机器学习与数据挖掘

Machine Learning & Data Mining

权小军 教授

中山大学数据科学与计算机学院

quanxj3@mail.sysu.edu.cn

Preface

Kaggle CEO gives you the Secret to winning Kaggle competitions

Source: https://www.kdnuggets.com/2016/01/anthony-goldbloom-secret-winning-kaggle-competitions.html

Secret to winning Kaggle competitions

According to Anthony, in the history of Kaggle competitions, there are only two Machine Learning approaches that win competitions: Handcrafted & Neural Networks.

- **Handcrafted feature engineering**
  - This approach works best if you already have an intuition as to what's in the data.
  - First, a competitor will take the data and plot histograms and such to explore what's in it.
  - Then they'll spend a lot time generating features and testing which ones really do correlate with the target variable.

Secret to winning Kaggle competitions

According to Anthony, in the history of Kaggle competitions, there are only two Machine Learning approaches that win competitions: Handcrafted & Neural Networks.

- **Handcrafted feature engineering**
  - The way was to test lots and lots and lots of hypotheses. The vast majority of them didn't work out, but the one that did won them the competition.
  - It has almost always been ensembles of **decision trees** that have won competitions.
  - Used to be random forest that was the big winner, but over the last six months XGboost has cropped up.

Secret to winning Kaggle competitions

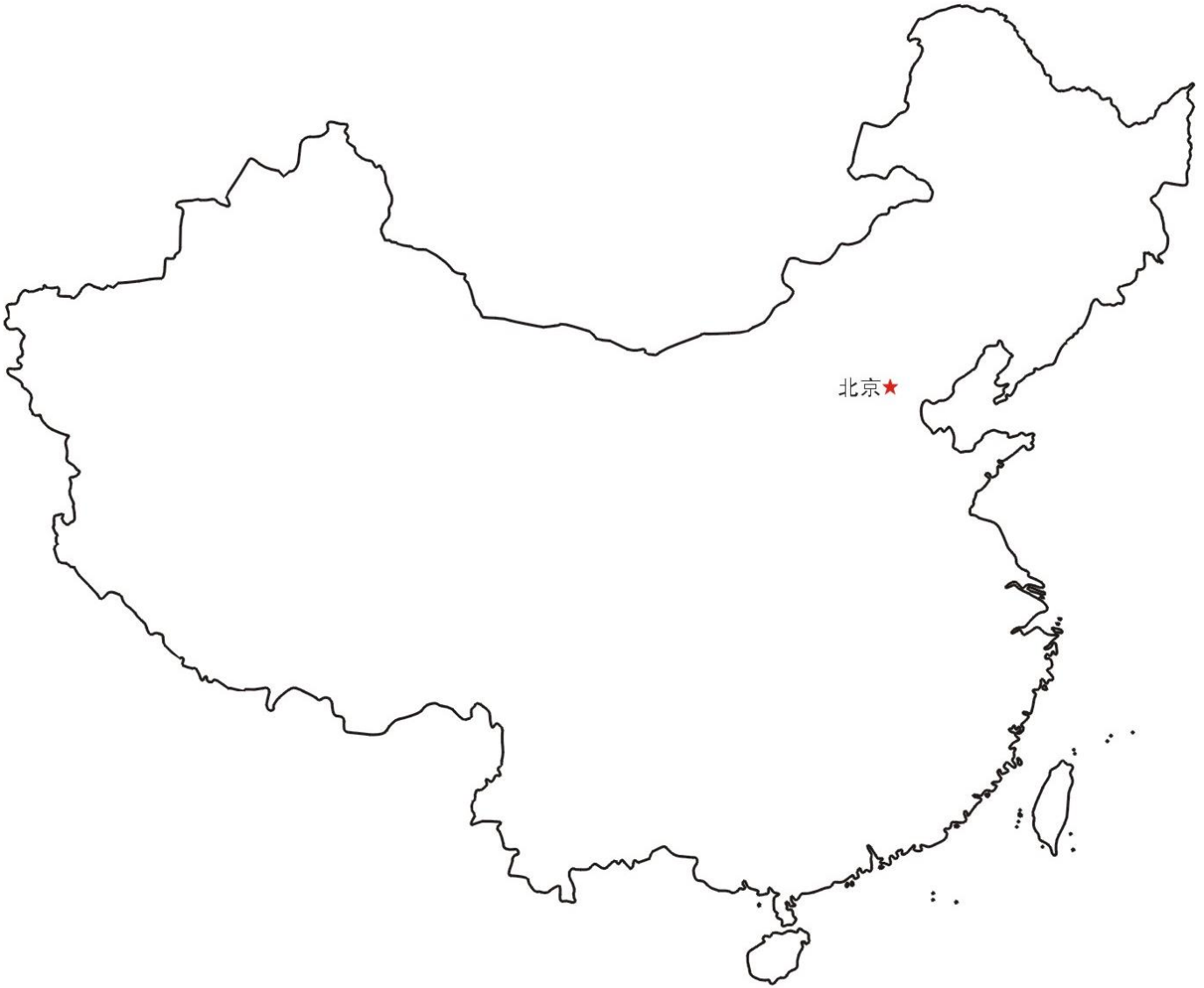
According to Anthony, in the history of Kaggle competitions, there are only two Machine Learning approaches that win competitions: Handcrafted & Neural Networks.

