3.1 决策树算法

- 决策树分类算法通常分为两个步骤：决策树生成和决策树修剪。

- （1）决策树生成算法
- 决策树生成算法的输入参数是一组带有类别标记的样本，输出是构造一颗决策树，该树可以是一棵二叉树或多叉树。二叉树的内部结点（非叶子结点）一般表示为一个逻辑判断，构造决策树的方法是采用自上而下的递归方法。

3.1 决策树算法

- （2）决策树修改算法
- 剪枝是一种克服噪声的基本技术，有两种基本的剪枝策略：

- （1）预先剪枝（Pre-Pruning）：在生成树的同时决定是继续对不纯的训练子集进行划分还是停止。
- （2）后剪枝（Post-pruning）：一种拟合-化简（Fitting-and-simplifying）的两阶段方法。首先生成与训练数据完全拟合的一棵决策树，然后从树的叶子开始剪枝，逐步向根的方向剪。剪枝时要用到一个测试数据集合（Tuning Set或Adjusting Set），如果存在某个叶子剪去后测试集上的准确度或其他测试度不降低，则剪去该叶子；否则停止。

3.1 决策树算法

- 三项关键技术

（1）选择最能区别数据集中实例属性的方法

（2）剪枝方法

（3）检验方法

1、 选择最能区别数据集中实例属性的方法

- 构造好的决策树的关键在于如何选择好的逻辑判断或属性。对于同样一组样本，可以有很多决策树符合这组样本。一般情况下，树越小则树的预测能力越强。要构造尽可能小的决策树，关键在于选择合适的产生分支的属性。

- C4.5使用了信息论（Information Theory）的方法，即使用增益率（Gain Ratio）的概念来选择属性；
  - 目的是使树的层次和节点数最小，使数据的概化程度最大化。
- C4.5选择的基本思想
  - 选择具有最大增益率的属性作为分支节点来分类实例数据。

1）信息熵（Information Entropy）

- 信息是个很抽象的概念。人们常常说信息很多，或者信息较少，但却很难说清楚信息到底有多少。比如一本五十万字的中文书到底有多少信息量。
- 1948年，克劳德·香农（Claude Shannon）提出“信息熵”（Information Entropy）的概念，才解决了对信息的量化度量问题。
- 信息变化的平均信息量称为“信息熵”（信息量化）
- 在信息论中，信息熵是信息的不确定程度的度量。熵越大，信息就越不容易搞清楚，需要的信息量就越大，能传输的信息就越多。

1）信息熵（Information Entropy）

- 不确定性函数f是概率P的单调递降函数；两个独立符号所产生的不确定性应等于各自不确定性之和，即

$$\mathrm{f}(\mathrm{p}1,\mathrm{p}2)=f(p1)+f(p2)$$

- 这称为可加性。
- 同时满足这两个条件的函数f是对数函数，即

$$\mathrm{f}(\mathrm{p})=\log\frac{1}{p}=-\log p$$

1）信息熵（Information Entropy）

- 在信源中，考虑的不是某一单个符号发生的不确定性，而要考虑这个信源所有可能发生情况的平均不确定性。
- 若信源符号有n种取值：$U_1 \cdots U_i \cdots U_n$，对应概率为：$P_1 \cdots P_i \cdots P_n$，且各种符号的出现彼此独立。这时，信源的平均不确定性应当为单个符号不确定性$-\log P_i$的统计平均值（E），可称为信息熵，即
- 式中对数一般取2为底，单位为比特。也可取其它对数底，采用其它相应的单位，它们间可用换底公式换算。

$$H(U) = E[-\log p_i] = -\sum_{i=1} p_i \log p_i$$

1）信息熵（Information Entropy）

- 设I是训练样本集，它包括n个类别的样本，这些类别分别用 $C_1, C_2, \ldots, C_n$ 表示，那么I的熵（entropy）或者期望信息为
- $$\text{Info(I)} = -\sum_{i=0}^{n} p_i \log_2 p_i$$
- 其中，p_i 表示类 C_i 的概率。如果将I中的n类训练样本看成n种不同的消息，那么I的熵表示对每一种消息编码需要的平均比特数，$|\mathrm{I}| \times Info(I)$ 就表示对I进行编码需要的比特数，其中 $|I|$ 表示I中的样本数目。

