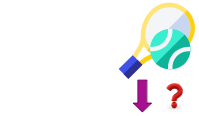
决策树

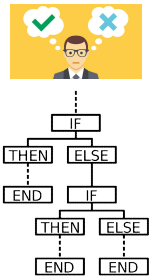
刘杰
人工智能学院
南开大学 Nankai University

目录

决策树模型

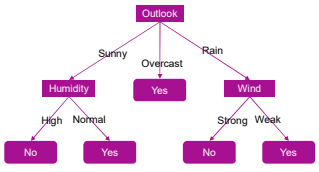
我们如何做“决策”？
If Then规则





Play Tennis?

- 分类决策树模型是一种描述对实例进行分类的树形结构。
- 决策树由结点(node) 和有向边(directed edge)组成。
- 结点有两种类型: 内部结点(internal node)和叶结点(leaf node)。
- 内部结点表示一个特征或属性，叶结点表示一个类。

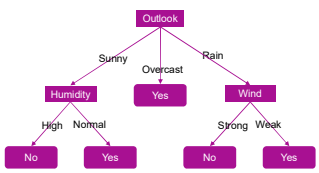


Play Tennis?

如图的决策可以看做一个IF-THEN规则的集合

根结点到叶结点的每一条路径构成一条规则：

- 路径上的内部结点对应着规则的条件
- 叶结点的类对应着规则的结论

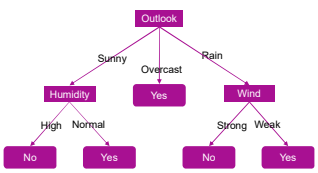


决策树

决策树是一种典型的分类方法

- 首先对数据进行处理，利用归纳算法生成可读的规则和决策树，
- 然后使用决策对新数据进行分析。

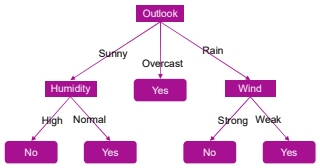
本质上决策树是通过一系列规则对数据进行分类的过程。



决策树

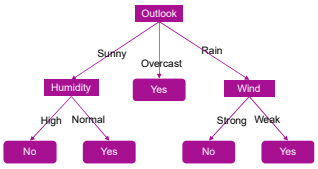
决策树的优点

- 1、推理过程容易理解，决策推理过程可以表示成为If --Then形式；
- 2、推理过程完全依赖于属性变量的取值特点；
- 3、可自动忽略目标变量没有贡献的属性变量，也为判断属性变量的重要性，减少变量的数目提供参考。



决策树

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

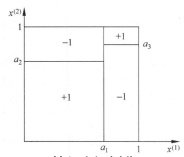


决策树与条件概率分布

决策树还表示给定特征条件下类的条件概率分布: $P(Y|X)$

这一条件概率分布定义在特征空间的一个划分（partition）上。

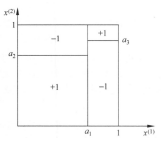
将特征空间划分为互不相交的单元(cell)或区域(region)，并在每个单元定义一个类的概率分布就构成了一个条件概率分布。



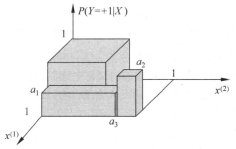
特征空间划分

决策树与条件概率分布

假设X为表示特征的随机变量， Y为表示类的随机变量，那么这个条件概率分布可以表示为$P(Y|X)$。 X取值于给定划分下单元的集合，Y取值于类的集合。各叶结点(单元)上的条件概率往往偏向某一个类，即属于某一类的概率较大。决策树分类时将该结点的实例分到条件概率大的那一类去。



