2.Base tree

1.信息熵，信息增益

1.1信息熵

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

其中P表示事件发生的概率。

1.2信息增益

用 a 属性对样本D进行划分所获得的"信息增益"(information gain)

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

其中D表示按照某个特征分裂之后得到的样本子集内样本个数，V则表示分为样本子集。注意要进行加权。

2.ID3决策树

2.1构建过程（原理）

下面以西瓜数据集为例, 该数据集包含17个样本,用以学习一棵能预测没刨开的是不是好瓜的决策树. 显然y = 2, 下图中可以看到,正例 占 8/17 , 反例占 9/17,

$$\mathrm{Ent}(D) = -\sum_{k=1}^{2} p_k \log_2 p_k = -\left(\frac{8}{17}\log_2\frac{8}{17} + \frac{9}{17}\log_2\frac{9}{17}\right) = 0.998$$

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是

7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	否

然后我们计算出当前属性集合{色泽,根蒂,敲声,纹理,脐部,触感} 中每个属性的信息增益.

以属性"色泽" 为例 ,它有三个可能的取值 :{青绿,乌黑,浅白} .

使用该属性对D进行划分, 则可得到 3个子集 ,分别记为 D1 (色泽=青绿) D2(色泽=乌黑) D3(色泽=浅白).

子集D1 包含的编号{1,4,6,10,13,17} , 正例(是)占 p1 = 3/6 ,反例(否) 占 p2 = 3/6 ;

子集D2 包含的编号 {2,3,7,8,9,15} , 正例占 p1 = 4/6 , 反例占 p2 = 2/6 ;

子集D3 包含的编号 {5,11,12,14,16} ,正例占p1 = 1/5 , 反例占p2 = 4/5 ;

可计算出"色泽"划分之后所获得的信息熵为:

$$\begin{aligned}
\mathrm{Ent}(D^1) &= -\left(\frac{3}{6}\log_2\frac{3}{6}+\frac{3}{6}\log_2\frac{3}{6}\right) = 1.000\ , \\
\mathrm{Ent}(D^2) &= -\left(\frac{4}{6}\log_2\frac{4}{6}+\frac{2}{6}\log_2\frac{2}{6}\right) = 0.918\ , \\
\mathrm{Ent}(D^3) &= -\left(\frac{1}{5}\log_2\frac{1}{5}+\frac{4}{5}\log_2\frac{4}{5}\right) = 0.722\ ,
\end{aligned}$$

于是 , 计算出属性 " 色泽"的信息增益为 :

$$\begin{aligned}\text{Gain}(D, \text{色泽}) &= \text{Ent}(D) - \sum_{v=1}^{} \frac{|D^v|}{|D|}\text{Ent}(D^v) \\ &= 0.998 - \left(\frac{6}{17} \times 1.000 + \frac{6}{17} \times 0.918 + \frac{5}{17} \times 0.722\right) \\ &= 0.109 .\end{aligned}$$

类似的, 我们可计算出其他属性的信息增益:

$$\text{Gain}(D, \text{根蒂}) = 0.143; \quad \text{Gain}(D, \text{敲声}) = 0.141;$$
$$\text{Gain}(D, \text{纹理}) = 0.381; \quad \text{Gain}(D, \text{脐部}) = 0.289;$$
$$\text{Gain}(D, \text{触感}) = 0.006.$$