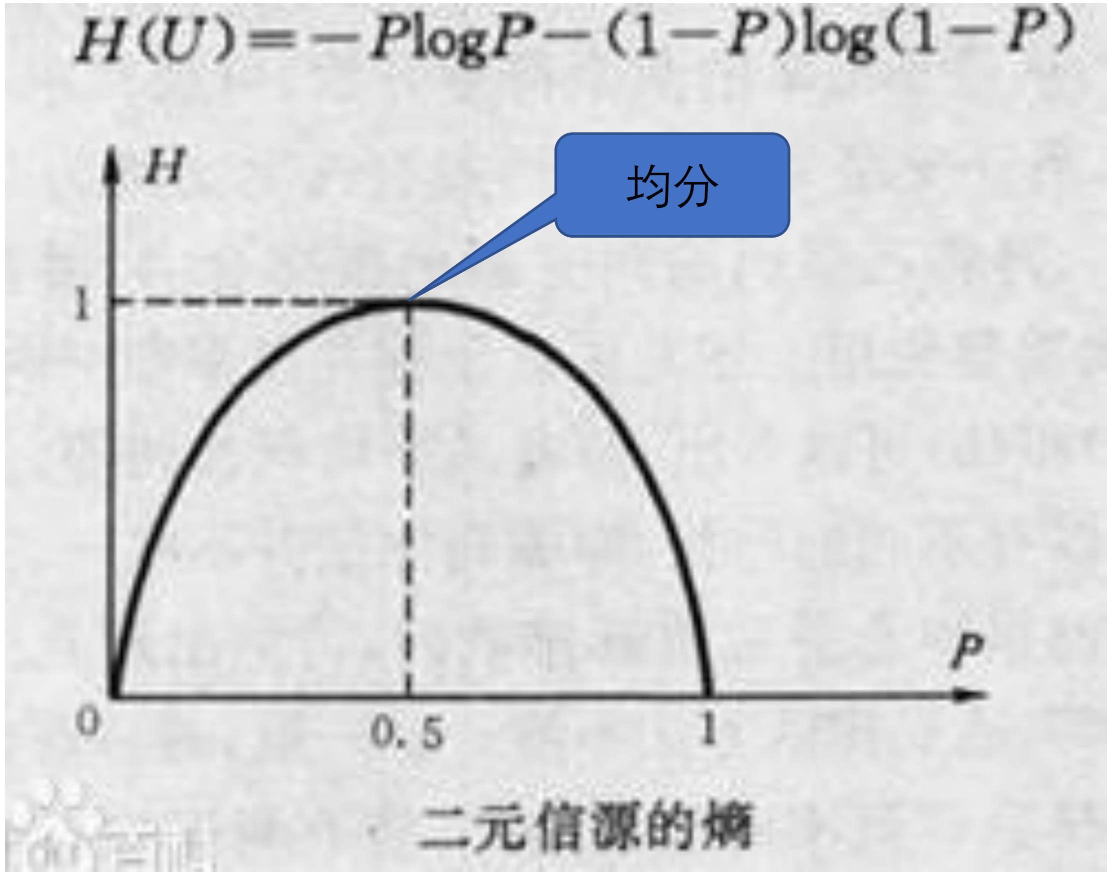
1）信息熵（Information Entropy）

- 如果n=2，p_1=0.5，p_2=0.5，则样本集I的熵为：
  - $Info(I)=-0.5\log_2 0.5-0.5\log_2 0.5=1$
- 如果n=2，p_1=0.67，p_2=0.33，则样本集I的熵为：
  - $Info(I)=-0.67\log_2 0.67-0.33\log_2 0.33=0.92$

