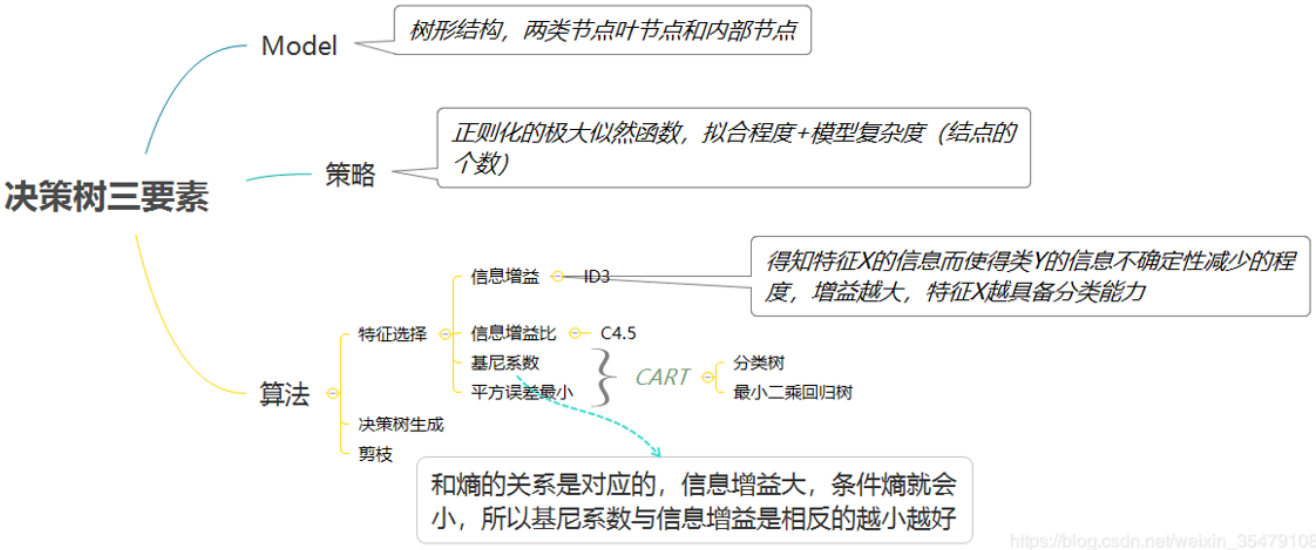
人工智能的数学基础

华东师范大学 数学科学学院 黎芳(教授) 2019年9月30日

Chapter 5 决策树



5.1 决策树模型与学习

决策树(decisiontree)是一种基本的分类与回归方法.

5.1.1 决策树模型

定义**5.1**（决策树）分类决策树模型是一种描述对实例进行分类的树形结构．决策树由结点(node)和有向边(directed edge)组成．结点有两种类型：内部结点(internal node)和叶结点(leaf node). 内部结点表示一个特征或属性，叶结点表示一个类.

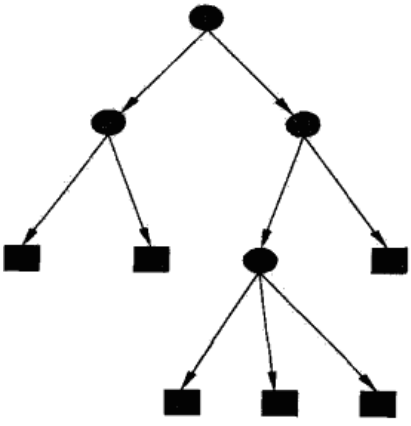


图 5.1　决策树模型

5.1.2 决策树与if-then规则

可以将决策树看成一个if-then规则的集合. 由决策树的根结点到叶结点的每一条路径构建一条规则；路径上内部结点的特征对应着规则的条件，而叶结点的类对应着规则的结论.

互斥且完备--每一个实例都被一条路径或一条规则所覆盖，而且只被一条路径或一条规则所覆盖.

5.1.3 决策树与条件概率分布

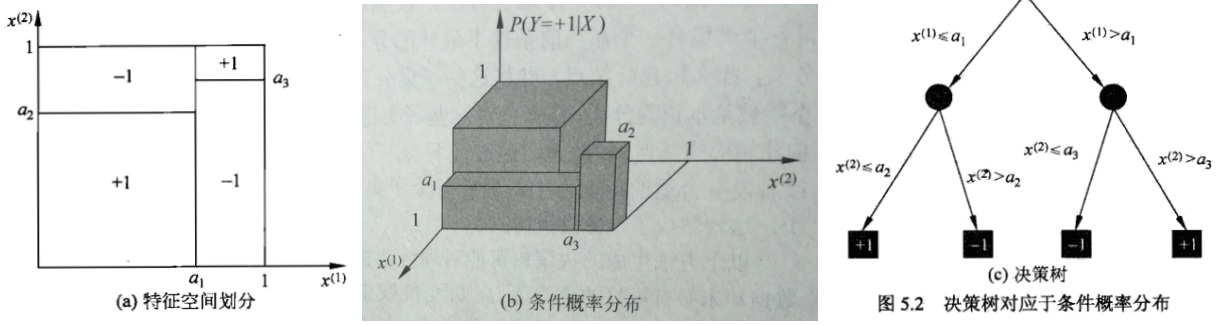


图 5.2 决策树对应于条件概率分布

$P(Y=+1|X=c)>0.5$

5.1.4 决策树学习

假设给定训练数据集$D=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}$,其中，$x_i=\left(x_i^{(1)},x_i^{(2)},\cdots,x_i^{(n)}\right)^{\mathrm{T}}$为输入实例（特征向量），n为特征个数，$y_i\in\{1,2,\cdots,K\}$为类标记，$i=l,2,\cdot,N$, N为样本容量.

学习的目标是根据给定的训练数据集构建一个决策树模型，使它能够对实例进行正确的分类.

- 决策树学习本质上是从训练数据集中归纳出一组分类规则.
- 决策树学习的算法通常是一个递归地选择最优特征，并根据该特征对训练数据进行分割，使得对各个子数据集有一个最好的分类的过程.
- 决策树学习算法包含特征选择、决策树的生成与决策树的剪枝过程

5.2 特征选择

5.2.1 特征选择问题

表 5.1　贷款申请样本数据表

ID	年龄	有工作	有自己的房子	信贷情况	类别
1	青年	否	否	一般	否
2	青年	否	否	好	否
3	青年	是	否	好	是
4	青年	是	是	一般	是
5	青年	否	否	一般	否
6	中年	否	否	一般	否
7	中年	否	否	好	否
8	中年	是	是	好	是
9	中年	否	是	非常好	是
10	中年	否	是	非常好	是
11	老年	否	是	非常好	是
12	老年	否	是	好	是
13	老年	是	否	好	是
14	老年	是	否	非常好	是
15	老年	否	否	一般	否