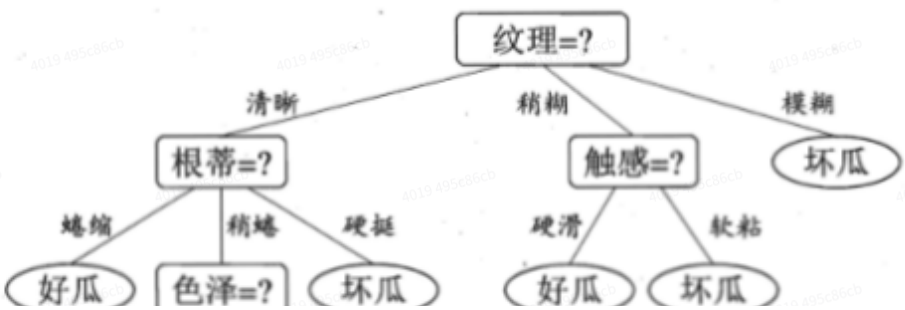
https://blog.csdn.net/qq_4166180

显然这里 Gain(D,纹理) = 0.381 信息增益最大, 于是他被选为划分属性. 根据属性将样本分为多个子集，在子集内继续重新划分，直到叶节点）

以图中的一个分支节点("纹理= 清晰") 为例, 该节点包含的样例集合D1中有编号{1,2,3,4,5,8,10,15} 的9 个样例,可用的属性集合为 { 色泽,根蒂,敲声,脐部,触感}; 基于 D1计算出各属性的信息增益:

$$\text{Gain}(D^1, \text{色泽}) = 0.043; \quad \text{Gain}(D^1, \text{根蒂}) = 0.458;$$
$$\text{Gain}(D^1, \text{敲声}) = 0.331; \quad \text{Gain}(D^1, \text{脐部}) = 0.458;$$
$$\text{Gain}(D^1, \text{触感}) = 0.458.$$

"根蒂" , "脐部" , "触感" 3 个属性均取得最大的信息增益 ,可用选择其中一个作为划分属性, 最终得到:



https://blog.csdn.net/qq_4166

2.2ID3的缺点

1. id3是多叉树，效率较低，并且只能处理离散特征;

2. 信息增益的衡量方式非常容易偏向取值数量特别多的特征.

   对于id3来说，依据信息增益的原则，取值越多的特征，会切分的越细，即每个分支的数据越少，因为每一个分支的数据越少，每一个节点的“纯度”会越高，整体的信息增益越大。

3. 信息熵的计算比较涉及到求和和对数变换，比较费时.