特征空间划分

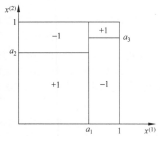


条件概率分布

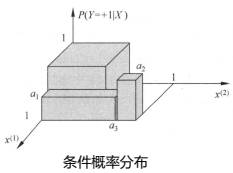
决策树与条件概率分布

决策树的一条路径对应于划分中的一个单元。

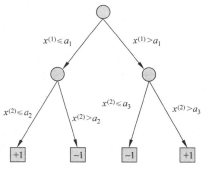
决策树所表示的条件概率分布由各个单元给定条件下类的条件概率分布组成。



特征空间划分



条件概率分布



决策树学习算法

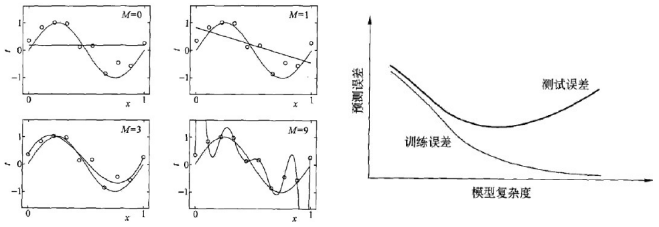
假设给定训练数据集

$$D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$$

其中，$x_i = (x_i^{(1)}, x_i^{(2)}, \cdots, x_i^{(n)})^{\mathrm{T}}$ 为输入实例（特征向量），n 为特征个数，$y_i \in \{1, 2, \cdots, K\}$ 为类标记，$i = 1, 2, \cdots, N$，N 为样本容量。决策树学习的目标是根据给定的训练数据集构建一个决策树模型，使它能够对实例进行正确的分类。

决策树学习算法

- 决策树学习本质上是从训练数据集中归纳出一组分类规则，与训练数据集不相矛盾的决策树。
- 能对训练数据进行正确分类的决策树可能有多个，也可能 一个也没有.我们需要的是一个与训练数据矛盾较小的决策树，同时具有很好的 泛化能力.
- 决策树学习是由训练数据集估计条件概率模型.基于特征空间划分的类的条件概率模型有无穷多个.
- 我们选择的条件概率模型应该不仅对训练数据有很好的拟合，而且对未知数据有很好的预测.



决策树学习算法

决策树学习策略：以损失函数为目标函数的最小化。

特征选择：递归地选择最优特征，并根据该特征对训练数据进行分割，使得对各个子数据集有一个最好的分类

决策树生成：与特征选择过程相对应，选择的特征依次形成决策树的结点，直至所有训练数据子集都被基本正确分类，或无合适特征可选。

决策树的剪枝：为避免过拟合，对已生成的树自下而上进行剪枝，将树变得更简单，从而使它具有更好的泛化能力。

特征选择

特征选择在于选取对训练数据具有分类能力的特征

特征选择是决定用哪个特征来划分特征空间。

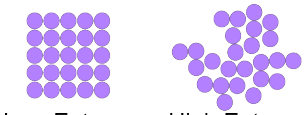
特征选择的准则是信息增益或信息增益比。

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

信息增益

- Shannon 1948年提出的信息论理论：
- 熵(entropy)：信息量大小的度量，即表示随机变量不确定性的度量。

$$I(a_i) = p(a_i)\log_2 \frac{1}{p(a_i)}$$

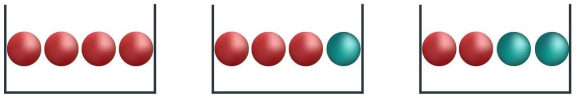


Low Entropy　　High Entropy

$$I(a_1, a_2, ..., a_n) = \sum_{i=1}^{n} I(a_i) = \sum_{i=1}^{n} p(a_i)\log_2 \frac{1}{p(a_i)}$$

