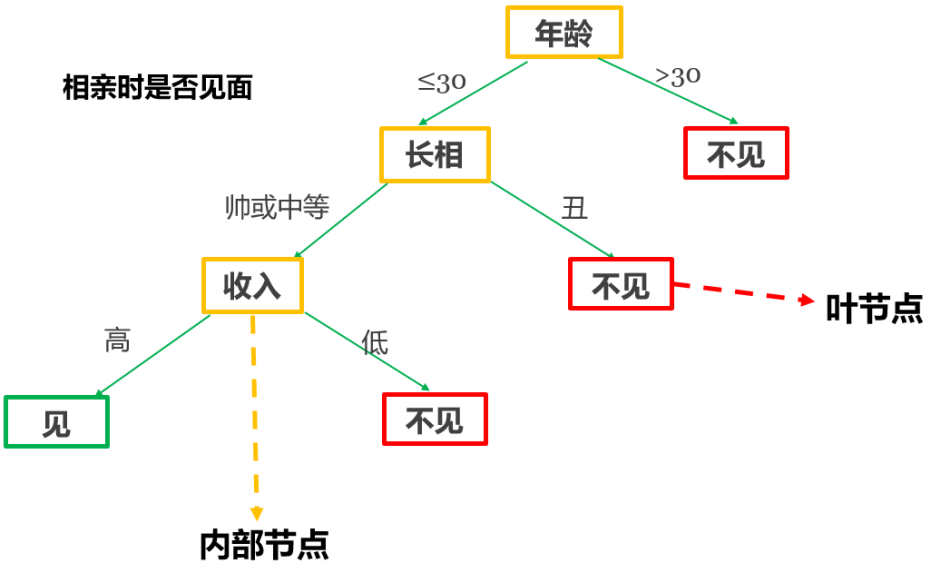
决策树

决策树：一种重要的分类&回归方法。可以看成是if-then 规则的集合，具有较好的可解读性。

1 决策树模型与学习

(1) 模型定义

定义：分类决策树是一种描述对实例进行分类的树形结构。决策树由节点(node)和有向边(directed edges)组成。节点有两种类型：内部节点(internal node)和叶节点(leaf node)。内部节点表示一个特征或者属性；叶节点表示一个类。



(2) 决策树与if-then规则集合

由决策树根节点到叶节点的每一条路径构成一条规则，路径上的内部节点对应规则的条件，叶节点对应规则的结论。

这些if-then规则集合具备性质：互斥且完备。即：每一个实例可以被一条规则覆盖；而且只能被一条规则覆盖。

(3) 决策树与条件概率分布

将特征空间划分为互不相交的单元，并在每个单元定义一个类的概率分布就构成了一个条件概率分布。决策树的一条路径对应于划分中的一个单元。各叶节点上的条件概率往往偏向于某一个类，决策树分类时将实例强行分到条件概率较大的那一类。

(4) 决策树学习

给定训练数据集

$$D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}, \tag{1.1}$$

其中， $x_i = (x_{i1}, \cdots, x_{ip})^\top \in \mathbb{R}^p$ 为输入实例， $y_i \in \{1, \cdots, K\}$ 为类别标记。

决策树学习目标：给定训练数据集，构建决策树模型，使得它对实例可以进行正确分类。

决策树生成： 构建根节点，将所有训练数据放在根节点；选择一个最优特征，按照这个最优特征对训练集进行划分；如果按照这个特征可以将训练数据集进行基本正确的分类，则构建叶节点；否则再递归地选择特征，按照特征进行训练集划分，直到达到停止条件为止。