3.C4.5

3.1 C4.5的特点

3.1.1信息增益率

针对信息增益对取值数目多的特征有偏好的问题，使用信息增益率替代信息增益

$$\text{Gain\_ratio}(D,a)=\frac{\text{Gain}(D,a)}{\text{IV}(a)}$$

知乎 @什么

分子部分就是信息增益没有变，分母部分是：

$$\text{IV}(a)=-\sum^{V}\frac{|D^v|}{|D|}\log_2\frac{|D^v|}{|D|}$$

3.1.2启发式方法

但是信息增益率存在的问题在于，会对取值很少的特征有所偏好，举个极端的例子，假设某个特征取值完全相同，则分母IV(a)的计算结果为0，则信息增益率为无穷大。

先从候选划分特征中找到信息增益高于平均值的特征

再从中选择增益率最高的

3.2 C4.5的缺点

1. C4.5和id3一样 用的是多叉树，效率较低，用二叉树效率更高；
2. C4.5 的信息增益率计算和信息熵一样都比较计算复杂而麻烦。

4.Cart

4.1Cart的改进

1. Cart摒弃了麻烦的多叉树，而使用二叉树进行替代；
2. Cart使用了gini指数作为分裂标准。

4.2基尼系数

基尼指数代表了模型的不纯度，基尼系数越小，不纯度越低，特征越好。这和信息增益（率）正好相反。

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

基尼指数的问题在于偏向于特征值较多的特征，对于特征取值较多的特征比较容易算出高的gini值，这个缺点和信息增益是一样的。

实例： 二分类

假设某个节点上的样本全是1，则其pk=1，gini=0；如果节点上的样本全是0，则pk=0，gini=0

如果节点上的样本一半是1，一半是0，则pk=0.5，gini=0.5*0.5=0.25达到最大，此时混沌程度最大，划分了等于没划分；

5.剪枝策略

5.1预剪枝

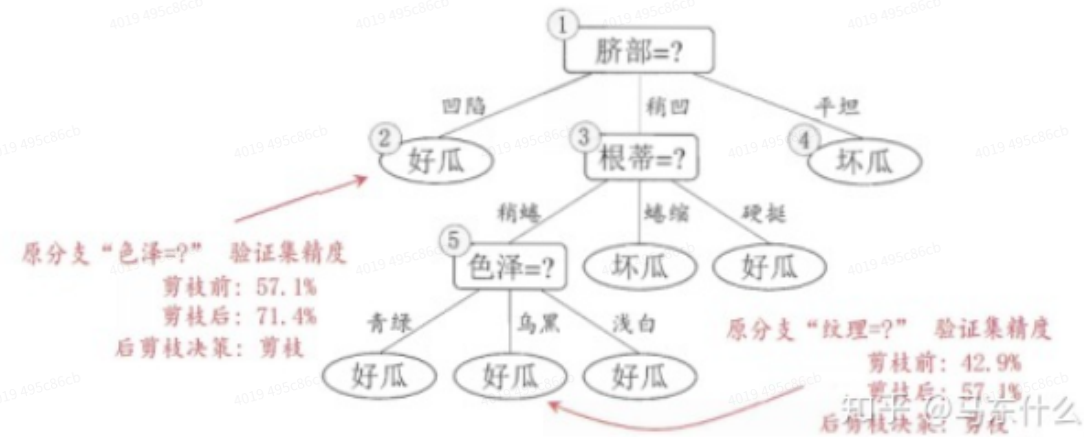
通过tree的最大深度或者叶节点的最小样本数量等超参数的调节就可以达到预剪枝的效果。但有可能带来会带来欠拟合的风险。

5.2后剪枝

引入验证集，对每一个叶子节点的父节点进行删除，然后观察验证集上的精度变化，例如下图：

知乎 @马东什么

去掉纹理这个叶子节点的父节点后：

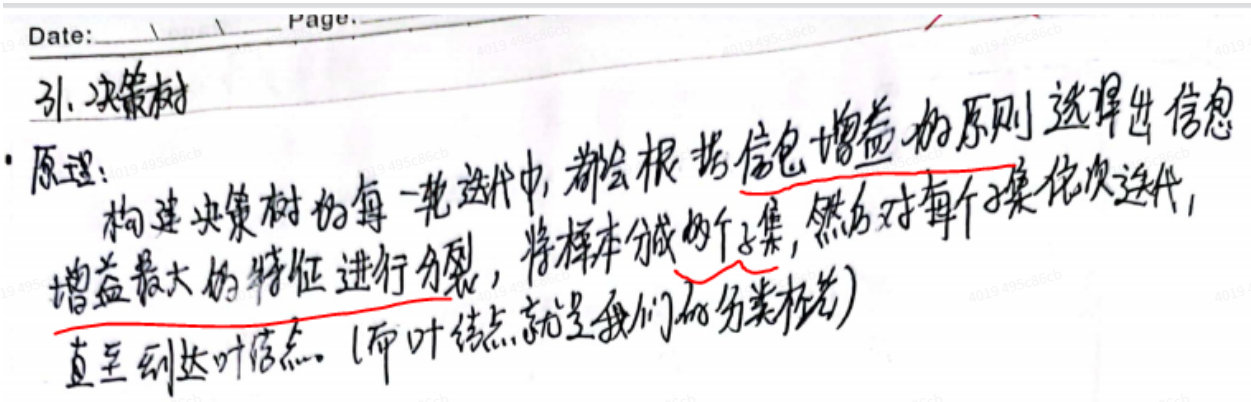


知乎 @马东什么

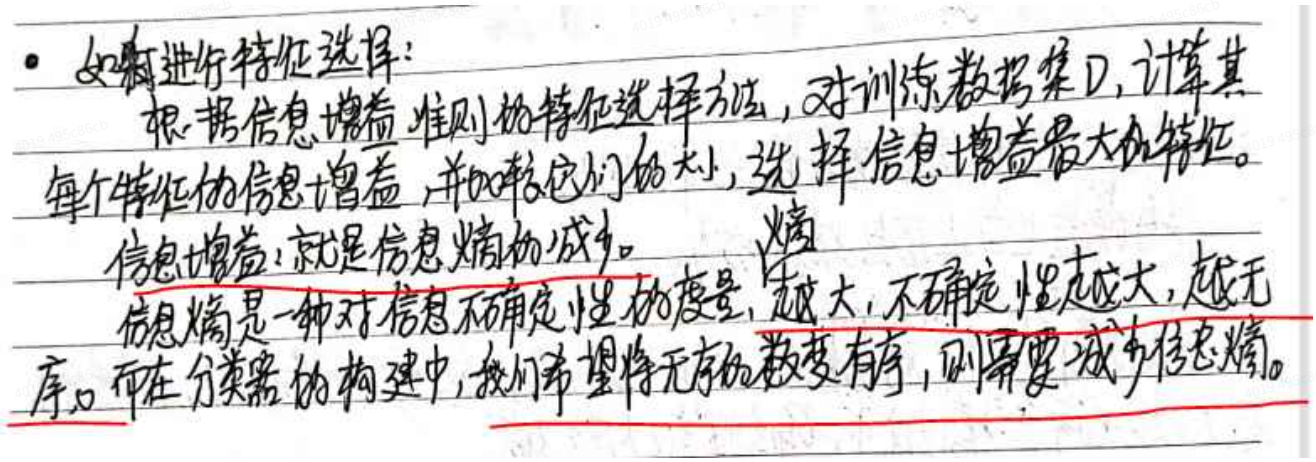
验证集的精度梯度提高，因此进行剪枝.