信息增益

- Shannon 1948年提出的信息论理论：
- 熵(entropy)：信息量大小的度量，即表示随机变量不确定性的度量。



High Knowledge
Low Entropy

Medium Knowledge
Medium Entropy

Low Knowledge
High Entropy

信息增益

•设X是一个取有限个值的离散随机变量，其概率分布为：

$$P(X=x_i)=p_i, i=1,2,...,n$$

•信息量：随机变量x_i的信息量$I(x_i)$可如下度量：

$$I(x_i)=log\frac{1}{p(x_i)}$$

•其中$p(x_i)$表示事件x_i发生的概率。

信息增益

•熵为各事件信息量的“数学期望”：

$$H(X)=E[I(x_i)]=-\sum_{i=1}^{n}p(x_i)\log p(x_i)$$

对数以2为底或以e为底(自然对数)，这时熵的单位分别称作比特(bit)或纳特(nat),熵只依赖于X的分布，与X的取值无关。

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

信息增益

•熵的理论解释：

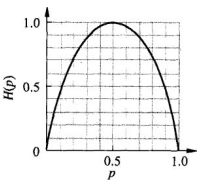
熵越大，随机变量的不确定性越大：

$$0\le H(p)\le \log n$$

当X为0,1分布时：

$$P(X=1)=p, P(X=0)=1-p, 0\le p\le 1$$

熵：$H(p)=-p\log_2 p-(1-p)\log_2(1-p)$



信息增益

设有随机变量(X,Y),其联合概率分布为：

$$P(X=x_i,Y=y_i)=p_{ij}, i=1,2,...,n; j=1,2,...,m$$

条件熵H(Y|X)：表示在已知随机变量X的条件下随机变量Y的不确定性，定义为X给定条件下Y的条件概率分布的熵对X的数学期望：

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

当熵和条件熵中的概率由数据估计(特别是极大似然估计)得到时，所对应的熵与条件熵分别称为经验熵(empirical entropy)和经验条件熵(empirical conditional entropy)。

信息增益

•定义 (信息增益):特征A对训练数据集D的信息增益，$g(D,A)$, 定义为集合D的经验熵$H(D)$与特征A给定条件下D的经验条件熵$H(D|A)$之差，即

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

•(Information gain)表示得知特征X的信息而使得类Y的信息的不确定性减少的程度.

•一般地，熵$H(Y)$与条件熵$H(Y|X)$之差称为互信息(mutual information)

•决策树学习中的信息增益等价于训练数据集中类与特征的互信息.

信息增益的算法

•设训练数据集为D

•$|D|$表示其样本容量，即样本个数

•设有K个类C_k, $k=1,2,...K$,

•$|C_k|$为属于类C_k的样本个数

•特征A有n个不同的取值$\{a_1,a_2...an\}$，根据特征A的取值将D划分为n个子集$D_1...D_n$

•$|Di|$为 D_i的样本个数

•记子集D_i中属于类C_k的样本集合为D_{ik}

•$|D_{ik}|$为D_{ik}的样本个数

信息增益的算法

- 输入：训练数据集D和特征A；
- 输出：特征A对训练数据集D的信息增益g(D,A)

1、计算数据集D的经验熵H(D)

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

2、计算特征A对数据集D的经验条件熵H(D|A)

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

3、计算信息增益

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

信息增益比

- 以信息增益作为划分训练数据集的特征，存在偏向于选择取值较多的特征的问题
- 使用信息增益比可以对这一问题进行校正
- 定义（信息增益比） 特征A对训练数据集D的信息增益比定义为信息增益与训练数据集D关于特征A的值的熵之比

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

$$H_A(D)=-\sum_{i=1}^{n}\frac{|D_i|}{|D|}\log_2\frac{|D_i|}{|D|}$$，n是特征A取值的个数

决策树ID3算法

输入：训练数据集 D，特征集 A 阈值 ε；

输出：决策树 T。

（1）若 D 中所有实例属于同一类 C_k，则 T 为单结点树，并将类 C_k 作为该结点的类标记，返回 T；

（2）若 $A=\varnothing$，则 T 为单结点树，并将 D 中实例数最大的类 C_k 作为该结点的类标记，返回 T；

（3）否则， **根据公式** 计算 A 中各特征对 D 的信息增益，选择信息增益最大的特征 A_g；

（4）如果 A_g 的信息增益小于阈值 ε，则置 T 为单结点树，并将 D 中实例数最大的类 C_k 作为该结点的类标记，返回 T；

（5）否则，对 A_g 的每一可能值 a_i，依 $A_g=a_i$ 将 D 分割为若干非空子集 D_i，将 D_i 中实例数最大的类作为标记，构建子结点，由结点及其子结点构成树 T，返回 T；

