end{aligned}$$

同样方法，计算其它特征的信息增益：

$$\begin{aligned}\mathsf{Gain}(D, \text{温度}) &= 0.029 \\ \mathsf{Gain}(D, \text{湿度}) &= 0.151 \\ \mathsf{Gain}(D, \text{风力}) &= 0.048\end{aligned}$$

例9.1

“天气”的信息增益最大，选择作为划分特征



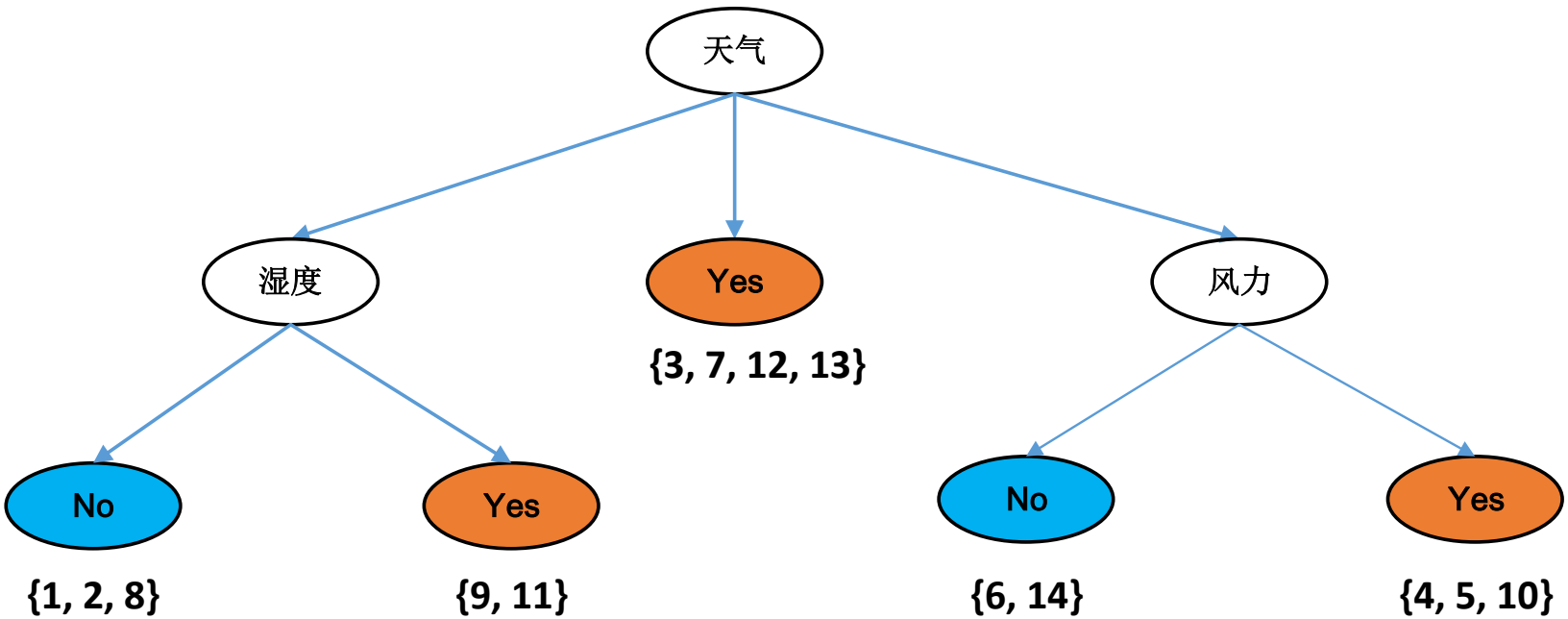
D^2全部为第1类，满足终止条件，标注为“Yes”；

D^1递归构造判定树：计算特征“温度”、“湿度”、“风力”的信息增益，选择“湿度”划分样本集；

D^3递归构造判定树：计算特征“温度”、“湿度”、“风力”的信息增益，选择“风力”划分样本集；

例9.1

ID3算法构造的完整判定树：



CART算法

- **基尼指数**
  - 基尼值是样本集D纯度的另外一种度量：

$$\text{Gini}(D) = \sum_{k=1}^{c} \sum_{k' \neq k} p_k p_{k'} = 1 - \sum_{k=1}^{c} p_k^2$$

