# 决策树之ID3算法

**D3算法选择特征的依据是信息增益、C4.5是信息增益比，而CART则是Gini指数。**

决策树：从根节点开始一步步走到叶子节点（决策）

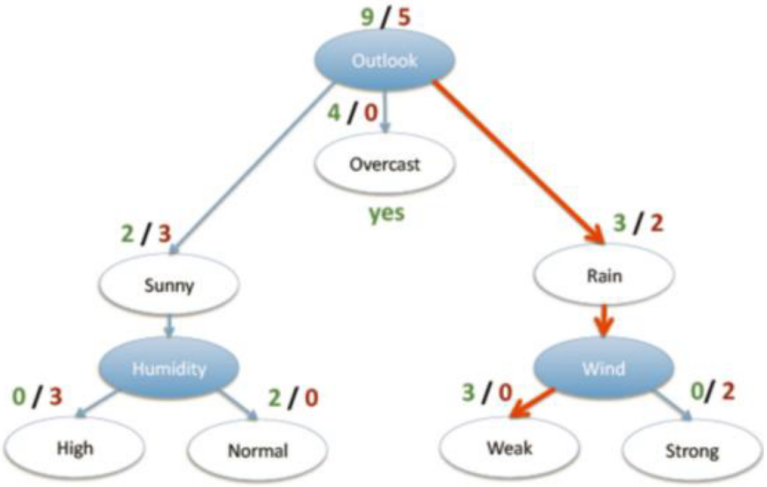
所有的数据最终都会落到叶子节点，既可以做分类也可以做回归

**树的组成**

根节点：第一个选择点决策树

非叶子节点与分支：中间过程

叶子节点：最终的决策结果




决策树学习通常包括 3 个步骤：**特征选择、决策树的生成、决策树的修剪。**最为关键的就是如何选择最优划分属性。一般而言，随着划分过程不断进行，我们希望决策树的分支结点所包含的样本尽可能属于同一类别，即结点的“纯度”(purity) 越来越高。

问题：根节点的选择该用哪个特征呢？接下来呢？如何切分呢？

**这里的关键在于如何选择最优特征对数据集进行划分：ID3算法采用信息增益**

在讲信息增益之前，这里我们必须先介绍下熵的概念。在信息论里面，熵是一种表示随机变量不确定性的度量方式。若离散随机变量X的概率分布为：

$$P(X = x_i) = p_i$$

则随机变量X的熵定义为：

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

意思是一个变量的变化情况可能越多，那么它携带的信息量就越大。

当给定随机变量X的条件下随机变量Y的熵可定义为条件熵H(Y|X)：

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

所谓信息增益就是数据在得到特征X的信息时使得类Y的信息不确定性减少的程度。假设数据集D的信息熵为H(D)，给定特征A之后的条件熵为H(D|A)，则特征A对于数据集的信息增益g(D,A)可表示为：

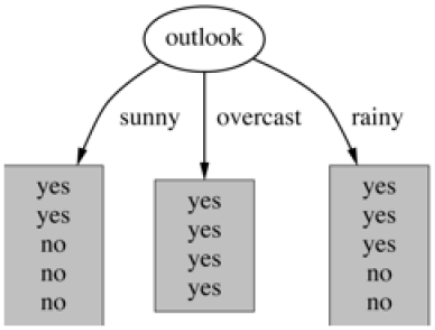
$$g(D,A) = H(D) - H(D|A)$$

信息增益越大，则该特征对数据集确定性贡献越大，表示该特征对数据有较强的分类能力。

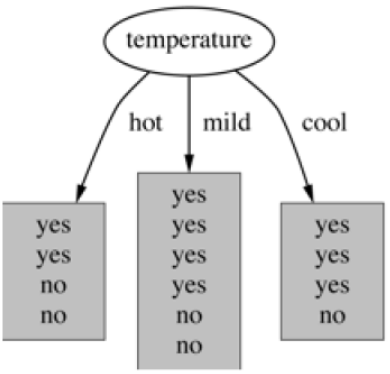
| Day | Outlook | Temperature | Humidity | Wind | PlayTennis |
|---|---|---|---|---|---|
| D1 | Sunny | Hot | High | Weak | No |
| D2 | Sunny | Hot | High | Strong | No |
| D3 | Overcast | Hot | High | Weak | Yes |
| D4 | Rain | Mild | High | Weak | Yes |
| D5 | Rain | Cool | Normal | Weak | Yes |
| D6 | Rain | Cool | Normal | Strong | No |
| D7 | Overcast | Cool | Normal | Strong | Yes |
| D8 | Sunny | Mild | High | Weak | No |
| D9 | Sunny | Cool | Normal | Weak | Yes |
| D10 | Rain | Mild | Normal | Weak | Yes |
| D11 | Sunny | Mild | Normal | Strong | Yes |
| D12 | Overcast | Mild | High | Strong | Yes |
| D13 | Overcast | Hot | Normal | Weak | Yes |
| D14 | Rain | Mild | High | Strong | No |