（6）对第 i 个子结点，以 D_i 为训练集，以 $A-\{A_g\}$ 为特征集，递归地调用步 (1)～步 (5)，得到子树 T_i，返回 T_i。 ■

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

第1步计算决策属性的熵

决策属性“PlayTennis”。
该属性分两类：Yes/No

$|C_1|(Yes)=9$
$|C_2|(No)=5$
$|D|=|C_1|+|C_2|=14$

$P_1=9/14=0.643$
$P_2=5/14=0.357$

$$H(D)=-P_1\mathrm{Log}_2P_1-P_2\mathrm{Log}_2P_2$$
$$=-(P_1\mathrm{Log}_2P_1+P_2\mathrm{Log}_2P_2)$$
$$=0.9401$$

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

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

第2步计算条件属性的熵

条件属性共有4个：
Outlook、Temperature、Humidity、Wind。
分别计算不同属性的信息增益。

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

第2步 计算Outlook中各个属性的条件熵

Outlook共分三个组：
Sunny、Overcast、Rain

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

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

第2-1步计算Outlook中Sunny的熵

Outlook共分三个组：
Sunny、Overcast、Rain
其中当Outlook为Sunny时，Yes与No比例为2/3

$|D_{11}|(Yes) = 2$
$|D_{12}|(No) = 3$
$|D_1| = 5$

$P_1 = 2/5 = 0.4$
$P_2 = 3/5 = 0.6$

$H(D_1) = -P_1\mathrm{Log}_2P_1 - P_2\mathrm{Log}_2P_2$
$= -(P_1\mathrm{Log}_2P_1 + P_2\mathrm{Log}_2P_2)$
$= 0.971$

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

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

第2-2步计算Outlook中Overcast的熵

Outlook共分三个组：
Sunny、Overcast、Rain
其中当Outlook为Overcast时，Yes与No比例为4/0

$|D_{21}|(Yes) = 4$
$|D_{22}|(No) = 0$
$|D_2| = 4$

$P_1 = 1$
$P_2 = 0$

$H(D_2) = -P_1\mathrm{Log}_2P_1 - P_2\mathrm{Log}_2P_2$
$= -(P_1\mathrm{Log}_2P_1 + P_2\mathrm{Log}_2P_2)$
$= 0$

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

第2-3步计算计算Outlook中Rain的熵

Outlook共分三个组：
Sunny、Overcast、Rain
其中当Outlook为Rain时，Yes与No比例为3/2

$|D_{31}|(Yes) = 3$
$|D_{32}|(No) = 2$
$|D_3| = 5$

$P_1 = 0.6$
$P_2 = 0.4$

$H(D_3) = -P_1\mathrm{Log}_2P_1 - P_2\mathrm{Log}_2P_2$
$= -(P_1\mathrm{Log}_2P_1 + P_2\mathrm{Log}_2P_2)$
$= 0.971$

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

第2-4步计算Outlook的熵

Outlook共分三个组：
Sunny、Overcast、Rain
所占比例
Sunny: $5/14 = 0.357$
Overcast: $4/14 = 0.286$
Rain: $5/14 = 0.357$

计算Outlook的平均信息期望
$E(Outlook) = 0.357 * 0.971 + 0.286 * 0 + 0.357 * 0.971 = 0.6932$
*Outlook*信息增益 $= 0.9401 - 0.6932 = 0.2469$ (1)

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

第3步 计算Temperature的熵

收入共分三个组：
Hot、Mild、Cool
$E(Temperature) = 0.9108$
*Temperature*信息增益 $= 0.9401 - 0.9108 = 0.0293$ (2)

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

第4步计算Humidity的熵

Humidity共分二个组：
High、Normal
$E(Humidity) = 0.7885$
*Humidity*信息增益 $= 0.9401 - 0.7885 = 0.1516$ (3)

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

第5步计算Wind的熵

Wind共分二个组：
Weak、Strong
$E(Wind) = 0.892$
$Wind$信息增益 $= 0.9401 - 0.892$
$= 0.0481$ (4)

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

第6步计算选择节点

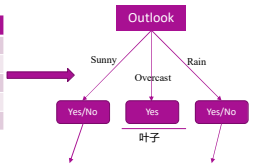
$Outlook$信息增益 $= 0.9401 - 0.6932$
$= 0.2469$ (1)

$Temperature$信息增益 $= 0.9401 - 0.9108$
$= 0.0293$ (2)

$Humidity$信息增益 $= 0.9401 - 0.7885$
$= 0.1516$ (3)