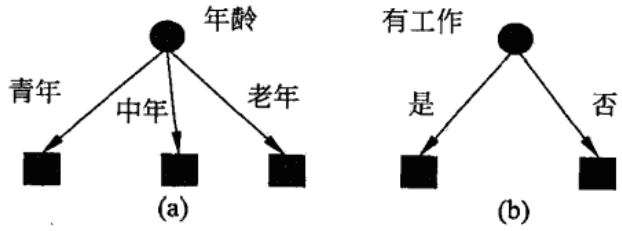
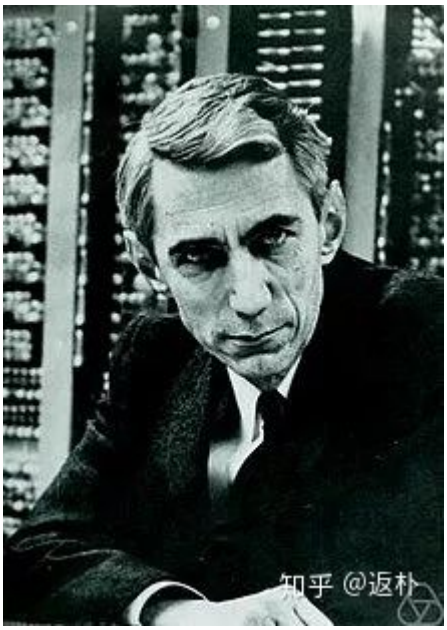


图 5.3　不同特征决定的不同决策树

到底选哪一个？分类能力强的

5.2.2 信息增益

信息熵（entropy）



- Shannon,

https://www.zhihu.com/question/22178202

设X是一个取有限个值的离散随机变量，其概率分布为

$P(X = x_i) = p_i, \quad i = 1, 2, \cdots, n$

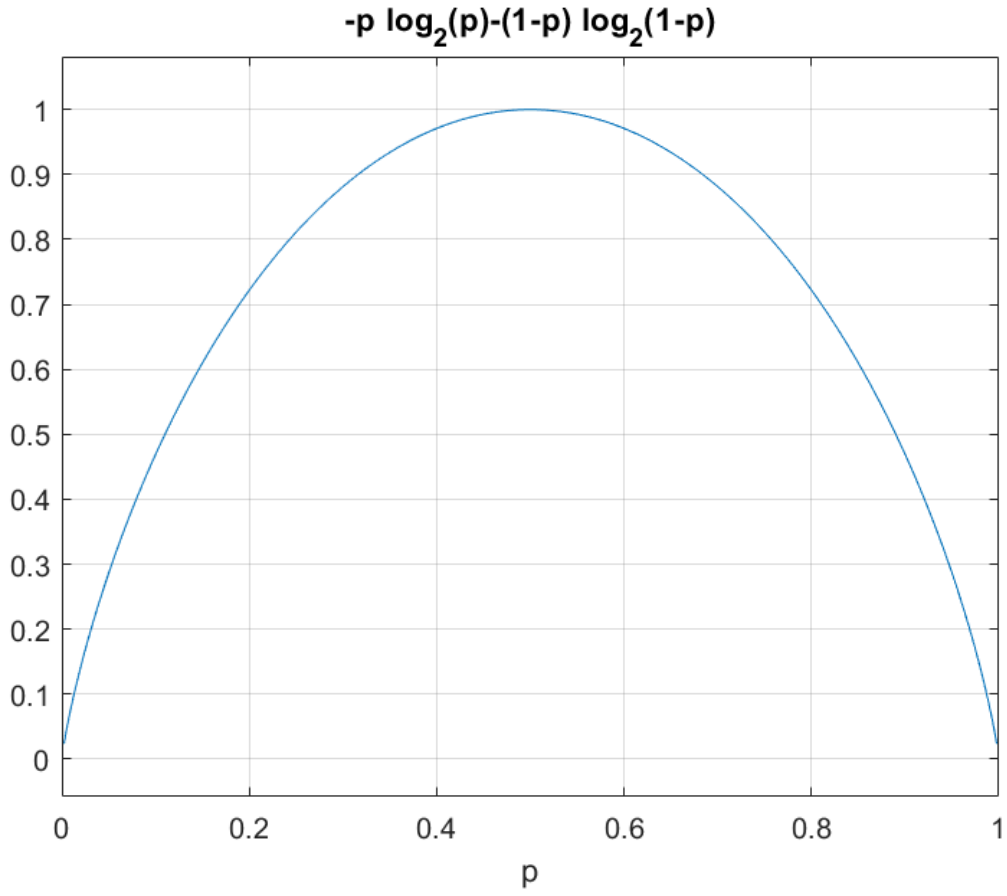
熵：$H(X) = -\sum_{i=1}^{n} p_i \log p_i$ 也可以记作 $H(p)$

$0 \le H(p) \le \log n$

例：$P(X = 1) = p, \quad P(X = 0) = 1 - p, \quad 0 \le p \le 1$

$H(p) = -p \log_2 p - (1 - p) \log_2 (1 - p)$

```
ezplot('-p*log2(p)-(1-p)*log2(1-p)',[0,1]); grid on
```



设有随机变量 (X, Y)，其联合概率分布为

$P(X = x_i, Y = y_j) = p_{ij}, \quad i = 1, 2, \cdots, n; \quad j = 1, 2, \cdots, m$

条件熵：$H(Y|X) = \sum_{i=1}^{n} p_i H(Y|X = x_i)$

经验熵和经验条件熵

信息增益 (information gain) 表示得知特征X 的信息而使得类Y的信息的不确定性减少的程度.

定义 **5.2** （信息增益） 特征A 对训练数据集D的信息增益$g(D, A)$, 定义为集合D的经验熵H(D)与特征A给定条件下D的经验条件熵 $H(D|A)$之差， 即

$$g(D, A) = H(D) - H(D|A)$$

记号

- D--训练数据集, $|D|$--样本个数.
- C_k--K个类, $|C_k|$ --类C_k的样本个数， $\sum_{k=1}^{K} |C_k| = |D|$.
- $\{a_1, a_2, \cdots, a_n\}$-- A有 n个不同的取值.
- D_i--根据特征A 将D划分为 n 个子集, $|D_i|$--D_i的样本个数， $\sum_{i=1}^{n} |D_i| = |D|$.
- D_{ik}-- D_i中属于类C_k 的样本的集合，即$D_{ik} = D_i \cap C_k$, $|D_{ik}|$-- D_{ik}的样本个数.