决策树剪枝：生成的决策树对训练数据集有很好的拟合效果，但可能泛化能力较差。此时可以对树进行剪枝，使它具备更好的泛化能力。

2 特征选择

2.1 特征选择问题

目的：选择对训练数据集有 分类能力 的特征。如以下贷款数据集，应该优先选择哪个特征首先进行分裂呢？这是特征选择讨论的问题。

“分类能力”：如果利用一个特征分类后与随机分类结果差异不大，则称该特征没有分类能力。通常用信息增益或者信息增益比作为选择的准则。

ID	年龄	有工作	有自己的房子	信贷情况	贷款审批
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

Figure 1: 贷款申请数据集

2.2 信息增益

为介绍信息增益的概念，先给出熵(entropy)与条件熵的定义。

熵是随机变量不确定性的度量。设X是一个取有限个值的离散随机变量，其概率分布为：

$$P(X = x_i) = p_i, i = 1, \cdots, n \tag{2.1}$$

则随机变量熵的定义为：

$$H(X) = -\sum_{i=1}^{n} p_i \log p_i \tag{2.2}$$

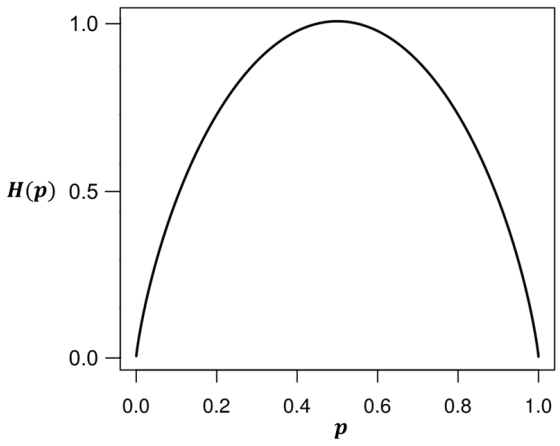
由上可知，随机变量的熵只与X的分布有关，所以也可以将X的熵记作$H(p)$,即

$$H(p) = -\sum_{i=1}^{n} p_i \log p_i \tag{2.3}$$

熵越大，随机变量不确定性越大。从定义可以验证 $0 \leq H(p) \leq \log n$. 以上对数以2为底或以自然对数为底时，这时熵的单位分别称为比特(bit)或者纳特(nat)。

例：当随机变量只取两个取值时：

$$P(X = 1) = p, \quad P(X = 0) = 1 - p \tag{2.4}$$



条件熵$H(Y|X)$(conditional entropy)：

$$H(Y|X) = \sum_i p_i H(Y|X = x_i) \tag{2.5}$$

其中，$p_i = P(X = x_i), i = 1, \cdots, n.$

当熵和条件熵由数据估计得到时，分别称作经验熵(empirical entropy)和经验条件熵(empirical conditional entropy)。如果有0概率，令$0 \log 0 = 0.$

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

$$g(D, A) = H(D) - H(D|A). \tag{2.6}$$

一般地，熵$H(Y)$与条件熵$H(Y|X)$之差称为互信息(mutual information), 以上定义的信息增益相当于训练数据集中类与特征的互信息。

计算方式： 设训练数据集为D, $|D|$表示样本容量。设有K个类别C_k, $|C_k|$为属于类C_k的样本个数， $\sum_k |C_k| = |D|$. 设特征A有n个不同的取值$\{a_1, \cdots, a_n\}$, 根据特征A的取值将D划分为n个子集 $D_1, D_2, \cdots, D_n$, $|D_i|$表示D_i的样本个数。 记子集D_i中属于C_k的样本集合为D_{ik}, 即$D_{ik} = D_i \cap C_k$, $|D_{ik}|$ 为D_{ik}的样本个数，信息增益的算法如下：

(1) 计算数据集D的经验熵$H(D)$:

$$H(D) = -\sum_{k=1}^{K} \frac{|C_k|}{|D|} \log_2 \frac{|C_k|}{|D|} \tag{2.7}$$

(2) 计算特征A对数据集D的经验条件熵$H(D|A)$

$$H(D|A) = -\sum_{i=1}^{n} \frac{|D_i|}{|D|} \sum_{k=1}^{K} \frac{|D_{ik}|}{|D_i|} \log_2 \frac{|D_{ik}|}{|D_i|}. \tag{2.8}$$

(3) 计算信息增益

$$g(D,A)=H(D)-H(D|A) \tag{2.9}$$

[练习] 给定贷款数据集，根据信息增益准则选择最优特征。

解：首先计算经验熵$H(D)$：

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

然后计算各特征对数据集D的信息增益。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，则：

（1）

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

这里D_1,D_2,D_3分别是D中A_1（年龄）取值为青年、中年和老年的样本子集。类似地，

（2）

$$\begin{aligned}g\left(D,A_2\right)&=H(D)-\left[\frac{5}{15}H\left(D_1\right)+\frac{10}{15}H\left(D_2\right)\right]\\&=0.971-\left[\frac{5}{15}\times0+\frac{10}{15}\left(-\frac{4}{10}\log_2\frac{4}{10}-\frac{6}{10}\log_2\frac{6}{10}\right)\right]=0.324\end{aligned}$$

