决策树分享

Ricardo

决策树

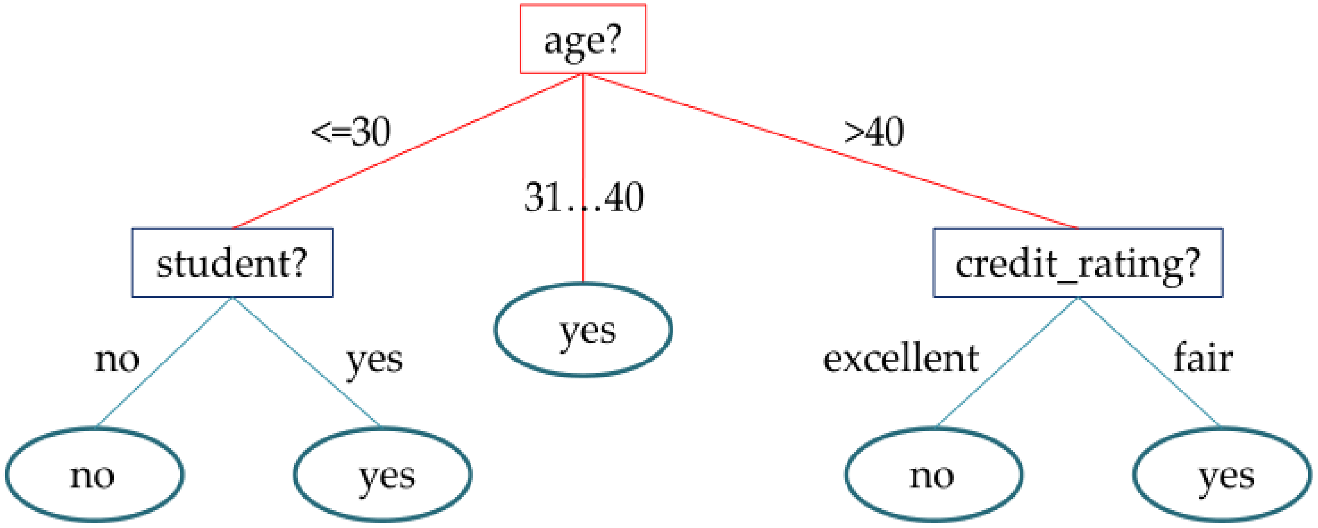
决策树是个既可以做分类也可以做回归的模型

内部节点是 属性(特征)

每条边上面是 属性值(特征值)

叶子结点是 分类结果(标签)

分类的过程就是顺着树从根节点走到叶结点

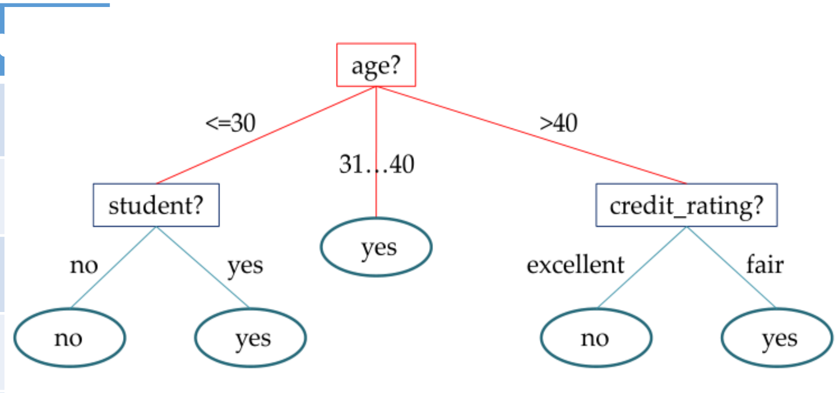


Age	Income	student	Credit_rating	Bus_computer
<=30	High	No	Fair	no
<=30	High	No	Excellent	No
31…40	High	No	Fair	Yes
>40	Medium	No	Fair	Yes
>40	Low	Yes	Fair	Yes
>40	Low	Yes	Excellent	No
31…40	Low	Yes	Excellent	Yes
<=30	Medium	No	Fair	No
<=30	Low	Yes	Fair	Yes
>40	Medium	Yes	Fair	Yes
<=30	Medium	Yes	Excellent	Yes
31…40	Medium	No	Excellent	Yes
31…40	High	Yes	Fair	Yes
>40	Medium	No	excellent	No