算法 **5.1** (信息增益的算法)

输入：训练数据集D和特征A:

输出：特征A对训练数据集D的信息增益$g(D, A)$

1. 计算 $H(D) = -\sum_{k=1}^{K} \frac{|C_k|}{|D|} \log_2 \frac{|C_k|}{|D|}$
2. 计算 $H(D|A) = \sum_{i=1}^{n} \frac{|D_i|}{|D|} H(D_i|A) = -\sum_{i=1}^{n} \frac{|D_i|}{|D|} \sum_{k=1}^{K} \frac{|D_{ik}|}{|D_i|} \log_2 \frac{|D_{ik}|}{|D_i|}$
3. 计算$g(D, A) = H(D) - H(D|A)$

例**5.2** 对表5.1所给的训练数据集D, 根据信息增益准则选择最优特征.

$$H(D) = -\frac{9}{15} \log_2 \frac{9}{15} - \frac{6}{15} \log_2 \frac{6}{15} = 0.971$$

A_1, A_2, A_3, A_4分别表示年龄，有工作，有自己的房子和信贷情况四个特征.

$$
\begin{aligned}
g(D, A_1) &= H(D) - \left[\frac{5}{15}H(D_1) + \frac{5}{15}H(D_2) + \frac{5}{15}H(D_3)\right] \\
&= 0.971 - \left[\frac{5}{15}\left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right)\right. \\
&\left. + \frac{5}{15}\left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right) + \frac{5}{15}\left(-\frac{4}{5}\log_2\frac{4}{5} - \frac{1}{5}\log_2\frac{1}{5}\right)\right] \\
&= 0.971 - 0.888 = 0.083
\end{aligned}
$$

$$
\begin{aligned}
g(D, A_2) &= H(D) - \left[\frac{5}{15}H(D_1) + \frac{10}{15}H(D_2)\right] \\
&= 0.971 - \left[\frac{5}{15} \times 0 + \frac{10}{15}\left(-\frac{4}{10}\log_2\frac{4}{10} - \frac{6}{10}\log_2\frac{6}{10}\right)\right] = 0.324
\end{aligned}
$$

$$
\begin{aligned}
g(D, A_3) &= 0.971 - \left[\frac{6}{15} \times 0 + \frac{9}{15}\left(-\frac{3}{9}\log_2\frac{3}{9} - \frac{6}{9}\log_2\frac{6}{9}\right)\right] \\
&= 0.971 - 0.551 = 0.420
\end{aligned}
$$

$$g(D, A_4) = 0.971 - 0.608 = 0.363$$

5.2.3 信息增益比

$$g_R(D, A) = \frac{g(D, A)}{H(D)}$$

5.3 决策树的生成

5.3.1 ID3算法

ID3由Ross Quinlan在1986年提出。

算法**5.2 (ID3**算法）

输入：训练数据集D,特征集A,阈值ϵ

输出：决策树T.

1. 若D中所有实例属于同一类C_k，则T为单结点树，并将类C_k作为该结点的类标记，返回T.
2. 若$A = \varnothing$，则T为单结点树，并将D中实例数最大的类C_k作为该结点的类标记，返回T.
3. 否则，按算法5.1计算A中各特征对D的信息增益，选择信息增益最大的特征A_g.
4. 如果A_g的信息增益小于阈值ϵ,则置T为单结点树，并将D中实例数最大的类C_k，作为该结点的类标记，返回T.
5. 否则，对A_g的每一可能值a_i，依$A_g = a_i$，将D分割为若干非空子集D_i，将 D_i中实例数最大的类作为标记，构建子结点，由结点及其子结点构成树T，返回T.
6. 对第i个子结点，以D_i为训练集，以$A - \{A_g\}$为特征集，递归地调用步(1)-步(5)，得到子树T_i，返回T_i.

例**5.3** 对表5.1，利用ID3建立决策树

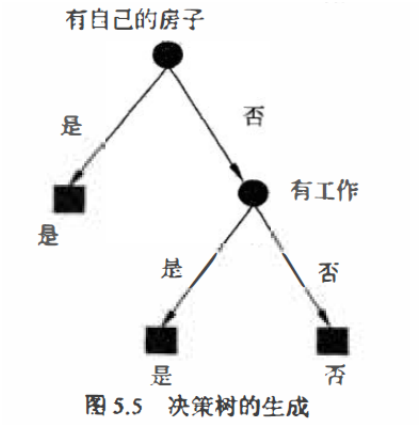


图 5.5 决策树的生成

$$g(D_2, A_1) = H(D_2) - H(D_2|A_1) = 0.918 - 0.667 = 0.251$$
$$g(D_2, A_2) = H(D_2) - H(D_2|A_2) = 0.918$$
$$g(D_2, A_4) = H(D_2) - H(D_2|A_4) = 0.474$$

5.3.2 C4.5的生成算法

C4.5是Ross Quinlan在1993年在ID3的基础上改进而提出的。ID3采用的信息增益度量存在一个缺点，它一般会优先选择有较多属性值的Feature,因为属性值多的Feature会有相对较大的信息增益?(信息增益反映的给定一个条件以后不确定性减少的程度,必然是分得越细的数据集确定性更高,也就是条件熵越小,信息增益越大).为了避免这个不足, C4.5中是用信息增益比率(gain ratio)来作为选择分支的准则。信息增益比率通过引入一个被称作分裂信息(Split information)的项来惩罚取值较多的Feature。除此之外，C4.5还弥补了ID3中不能处理特征属性值连续的问题。

信息增益比: $g_R(D, A) = \frac{g(D, A)}{H_A(D)}$, 其中 $H_A(D) = -\sum_{i=1}^{n} \frac{|D_i|}{|D|} \log_2 \frac{|D_i|}{|D|}$(D A 关于特征 的熵)

5.4 决策树的剪枝

上述生成算法存在过拟合的缺点,剪枝可以简化模型.

设树T的叶结点个数为$|T|$, t是树T的叶结点，叶结点有N_t个样本点，其中K类的样本点有N_{tk}, $H_t(T)$为叶结点t上的经验熵，$\alpha \geq 0$为参数，则决策树学习的损失函数可以定义为

$$C_\alpha(T) = \sum_{t=1}^{|T|} N_t H_t(T) + \alpha|T| = C(T) + \alpha|T|$$

其中经验熵为

$$H_t(T) = -\sum_k \frac{N_{tk}}{N_t} \log \frac{N_{tk}}{N_t}$$

$C(T)$表示模型对训练数据的预测误差, 即模型与训练数据的拟合程度, $|T|$表示模型复杂度.

算法5.4（树的剪枝算法）

输入：生成算法产生的整个树T，参数α.

输出：修剪后的子树T_α.

1. 计算每个结点的经验熵.
2. 递归地从树的叶结点向上回缩. 设一组叶结点回缩到其父结点之前与之后的整体树分别为T_B与T_A，其对应的损失函数值分别是$C_\alpha(T_B)$与$C_\alpha(T_A)$，如果$C_\alpha(T_A) \leq C_\alpha(T_B)$ 则进行剪枝， 即将父结点变为新的叶结点.
3. 返回(2)，直至不能继续为止， 得到损失函数最小的子树T_α.

