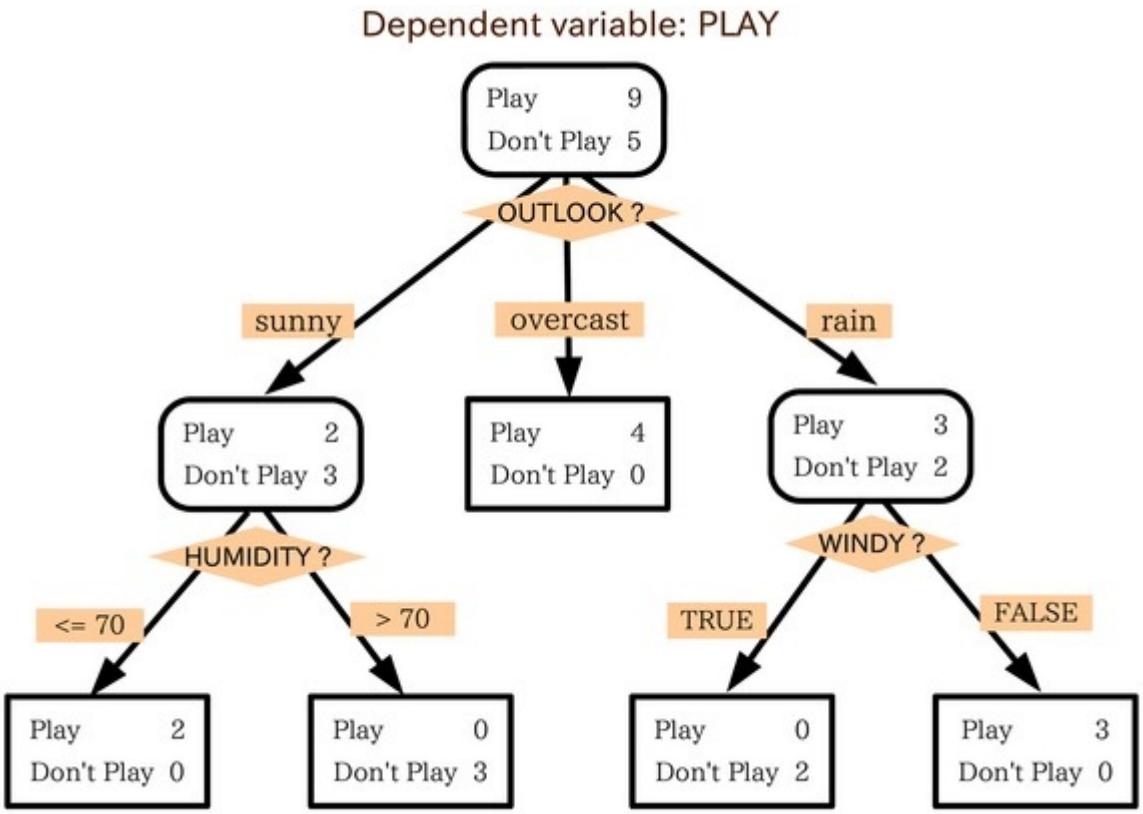
决策树

1.1 原理

顾名思义，决策树就是用**一棵树**来表示我们的整个决策过程。这棵树可以是二叉树（比如CART只能是二叉树），也可以是多叉树（比如ID3、C4.5可以是多叉树或二叉树）。

根节点包含整个样本集，每个**叶节点**都对应一个**决策结果**（注意，不同的叶节点可能对应同一个决策结果），每一个**内部节点**都对应一次决策过程或者说是一次**属性测试**。从根节点到每个叶节点的路径对应一个**判定测试序列**。

举个例子：



就像上面这个例子，训练集由三个特征：outlook(天气)，humidity（湿度），windy（是否有风）。那么我们该如何选择特征对训练集进行划分那？连续型特征（比如湿度）划分的阈值又是如何确定的那？

决策树的生成就是不断的**选择最优的特征**对训练集进行划分，是一个**递归**的过程。递归返回的条件有三种：

（1）当前节点包含的样本属于同一类别，无需划分

（2）当前属性集为空，或所有样本在属性集上取值相同，无法划分

（3）当前节点包含样本集合为空，无法划分

1.2 ID3、C4.5、CART

> 这三个是非常著名的决策树算法。简单粗暴来说，ID3使用**信息增益**作为选择特征的准则；C4.5使用**信息增益比**作为选择特征的准则；CART使用**Gini指数**作为选择特征的准则。

ID3

熵表示的是数据中包含的信息量大小。熵越小，数据的纯度越高，也就是说数据越趋于一致，这是我们希望的划分之后每个子节点的样子。

`信息增益 = 划分前熵 - 划分后熵`。**信息增益越大，则意味着使用属性a来进行划分所获得的"纯度提升"越大。**也就是说，用属性a来划分训练集，得到的结果中纯度比较高。

ID3仅仅能够处理离散属性。

信息熵：

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

条件熵：

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

信息增益：

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

	年龄	长相	工资	写代码	类别
小A	老	帅	高	不会	不见
小B	年轻	一般	中等	会	见
小C	年轻	丑	高	不会	不见
小D	年轻	一般	高	会	见
小L	年轻	一般	低	不会	不见