只要建立了决策树，
预测的过程是很简单的。

关键是怎么根据训练集
建立决策树。

Age	Income	student	Credit_rating	Bus_compu
<=30	High	No	Fair	no
<=30	High	No	Excellent	No
31…40	High	No	Fair	Yes
>40	Medium	No	Fair	Yes
>40	Low	Yes	Fair	Yes
>40	Low	Yes	Excellent	No
31…40	Low	Yes	Excellent	Yes
<=30	Medium	No	Fair	No
<=30	Low	Yes	Fair	Yes
>40	Medium	Yes	Fair	Yes
<=30	Medium	Yes	Excellent	Yes
31…40	Medium	No	Excellent	Yes
31…40	High	Yes	Fair	Yes
>40	Medium	No	excellent	No



注意到树的内部结点就是属性，现在问题是。**怎样决定用哪一个属性做根节点**

信息熵

假设你有一个八个面的色子，要用多少bit才能表示出所有可能性呢？

$$3bits = \log_2 8 = -\sum_{i=1}^{8} \frac{1}{8} \log_2 \frac{1}{8} = -\sum_{i=1}^{8} p(i) \log_2 p(i) = H(X)$$

答案是3bit，很简单，就是000，001，010，011，100，101，110，111。
用着3bit的排列组合可以表示八个数，把八种情况都包含了。

除了只接用log2（8）来求，还可以表示为右边的形式，因为色子是均匀的，所以投掷到每一面的概率都是1/8，右边式子的实质就是把发生各个事件的不确定性进行求和，得到的就是整个投掷色子时间的不确定性H(X)，也即投掷色子的信息熵。

信息熵

$$H(X) = -\sum_{x\in X} p(x)\log_2 p(x)$$

$$= \sum_{x\in X} p(x)\log_2 \frac{1}{p(x)}$$

$$= E\left(\log_2 \frac{1}{p(X)}\right)$$

熵就是这么一个用来描述**离散变量不确定性**的概念。

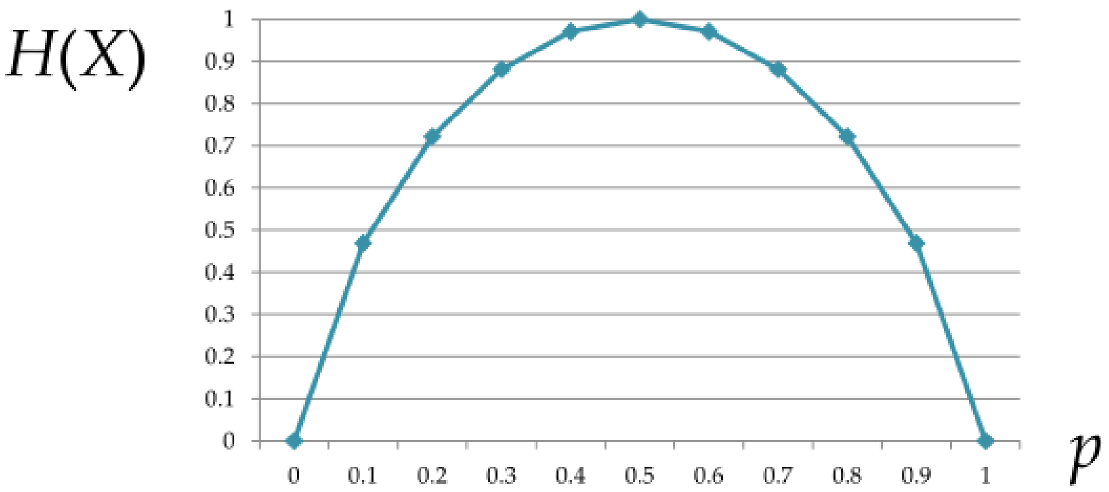
比方说抛硬币, 投色子等等...

再举个抛硬币的例子，一般来说正面背面都是二分之一的概率。这时H(X)就是1。

如果有人作弊，做了一个百分百正面的硬币。
这时按左式一算，H(x)就是0了，也即不确定性为0.
这是很显然的，都百分百正面了，
不确定性当然为0。

信息熵

$$H(X) = -p\log_2 p - (1-p)\log_2(1-p)$$



设抛到正面的概率是p，则反面的概率是1-p

可以看到概率分布是一个钟形函数。

当各个事件发生的概率相等时，总的不确定性越高，我们越难猜出会发生哪一个事件。

联系实际，如果有一个事件发生的概率比较大，我们往往就选择猜这个事件了。

条件熵和联合熵 Conditional&joint entropy

条件熵：

$$
\begin{aligned}
H(Y \mid X) &= \sum_{x \in X} p(x) H(Y \mid X = x) \\
&= \sum_{x \in X} p(x) \left[-\sum_{y \in Y} p(y \mid x) \log_2 p(y \mid x) \right] \\
&= -\sum_{x \in X} \sum_{y \in Y} p(x, y) \log_2 p(y \mid x) \\
&= -\sum_{x \in X} \sum_{y \in Y} p(x) p(y \mid x) \log_2 p(y \mid x)
\end{aligned}
$$