$Wind$信息增益 $= 0.9401 - 0.892$
$= 0.0481$ (4)

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes



Day	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

Sunny中Yes与No比例为2/3

$|C_1|(Yes) = 2$
$|C_2|(No) = 3$
$|D| = 5$

$P_1 = 2/5 = 0.4$
$P_2 = 3/5 = 0.6$

$H(DS_{\text{unny}}) = -P_1Log_2P_1 - P_2Log_2P_2$
$= -(P_1Log2P1 + P2Log2P2)$
$= 0.9710$

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

如果选择Temperature作为节点
分为Hot、Mild、Cool
$H(D_1) = 0$
比例: 2/5
$H(D_2) = 1$
比例: 2/5
$H(D_3) = 0$
比例: 1/5

平均信息期望(加权总和):
$E(H(D_{Temperature|\text{Sunny}})) = 0.4 * 0 + 0.4 * 1 + 0.2 * 0 = 0.4$
$Gain(Temperature) = H(D_{Sunny}) - E(H(D_{Temperature|\text{Sunny}})) = 0.9710 - 0.4 = 0.5710$
注意

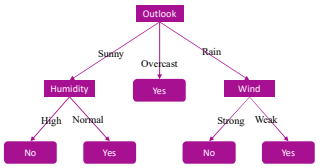
Day	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

如果选择Humidity作为节点
分为High、Normal

$H(D_1) = 0$
比例: 3/5
$H(D_2) = 0$
比例: 2/5

平均信息期望(加权总和):
$E(H(D_{Humidity|\text{Sunny}})) = 0$
$Gain(Humidity) = H(D_{Sunny}) - E(H(D_{Humidity|\text{Sunny}})) = 0.9710 - 0 = 0.9710$
由于信息增益已然达到最极值，所以选择Humidity。
其他结点的增益也按照上述的方法计算。

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No



43

决策树ID3算法-小结

ID3算法的基本思想是，以信息熵为度量，用于决策树节点的属性选择

每次优先选取信息量最多的属性，亦即能使熵值变为最小的属性，以构造一棵熵值下降最快的决策树，到叶子节点处的熵值为0。

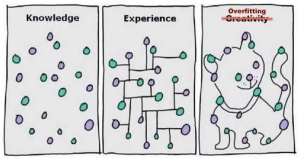
此时，每个叶子节点对应的实例集中的实例属于同一类。

44

决策树面临的问题

过度拟合

- 决策树算法增长树的每一个分支的深度，直到恰好能对训练样例比较完美地分类。实际应用中，当数据中有噪声或训练样例的数量太少以至于不能产生目标函数的有代表性的采样时，该策略可能会遇到困难。
- 在以上情况发生时，这个简单的算法产生的树会过拟合训练样例(过拟合：Over Fitting)。