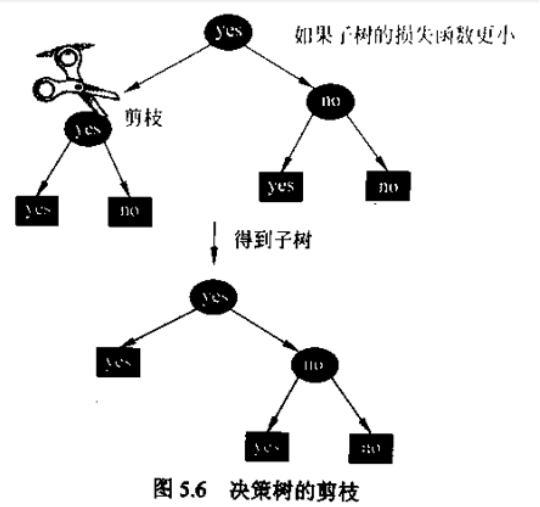


图 5.6 决策树的剪枝

5.5 CART算法

分类与回归树(classification and regression tree, CART)模型由Breiman等人在1984年提出， 是应用广泛的决策树学习方法. CART同样由特征选择、树的生成及剪枝组成， 既可以用于分类也可以用于回归.

5.5.1 CART生成

1. 回归树的生成

一颗回归树对应着输入空间（即特征空间）的一个划分以及在划分的单元上的输出值. 假设已将输入空间划分为M个单元$R_1, R_2, \cdots, R_M$，并且在每个单元R_m上有一个固定的输出值c_m，于是回归树模型可以表示为

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

最小二乘回归：平方误差 $\sum_{x_i \in R_m} (y_i - f(x_i))^2$

算法 5.5（最小二乘回归树生成算法）

输入：训练数据集 D；

输出：回归树 $f(x)$.

在训练数据集所在的输入空间中，递归地将每个区域划分为两个子区域并决定每个子区域上的输出值，构建二叉决策树：

（1）选择最优切分变量 j 与切分点 s，求解

$$\min_{j,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2\right] \tag{5.21}$$

遍历变量 j，对固定的切分变量 j 扫描切分点 s，选择使式 (5.21) 达到最小值的对 (j,s).

（2）用选定的对 (j,s) 划分区域并决定相应的输出值：

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

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

（3）继续对两个子区域调用步骤 (1)，(2)，直至满足停止条件.

（4）将输入空间划分为 M 个区域 $R_1,R_2,\cdots,R_M$，生成决策树：

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

■

CART回归树简单实例

臂长（m）	年龄(岁)	体重（kg）	身高（m）(标签值)
0.5	5	20	1.1
0.7	7	30	1.3
0.9	21	70	1.7

训练数据中臂长，年龄，体重为特征变量X，身高为标签值Y，下面开始种树

1、首先将第一个特征的第一个值作为切割点（0.5），则划分的两个空间记为R_1,R_2

$R_1=\{0.5,5,20\},\ R_2=\{(0.7,7,30),(0.9,21,70)\}$

$c_1=\{1.1\},\ c_2=\frac{1}{2}(1.3+1.7)=1.5$

则平方误差（（真实值-预测值）的平方）为

$m(0.5)=(1.1-1.1)^2+(1.5-1.3)^2+(1.5-1.7)^2=0.08$

2、将第一个特征的第二个变量（0.7）作为切割点，类比第一步，划分的两个空间记为R_1,R_2

$R_1=\{(0.5,5,20),(0.7,7,30)\},\ R_2=\{(0.9,21,70)\}$

$c_1=\frac{1}{2}(1.1+1.3)=1.2,\ c_2=\{1.7\}$

则平方误差

$m(0.5)=0.02+0=0.02$

所以对于固定了特征后，从上面的MSE得出，所以特征“臂长=0.7”为切分点。同理。对于特征年龄，也可以采取上述的方式寻找最佳切分点，这样遍历了所有的特征，寻找平方误差最小的对(j,s)，j表示第j个特征，s表示第j个特征的第s个值。本例中最佳切分点为0.7，所以以此将特征空间划分为两个区域R_1, R_2.

3、对于第二步得到的R_1和R_2，分别再次求最佳切分点，递归操作，过程同1~2。

2 分类树生成

定义5.4（基尼指数）分类问题中，假设有K个类，样本点属于第k类的概率为p_k，则概率分布的基尼指数定义为

$$\text{Gini}(p) = \sum_{k=1}^{K} p_k(1-p_k) = 1 - \sum_{k=1}^{K} p_k^2$$

对于二类分类问题，若样本点属于第1个类的概率是p，则概率分布的基尼指数为

$$\text{Gini}(p) = 2p(1-p)$$

对于给定的样本集合D，其基尼指数为

$$\text{Gini}(D) = 1 - \sum_{k=1}^{K} \left(\frac{|C_k|}{|D|} \right)^2$$

这里，C_k是D中属于第k类的样本子集，K是类的个数.

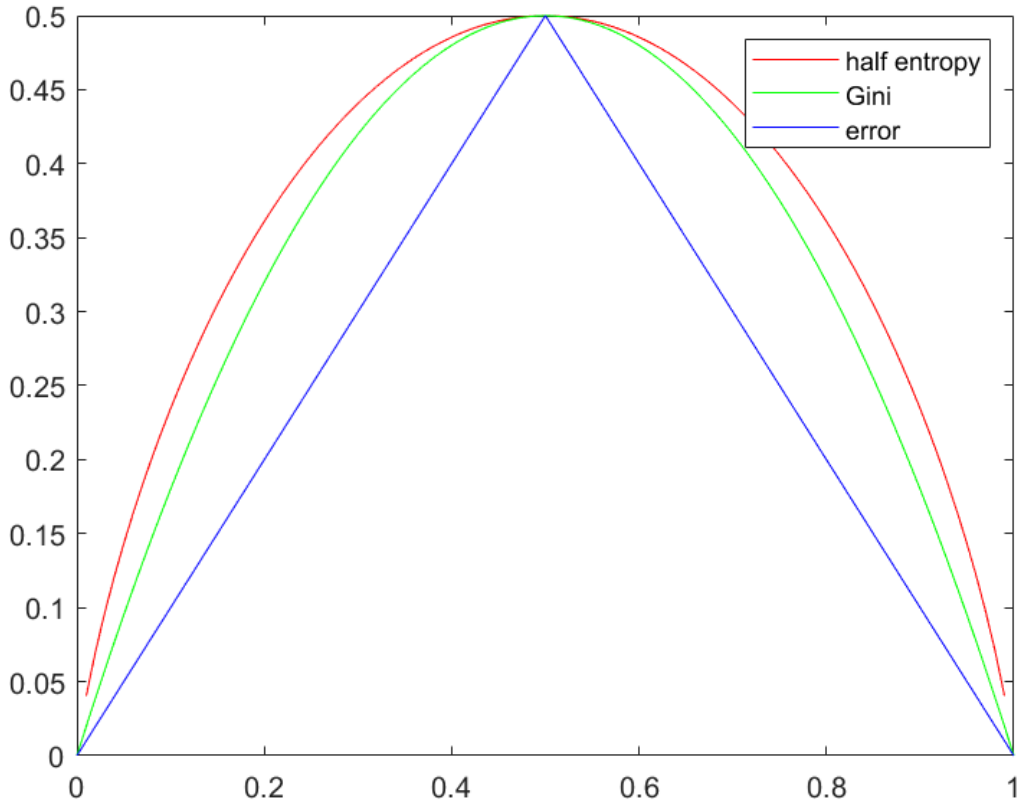
如果样本集合D 根据特征A 是否取某一可能值a被分割成D_1和 D_2两部分，即