和条件概率类似，也有条件熵这个概念。
即已知发生事件X时，Y的不确定性。

注意这个概念很重要，我们可以把X理解为前面例子的属性，构建决策树时，我们希望每次选择结点都能**尽可能减少事情的不确定性。**

所以我们每次都会选能够让条件熵最小的一个属性作为新的节点。

条件熵和联合熵 Conditional&joint entropy

联合熵：　$$H(X,Y) = -\sum_{x\in X}\sum_{y\in Y} p(x,y)\log_2 p(x,y)$$

$$\begin{aligned}
H(X,Y) &= -E_{p(x,y)}\log_2 p(x,y) \\
&= -E_{p(x,y)}\left(\log_2(p(x)p(y\,|\,x))\right) \\
&= -E_{p(x,y)}\left(\log_2 p(x) + \log_2 p(y\,|\,x)\right) \\
&= -E_{p(x)}\log_2 p(x) - E_{p(x,y)}\log_2 p(y\,|\,x) \\
&= H(X) + H(Y\,|\,X)
\end{aligned}$$

和联合概率类似，也有联合熵这个概念。
即同时发生事件X和Y的不确定性。

注意联合熵的计算方式，两个离散变量X和Y的联合熵，或者说联合出现的不确定性 = X的熵加上 给定X，出现Y的条件熵 = X的不确定性加上 给定X，出现Y的不确定性。

X和Y的顺序是可以调转的。

互信息 Mutual information

因为：$H(X,Y)=H(X)+H(Y|X)=H(Y)+H(X|Y)$

所以：$H(Y)-H(Y|X)=H(X)-H(X|Y)=I(X;Y)$

一个连续变量X的不确定性，用方差Var(X)来度量

一个离散变量X的不确定性，用熵H(X)来度量

两个连续变量X和Y的相关度，用协方差或相关系数来度量

两个离散变量X和Y的相关度，用互信息I(X;Y)来度量
(直观地，X和Y的相关度越高，X对分类的作用就越大)

互信息用来衡量两个离散变量X和Y的相关度。

除了互信息还有一种叫法是

信息增益(information gain)

记作：

g(Y, X) = I(Y, X) = H(Y) – H(Y|X)

接下来就是如何利用上面
提及的概念构建决策树了

ID3决策树

ID	Age	Income	student	Credit_rating	Bus_computer
1	<=30	High	No	Fair	No
2	<=30	High	No	Excellent	No
3	31…40	High	No	Fair	Yes
4	>40	Medium	No	Fair	Yes
5	>40	Low	Yes	Fair	Yes
6	>40	Low	Yes	Excellent	No
7	31…40	Low	Yes	Excellent	Yes
8	<=30	Medium	No	Fair	No
9	<=30	Low	Yes	Fair	Yes
10	>40	Medium	Yes	Fair	Yes
11	<=30	Medium	Yes	Excellent	Yes
12	31…40	Medium	No	Excellent	Yes
13	31…40	High	Yes	Fair	Yes
14	>40	Medium	No	excellent	No

继续看这个例子，类标签是最后一列。

下面用D表示类标签，A表示属性。

$H(D)=0.940$

14个训练样本中，9个买了电脑

$$H(D)=-\frac{9}{14}\log_2\frac{9}{14}-(1-\frac{9}{14})\log_2(1-\frac{9}{14})$$

可以先算出买电脑这件事的不确定性。

ID3决策树

ID	Age	Income	student	Credit_rating	Bus_computer
1	<=30	High	No	Fair	No
2	<=30	High	No	Excellent	No
3	31…40	High	No	Fair	Yes
4	>40	Medium	No	Fair	Yes
5	>40	Low	Yes	Fair	Yes
6	>40	Low	Yes	Excellent	No
7	31…40	Low	Yes	Excellent	Yes
8	<=30	Medium	No	Fair	No
9	<=30	Low	Yes	Fair	Yes
10	>40	Medium	Yes	Fair	Yes
11	<=30	Medium	Yes	Excellent	Yes
12	31…40	Medium	No	Excellent	Yes
13	31…40	High	Yes	Fair	Yes
14	>40	Medium	No	excellent	No