- **Neural Networks and Deep Learning**
  - For any dataset that contains images or speech problems, deep learning is the way to go.
  - The people who are winning these competitions (the ones without well-structured data) are spending almost none of their time doing feature engineering. **Instead, they spend their time constructing neural networks.**

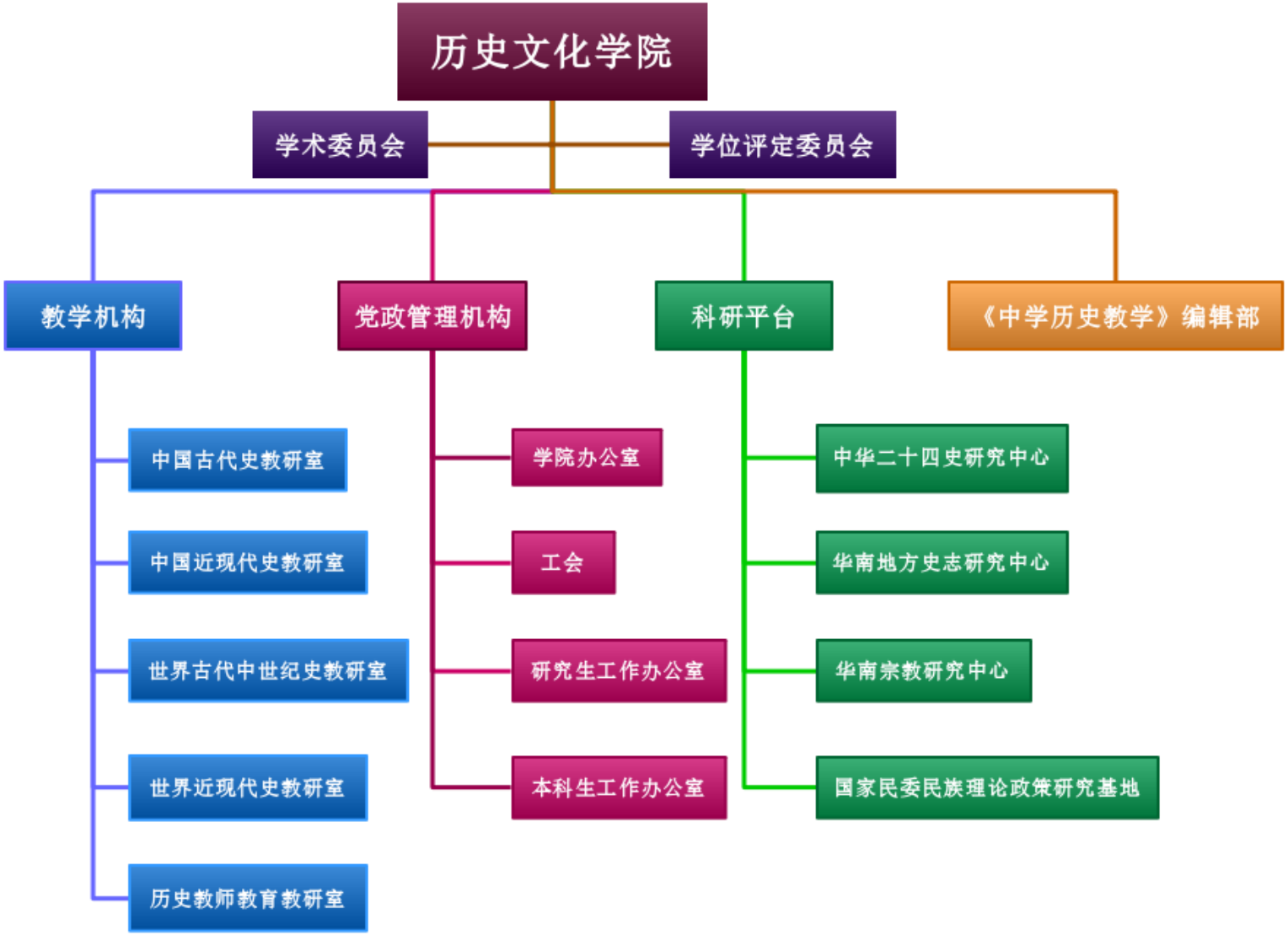
Secret to winning Kaggle competitions

According to Anthony, in the history of Kaggle competitions, there are only two Machine Learning approaches that win competitions: Handcrafted & Neural Networks.

- **Which method should you use?**
  - For most competitions it's pretty obvious. If you have lots of structured data, the handcrafted approach is your best bet;
  - If you have unusual or unstructured data your efforts are best spent on neural networks.