45

决策树面临的问题

- 理想的决策树：
  (1)叶子结点数最少；
  (2)叶子结点深度最小；
  (3)叶子结点数最少且叶子结点深度最小。

46

决策树的剪枝

通过极小化决策树整体的损失函数或代价函数来实现。

设树T的叶结点个数为$|T|$，t是树T的叶结点，该叶结点有N_t个样本点，其中k类的样本点有N_{tk}个，$k = 1,2..K$，

$H_t(T)$为叶结点t上的经验熵，$\alpha \geq 0$为参数，损失函数：

$$C_\alpha(T) = \sum_{t=1}^{|T|} N_t H_t(T) + \alpha|T|$$

47

决策树的剪枝

通过极小化决策树整体的损失函数或代价函数来实现。

设树T的叶结点个数为$|T|$，t是树T的叶结点，该叶结点有N_t个样本点，其中k类的样本点有N_{tk}个，$k = 1,2..K$，

$H_t(T)$为叶结点t上的经验熵，$\alpha \geq 0$为参数，损失函数：

$$C_\alpha(T) = \sum_{t=1}^{|T|} N_t H_t(T) + \alpha|T|$$

预测误差损失　　模型复杂度损失

48

决策树的剪枝

通过极小化决策树整体的损失函数或代价函数来实现。

设树T的叶结点个数为$|T|$，t是树T的叶结点，该叶结点有N_t个样本点，其中k类的样本点有N_{tk}个，$k = 1,2..K$，

$H_t(T)$为叶结点t上的经验熵，$\alpha \geq 0$为参数，损失函数：

$$C_\alpha(T) = \sum_{t=1}^{|T|} N_t H_t(T) + \alpha|T|$$

经验熵：$$H_t(T) = -\sum_k \frac{N_{tk}}{N_t} \log \frac{N_{tk}}{N_t}$$

原式第一项：$$C(T) = \sum_{t=1}^{|T|} N_t H_t(T) = -\sum_{t=1}^{|T|} \sum_{k=1}^{K} N_{tk} \log \frac{N_{tk}}{N_t}$$

则：$$C_\alpha(T) = C(T) + \alpha|T|$$

49

决策树的剪枝

•树的剪枝算法：

输入：生成算法产生的整个树 T，参数 α;

输出：修剪后的子树 T_α。

(1)计算每个结点的经验熵。

(2)递归地从树的叶结点向上回缩。

•设一组叶结点回缩到其父结点之后与之前的损失函数分别为：$C_\alpha(T_A)$与 $C_\alpha(T_B)$

如果：$C_\alpha(T_A) \leq C_\alpha(T_B$则进行剪枝

(3)返回(2)，直至不能继续为止，得到损失函数最小的子树 T_α。

50

CART树

CART：分类与回归树，Classification And Regression Tree

算法由两部分组成：

- 决策树生成
- 决策树剪枝

回归树：平方误差最小化

分类树：Gini Index

51

CART算法

- CART算法采用一种二分递归分割的技术，算法总是将当前样本集分割为两个子样本集
- 生成的决策树的每个非叶结点都只有两个分枝，因此CART算法生成的决策树是结构简洁的二叉树。
- CART算法适用于样本特征的取值为是或非的场景。
- 剪枝过程特别重要，所以在最优决策树生成过程中占有重要地位。有研究表明，剪枝过程的重要性要比树生成过程更为重要

52

CART的生成

回归树的生成

设Y是连续变量，给定训练数据集：

$$D = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$$

假设已将输入空间划分为M个单元$R_1, R_2..Rm$,并且每个单元R_m上有一个固定的输出C_m，回归树表示为：

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m)$$

平方误差来表示预测误差，用平方误差最小准则求解每个单元上的最优输出值

$$\sum_{x_i \in R_m} (y_i - f(x_i))^2$$

R_m上的C_m的最优值：$$\hat{c}_m = ave(y_i | x_i \in R_m)$$

53

CART的生成

输入空间的划分

启发式：选择第j维变量$x^{(j)}$和它取的值s，作为切分变量和切分点，定义两个区域：

$$R_1(j,s) = \{x | x^{(j)} \leq s\}$$和 $$R_2(j,s) = \{x | x^{(j)} > s\}$$

然后寻找最优切分变量和切分点：

$$\min_{j,s}[\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2]$$

且：$\hat{c}_1 = ave(y_i | x_i \in R_1(j,s))$ 和 $\hat{c}_2 = ave(y_i | x_i \in R_2(j,s))$

再对两个区域重复上述划分，直到满足停止条件。

54

CART的生成

最小二乘回归树生成算法

输入：训练数据集D;