接下来我们要算的是条件熵，找出能使条件熵最小的属性，那么根节点就是它了。

举age属性为例子：

- $H(D\,|\,A=\text{"}age\text{"})=0.694$

$$H(D\,|\,A=\text{"}age\text{"})=\frac{5}{14}\times\left(-\frac{2}{5}\log_2\frac{2}{5}-\frac{3}{5}\log_2\frac{3}{5}\right)$$

$$+\frac{4}{14}\times\left(-\frac{4}{4}\log_2\frac{4}{4}-\frac{0}{4}\log_2\frac{0}{4}\right)+\frac{5}{14}\times\left(-\frac{3}{5}\log_2\frac{3}{5}-\frac{2}{5}\log_2\frac{2}{5}\right)$$

ID3决策树

条件熵：

$$H(Y|X)=\sum_{x\in X}p(x)H(Y|X=x)$$

$$=\sum_{x\in X}p(x)\left[-\sum_{y\in Y}p(y|x)\log_2 p(y|x)\right]$$

$$=-\sum_{x\in X}\sum_{y\in Y}p(x,y)\log_2 p(y|x)$$

$$=-\sum_{x\in X}\sum_{y\in Y}p(x)p(y|x)\log_2 p(y|x)$$

- $H(D|A="age")=0.694$

$$H(D|A="age")=\frac{5}{14}\times\left(-\frac{2}{5}\log_2\frac{2}{5}-\frac{3}{5}\log_2\frac{3}{5}\right)$$

$$+\frac{4}{14}\times\left(-\frac{4}{4}\log_2\frac{4}{4}-\frac{0}{4}\log_2\frac{0}{4}\right)+\frac{5}{14}\times\left(-\frac{3}{5}\log_2\frac{3}{5}-\frac{2}{5}\log_2\frac{2}{5}\right)$$

把条件熵的公式拉过来比较一下。

可以看到Age属性有三种取值，
<=30, 31...40, 和>40

分别有5个，4个，5个实例。

<=30的有2人买了电脑，3人没买
31...40的有4人买了电脑，0人没买
>40的有3人买了电脑，2人没买

对照式子，搞清楚计算的过程。

ID3决策树

$$H(D)=0.940$$

$$g(D,A)=I(D;A)=H(D)-H(D\mid A)$$

$$g(D,A=\text{“}age\text{”})=0.246$$
$$g(D,A=\text{“}income\text{”})=0.029$$
$$g(D,A=\text{“}student\text{”})=0.151$$
$$g(D,A=\text{“}credit\_rating\text{”})=0.048$$

计算出所有属性的条件熵以后，下一步是**计算所有属性的信息增益。**

和条件熵相反，因为信息增益是用来衡量相关度，如果一个属性和类标签的相关度越高，说明该属性对于分类越有帮助。

所以选节点的条件是**条件熵越小约好，信息增益越大越好**！

“age” 这个属性的条件熵最小（等价于信息增益最大），因而首先被选出作为根节点
注意每次选出结点后，数据集根据结点属性值分为若干子集，新的子集要重新计算信息增益，**而非**采用上一次的信息增益继续做决策。

C4.5决策树

ID3方法虽好，但是有一个问题就是，如果数据集包含ID这样的属性，构建出的决策树就会完全无效，因为ID的信息增益永远是最大的。

试想一下，你有14个样本，结果构建了一颗14叉树，这样确实模型结果是百分百正确，
但是这样的树又有什么用呢？ 如果来了一个ID为15的样本，根本没有办法进行预测。

关于ID的信息增益，大家不妨自己返回前面几页算一算，这样就明白了。

于是ID3的发明人又提出了一种方法，就是C4.5，这个方法只是进行了小小改进，就解决了ID3的问题。不同于ID3采用信息增益进行划分，C4.5用的是**信息增益率**。

C4.5决策树

ID	Age	Income	student	Credit_rating	Bus_co
1	<=30	High	No	Fair	No
2	<=30	High	No	Excellent	No
3	31…40	High	No	Fair	Yes
4	>40	Medium	No	Fair	Yes
5	>40	Low	Yes	Fair	Yes
6	>40	Low	Yes	Excellent	No
7	31…40	Low	Yes	Excellent	Yes
8	<=30	Medium	No	Fair	No
9	<=30	Low	Yes	Fair	Yes
10	>40	Medium	Yes	Fair	Yes
11	<=30	Medium	Yes	Excellent	Yes
12	31…40	Medium	No	Excellent	Yes
13	31…40	High	Yes	Fair	Yes
14	>40	Medium	No	excellent	No

$$\mathrm{GainRatio}_A(D)=\mathrm{Gain}_A(D)/\mathrm{SplitInfo}_A(D)$$

$$SplitInfo_A(D) = -\sum_{j=1}^{v}\frac{|D_j|}{|D|}\times\log_2\left(\frac{|D_j|}{|D|}\right)$$