（3）

$$\begin{aligned}g\left(D,A_3\right)&=0.971-\left[\frac{6}{15}\times0+\frac{9}{15}\left(-\frac{3}{9}\log_2\frac{3}{9}-\frac{6}{9}\log_2\frac{6}{9}\right)\right]\\&=0.971-0.551=0.420\end{aligned}$$

（4）

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

最后，比较各特征的信息增益。由于特征A_3（有自己的房子）的信息增益最大，所以选择特征A_3作为最优特征。

2.3 信息增益比

在贷款数据集中，如果采用ID列进行划分，则信息增益更大，但这显然不合理。

信息增益倾向于选择取值较多的特征，可以使用信息增益比(information gain ratio)对这一问题进行校正。

定义： 特征A对训练数据集D的信息增益比$g_R(D, A)$定义如下：

$$g_R(D, A)=\frac{g(D, A)}{H_A(D)}, \tag{2.10}$$

其中假设经过特征$A=\{a_1, \cdots, a_n\}$划分后的子集为$D_1, \cdots, D_n$， 则$H_A(D)=\sum_{i=1}^n \frac{|D_i|}{|D|} \log_2 \frac{|D_i|}{|D|}$.

3 决策树的生成

3.1 ID3算法

ID3算法是在决策树的各个节点上应用信息增益准则选择特征，**递归**地构建决策树。

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为单节点树，将C_k 作为该节点的类标记，返回T;

(2) 若$A=\emptyset$，则T为单节点树，将D中实例数最大的类C_k作为该节点的类标记，返

回T.

(3) 否则，对A中各个特征计算信息增益，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，则设置T为单节点树，将D中实例数最大的类C_k作为该节点的类标记，返回T.

(5) 否则，对A_g的每一个特征值 a_i，依$A_g = a_i$将D分割为若干非空子集D_i，将D_i中实例最大的类作为标记，构建子节点，由节点及子节点构成树T，返回T; (6) 对第i个子节点，以D_i为训练集，以$A - \{A_g\}$为特征集，递归地调用步(1) (5)步，得到子树T_i，返回T_i;

[练习] 对贷款数据，用ID3算法建立决策树。

解：利用例5.2的结果，由于特征A_3（有自己的房子）的信息增益值最大，所以选择特征A_3作为根节点的特征。它将训练数据集D划分为两子集D_1（A_3取值为“是”）和D_2（A_3取值为“否”）。由于D_1只有同一类的样本点，所以它成为一个叶结点，结点的类标记为“是”。

对D_2则需要从特征A_1（年龄）、A_2（有工作）和A_4（信贷情况）中选择新的特征。计算各个特征的信息增益：

$$g\left(D_2, A_1\right) = H\left(D_2\right) - H\left(D_2|A_1\right) = 0.918 - 0.667 = 0.251$$

$$g\left(D_2, A_2\right) = H\left(D_2\right) - H\left(D_2|A_2\right) = 0.918$$

$$g\left(D_2, A_4\right) = H\left(D_2\right) - H\left(D_2|A_4\right) = 0.474$$

选择信息增益最大的特征A_2（有工作）作为结点的特征。由于A_2有两个可能取值，从这一结点引出两个子结点：一个对应“是”（有工作）的子结点，包含3个样本，它们属于同一类，所以这是一个叶结点，类标记为“是”；另一个是对应“否”（无工作）的子结点，包含6个样本，它们也属于同一类，所以这也是一个叶结点，类标记为“否”。

这样生成一颗如图2所示的决策树，该决策树只用了两个特征（有两个内部结点）。

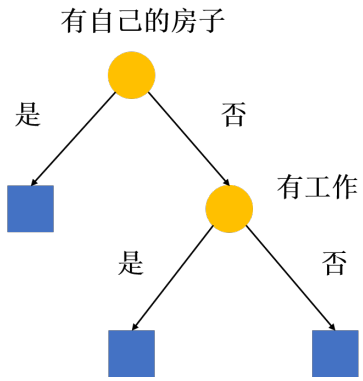


Figure 2: 决策树的生成

ID3算法只有树的生成，所以该算法生成的树容易产生过拟合。

3.2 C4.5算法

思考：ID3算法有何不足？
如果以编号作为划分依据？

C4.5算法：使用信息增益比作为划分依据：在ID3中用信息增益选择属性时偏向于选择分枝比较多的属性值，即取值多的属性；除以$H_A(D)$，可以削弱这种作用.