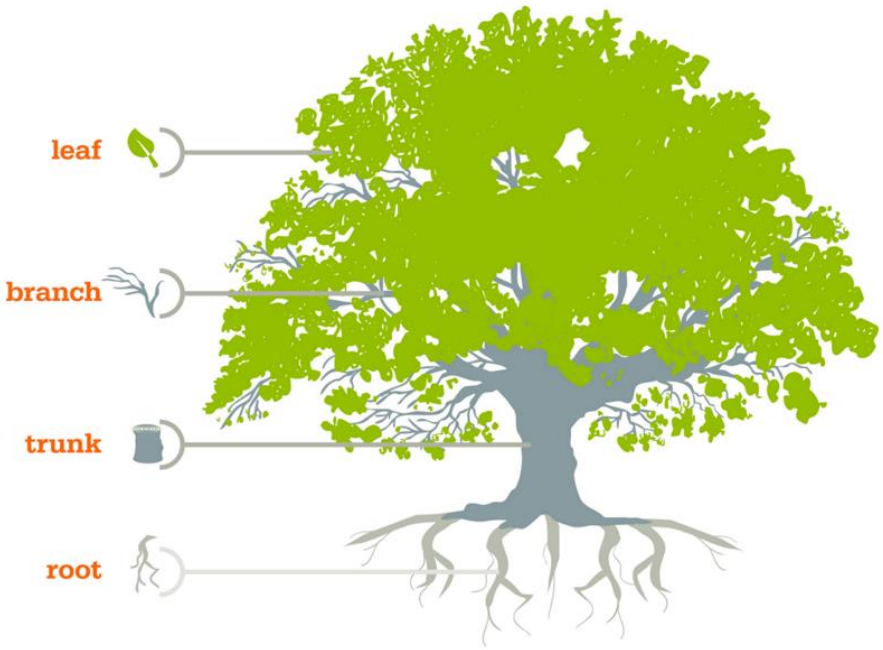

北京★

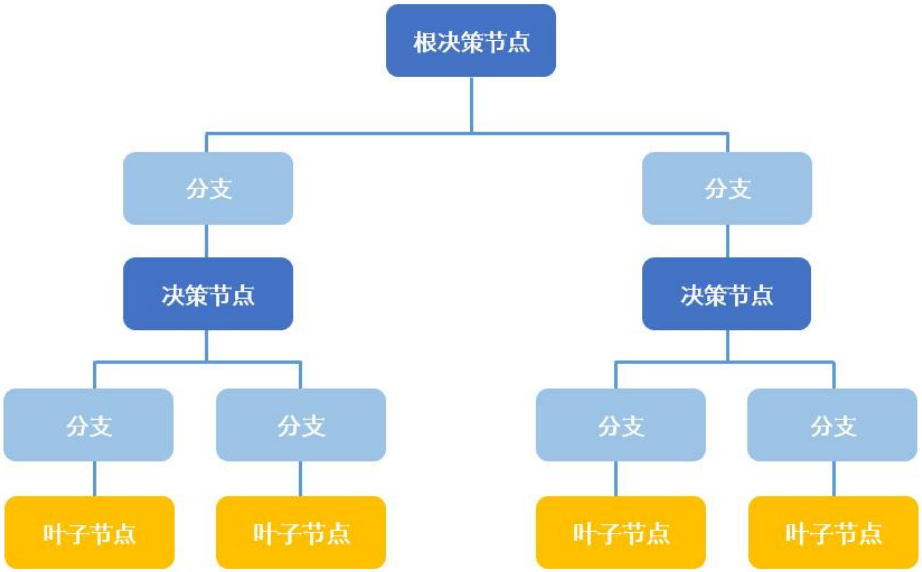

黑龙江
吉林
新疆
辽宁
内蒙古
北京
河北
天津
宁夏
山西
山东
青海
甘肃
河南
江苏
陕西
西藏
上海
四川
安徽
湖北
重庆
浙江
江西
湖南
福建
贵州
云南
广东
台湾
广西
香港
澳门
海南


历史文化学院
学术委员会
学位评定委员会
教学机构
党政管理机构
科研平台
《中学历史教学》编辑部
中国古代史教研室
中国近现代史教研室
世界古代中世纪史教研室
世界近现代史教研室
历史教师教育教研室
学院办公室
工会
研究生工作办公室
本科生工作办公室
中华二十四史研究中心
华南地方史志研究中心
华南宗教研究中心
国家民委民族理论政策研究基地