$$D_1 = \{(x, y) \in D | A(x) = a\}, \quad D_2 = D - D_1$$

则在特征A的条件下， 集合D的基尼指数定义为

$$\text{Gini}(D, A) = \frac{|D_1|}{|D|}\text{Gini}(D_1) + \frac{|D_2|}{|D|}\text{Gini}(D_2)$$

```
p = 0:0.01:1;
entropy = -p.*log2(p)-(1-p).*log2(1-p);
Gini = 2*p.*(1-p);
error = min(p,1-p);
figure
plot(p,entropy/2,'r',p, Gini, 'g', p,error,'b')
legend('half entropy','Gini','error')
```



算法 **5.6 (CART** 生成算法）

输入： 训练数据集D, 停止计算的条件

输出： CART决策树

根据训练数据集， 从根结点开始， 递归地对每个结点进行以下操作， 构建二叉决策树：

- (1)设结点的训练数据集为D, 计算现有特征对该数据集的基尼指数. 此时对每一个特征A, 对其可能取的每个值a, 根据样本点对A=a的测试为 “是 ” 或 “否“ 将D分割成D_1和D_2, 两部分， 计算A=a 时的基尼指数.
- (2)在所有可能的特征A 以及它们所有可能的切分点a中， 选择基尼指数最小的特征及其对应的切分点作为最优特征与最优切分点。依最优特征与最优切分点， 从现结点生成两个子结点， 将训练数据集依特征分配到两个子结点中去.
- (3)对两个子结点递归地调用 (1), (2), 直至满足停止条件.
- (4)生成CART决策树

停止条件：结点中的样本个数小于阈值，或样本集的Gini指数小于预定阈值.

例**5.4** 根据表5.1所给训练数据集，应用CART算法生成决策树

解 首先计算各特征的基尼指数，选择最优特征以及其最优切分点. 仍采用例5.2的记号， 分别以A_1, A_2, A_3, A_4 表示年龄、 有工作、有自己的房子和信贷情况 4个特征， 并以1. 2, 3表示年龄的值为青年、 中年和老年， 以1, 2表示有工作和有自己的房子的值为是和否， 以1, 2, 3表示信贷情况的值为非常好、 好和一般.

求特征A_1的基尼指数

$$\text{Gini}(D, A_1 = 1) = \frac{5}{15}\left(2 \times \frac{2}{5} \times \left(1 - \frac{2}{5}\right)\right) + \frac{10}{15}\left(2 \times \frac{7}{10} \times \left(1 - \frac{7}{10}\right)\right) = 0.44$$

$$\text{Gini}(D, A_1 = 2) = 0.48$$

$$\text{Gini}(D, A_1 = 3) = 0.44$$

$$\text{Gini}(D, A_2 = 1) = 0.32$$

$$\text{Gini}(D, A_3 = 1) = 0.27$$

$$\text{Gini}(D, A_4 = 1) = 0.36$$

$$\text{Gini}(D, A_4 = 2) = 0.47$$

$$\text{Gini}(D, A_4 = 3) = 0.32$$

本例中，CART和ID3生成的决策树完全一致.

5.5.2 CART剪枝

1.剪枝，形成一个子树序列

$C_\alpha(T) = C(T) + \alpha|T|$，对应的最优树记为$T_\alpha$

$\alpha = 0$,整体树最优；$\alpha \to \infty$，根节点组成的单结点树最优；α小树大.

Breiman等人证明，可以用递归的方法对树进行剪枝，将α逐渐增大，$0 = \alpha_0 < \alpha_1 < \cdots < \alpha_n < +\infty$.

从T_0开始，对T_0的任意内部结点t，以t为单结点的树的损失函数

$$C_\alpha(t) = C(t) + \alpha$$

以t为根结点的子树T_t的损失函数为

$$C_a(T_t) = C(T_t) + \alpha|T_t|$$

$\alpha = 0$，$C_a(T_t) < C_a(t)$

α增大时，存在某一α，使得$C_a(T_t) = C_a(t) \Rightarrow \alpha = \dfrac{C(t) - C(T_i)}{|T_i| - 1}$

算法5.7（CART剪枝算法）

输入：CART算法生成的决策树T_0.

输出：最优决策树T_α.

1. 设$k = 0, T = T_0$.
2. 设$\alpha = +\infty$.
3. 自下而上地对各内部结点t计算$C(T_t)$，$|T_t|$以及$g(t) = \dfrac{C(t) - C(T_t)}{|T_t| - 1}, \alpha = \min(\alpha, g(t))$，这里，$T_t$表示以$t$为根结点的子树，$C(T_t)$是对训练数据的预测误差，$|T_t|$是$T_t$的叶结点个数.

4. 对$g(t)=\alpha$的内部结点t进行剪枝，并对叶结点t以多数表决法决定其类，得到树T.
5. 设$k=k+1, \alpha_k=\alpha, T_k=T$
6. 如果T_k不是由根结点及两个叶结点构成的树，则回到步骤(3)；否则，令$T_k=T$.
7. 采用交叉验证法在子树序列$T_0, T_1, \cdots, T_n$中选取最优子树T_α.

ID3, C4.5, CART比较

算法	支持模型	树结构	特征选择	连续值处理	缺失值处理	剪枝
ID3	分类	多叉树	信息增益	不支持	不支持	不支持
C4.5	分类	多叉树	信息增益比	支持	支持	支持
CART	分类，回归	二叉树	基尼系数，均方差	支持	支持	支持