同样用例子来看看，分子是信息增益就不多说了，算法和前面一样，关键是分母。

公式很乱，又引入了SplitInfo这样一个词，但其实这个SplitInfo就是H(A)，也即属性自己的熵！ 只要这样记就可以了。 SplitInfo的算法就和前面提到的计算熵的方式一样。

$$SplitInfo_{A="income"}(D)$$
$$=-\frac{4}{14}\times\log_2\left(\frac{4}{14}\right)-\frac{6}{14}\times\log_2\left(\frac{6}{14}\right)-\frac{4}{14}\times\log_2\left(\frac{4}{14}\right)$$
$$=0.926$$

C4.5决策树

$$\mathrm{GainRatio}_A(D)=\mathrm{Gain}_A(D)/\mathrm{SplitInfo}_A(D)$$

$$SplitInfo_A(D) = -\sum_{j=1}^{v}\frac{|D_j|}{|D|}\times\log_2\left(\frac{|D_j|}{|D|}\right)$$

$$SplitInfo_{A="income"}(D)$$
$$= -\frac{4}{14}\times\log_2\left(\frac{4}{14}\right)-\frac{6}{14}\times\log_2\left(\frac{6}{14}\right)-\frac{4}{14}\times\log_2\left(\frac{4}{14}\right)$$
$$= 0.926$$

为什么用信息增益率能够避免ID被选为结点的情况呢？

可以这样理解，因为信息增益率=信息增益/属性本身的熵。

按照熵的定义，属性的不确定性越高，熵值就越大。

显然ID的不确定性是最高的，因为每一个ID都不一样且只出现一次，所以这样即使ID有一个较大的信息增益，也因为要除以自身的熵而使得它的信息增益率比不上其他属性了。

CART决策树

$$gini(D)=\sum_{j=1}^{n}p_j(1-p_j)=1-\sum_{j=1}^{n}p_j^2$$

$$gini_{split}(D)=\frac{N_1}{N}gini(D_1)+\frac{N_2}{N}gini(D_2)$$

最后介绍一种CART决策树，它和前面两种是不同的，它衡量的标准叫做 gini index 基尼指数。

根据$gini_{split}$(D)的值找根节点。

CART决策树

ID	Age	Income	student	Credit_rating	Bus_computer
1	<=30	High	No	Fair	No
2	<=30	High	No	Excellent	No
3	31…40	High	No	Fair	Yes
4	>40	Medium	No	Fair	Yes
5	>40	Low	Yes	Fair	Yes
6	>40	Low	Yes	Excellent	No
7	31…40	Low	Yes	Excellent	Yes
8	<=30	Medium	No	Fair	No
9	<=30	Low	Yes	Fair	Yes
10	>40	Medium	Yes	Fair	Yes
11	<=30	Medium	Yes	Excellent	Yes
12	31…40	Medium	No	Excellent	Yes
13	31…40	High	Yes	Fair	Yes
14	>40	Medium	No	excellent	No

$$gini(D) = \sum_{j=1}^{n} p_j(1-p_j) = 1 - \sum_{j=1}^{n} p_j^2$$

$$gini(D) = 1 - (\frac{9}{14})^2 - (\frac{5}{14})^2 = 0.459$$

先算出购买电脑事件的基尼指数。

CART决策树

ID	Age	Income	student	Credit_rating	Bus_computer
1	<=30	High	No	Fair	No
2	<=30	High	No	Excellent	No
3	31…40	High	No	Fair	Yes
4	>40	Medium	No	Fair	Yes
5	>40	Low	Yes	Fair	Yes
6	>40	Low	Yes	Excellent	No
7	31…40	Low	Yes	Excellent	Yes
8	<=30	Medium	No	Fair	No
9	<=30	Low	Yes	Fair	Yes
10	>40	Medium	Yes	Fair	Yes
11	<=30	Medium	Yes	Excellent	Yes
12	31…40	Medium	No	Excellent	Yes
13	31…40	High	Yes	Fair	Yes
14	>40	Medium	No	excellent	No