在这个问题中，

$$H(D)=-\frac{3}{5}\log_2\frac{3}{5}-\frac{2}{5}\log_2\frac{2}{5}=0.97$$

根据年龄进行划分：

$$H(D|\text{年龄})=\frac{1}{5}(-0)+\frac{4}{5}\left(-\frac{2}{4}\log_2\frac{2}{4}-\frac{2}{4}\log_2\frac{2}{4}\right)=0.8$$

C4.5

C4.5克服了ID3仅仅能够处理离散属性的问题，以及信息增益偏向选择取值较多特征的问题，使用信息增益比来选择特征。`信息增益比 = 信息增益 /划分前熵` 选择信息增益比最大的作为最优特征。

C4.5处理连续特征是先将特征取值排序，以连续两个值中间值作为划分标准。尝试每一种划分，并计算修正后的信息增益，选择信息增益最大的分裂点作为该属性的分裂点。

信息增益比：

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

其中分母是取值熵：

$$H_A(D)=-\sum_{i=1}^{n}\frac{|D_i|}{|D|}\log_2\frac{|D_i|}{|D|}$$

我们计算长相的取值熵：

$$H_{\text{长相}}(D)=-\frac{1}{5}\log_2\frac{1}{5}-\frac{3}{5}\log_2\frac{3}{5}-\frac{1}{5}\log_2\frac{1}{5}=1.371$$

那么，特征长相的信息增益比是：

$$g_R(D,\text{长相})=\frac{0.42}{1.371}=0.306$$

CART

CART与ID3，C4.5不同之处在于CART生成的树必须是**二叉树**。也就是说，无论是回归还是分类问题，无论特征是离散的还是连续的，无论属性取值有多个还是两个，内部节点只能根据属性值进行**二分**。

CART的全称是分类与回归树。从这个名字中就应该知道，CART既可以用于分类问题，也可以用于回归问题。

回归树中，使用**平方误差最小化准则**来选择特征并进行划分。每一个叶子节点给出的预测值，是划分到该叶子节点的所有样本目标值的**均值**，这样只是在给定划分的情况下最小化了平方误差。要确定**最优化分**，还需要遍历所有属性，以及其所有的取值来分别尝试划分并计算在此种划分情况下的**最小平方误差**，选取最小的作为此次划分的依据。由于回归树生成使用**平方误差最小化准则**，所以又叫做**最小二乘回归树**。

分类树种，使用**Gini指数最小化准则**来选择特征并进行划分；

Gini指数表示集合的不确定性，或者是不纯度。基尼指数越大，集合不确定性越高，不纯度也越大。这一点和熵类似。另一种理解基尼指数的思路是，**基尼指数是为了最小化误分类的概率。**

我们通过百面书上的例子来说明一下Gini系数的计算过程：

Gini纯度公式：

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

按特征A切成两份后的Gini指数公式：

$$\text{Gini}(D|A)=\sum_{i=1}^{n}\frac{|D_i|}{|D|}\text{Gini}(D_i)$$

	年龄	长相	工资	写代码	类别
小A	老	帅	高	不会	不见
小B	年轻	一般	中等	会	见
小C	年轻	丑	高	不会	不见
小D	年轻	一般	高	会	见
小L	年轻	一般	低	不会	不见

当A为年龄=老时：

$Gini(D|_{\text{年龄}=\text{老}}) = \frac{1}{5}(1-(\frac{1}{1})^2+(\frac{0}{1})^2)+\frac{4}{5}(1-(\frac{1}{2})^2-(\frac{1}{2})^2) = 0.4$

当A为工资=高时：

$Gini(D|_{\text{工资}=\text{高}}) = \frac{3}{5}(1-(\frac{2}{3})^2+(\frac{1}{3})^2)+\frac{2}{5}(1-(\frac{1}{2})^2-(\frac{1}{2})^2) = 0.47$