**其中4个条件属性，一个决策属性（PlayTennis）**

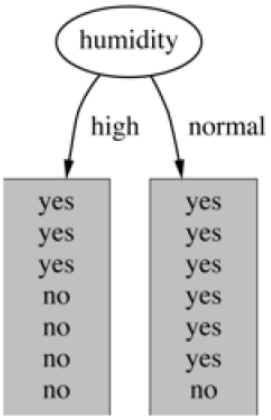
## 1.基于天气的划分



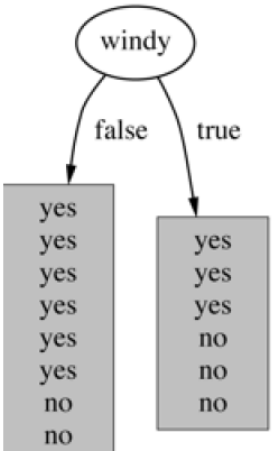

## 2.基于温度的划分




## 3.基于湿度的划分




## 4.基于有风的划分




**划分方式：4种决策树**

**问题：谁当根节点呢?**

**依据：信息增益**

**决策属性PlayTennis的信息熵Entropy：**

$$\text{Entropy(PT)} = \text{Entropy(Yes9, No5)} = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940286$$

**Outlook = sunny时，熵值为：**

$$\text{Entropy(Sunny)} = \text{Entropy(Yes2, No3)} = -\frac{2}{5}\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\log_2\left(\frac{3}{5}\right) = 0.97095$$

**Outlook = overcast时，熵值为：**

$$\text{Entropy(Overcast)} = \text{Entropy(Yes4, No0)} = -\frac{4}{4}\log_2\left(\frac{4}{4}\right) - \frac{0}{4}\log_2\left(\frac{0}{4}\right) = 0.0$$

**Outlook = rain时，熵值为：**

$$\text{Entropy(Rain)} = \text{Entropy(Yes3, No2)} = -\frac{3}{5}\log_2\left(\frac{3}{5}\right) - \frac{2}{5}\log_2\left(\frac{2}{5}\right) = 0.97095$$

**Outlook的信息熵Entropy：**

根据数据统计,outlook取值分别为**sunny,overcast,rainy**的概率分别为：**5/14, 4/14, 5/14**

**Entropy(Outlook) = Entropy(Sunny)+Entropy(Overcast)+Entropy(Rain)**

**= 5/14 * 0.971 + 4/14 * 0 + 5/14 * 0.971 = 0.693**

## Outlook 的信息增益 Gain(PT,Ol)：

$$\text{Gain(PT, Ol)} = \text{Entropy(PT)} - \frac{5}{14}\text{Entropy(Sunny)} - \frac{4}{14}\text{Entropy(Overcast)} - \frac{5}{14}\text{Entropy(Rain)}$$

$$= 0.940286 - \frac{5}{14} \times 0.97095 - 0.0 - -\frac{5}{14} \times 0.97095 \ = 0.24675$$

## 同理可得其他属性的信息增益：

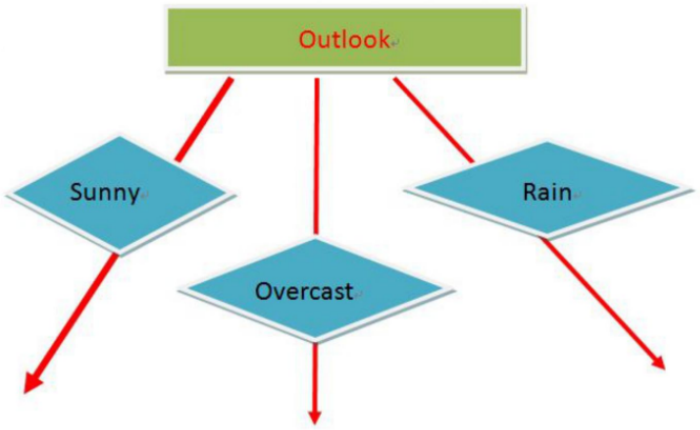
**Gain(PT,Temp)=0.029**

**Gain(PT,Humidity)=0.151**

**Gain(PT,Wind)=0.048**

信息增益：系统的熵值从原始的0.940下降到了0.693，增益为0.247,同样的方式可以计算出其他特征的信息增益，那么我们选择最大的那个就可以啦，相当于是遍历了一遍特征，找出来了大当家，然后再其余的中继续通过信息增益找二当家！