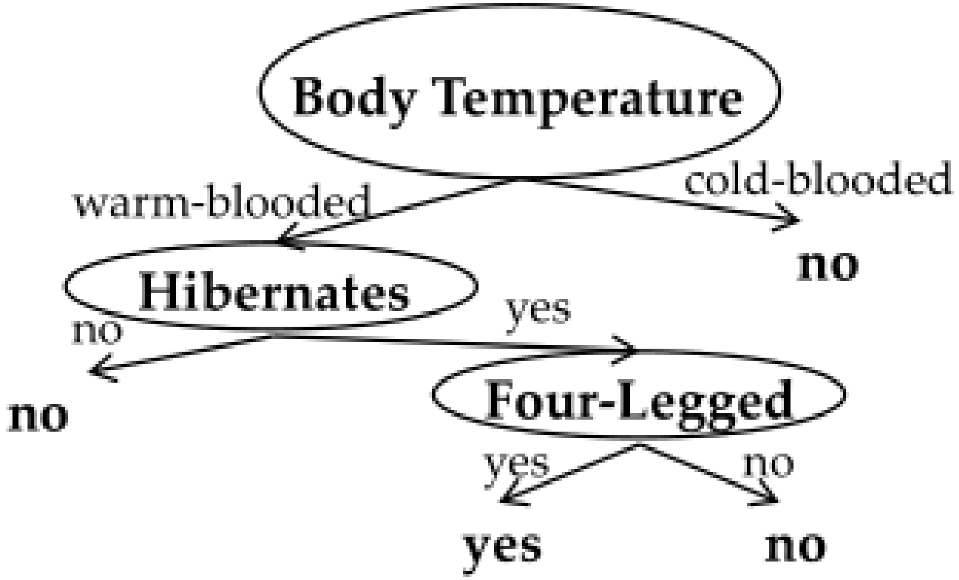
$$gini_{split}(D) = \frac{N_1}{N} gini(D_1) + \frac{N_2}{N} gini(D_2)$$

然后根据公式可以算出各个属性的$gini_{split}(D)$。
然后看看谁的最大，就用谁作为划分依据。

$$gini_{income\in\{\mathrm{medium,high}\}}(D) = \frac{10}{14} gini(D_1) + \frac{4}{14} gini(D_2)$$
$$= \frac{10}{14}\left(1-(\frac{6}{10})^2-(\frac{4}{10})^2\right)+\frac{4}{14}\left(1-(\frac{1}{4})^2-(\frac{3}{4})^2\right)$$
$$= 0.450 = gini_{income\in\{low\}}(D)$$

过拟合 Overfitting

Name	Body Temperature	Four-Legged	Hibernates	Mammals?
salamander	cold-blooded	yes	yes	no
guppy	cold-blooded	no	no	no
eagle	warm-blooded	no	no	no
poorwill	warm-blooded	no	yes	no
platypus	warm-blooded	yes	yes	yes



过拟合是构建决策树很容易产生的一个问题。
过拟合意思就是说在训练集上效果很好，
但是一到测试集就悲剧了。

这样的问题很常见，比方说商城做推荐系统的，离线训练的效果上线后得到的反馈可能很不乐观。

比方说上面这个数据集，由于恒温不冬眠的动物，只有鹰这一种，而鹰不是哺乳动物，所以左边构建出的决策树就认为这一分支的标签是no。

如果新加入一个样本：大象， 显然得到的答案就不正确了。

过拟合 Overfitting

解决过拟合有两种方法：

一是增大训练集的规模，使得它能包含足够多的差异性。 但这样可能不太现实，因为我们实际使用的训练集一般都是很大的，但差异性是很多的，不可能包括进去，并且在有监督学习中我们要给每个训练样本贴标签，花费是比较大的。所以这不是一个好的解决方案。

二是剪枝，把复杂的树变得简单一些，这样就能避免一些极端情况，这是我们相对能容易点完成的解决方案。

过拟合 Overfitting

Name	Body Temperature	Gives Birth	Four-Legged	Hibernates	Mammals?
porcupine	warm-blooded	yes	yes	yes	yes
cat	warm-blooded	yes	yes	no	yes
bat	warm-blooded	yes	no	yes	no
whale	warm-blooded	yes	no	no	no
salamander	cold-blooded	no	yes	yes	no
komodo dragon	cold-blooded	no	yes	no	no
python	cold-blooded	no	no	yes	no
salmon	cold-blooded	no	no	no	no
eagle	warm-blooded	no	no	no	no
guppy	cold-blooded	yes	no	no	no