1.3 信息增益 vs 信息增益比

之所以引入了信息增益比，是由于信息增益的一个缺点。那就是：**信息增益总是偏向于选择取值较多的属性。**信息增益比在此基础上增加了一个罚项，解决了这个问题。

1.4 Gini指数 vs 熵

既然这两个都可以表示数据的不确定性，不纯度。那么这两个有什么区别那？

- Gini指数的计算**不需要对数运算**，更加高效
- Gini指数更偏向于连续属性，熵更偏向于离散属性

1.5 剪枝

决策树算法很容易过拟合（overfitting），剪枝算法就是用来防止决策树过拟合，提高泛华性能的方法。

剪枝分为**预剪枝**与**后剪枝。**

预剪枝是指在决策树的生成过程中，对每个节点在划分前先进行评估，若当前的划分不能带来泛化性能的提升，则停止划分，并将当前节点标记为叶节点。

后剪枝是指先从训练集生成一颗完整的决策树，然后自底向上对非叶节点进行考察，若将该节点对应的子树替换为叶节点，能带来泛化性能的提升，则将该子树替换为叶节点。

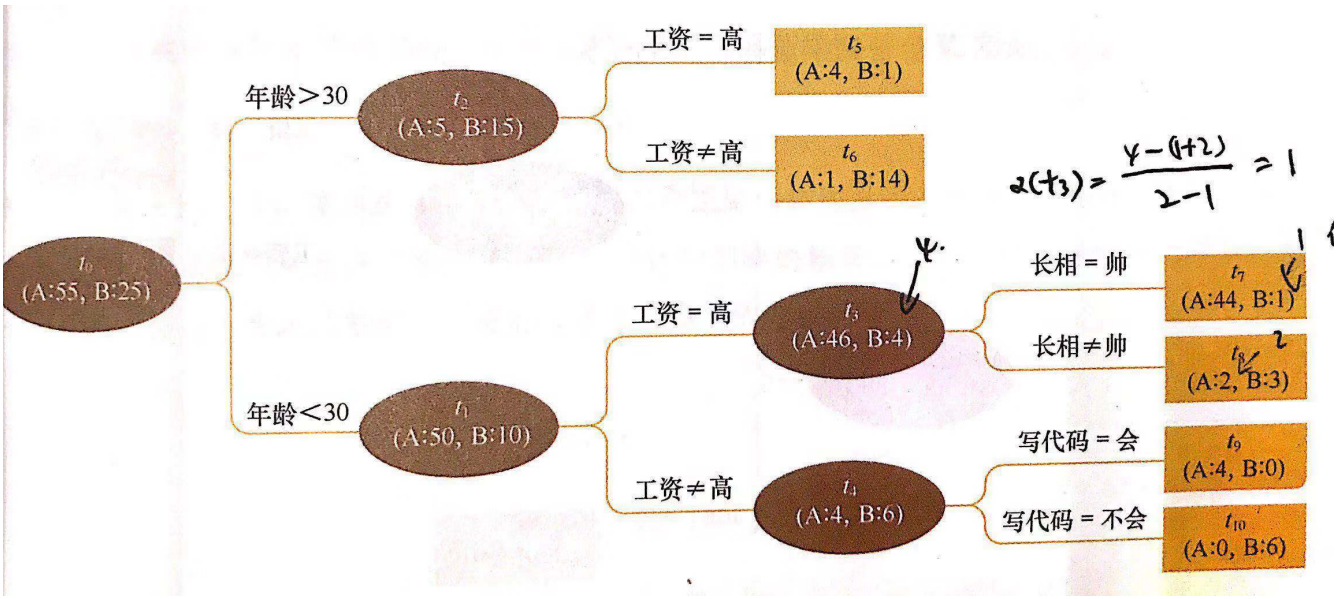
那么怎么来判断是否带来泛化性能的提升那？最简单的就是**留出法**，即预留一部分数据作为**验证集**来进行性能评估。

我们讲述一下百面书上的代价复杂剪枝：

女孩需要对80个人进行见或不见的分类。假设根据某种规则，已经得到了一棵CART决策树T_0

从T_0开始，裁剪T_i中关于训练数据集合误差增加最小的分支以得到T_{i+1}。具体地，当一棵树T在结点t处剪枝时，它的误差增加可以用$R(t)-R(T_t)$表示，其中$R(t)$表示进行剪枝之后的该结点误差，$R(T_t)$表示未进行剪枝时子树T_t的误差。考虑到树的复杂性因素，我们用$|L(T_t)|$表示子树T_t的叶子结点个数，则树在结点t处剪枝后的误差增加率为

$$\alpha=\frac{R(t)-R\left(T_t\right)}{\left|L\left(T_t\right)\right|-1}$$



在t_3处剪枝，剪枝之前误差是1+2（类别中较少的样本数），剪枝之后误差是4。子树叶节点个数为2。误差增加率为

$\alpha(t_3)=\frac{4-(1+2)}{2-1}=1$

其他的误差增加率依次计算。

1.6 总结

决策树算法主要包括三个部分：特征选择、树的生成、树的剪枝。常用算法有ID3、C4.5、CART。

- 特征选择。特征选择的目的是选取能够对训练集分类的特征。特征选择的关键是准则：信息增益、信息增益比、Gini指数。
- 决策树的生成。通常是利用信息增益最大、信息增益比最大、Gini指数最小作为特征选择的准则。从根节点开始，递归的生成决策树。相当于是不断选取局部最优特征，或将训练集分割为基本能够正确分类的子集。
- 决策树的剪枝。决策树的剪枝是为了防止树的过拟合，增强其泛化能力。包括预剪枝和后剪枝。

问：决策树中连续值和缺失值特征是如何处理的？

答：决策树中，对于连续属性，假设有n个样本，那么首先按照取值从小到大进行排序。取每两个值的中值作为候选的划分点进行划分。n个样本，对应有n-1个区间，也就是n-1个候选划分点。尝试所有划分点之后，分别计算信息增益，选取信息增益最大的划分点即可。对于属性有缺失值的情况，划分过程中计算属性信息增益的时候，只使用属性没有缺失值的样本进行信息增益的计算。确定好分类之后，对于在该属性值有缺失的样本，将被归入所有的分支节点。