MATLAB函数使用

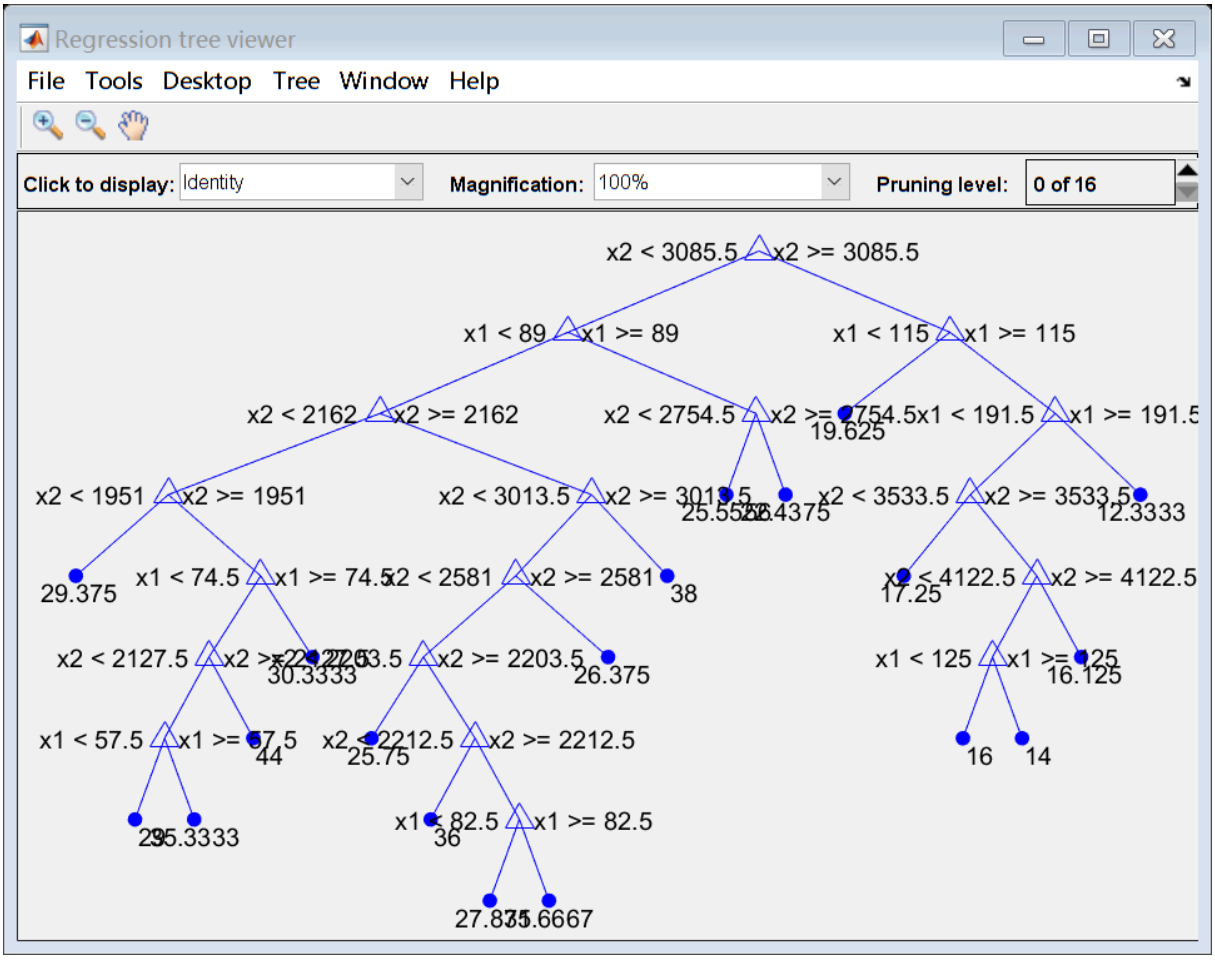
1.创建分类决策树或回归决策树

```
load carsmall % contains Horsepower, Weight, MPG
X = [Horsepower Weight];
rtree = fitrtree(X,MPG);% create regression tree
view(rtree)
```

```
Decision tree for regression
 1  if x2<3085.5 then node 2 elseif x2>=3085.5 then node 3 else 23.7181
 2  if x1<89 then node 4 elseif x1>=89 then node 5 else 28.7931
 3  if x1<115 then node 6 elseif x1>=115 then node 7 else 15.5417
 4  if x2<2162 then node 8 elseif x2>=2162 then node 9 else 30.9375
 5  if x2<2754.5 then node 10 elseif x2>=2754.5 then node 11 else 24.0882
 6  fit = 19.625
 7  if x1<191.5 then node 12 elseif x1>=191.5 then node 13 else 14.375
 8  if x2<1951 then node 14 elseif x2>=1951 then node 15 else 33.3056
 9  if x2<3013.5 then node 16 elseif x2>=3013.5 then node 17 else 29
10  fit = 25.5556
11  fit = 22.4375
12  if x2<3533.5 then node 18 elseif x2>=3533.5 then node 19 else 15.3421
13  fit = 12.3333
14  fit = 29.375
15  if x1<74.5 then node 20 elseif x1>=74.5 then node 21 else 34.4286
16  if x2<2581 then node 22 elseif x2>=2581 then node 23 else 28.5714
17  fit = 38
18  fit = 17.25
19  if x2<4122.5 then node 24 elseif x2>=4122.5 then node 25 else 14.8333
20  if x2<2127.5 then node 26 elseif x2>=2127.5 then node 27 else 35.5455
21  fit = 30.3333
22  if x2<2203.5 then node 28 elseif x2>=2203.5 then node 29 else 29.9231
23  fit = 26.375
24  if x1<125 then node 30 elseif x1>=125 then node 31 else 14.3636
25  fit = 16.125
26  if x1<57.5 then node 32 elseif x1>=57.5 then node 33 else 34.7
27  fit = 44
28  fit = 25.75
29  if x2<2212.5 then node 34 elseif x2>=2212.5 then node 35 else 30.6818
30  fit = 16
```

```
31  fit = 14
32  fit = 29
33  fit = 35.3333
34  fit = 36
35  if x1<82.5 then node 36 elseif x1>=82.5 then node 37 else 30.15
36  fit = 27.875
37  fit = 31.6667
```

```
view(rtree,'Mode','graph')  % 查看决策树
```




```
load fisheriris % load the sample data
ctree = fitctree(meas,species); % create classification tree
view(ctree) % text description
```

```
Decision tree for classification
1  if x3<2.45 then node 2 elseif x3>=2.45 then node 3 else setosa
2  class = setosa
3  if x4<1.75 then node 4 elseif x4>=1.75 then node 5 else versicolor
4  if x3<4.95 then node 6 elseif x3>=4.95 then node 7 else versicolor
5  class = virginica
6  if x4<1.65 then node 8 elseif x4>=1.65 then node 9 else versicolor
7  class = virginica
8  class = versicolor
```

```
 9  class = virginica
```