Lecture 3: Decision Trees

3.1 Basics


leaf
branch
trunk
root


根决策节点
分支
分支
决策节点
决策节点
分支
分支
分支
分支
叶子节点
叶子节点
叶子节点
叶子节点

Examples

- 决策树分类的思想类似于找对象，想象一个女孩的母亲要给这个女孩介绍男朋友，于是有了下面的对话：

女儿：多大年纪了？

母亲：26。

女儿：长的帅不帅？

母亲：挺帅的。

女儿：收入高不？

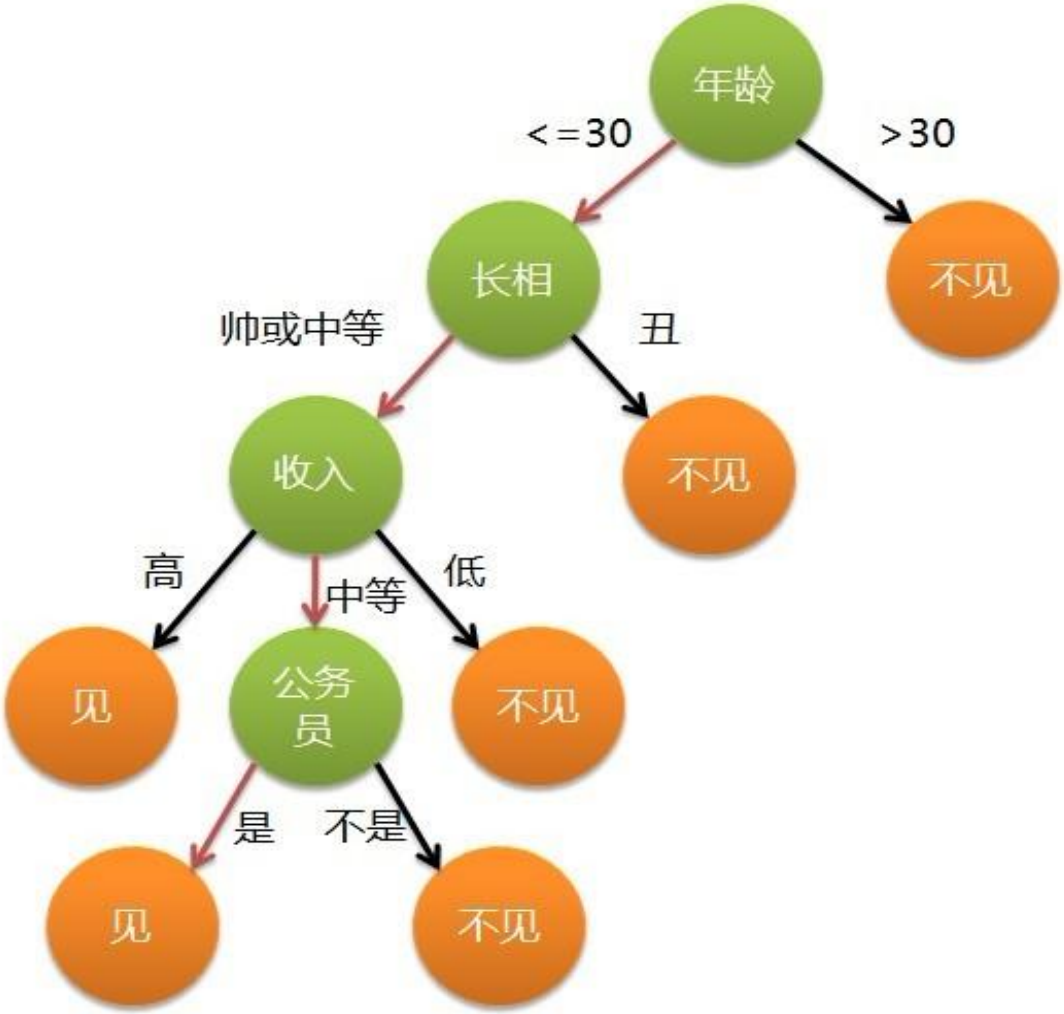