[练习]：使用C4.5算法对贷款数据集建立决策树。

4 决策树的剪枝

决策树在生成过程中易产生过拟合的问题。可以对决策树进行剪枝（pruning），降低决策树的复杂度。

设树T的叶节点个数为$|T|$, 对于叶节点t，设该叶节点有N_t个样本，其中k类的样本点有N_{tk}, $k = 1, \cdots, K$, $H_t(T)$为叶节点t上的经验熵， $\alpha > 0$为参数，则决策树学习的损失函数可以定义为：

$$C_\alpha(T) = \sum_{t=1}^{|T|} N_t H_t(T) + \alpha|T| \stackrel{\text{def}}{=} C(T) + \alpha|T|. \tag{4.1}$$

其中

$$H_t(T) = -\sum_k \frac{N_{tk}}{N_t} \log \frac{N_{tk}}{N_t}. \tag{4.2}$$

其中, $C(T)$表示对训练数据集的拟合误差， $|T|$表示模型复杂度。越大的α则意味着更加简单的树。

5 CART 算法

CART: classification and regression tree (分类回归树)，由Breiman 提出

CART是二叉树。由两步组成：

(1) 决策树生成：利用训练数据集生成决策树

(2) 决策树剪枝：用验证数据集对已生成的树进行剪枝

5.1 CART 生成

回归树：平方误差最小化

分类树：基尼指数（Gini Index）最小化

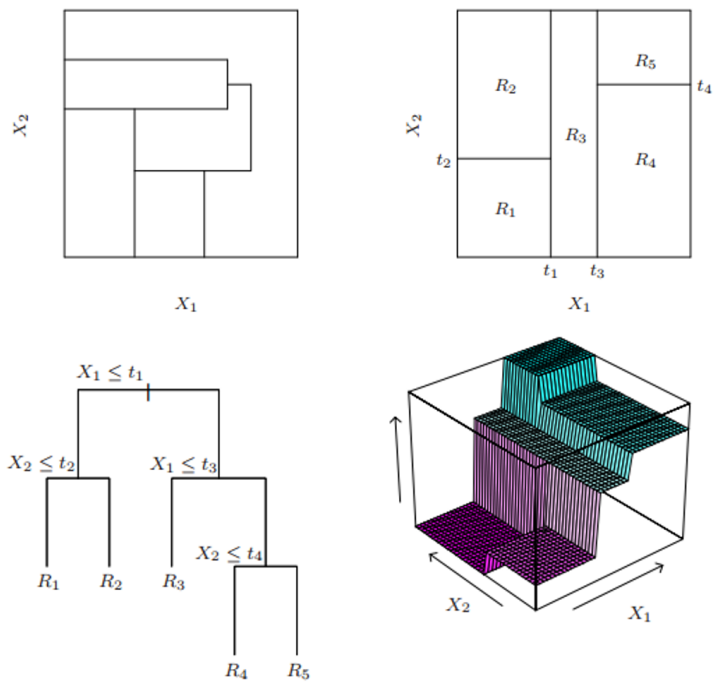
(1) 回归树的生成

给定划分得到预测值： 回归树对应着输入空间（即特征空间）的划分以及划分单元上的输出值。 假设将特征空间划分为M个单元$R_1, \cdots, R_M$，并且每个单元R_m上有固定的输出值 c_m，则回归树模型可表示为

$$f(x) = \sum_m c_m I(x \in R_m). \tag{5.1}$$

当输入空间划分确定，可用平方误差 $\sum_{x_i \in R_m}(y_i - f(x_i))^2$表示回归树对训练数据集的预测误差。 因此，在划分$R_m$上做出的预测为

$$\hat{c}_m = \text{ave}(y_i | x_i \in R_m). \tag{5.2}$$



如何得到划分： 采用启发式算法，先寻找切分变量(splitting variable)，再寻找切分点(splitting point)。 定义两个区域：

$$R_1(j,s) = \left\{x|x^{(j)} \leqslant s\right\}, R_2(j,s) = \left\{x|x^{(j)} > s\right\} \tag{5.3}$$

通过求解以下目标函数得到最优切分变量和切分点：

$$\min_{j,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2\right] \tag{5.4}$$

(2) 分类树的生成

分类树用基尼指数选择最优特征，同时决定该特征的最优切分点。

定义（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

$$\text{Gini}(p) = \sum_{k=1}^{K} p_k \left(1 - p_k\right) = 1 - \sum_{k=1}^{K} p_k^2 \tag{5.5}$$

对于二分类问题，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

$$\text{Gini}(p) = 2p(1-p) \tag{5.6}$$

对给定的样本集合D，其基尼指数为

$$\text{Gini}(D) = 1 - \sum_{k=1}^{K} \left(\frac{|C_k|}{|D|}\right)^2 \tag{5.7}$$