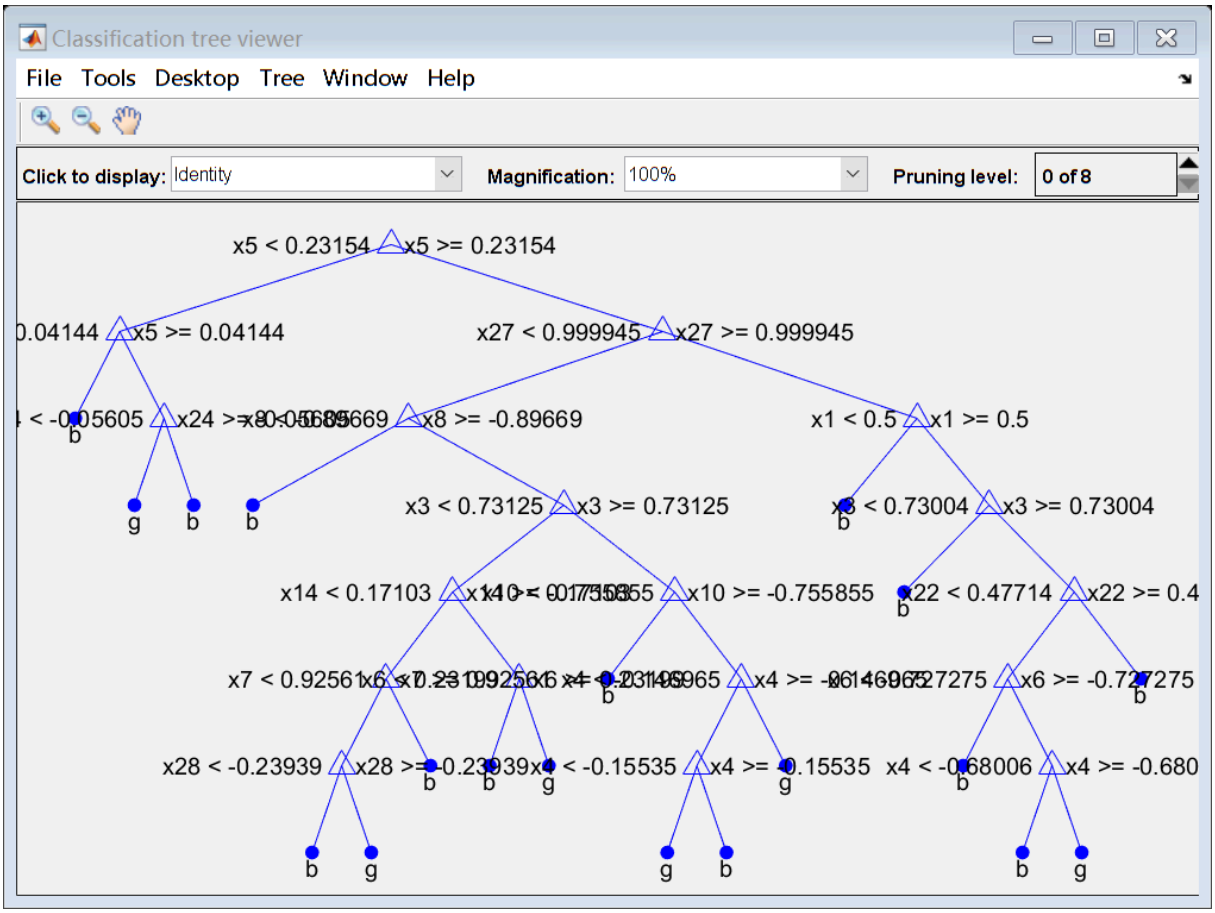
顺便提一下，MATLAB中默认的划分方法是基于基尼系数的。

2.用训练好的决策树做数据分类

```
load ionosphere % contains X and Y variables
ctree = fitctree(X,Y);
view(ctree)
```

```
Decision tree for classification
 1  if x5<0.23154 then node 2 elseif x5>=0.23154 then node 3 else g
 2  if x5<0.04144 then node 4 elseif x5>=0.04144 then node 5 else b
 3  if x27<0.999945 then node 6 elseif x27>=0.999945 then node 7 else g
 4  class = b
 5  if x24<-0.05605 then node 8 elseif x24>=-0.05605 then node 9 else b
 6  if x8<-0.89669 then node 10 elseif x8>=-0.89669 then node 11 else g
 7  if x1<0.5 then node 12 elseif x1>=0.5 then node 13 else b
 8  class = g
 9  class = b
10  class = b
11  if x3<0.73125 then node 14 elseif x3>=0.73125 then node 15 else g
12  class = b
13  if x3<0.73004 then node 16 elseif x3>=0.73004 then node 17 else b
14  if x14<0.17103 then node 18 elseif x14>=0.17103 then node 19 else g
15  if x10<-0.755855 then node 20 elseif x10>=-0.755855 then node 21 else g
16  class = b
17  if x22<0.47714 then node 22 elseif x22>=0.47714 then node 23 else g
18  if x7<0.92561 then node 24 elseif x7>=0.92561 then node 25 else g
19  if x6<0.23199 then node 26 elseif x6>=0.23199 then node 27 else b
20  class = b
21  if x4<-0.146965 then node 28 elseif x4>=-0.146965 then node 29 else g
22  if x6<-0.727275 then node 30 elseif x6>=-0.727275 then node 31 else g
23  class = b
24  if x28<-0.23939 then node 32 elseif x28>=-0.23939 then node 33 else g
25  class = b
26  class = b
27  class = g
28  if x4<-0.15535 then node 34 elseif x4>=-0.15535 then node 35 else g
29  class = g
30  class = b
31  if x4<-0.68006 then node 36 elseif x4>=-0.68006 then node 37 else g
32  class = b
33  class = g
34  class = g
35  class = b
36  class = b
37  class = g
```

```
view(ctree,'Mode','graph')
```




```
Ynew = predict(ctree,mean(X))
```

```
Ynew = 1×1 cell array
    {'g'}
```

3.检验决策树性能并修正

最简单的性能检测就是生成一颗决策树后，输入所有数据样本，进行误差检验。 这种检测方法输入数据样本中包含了训练数据，因此得到结果比实际结果要好的多，泛化能力差，过于乐观。

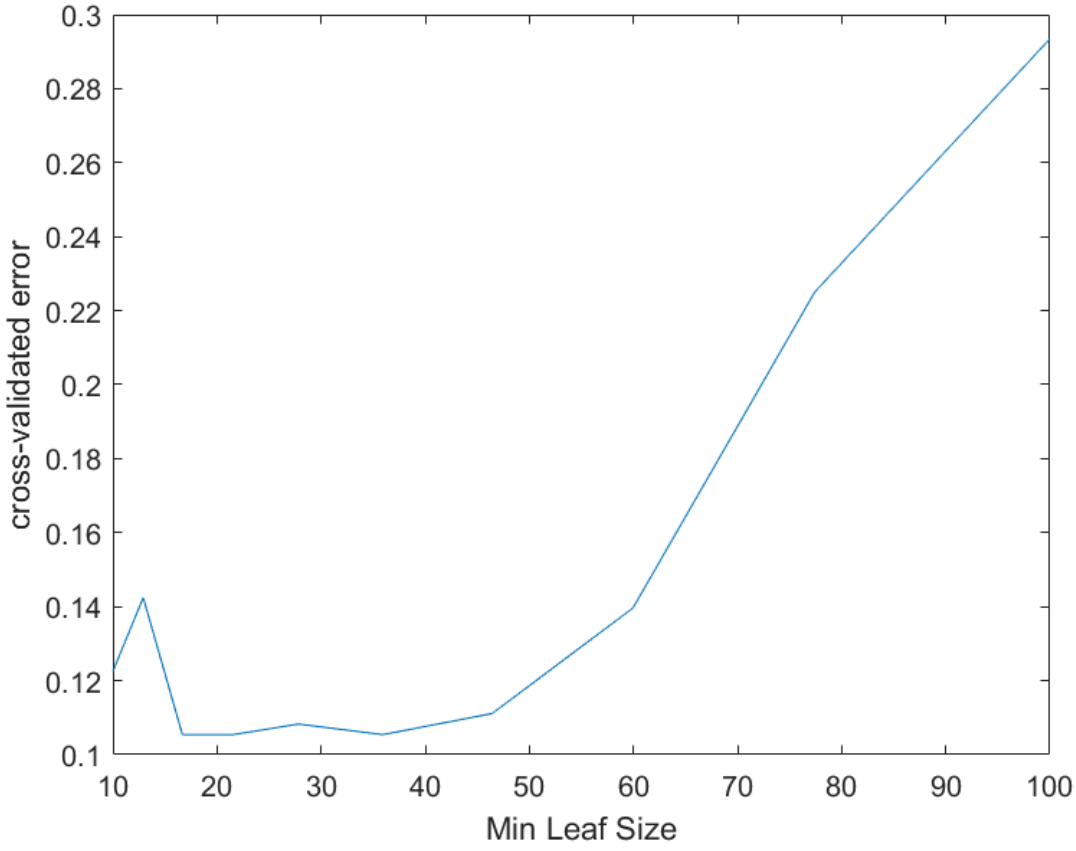
因此常常采用交叉检验法，因为交叉检验法使用的测试数据不同于训练数据，且是一个多次平均的结果，因此其对性能的估计比较准确。并根据交叉检验法进行决策树的修改，使得其性能表现更好。