母亲：不算很高，中等情况。

女儿：是公务员不？

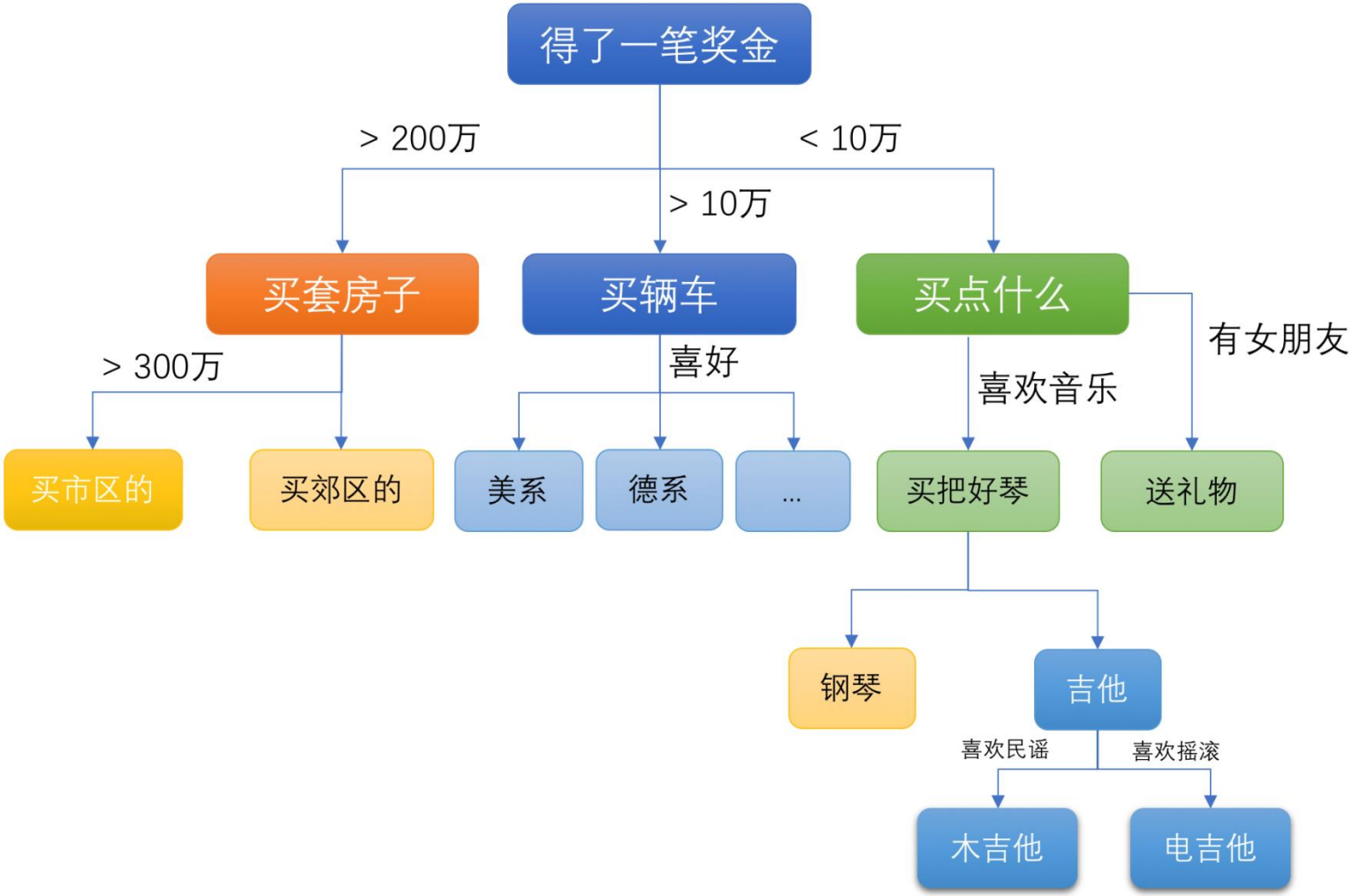
母亲：是，在税务局上班呢。

女儿：那好，我去见见。

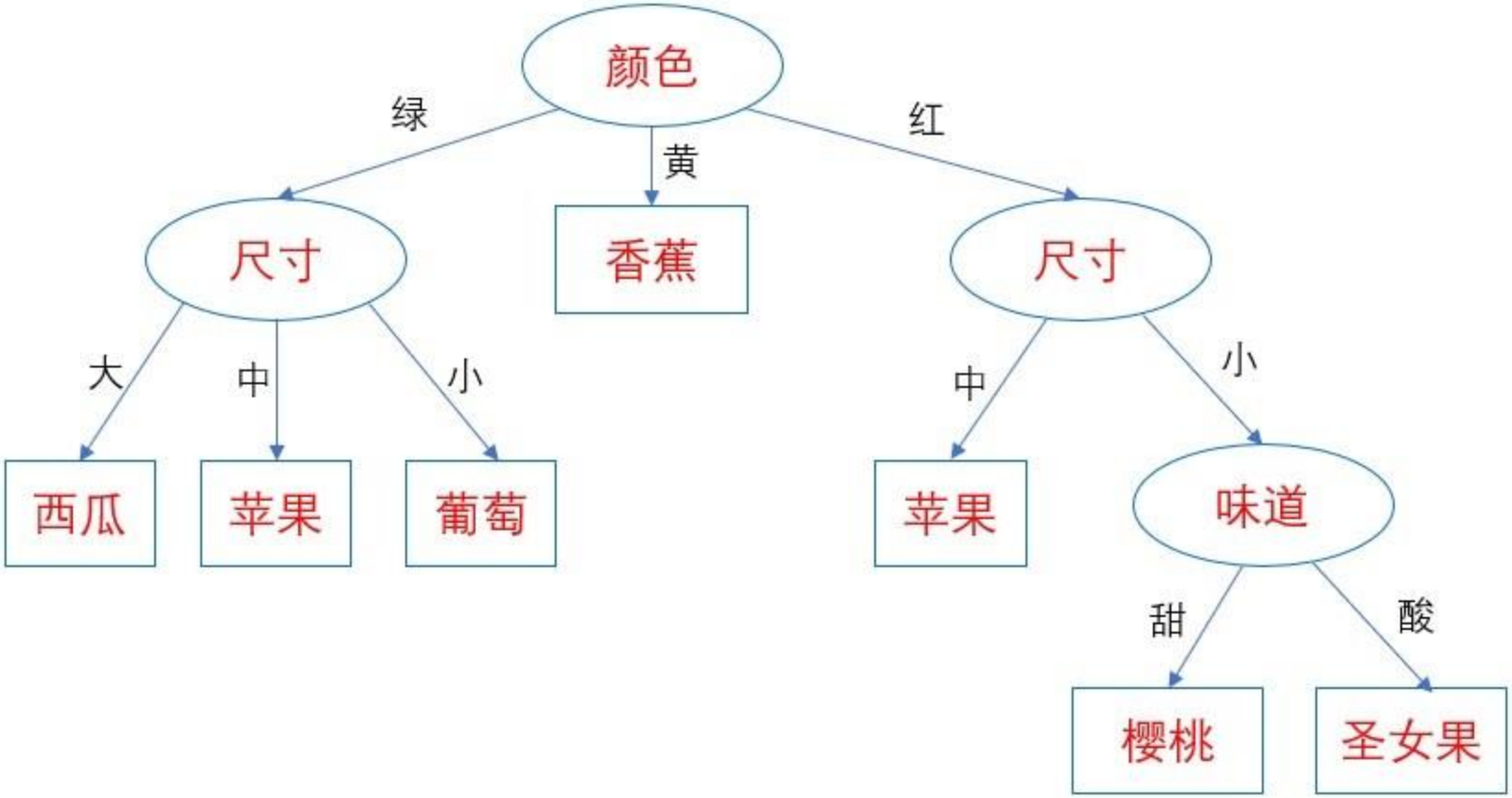
Examples



Examples


得了一笔奖金
> 200万
> 10万
< 10万
买套房子
买辆车
买点什么
> 300万
喜好
喜欢音乐
有女朋友
买市区的
买郊区的
美系
德系
...
买把好琴
送礼物
钢琴
吉他
喜欢民谣
喜欢摇滚
木吉他
电吉他