  - 基尼值反映了从样本集D中随机抽取两个样本，其类别不一致的概率，值越小纯度越高；
  - CART算法选择基尼指数最小的特征作为划分特征：

$$\text{Gini\_index}(D, a) = \sum_{v=1}^{V} \frac{|D^v|}{|D|} \text{Gini}(D^v)$$

9.3 剪枝

剪枝

- **决策树是容易过拟合的模型**
  - 一般来说，决策树可以完美地预测训练数据，但对测试数据可能预测效果会很差；
  - 决策树的分支过多，从根节点到叶节点的路径过长都是过拟合的表现；
- **剪枝处理**
  - 剪枝是控制决策树过拟合的主要手段；
  - 预剪枝：决策树生成过程中，估计当前节点的划分是否能够带来泛化性能的提升，决定是否停止划分节点；
  - 后剪枝：先生成完整的决策树，然后自底向上考察将非叶节点替换为叶节点，能否带来泛化性能的提升；

预剪枝

- **分支停止的条件**
  - 信息增益阈值：设置阈值β，当$\max_a \mathsf{Gain}(D, a) < \beta$时停止当前节点继续分支;
  - 最小化全局目标：

$$a \times \mathsf{size} + \sum_{v \in \text{所有叶节点}} \mathsf{Ent}(D^v)$$

    其中，size度量决策树的复杂程度，如分支数、节点数等;
  - 验证技术：用部分训练样本作为验证集，持续节点分支，直到验证集的分类误差最小为止;
- **标注节点**
  - 停止分支的节点，标注为当前训练集中样本最多的类别;

后剪枝

- **后剪枝过程**
  - 判定树充分生长，直到叶节点都有最小的不纯度为止;
  - 对所有具有公共父节点的叶节点，考虑是否可以合并：
- **合并的条件**
  - 不纯度：合并后只引起很小的不纯度增加，则合并叶节点；
  - 验证技术：如果合并后能够提高验证集的分类正确率，则合并叶节点；

9.4 连续特征判定树

连续值特征

- **连续值的处理**
  - 基本思路：连续特征离散化；
  - 二分法：连续特征a设定划分点t，将样本集D划分为两个子集，$a \geq t$的样本作为D_t^+，$a < t$的样本作为D_t^-；
  - 特征a在D中有m个取值，由小到大排列为$\{a^1, \cdots, a^m\}$；
  - $t \in [a^i, a^{i+1})$的任意取值，划分效果相同，包含$m-1$个候选划分点的集合：

$$T_a = \left\{ \frac{a^i + a^{i+1}}{2} \mid 1 \leq i \leq m-1 \right\}$$

连续值特征

- **连续值特征的信息增益**
  - 特征a以t为划分点的信息增益：

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

  - 选择最优的划分点t来划分，特征a的信息增益：

$$\begin{aligned}\mathsf{Gain}(D,a)&=\max_{t\in T_a}\mathsf{Gain}(D,a,t)\\&=\max_{t\in T_a}\left\{\mathsf{Ent}(D)-\sum_{\lambda\in\{-,+\}}\frac{|D_t^\lambda|}{|D|}\mathsf{Ent}(D_t^\lambda)\right\}\end{aligned}$$

- **连续值特征的选择**
  - 每个节点按照$\mathsf{Gain}(D,a)$来选择划分特征；
  - 与离散特征不同，节点划分特征为连续特征，该特征仍可作为后代节点的划分特征；

例9.2

包含16个样本的数据集

编号	特征1	特征2	类别	编号	特征1	特征2	类别
1	0.15	0.83	ω_1	9	0.10	0.20	ω_2
2	0.09	0.55	ω_1	10	0.08	0.15	ω_2
3	0.29	0.35	ω_1	11	0.23	0.16	ω_2
4	0.38	0.70	ω_1	12	0.70	0.19	ω_2
5	0.52	0.48	ω_1	13	0.62	0.47	ω_2
6	0.57	0.73	ω_1	14	0.91	0.27	ω_2
7	0.73	0.75	ω_1	15	0.65	0.90	ω_2
8	0.47	0.08	ω_1	16	0.75	0.36	ω_2

例9.2

训练集的信息熵：

$$\mathsf{Ent}(D) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$$

特征1由小到大排序：0.08, 0.09, 0.10, 0.15, 0.23, 0.29, 0.38, 0.47, 0.52, 0.57, 0.62, 0.65, 0.70, 0.73, 0.75, 0.91

得到特征1的划分点集合：

$$T_{x_1} = \{0.085, 0.095, 0.125, 0.19, 0.26, 0.335, 0.425, 0.495, 0.545, 0.595, 0.635, 0.675, 0.715, 0.74, 0.83\}$$

当划分点$t = 0.595$时，特征1的信息增益最大：

$$\begin{aligned}\mathsf{Gain}(D, x_1, 0.595) = 1 &- \frac{10}{16}\left[-\frac{7}{10}\log_2\frac{7}{10} - \frac{3}{10}\log_2\frac{3}{10}\right] \\ &- \frac{6}{16}\left[-\frac{1}{6}\log_2\frac{1}{6} - \frac{5}{6}\log_2\frac{5}{6}\right] = 0.0592\end{aligned}$$