（1）限定每个叶节点包含的最少数据量

如果不进行限定，每个叶节点包含的最小数据量为1，过多的叶子结点必然造成决策树泛化能力的降低，因此应该求得一个Leaf(min)，从而使得计算出交叉误差最小。

```
% X = meas;
% Y = species;
leafs = logspace(1,2,10);
rng('default')
```

```
N = numel(leafs);
err = zeros(N,1);
for n=1:N
    t = fitctree(X,Y,'CrossVal','On',...
        'MinLeaf',leafs(n));
    err(n) = kfoldLoss(t);
end
plot(leafs,err);
xlabel('Min Leaf Size');
ylabel('cross-validated error');
```

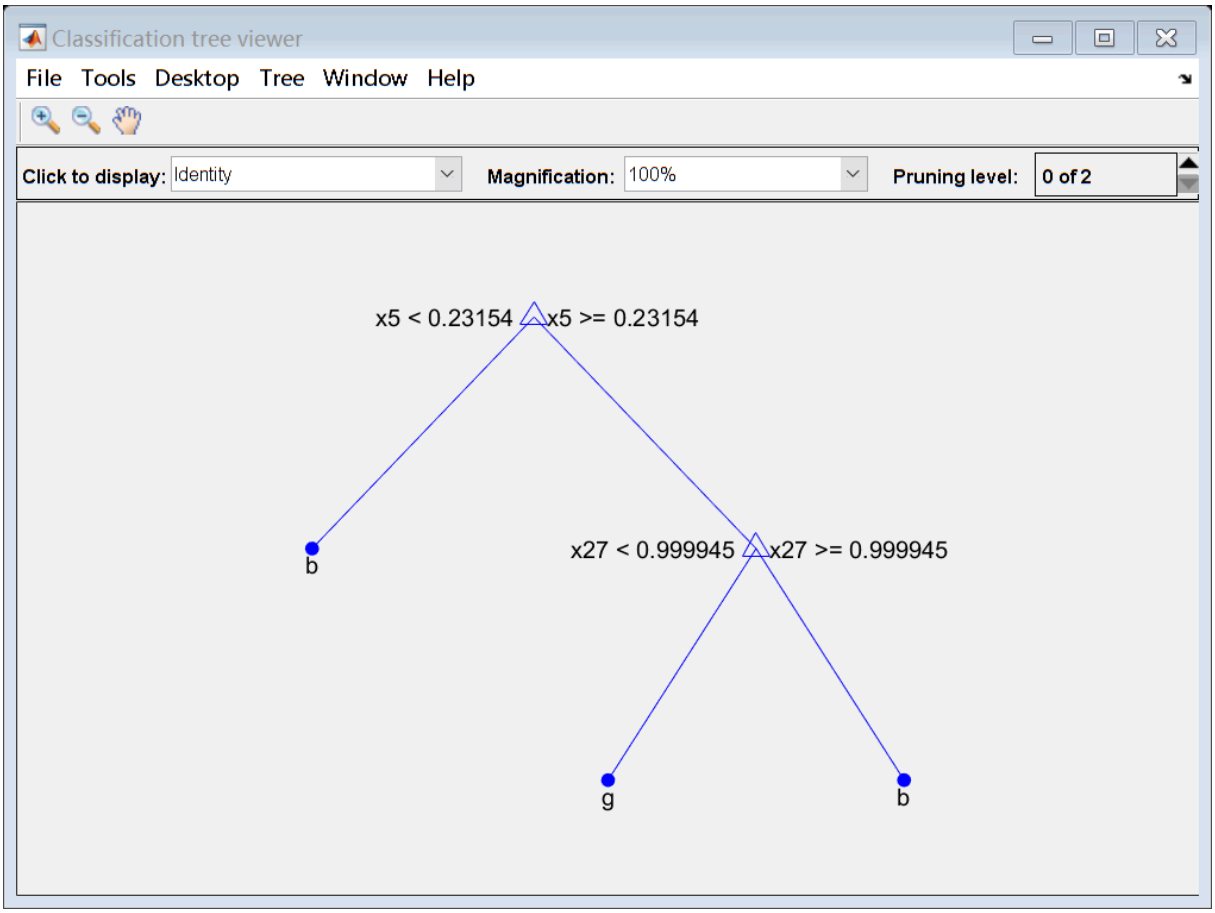


得到最小的叶子尺寸，再进行决策树生成

```
OptimalTree = fitctree(X,Y,'minleaf',40);
view(OptimalTree)
```

```
Decision tree for classification
1  if x5<0.23154 then node 2 elseif x5>=0.23154 then node 3 else g
2  class = b
3  if x27<0.999945 then node 4 elseif x27>=0.999945 then node 5 else g
4  class = g
5  class = b
```

```
view(OptimalTree,'Mode','graph')
```



（2）剪枝

与上面类似，计算不同剪枝下的交叉检测误差，选择最小误差处剪枝。

```
[~,~,~,bestlevel] = cvLoss(ctree,...
    'SubTrees','All','TreeSize','min')
```

```
bestlevel = 2
```

```
tree = prune(ctree,'Level',bestlevel)
```

```
tree =
  ClassificationTree
             ResponseName: 'Y'
    CategoricalPredictors: []
               ClassNames: {'b'  'g'}
           ScoreTransform: 'none'
          NumObservations: 351


  Properties, Methods
```

作业

习题5.1 根据表5.1所给的训练数据集，利用信息增益比（C4.5算法）生成决策树.

注：手算或者编程计算皆可.