Examples



Decision Trees (决策树)

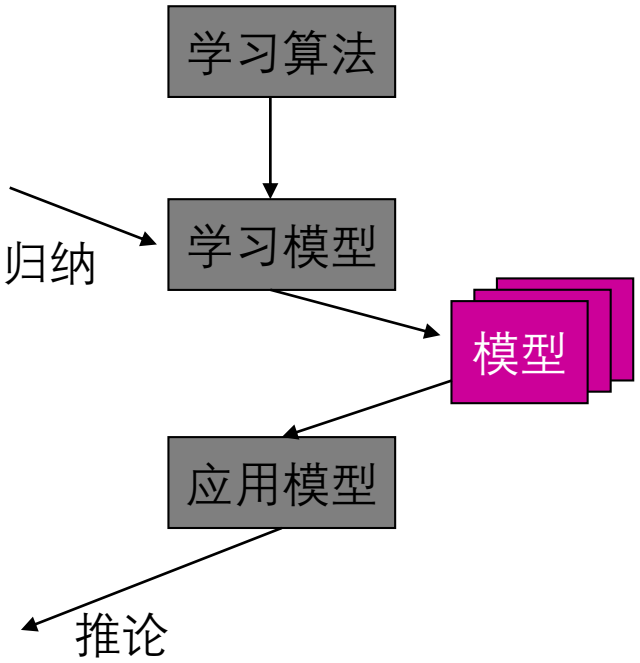
决策树解决分类问题的一般方法

训练集（类标号已知）

TID	A1	A2	A3	类
1	Y	100	L	N
2	N	125	S	N
3	Y	400	L	Y
4	N	415	M	N

验证集（类标号未知）

TID	A1	A2	A3	类
1	Y	100	L	?
2	N	125	S	?
3	Y	400	L	?
4	N	415	M	?



Decision Trees (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。

- 本质上是通过一系列规则对数据进行分类的过程

Decision Trees (决策树)

- 决策树的优点
  - 1、推理过程易理解，决策推理过程可以表示成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。

3.2 基本流程

基本流程

- 决策过程中提出的每个判定问题都是对某个属性的“测试”
- 决策过程的最终结论对应了我们所希望的判定结果
- 每个测试的结果或是导出最终结论，或者导出进一步的判定问题，其考虑范围是在上次决策结果的限定范围之内
- 从根结点到每个叶结点的路径对应了一个判定测试序列

决策树学习的目的是为了产生一棵泛化能力强，即处理未见示例能力强的决策树

基本流程

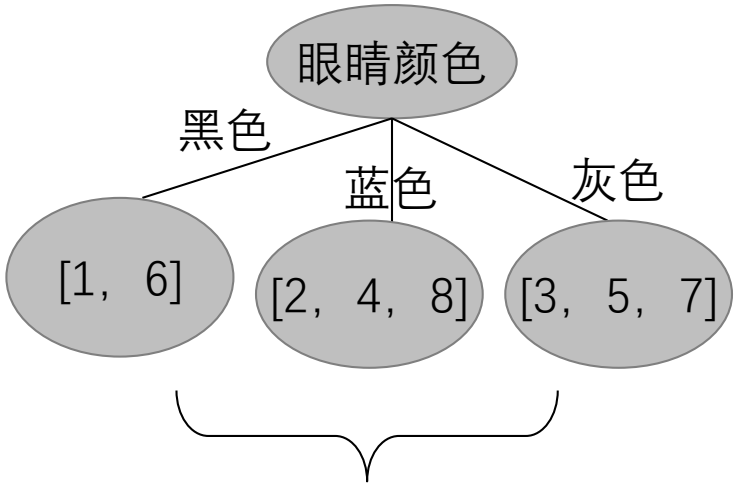
训练集：

人员	眼睛颜色	头发颜色	所属人种
1	黑色	黑色	黄种人
2	蓝色	金色	白种人
3	灰色	金色	白种人
4	蓝色	红色	白种人
5	灰色	红色	白种人
6	黑色	金色	混血
7	灰色	黑色	混血
8	蓝色	黑色	混血