- 样本的概率分布越均衡，它的信息量（熵）就越大，样本集的混杂程度也越高。
- 因此，熵可以作为训练集的不纯度（impurity）的一个度量，熵越大，不纯度就越高。

1）信息熵（Information Entropy）

- 当n=2，熵的分布为：



2）信息增益（Information Gain）

- 设属性A将I划分成m份，根据A划分的子集的熵或期望信息由下式给出：
- $$\mathrm{Info}(\mathrm{I}, A) = \sum_{i=0}^{m} \frac{|I_i|}{|I|} Info(I_i)$$
- 其中， I_i 表示根据属性A划分的I的第i个子集， $|\mathrm{I}|$ 和 $|\mathrm{I}_i|$ 分别表示 I 和 I_i 中的样本数目。
- 信息增益用来衡量熵的期望减少值，因此，使用属性A对I进行划分获得的信息增益为

  $$\mathrm{Gain}(A) =$$
- Gain(A)是指因为知道属性A的值后导致的熵的期望压缩。
- Gain(A)越大说明

3）信息增益率（Information Gain Ratio）

- 信息增益表示当x取属性x_i值时，其对降低x的熵的贡献大小。
- 信息增益值越大，越适于对x进行分类。
- C4.5使用信息量和信息增益的概念计算所有属性的增益，并计算所有属性的增益率，选择值最大的属性来划分数据实例。

$$\mathrm{GainRatio}(A) = \frac{\mathrm{Gain}(A)}{\mathrm{SplitsInfo}(A)}$$

计算属性*A*的增益率的公式

- SplitsInfo(A) 是对A属性的增益值的标准化，目的是消除属性选择上的偏差（Bias），即在所有实例的属性A的取值只有1个时，该属性总被优先选中的情况。

3）信息增益率（Information Gain Ratio）

- 训练样本关于属性值的信息量（熵）SplitsInfo（A），信息量的计算公式如下：

$$\text{SplitsInfo}(A) = -\sum_{i=0}^{m} \frac{|I_i|}{|I|} \log_2 \frac{|I_i|}{|I|}$$

- 其中，I代表训练样本集，A代表属性，其中，$\mathrm{I_i}$ 表示根据属性A划分的I的第i个样本子集。
- 样本在A上的取值分布越均匀，SplitsInfo的值也就越大。SplitsInfo用来衡量属性分裂数据的广度和均匀性。

3）信息增益率（ Information Gain Ratio）

- Info(*I*) 为当前数据集所有实例所表达的信息量

$$\mathrm{Info}(I) = -\sum_{i=1}^{n} \frac{出现在i类中的实例个数}{所有实例总数} \log_2\left(\frac{出现在i类中的实例个数}{所有实例总数}\right)$$

- Info(I,A) 为根据属性 A 的 k 个可能取值分类 I 中实例之后所表达的信息量

$$\mathrm{Info}(I, A) = \sum_{j=1}^{k} \frac{出现在j类中的实例个数}{所有实例总数} \mathrm{Info}(j类)$$

- SplitsInfo(A) 是对A属性的增益值的标准化，目的是消除属性选择上的偏差（Bias），

$$\mathrm{SplitsInfo}(A) = -\sum_{j=1}^{k} \frac{出现在j类中的实例个数}{所有实例总数} \log_2\left(\frac{出现在j类中的实例个数}{所有实例总数}\right)$$

表2.1 一个假想的打篮球数据集

序号	Weather	Temperature/°C	Courses	Partner	Play
1	Sunny	20~30	4	Yes	Yes
2	Sunny	20~30	4	No	Yes
3	Rain	10~0	1	Yes	Yes
4	Sunny	30~40	5	Yes	Yes
5	Rain	20~30	8	No	No
6	Sunny	-10~0	5	Yes	Yes
7	Sunny	-10~0	7	No	No
8	Rain	20~30	2	Yes	Yes
9	Rain	20~30	6	Yes	No
10	Sunny	10~20	6	Yes	No
11	Rain	10~20	3	No	No
12	Rain	10~20	1	Yes	No
13	Sunny	10~20	8	Yes	No
14	Sunny	0~10	3	Yes	Yes
15	Rain	0~10	2	Yes	No