Training set

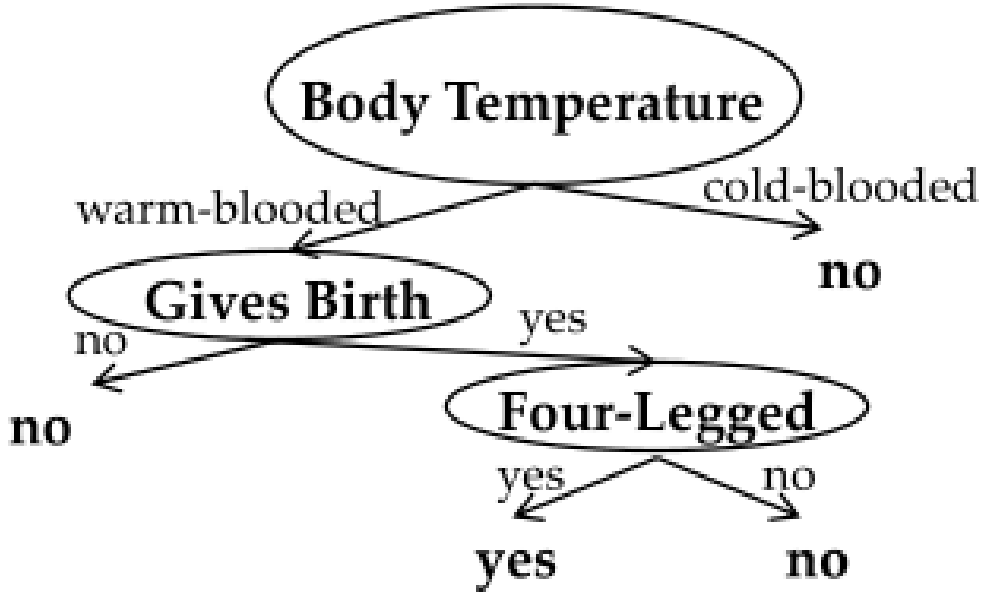
Name	Body Temperature	Gives Birth	Four-Legged	Hibernates	Mammals?
human	warm-blooded	yes	no	no	yes
pigeon	warm-blooded	no	no	no	no
elephant	warm-blooded	yes	yes	no	yes
leopard shark	cold-blooded	yes	no	no	no
turtle	cold-blooded	no	yes	no	no
penguin	warm-blooded	no	no	no	no
eel	cold-blooded	no	no	no	no
dolphin	warm-blooded	yes	no	no	yes
spiny anteater	warm-blooded	no	yes	yes	yes
gila monster	cold-blooded	no	yes	yes	no