基本流程

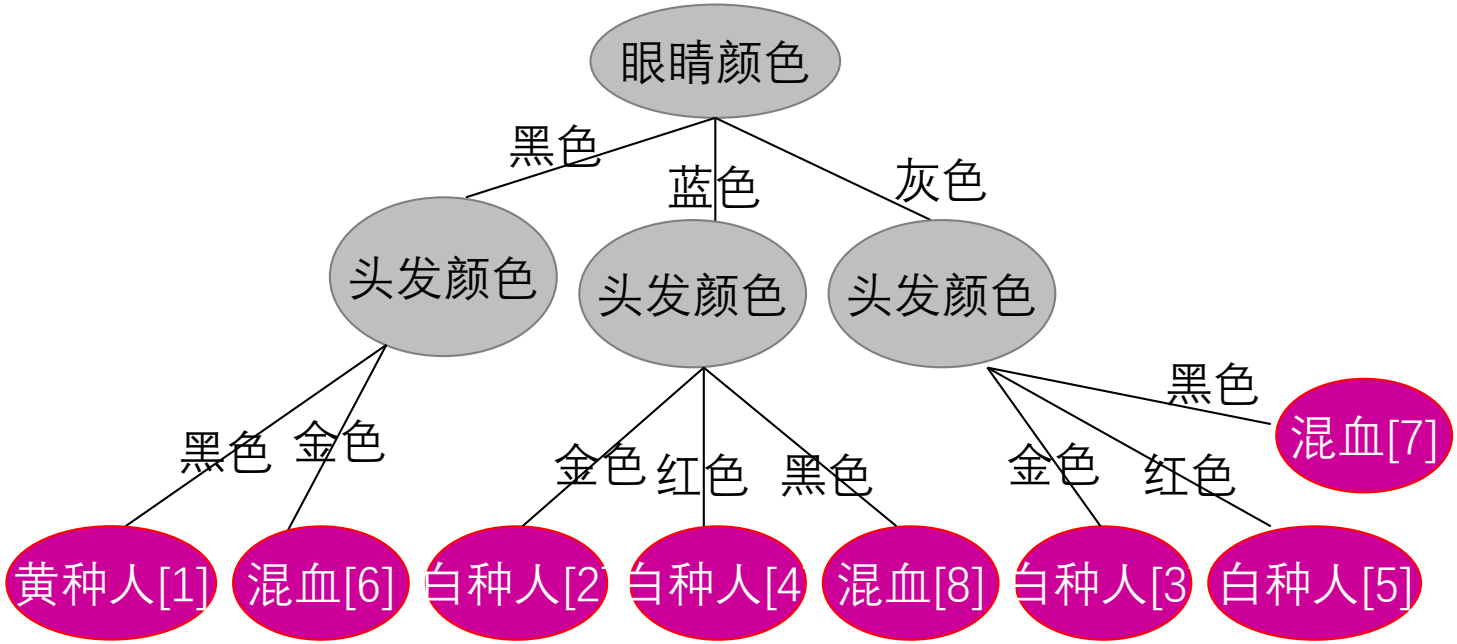
决策树的构建

人员	眼睛颜色	头发颜色	所属人种
1	黑色	黑色	黄种人
2	蓝色	金色	白种人
3	灰色	金色	白种人
4	蓝色	红色	白种人
5	灰色	红色	白种人
6	黑色	金色	混血
7	灰色	黑色	混血
8	蓝色	黑色	混血



不属于同一类，非叶结点

基本流程


眼睛颜色
黑色
蓝色
灰色
头发颜色
头发颜色
头发颜色
黑色
金色
金色
红色
黑色
黑色
金色
红色
混血[7]
黄种人[1]
混血[6]
白种人[2
白种人[4
混血[8]
白种人[3
白种人[5]

3.3 Decision Tree Algorithms

决策树算法

- 与决策树相关的重要算法包括:
  - **CLS, ID3, C4.5, CART**

- 算法的发展过程
  - Hunt,Marin和Stone 于1966年研制的CLS学习系统，用于学习单个概念。
  - 1979年, J.R. Quinlan 给出ID3算法，并在1983年和1986年对ID3 进行了总结和简化，使其成为决策树学习算法的典型。
  - Schlimmer 和Fisher 于1986年对ID3进行改造，在每个可能的决策树节点创建缓冲区，使决策树可以递增式生成，得到ID4算法。
  - 1988年，Utgoff 在ID4基础上提出了ID5学习算法，进一步提高了效率。
  - 1993年，Quinlan 进一步发展了ID3算法，改进成C4.5算法。
  - 另一类决策树算法为CART，与C4.5不同的是，CART的决策树由二元逻辑问题生成，每个树节点只有两个分枝，分别包括学习实例的正例与反例。

决策树CLS算法

- **CLS（Concept Learning System）算法**
  - CLS算法是早期的决策树学习算法。它是许多决策树学习算法的基础
- CLS基本思想
  - 从一棵空决策树开始，选择某一属性（分类属性）作为测试属性。该测试属性对应决策树中的决策结点。根据该属性的值的不同，可将训练样本分成相应的子集：
    - 如果该子集为空，或该子集中的样本属于同一个类，则该子集为叶结点；
    - 否则该子集对应于决策树的内部结点，需要选择一个新的分类属性对该子集进行划分，直到所有的子集都为空或者属于同一类。

基本流程

Algorithm 1 决策树学习基本算法

输入:

- 训练集 $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$;
- 属性集 $A = \{a_1, \ldots, a_d\}$.