6.决策树面试版

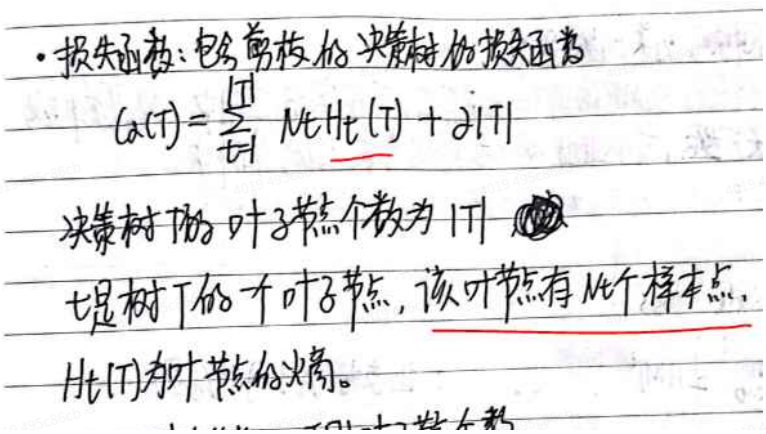
6.1原理



6.2如何进行特征选择



6.3损失函数



2:正则化参数，正则项系数.

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

$$\text{或} L(w,d) = \frac{1}{N} \sum_{i=1}^{N} (y_i - f(x_i))^2 + r(d)$$

6.4回归树

• 回归树：

过程：(1)将预测变量空间 $(X_1, X_2, X_3, \cdots, X_p)$ 的可能取值构成的集合成J个互不重叠的区域 $\{R_1, R_2, R_3, \cdots, R_J\}$。

(2)对落入区域 R_j 的每个观测值作同样的预测，预测值等于 R_j 上训练集的各个样本取值的算术平均数。

如：在第(1)步中，得到两个区域 R_1 与 R_2，R_1 中训练集的各个样本取值的算术平均数为10，R_2 中训练集的各个样本取值的算术平均数为20。

则，对给定的观测值 $X=x$，若 $x \in R_1$，给出的预测值为10，若 $x \in R_2$，则预测值为20。

扫描全能王 创建

回归树：

在训练数据集所在的输入空间中，递归的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：（c_1 与 c_2 为对应的两个类别）

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

遍历变量 j，对固定的切分变量 j，扫描切分点 s，选择使得上式达到最小值的对 (j,s)　[j 是指的一个输入空间，s 是对空间的划分]

(2) 用选定的 (j,s) 划分区域并决定相应的输出值：

$R_1(j,s)=\{x \mid x^{(j)} \le s\}$，在区间 j 内的输入 x，若小于 s，则输出 R

$R_2(j,s)=\{x \mid x^{(j)} > s\}$，、、、、、、，大于 s，…… R

$\hat{c}_m=\frac{1}{N}\sum_{x_i \in R_m(j,s)} y_i,\ x \in R_m,\ m=1,2$　(下一个类别是)

(3) 继续对两个子区域调用步骤(1)与(2)，直到满足停止条件.

notebo

6.5优缺点

· 优缺点

1) 计算复杂度不高

2) 可以对树形结构表示，结果容易被理解

3) 决策树的预测准确性如一般，回归方法较弱，但可以通过集成学习方法组合大量决策树，显著提升树模型的性能。