## ****ID3、C4.5和CART的区别****

此为决策树的三个类别，区分如下

相同点：

CART与ID3和C4.5相同都由特征选择，树的生成，剪枝组成。

不同点：

①ID3和C4.5用于分类，CART可用于分类与回归

②ID3和C4.5生成的决策树可以是**多叉**的，每个节点下的叉树由该节点特征的取值种 类而定，比如特征年龄分为（青年，中年，老年），那么改节点下可分为3叉。而CART为假设决策树为**二叉树**，内部结点特征取值为”是”和”否”

③CART分类树通过**基尼指数**选择最优特征，同时**决定该特征的最优二值切分点**，而ID3和C4.5直接选择最优特征，不用划分。

ID3通过信息增益来选择特征，C4.5通过信息增益比来选择特征

## 决策树的剪枝

决策树的剪枝氛围预剪枝和后剪枝两种，其中后剪枝用得较多。

### 区分预剪枝、后剪枝

预剪枝：在构造决策树的同时进行剪枝。所有决策树的构建方法，都是在无法进一步降低熵的情况下才会停止创建分支的过程，为了避免过拟合，可以设定一个阈值，熵减小的数量小于这个阈值，即使还可以继续降低熵，也停止继续创建分支。但是这种方法实际中的效果并不好。

**常用的预剪枝的依据：**

①作为叶结点或作为根结点需要含的最少样本个数

②决策树的层数

③结点的经验熵小于某个阈值才停止

**常用的后剪枝算法：**

①Reduced-Error Pruning (REP,错误率降低剪枝）

这个思路很直接，完全的决策树不是过度拟合么，我再搞一个测试数据集来纠正它。对于完全决策树中的每一个非叶子节点的子树，我们尝试着把它替换成一个叶子节点，该叶子节点的类别我们用子树所覆盖训练样本中存在最多的那个类来代替，这样就产生了一个简化决策树，然后比较这两个决策树在测试数据集中的表现，如果简化决策树在测试数据集中的错误比较少，那么该子树就可以替换成叶子节点。该算法以**bottom-up**的方式遍历所有的子树，直至没有任何子树可以替换使得测试数据集的表现得以改进时，算法就可以终止。

②Pessimistic Error Pruning(PEP，悲观剪枝)

PEP剪枝算法是在C4.5决策树算法中提出的， 把一颗子树（具有多个叶子节点）用一个叶子节点来替代（我研究了很多文章貌似就是用子树的根来代替）的话，比起REP剪枝法，它**不需要一个单独的测试数据集**。

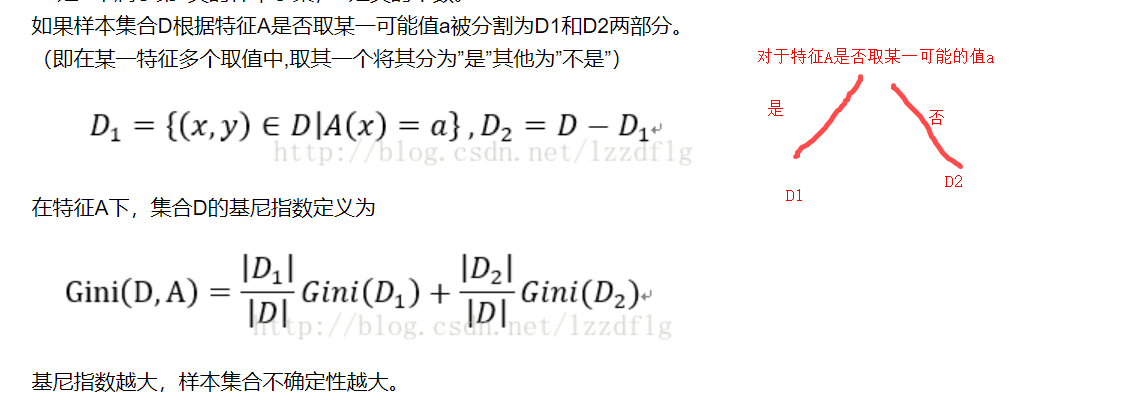
## CART分类回归树

### 基尼指数

注意基尼指数的定义：正确被分为该类别的概率×错误被分为该类别的概率，所有类别求和

基尼指数值越大，样本集合的不确定性也就越大，这一点和熵类似。

对于特征选择时，基尼指数的计算：



### **CART树生成**

(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，**对于每个特征，每次取其中一个每个可能取得值(因为此处为二叉树，并且分支只能为是/不是)**，根据样本点对该值的测试结果为”是”或”否”将D分成2部分，并计算基尼指数

(2) 在所有可能的特征，以及他们所有可能的切分点中，选择基尼指数最小的特征，该选择的特征为最优特征，该特征中的切分点为最优切分点，依照最优特征和最优切分点生成二叉树，并把训练数据集分配到子节点中

(3) 对两个子节点递归调用 (1) (2) ,直至满足停止条件

(4) 生成CART决策树

停止条件是样本个数小于预定阈值，或样本基尼指数小于预定阈值，或没有更多特征。

#### CART构建决策树与ID3、C4.5的不同

|  |  |  |  |
| --- | --- | --- | --- |
| 种类 | CART | ID3 | C4.5 |
| 划分依据 | 基尼指数 | 信息增益 | 信息增益比 |
| 切分点 | 针对每个特征下的所有可能值单独求基尼指数，即存在特征取值切分点 | 针对每一个特征，不将特征进行划分。 | |

### **CART剪枝**

CART剪枝分为两个步骤：

①剪枝成子树序列；

②通过交叉验证选取最优子树。

参考链接：

https://www.cnblogs.com/luban/p/9412339.html