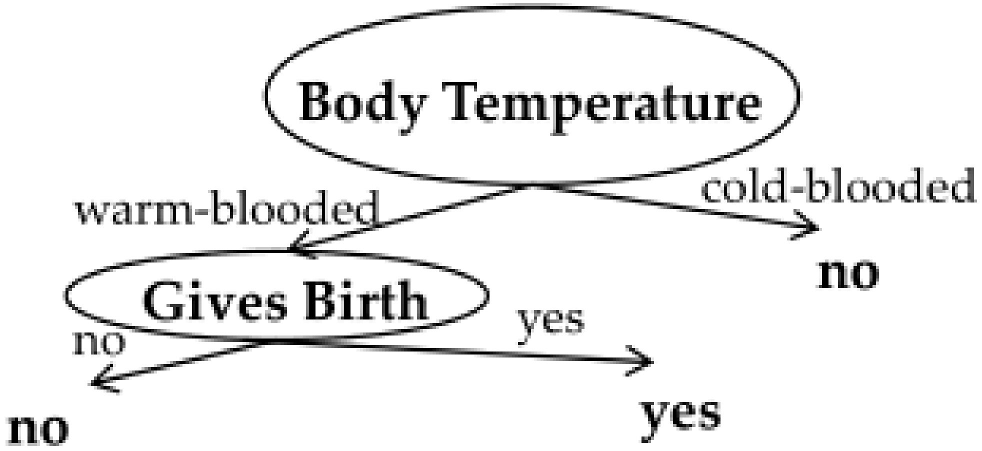
Testing set

不妨用左边的数据集验证一下剪枝的效果。

过拟合 Overfitting

尽管训练集的错误率提高了，但剪枝后测试集的错误率减少了很多，所以剪枝后效果更好。





The training error is 0
The error rate on the testing set is 30%

The training error rate is 20%
The error rate on the testing set is 10%

过拟合 Overfitting

那么问题来了，说道剪枝，到底要怎么剪呢？

我们在使用决策树时，一般都是希望目标函数(也有说损失函数/误差函数/价值函数的)最小。

传统地有： e = 错误个数 / 总个数

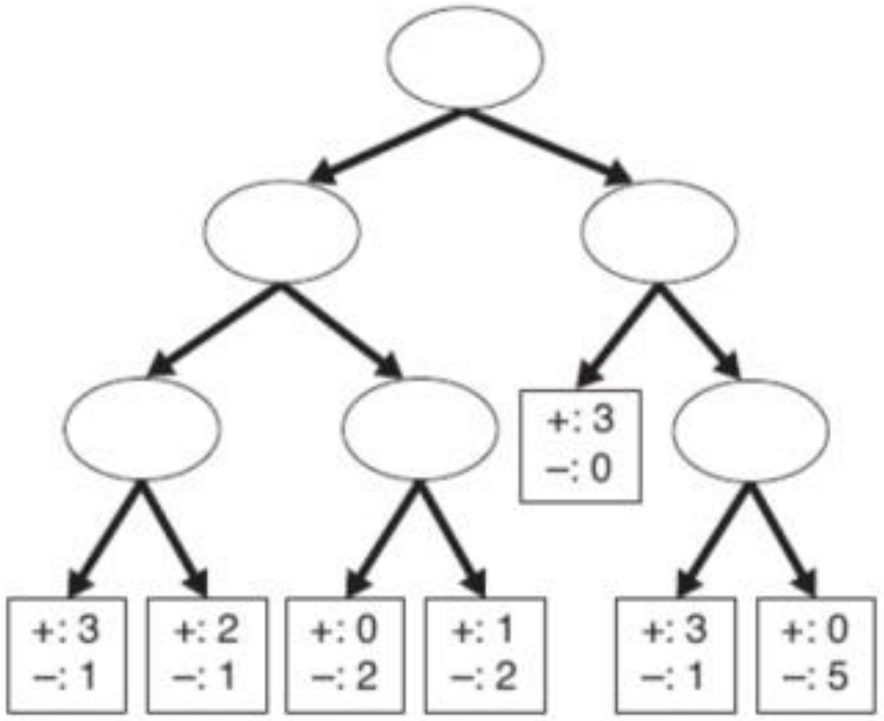
但现在我们希望找到一种方法能够**让目标函数在剪枝后有可能比剪枝前要小。**

这是一个还在不断研究的课题，这里演示一种简单的方法，引入**惩罚项α**

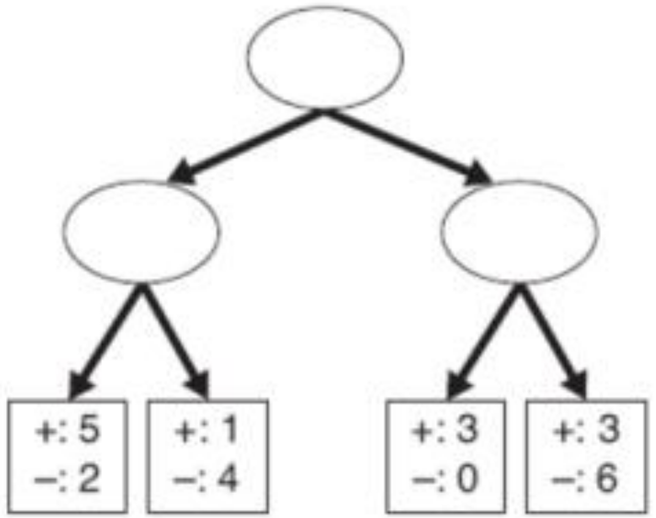
剪枝方法： e = （错误个数 + α*叶子数）/ 总个数

这种方法的直观理解是，每多一片叶子就多α个错误个数。

过拟合 Overfitting



Decision Tree, T_L



Decision Tree, T_R

直接看例子，假设惩罚项α=0.5

左边： e = （1+1+1+1 + 7*0.5）/ 24 = 0.3125

右边：e = （2+1+3 + 4*0.5）/ 24 = 0.3333

这样一算，左边未剪枝效果较好，所以就不剪了。

直观地，惩罚项为0.5意味着如果分裂两个子结点后能够多出一个正确的样本就进行分裂。

决策树基本的内容就是以上这些，但是这些仅仅是基础。

决策树模型虽然简单，但是也是比较常用的，在各大竞赛以及一些顶尖会议的论文中不乏出现。像RF，GBDT这些更高级的方法大家不妨再作了解。

Written by: Ricardo
2016.4.3