输出：回归树$f(x)$。

在训练数据集所在的输入空间中，递归地将每个区域划分成两个子区域并决定每个子区域上地输出值，构建二叉决策树；

(1)选择最优切分变量j与切分点s，求解

$$\min_{j,s}[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2]$$

遍历变量j，对固定地切分变量j扫描切分点s，选择使上式达到最小值地对(j,s)。

CART的生成

•最小二乘回归树生成算法

(2)用选定地对(j,s)划分区域并决定相应地输出值：

$$R_1(j,s)=\{x|x^{(j)}\le s\},R_2(j,s)=\{x^{(j)}>s\}$$

$$\hat{c}_m=\frac{1}{N_m}\sum_{x_i\in R_m(j,s)}y_i,x\in R_m,m=1,2$$

(3)继续对两个子区域调用(1),(2),直至满足停止条件。

(4)将输出空间划分为M个区域$R_1,R_2,...,R_M$生成决策树；

$$f(x)=\sum_{}^{M}\hat{c}_mI(x\in R_m)$$

CART的生成

•分类树的生成：

•基尼指数

•分类问题中，假设有k个类，样本点属于k的概率P_k，则概率分布的基尼指数：

$$Gini(p)=\sum_{k=1}^{K}p_k(1-p_k)=1-\sum_{k=1}^{K}p_k^2$$

•二分类问题：$Gini(p)=2p(1-p)$

•对给定的样本集合D，基尼指数

$$Gini(D)=1-\sum_{k=1}^{K}(\frac{|C_k|}{|D|})^2$$

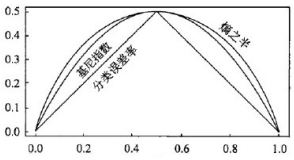
CART的生成

如果样本集合D根据特征A是否为a被分割成D_1和D_2，即

$$D_1=\{(x,y)\in D|A(x)=a\},D_2=D-D_1$$

则在特征A的条件下，集合D的基尼指数：

$$Gini(D,A)=\frac{|D_1|}{|D|}Gini(D_1)+\frac{|D_2|}{|D|}Gini(D_2)$$



CART的生成

•CART分类树生成算法

输入：训练数据集D

输出：CART分类树

从根节点开始，递归对每个结点操作

1、设结点数据集为D，对每个特征A，对其每个值a，根据样本点对$A=a$的测试为是或否，将D分为D_1，D_2，计算$A=a$的基尼指数

2、在所有的特征A以及所有可能的切分点a中，选择基尼指数最小的特征和切分点，将数据集分配到两个子结点中。

3、对两个子结点递归调用1，2步骤

4、生成CART分类树

CART树剪枝

•CART剪枝

两步

1、从生成算法产生的决策树T_0底端开始不断剪枝，直到T_0的根结点，形成子树序列$\{T_0,T_1..Tn\}$,

2、通过交叉验证法在独立的验证数据集上对子树序列进行测试，从中选择最优子树

CART树剪枝

•CART剪枝

1、剪枝，形成子树序列

剪枝过程中，计算子树的损失函数：

$$C_\alpha(T) = C(T) + \alpha|T|$$

对固定的α一定存在损失函数最小的子树，表示为T_α，

当α变大时，最优子树T_α偏小，

$\alpha = 0$时，整体树最优，α趋近无穷大，单结点最优

将a从小增大，$0 = \alpha_0 < \alpha_1 < ... < \alpha_n < +\infty$

最优子树序列$\{T_0, T_1, ..., T_n\}$

CART算法剪枝

•CART剪枝

1、剪枝，形成子树序列

具体：从整树T_0开始剪枝，以其中任意内部结点t为单结点树的损失函数：

$$C_\alpha(t) = C(t) + \alpha$$

以t为根结点的子树T_t的损失函数：

$$C_\alpha(T_t) = C(T_t) + \alpha|T_t|$$

当α = 0及α很小时，$C_\alpha(T_t) < C_\alpha(t)$

不断增大α，当 $C_\alpha(T_t) = C_\alpha(t)$ $\alpha = \dfrac{C(t) - C(T_t)}{|T_t| - 1}$

T_t与t有相同损失函数值，但t结点更少，所以剪枝T_t。

CART的生成

•CART剪枝

1、剪枝，形成子树序列

对整树T_0中每个内部结点t，计算：

$$g(t) = \frac{C(t) - C(T_t)}{|T_t| - 1}$$

在T_0中剪去$g(t)$最小的T_t，将得到的子树作为T_1，同时将最小的$g(t)$设为α_1，T_1为区间$[\alpha_1, \alpha_2)$的最优子树

如此剪枝下去，直到根节点，不断增加α的值，产生新的区间。

CART的生成

•CART剪枝

2、在剪枝得到的子树序列$\{T_0, T_1..Tn\}$中通过交叉验证选取最优子树T_α

利用独立的验证数据集，测试子树序列中各子树的平方误差或基尼指数，最小的决策树就是最优决策树。

Homework

习题5.2

PlayTennis用CART的流程做一遍

END