以Weather作为根节点

（1）Info(*I*)=

（2）Info(*I*,Weather)=

其中：Info(Sunny)= -

Info(Rain)=

（3）SplitsInfo(Weather)= = 0.9968

（4）Gain(Weather) = Info(

（5）GainRatio(Weather) = Gain(Weather))/ Spl

= 0.085 / 0.9968 = 0.085

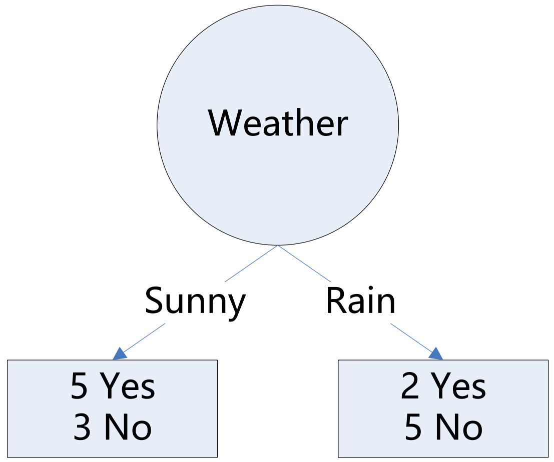


图2.2 Weather作为根节点的局部决策树

二元分裂点（Binary Splits）

- 数值型属性Courses的增益值如何计算呢？
  - C4.5算法对这些数值型数据进行排序，计算每个可能的二元分裂点的增益率值来离散化这个属性值。

1	1	2	2	3	3	4	4	5	5	6	6	7	8	8
Yes	No	Yes	No	No	Yes	Yes	Yes	Yes	Yes	No	No	No	No	No