**Outlook的信息增益最大作为决策树的第一个根节点：**




从Outlook下面出来的三个分支，最左边的Sunny，是Outlook中信息增益最大的那个，以此类推

# C4.5算法

## 分离信息（Split Informatiion）

数据集通过条件属性A的分离信息

分离信息的计算方法公式为：

$$SplitInformation(\mathrm{S,A}) = -\sum_{i=1}^{c}\frac{|S_i|}{S}\log_2\left(\frac{|S_i|}{|S|}\right)$$

数据集通过Outlook这个条件属性的分离信息，Outlook有三个属性值分别为：sunny,overcast,rain，它们各占5，4，5，所以

$$SplitInformation(PT,\text{Outlook}) = -\frac{5}{14}\log_2\left(\frac{5}{14}\right) - \frac{4}{14}\log_2\left(\frac{4}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 1.577$$

**同理可得其他属性的分离信息：**

**SplitInformation(PT,Temp)=1.5566**

**SplitInformation(PT,Humidity)=1.0**

**SplitInformation(PT,Wind)=0.9402**

**信息增益比（information gain ratio）**

信息增益比的计算方法公式为：

$$GainRatio(S,\text{A}) = \frac{Gain(S,A)}{SplitInformation(S,A)}$$

**根据公式可得：**

**GainRatio(PT,Outlook)=0.1564**

**GainRatio(PT,Temp)=0.0187**

**GainRatio(PT,Humidity)=0.1518**

**GainRatio(PT,Wind)=0.1087**

**基尼指数（Gini index）**

ID3还是C4.5都是基于信息论的熵模型的，这里面会涉及大量的对数运算，为了简化模型同时也完全丢失熵模型的优点，**在CART算法中使用基尼系数来代替信息增益比，基尼系数代表了模型的不纯度，基尼系数越小，则不纯度越低，特征越好。这和信息增益(比)是相反的。**通过子集计算基尼不纯度，即随机放置的数据项出现于错误分类中的概率。以此来评判属性对分类的重要程度。

$$Gini(D) = \sum_{k=1}^{K} p_k(1-p_k)$$

其中pk为任一样本点属于第k类的概率，也可以说成样本数据集中属于k类的样本的比例。

集合D的基尼指数为Gini(D)，在特征A条件下的集合D的基尼指数为

$$Gini(D|A) = \sum_{k=1}^{K} \frac{|D_k|}{|D|} Gain(D_k)$$

**其中 |Dk|为特征A取第i个值时对应的样本个数。|D|为总样本个数**

**CART算法中对于分类树采用的是上述的基尼指数最小化准则。**

**对于回归树，CART采用的是平方误差最小化准则。**

树的建立过程明显是一个递归的过程!

**输入:** 训练集 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$;
属性集 $A = \{a_1, a_2, \ldots, a_d\}$.

**过程:** 函数 TreeGenerate($D$, $A$)

1: 生成结点 node;
2: **if** $D$ 中样本全属于同一类别 $C$ **then**
3: 将 node 标记为 $C$ 类叶结点; **return** 递归返回, 情形(1).
4: **end if**
5: **if** $A = \varnothing$ **OR** $D$ 中样本在 $A$ 上取值相同 **then**
6: 将 node 标记为叶结点, 其类别标记为 $D$ 中样本数最多的类; **return** 递归返回, 情形(2).
7: **end if**
8: 从 $A$ 中选择最优划分属性 $a_*$; 我们将在下一节讨论如何获得最优划分属性.
9: **for** $a_*$ 的每一个值 $a_*^v$ **do**
10: 为 node 生成一个分支; 令 $D_v$ 表示 $D$ 中在 $a_*$ 上取值为 $a_*^v$ 的样本子集;
11: **if** $D_v$ 为空 **then**
12: 将分支结点标记为叶结点, 其类别标记为 $D$ 中样本最多的类; **return** 递归返回, 情形(3).
13: **else**
14: 以 TreeGenerate($D_v$, $A \setminus \{a_*\}$)为分支结点 从 $A$ 中去掉 $a_*$.
15: **end if**
16: **end for**

**输出:** 以 node 为根结点的一棵决策树

**图 4.2** 决策树学习基本算法