这里，C_k是D中属于第k类的样本子集，K是类的个数。

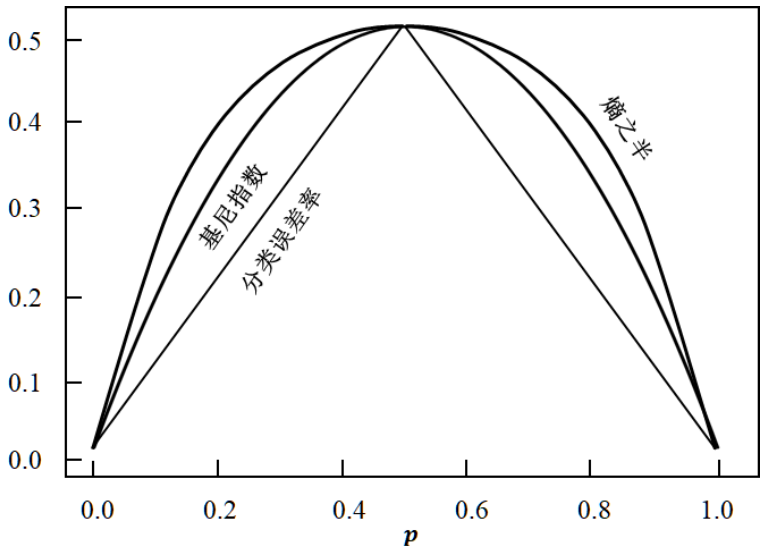


Figure 3: 二分类中基尼指数、熵之半和分类误差率的关系

如果样本集合根据特征A是否取某个值a被分割成D_1 和 D_2，则在特征A的条件下，集合D的基尼指数定义为：

$$\text{Gini}(D, A) = \frac{|D_1|}{|D|} \text{Gini}\left(D_1\right) + \frac{|D_2|}{|D|} \text{Gini}\left(D_2\right) \tag{5.8}$$

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，对其可能的每个取值a，根据样本点对$A = a$的测试为“是”或“否”将D分割成D_1, D_2两部分，利用式（5.8）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。根据最优特征和最优切分点，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。

算法停止的条件是结点中的样本个数小于预定阈值，或样本集的基尼指数小于预定阈值（样本基本属于同一类），或者没有更多特征。

5.2 CART 剪枝

CART剪枝算法从底端开始，减去一些子树，使决策树变小，从而对未知数据有更好的预测性。 CART剪枝由以下步骤组成：

(1) 剪枝，形成子树序列

子树的损失函数：

$$C_\alpha(T) = C(T) + \alpha|T|. \tag{5.9}$$

其中 $C(T)$ 为对训练数据的预测误差度量（如基尼指数），$|T|$ 为子树的叶节点个数。

随着α递增，剪枝得到的树越来越小。

具体而言，从整体T_0开始剪枝。对任意内部节点t, 以t为单节点的树的损失函数为：

$$C_\alpha(t) = C(t) + \alpha \tag{5.10}$$

以t为根节点的子树T_t的损失函数是：

$$C_\alpha(T_t) = C(T_t) + \alpha|T_t| \tag{5.11}$$

若α 充分小， 则$C_\alpha(T_t)$充分小，此时有 $C_\alpha(T_t) < C_\alpha(t)$. 随着$\alpha$ 增大，在某一α 有 $C_\alpha(T_t) = C_\alpha(t)$. 此时$\alpha$的取值为：

$$\alpha = \frac{C(t) - C(T_t)}{|T_t| - 1} \tag{5.12}$$

对T_0中每个点 t，计算

$$g(t) = \frac{C(t) - C(T_t)}{|T_t| - 1} \tag{5.13}$$

将最小的$g(t)$设为α_1，则T_1为在区间$[\alpha_1, \alpha_2)$ 的最优子树。 不断增加α的值，产生新的最优区间。

(2) 在剪枝得到的子树序列中，通过交叉验证选择最优子树

计算验证集上的每颗子树对应的平方误差，选择最优子树。

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

$$g(t) = \frac{C(t) - C\left(T_t\right)}{|T_t| - 1}$$

$$\alpha = \min(\alpha, g(t))$$

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（2）；否则令$T_k = T_n$。

（7）采用交叉验证法在子树序列$T_0, T_1, ..., T_n$中选取最优子树T_α。