表2.2 打篮球数据集中数值型属性Courses的排序结果

Courses属性作为根节点

- 计算4个属性的增益率值后，发现Courses属性的 ≤5 和 >5 分裂点处具有最佳增益率值，为0.4457。

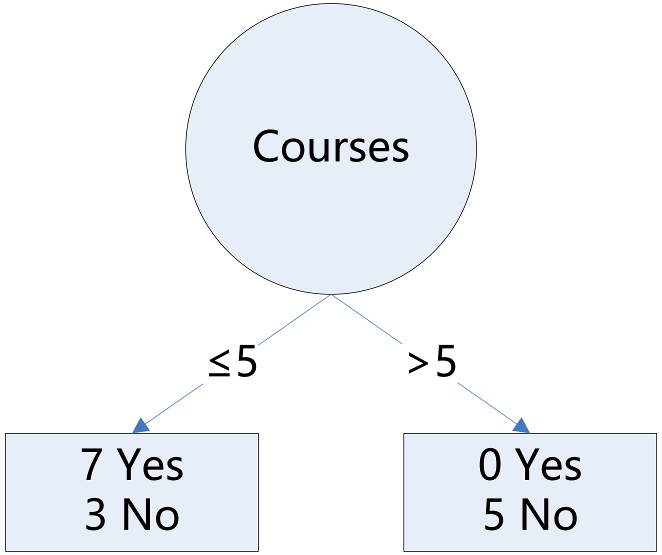


图2.3 Courses作为根节点的局部决策树

完整决策树

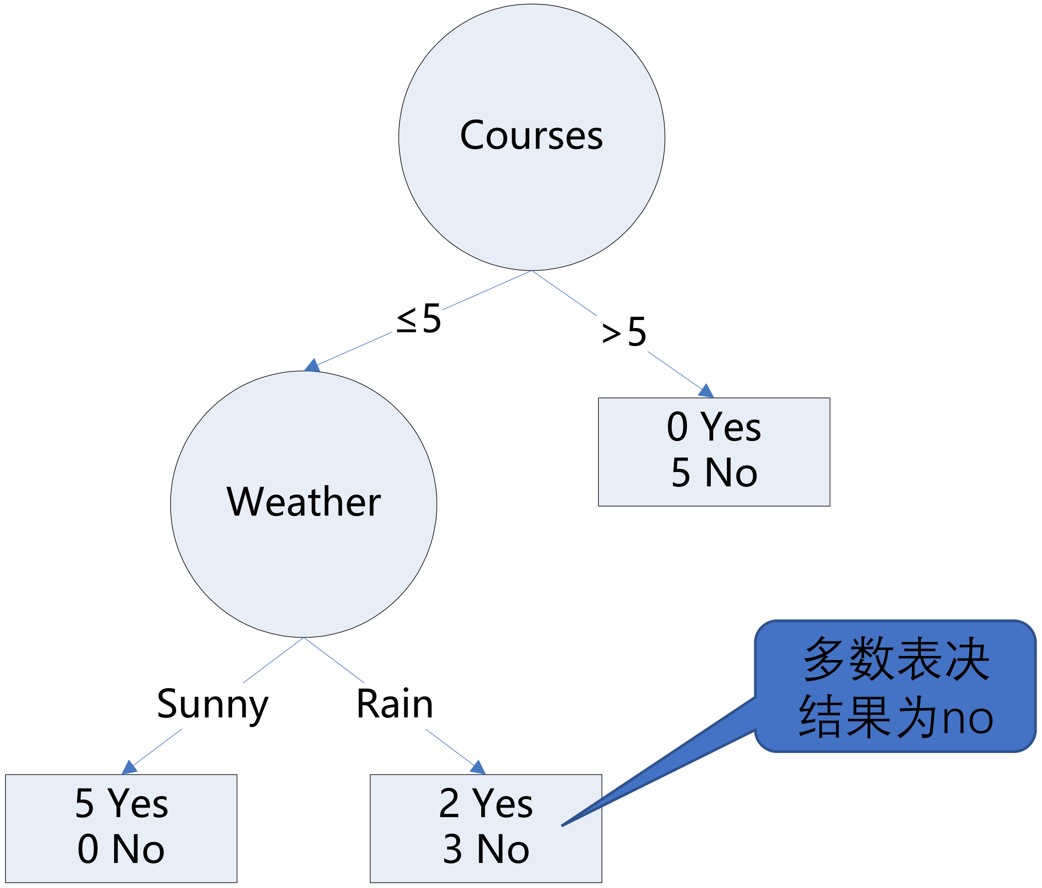


图2.4 *Courses*作为根节点的完整决策树

2、决策树剪枝

- 剪枝（Pruning）
  - 为控制决策树规模，优化决策树而采取的剪除部分分支的方法。
- 剪枝分为两种
  - 预剪枝（Pre-Pruning）
  - 后剪枝（Post-Pruning）

3、决策树检验

- **常用的4种检验方法**

（1）use training set：使用在训练集实例上的预测效果进行检验。

（2）supplied test set：使用另外提供的检验集实例进行检验。

（3）cross-validation：使用交叉验证（Cross Validation，简称CV）来检验分类器。

（4）percent split：百分比检验。从数据集中按一定百分比取出部分数据作为检验集实例用，根据分类器在这些实例上的预测效果来检验分类器的质量。

检验方法

- 当数据集规模较大时
  - 可将训练集与测试集之比设为6:4或7:3等。
- 当数据集规模较小时
  - K-折交叉验证

K-折交叉检验（k-CV）

- 检验分类器性能的一种最为常用的统计分析方法，
- 基本思想
  - 将数据集随机划分成k组
  - 之后执行k次循环，在第i次循环中，将第i组数据样本作为测试集，其余的k-1组数据作为训练集，最终的精度为k个精度的平均值。

- 如果数据集规模极小时，每次交叉验证时，只选择一条数据作为测试集，其余数据作为训练集。