例9.2

特征2由小到大排序：0.08, 0.15, 0.16, 0.19, 0.20, 0.27, 0.35, 0.36, 0.47, 0.48, 0.55, 0.70, 0.73, 0.75, 0.83, 0.90

得到特征2的划分点集合：

$$T_{x_2} = \{0.115, 0.155, 0.175, 0.195, 0.235, 0.31, 0.355, 0.415, 0.475, 0.515, 0.625, 0.715, 0.74, 0.79, 0.865\}$$

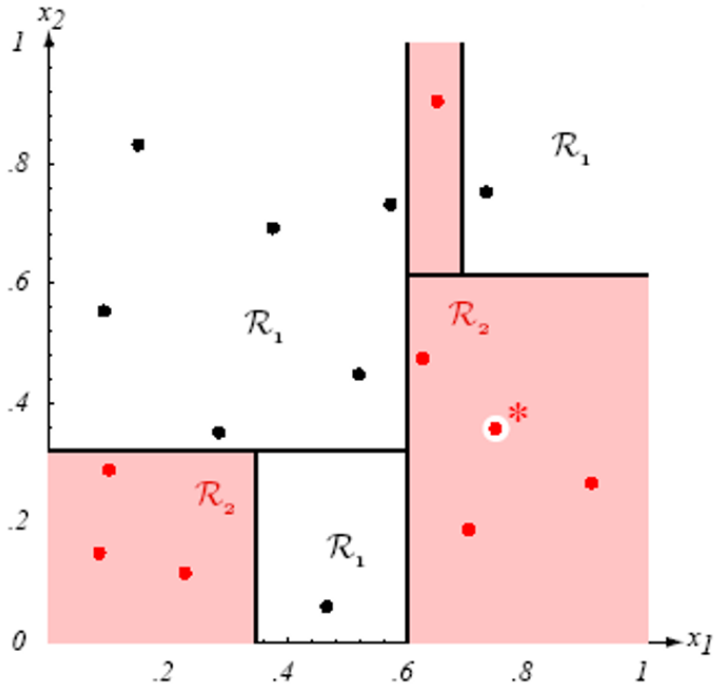
当划分点$t = 0.31$时，特征2的信息增益最大：

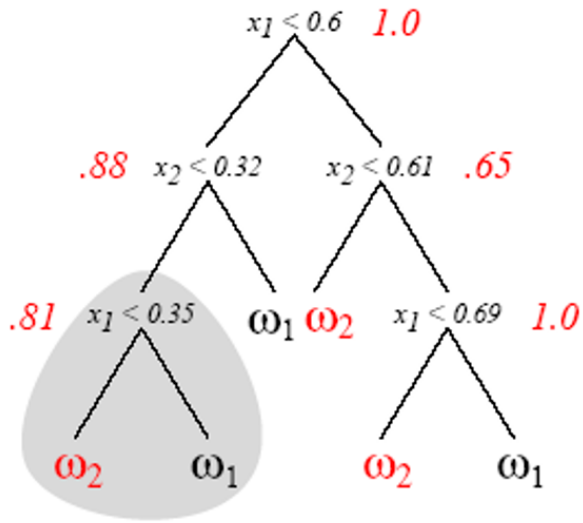
$$\begin{aligned}\text{Gain}(D, x_2, 0.31) = 1 &- \frac{10}{16}\left[-\frac{7}{10}\log_2\frac{7}{10} - \frac{3}{10}\log_2\frac{3}{10}\right] \\ &- \frac{6}{16}\left[-\frac{1}{6}\log_2\frac{1}{6} - \frac{5}{6}\log_2\frac{5}{6}\right] = 0.0592\end{aligned}$$

两个特征的最优信息增益相同，可以选择x_1作为划分特征，划分点为$t = 0.595$；

... ...

例9.2


x2
1
.8
.6
.4
.2
0
.2
.4
.6
.8
1
x1
R1
R1
R2
R2
R1
*


x1 < 0.6
1.0
.88
x2 < 0.32
x2 < 0.61
.65
.81
x1 < 0.35
ω1
ω2
x1 < 0.69
1.0
ω2
ω1
ω2
ω1

多变量决策树

- **斜线分类边界**
  - 每个节点采用多个特征的线性组合来划分，也就是用一个线性分类器来判别不同分支；
  - 可以用一个简单的决策树和对应的斜线分类面近似真实分类边界；