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
4: end if
5: if A = ∅ OR D 中样本在 A 上取值相同 then
6:     将 node 标记叶结点，其类别标记为 D 中样本数最多的类; return
7: end if
8: 从 A 中选择最优划分属性 a_*;
9: for a_* 的每一个值 a_*^v do
10:    为 node 生成每一个分枝; 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:        将分枝结点标记为叶结点, 其类别标记为 D 中样本最多的类; return
13:    else
14:        以 TreeGenerate(D_v, A − {a_*}) 为分枝结点
15:    end if
16: end for
```

输出: 以 node 为根结点的一棵决策树

（1）当前结点包含的样本全部属于同一类别

（2）当前属性集为空，或所有样本在所有属性上取值相同

（3）当前结点包含的样本集合为空

决策树CLS算法

- CLS算法问题：
  - 在步骤8中，根据某种策略从训练样本属性表中选择属性a_*作为测试属性。没有规定采用何种测试属性。实践表明，测试属性集的组成以及测试属性的先后对决策树的学习具有举足轻重的影响。

决策树ID3算法

- ID3算法是一种经典的决策树学习算法，由Quinlan于1979年提出；
- ID3算法主要针对属性选择问题，是决策树学习方法中最具影响和最为典型的算法；
- 该方法使用**信息增益**选择测试属性；
- 从直觉上讲，小概率事件比大概率事件包含的信息量大。如果某件事情是“百年一见”则肯定比“习以为常”的事件包含的信息量大。

- **如何度量信息量的大小？**

划分选择-信息增益

- “信息熵”是度量样本集合纯度最常用的一种指标，假定当前样本集合D中第k类样本所占的比例为$p_k\,(K=1,2,...,|\mathcal{Y}|)$，则$D$的信息熵定义为

$$\mathrm{Ent}(D)=-\sum_{k=1}^{|\mathcal{Y}|}p_k\log_2 p_k$$

  $\mathrm{Ent}(D)$的值越小，则D的纯度越高

- 计算信息熵时约定：若$p=0$，则$p\log_2 p=0$
- $\mathrm{Ent}(D)$的最小值为0，最大值为$\log_2|\mathcal{Y}|$

划分选择-信息增益

- 离散属性 a 有 V 个可能的取值 $\{a^1, a^2, ..., a^V\}$，用 a 来进行划分，则会产生 V 个分支结点，其中第 v 个分支结点包含了 D 中所有在属性 a 上取值为 a^v 的样本，记为 D^v。则可计算出用属性 a 对样本集 D 进行划分所获得的“信息增益”：

$$\mathrm{Gain}(D, a) = \mathrm{Ent}(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} \mathrm{Ent}(D^v)$$

为分支结点权重，样本数越多的分支结点的影响越大

- 一般而言，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

划分选择-信息增益

存在的问题:

- 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

信息增益对可取值数目较多的属性有所偏好

信息增益比

- 增益率定义：

$$\text{Gain\_ratio}(D,a) = \frac{\text{Gain}(D,a)}{\text{IV}(a)}$$

其中

$$\text{IV}(a) = -\sum_{v=1}^{V} \frac{|D^v|}{|D|} \log_2 \frac{|D^v|}{|D|}$$

称为属性 a 的“固有值” [Quinlan, 1993]，属性 a 的可能取值数目越多（即 V 越大），则 $\text{IV}(a)$ 的值通常就越大

- 存在的问题

增益率准则对可取值数目较少的属性有所偏好

信息增益比

C4.5 [Quinlan, 1993]使用了一个启发式：

- 先从候选划分属性中找出信息增益高于平均水平的属性，再从中选取增益率最高的

计数	年龄	收入	学生	信誉	归类：买计算机？
64	青	高	否	良	不买
64	青	高	否	优	不买
128	中	高	否	良	买
60	老	中	否	良	买
64	老	低	是	良	买
64	老	低	是	优	不买
64	中	低	是	优	买
128	青	中	否	良	不买
64	青	低	是	良	买
132	老	中	是	良	买
64	青	中	是	优	买
32	中	中	否	优	买
32	中	高	是	良	买
63	老	中	否	优	不买
1	老	中	否	优	买

计数	年龄	收入	学生	信誉	归类：买计算机？
64	青	高	否	良	不买
64	青	高	否	优	不买
128	中	高	否	良	买
60	老	中	否	良	买
64	老	低	是	良	买
64	老	低	是	优	不买
64	中	低	是	优	买
128	青	中	否	良	不买
64	青	低	是	良	买
132	老	中	是	良	买
64	青	中	是	优	买
32	中	中	否	优	买
32	中	高	是	良	买
63	老	中	否	优	不买
1	老	中	否	优	买

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

第1步计算决策属性的熵

决策属性“买计算机？”。
该属性分两类：买/不买

|C1|(买)=641
|C2|（不买）= 383
|D|=|C1|+|C2|=1024

P1=641/1024=0.6260
P2=383/1024=0.3740

$H(D)=-P1Log_2P1-P2Log_2P2$
$=-(P1Log_2P1+P2Log_2P2) =0.9537$

计数	年龄	收入	学生	信誉	归类：买计算机？
64	青	高	否	良	不买
64	青	高	否	优	不买
128	中	高	否	良	买
60	老	中	否	良	买
64	老	低	是	良	买
64	老	低	是	优	不买
64	中	低	是	优	买
128	青	中	否	良	不买
64	青	低	是	良	买
132	老	中	是	良	买
64	青	中	是	优	买
32	中	中	否	优	买
32	中	高	是	良	买
63	老	中	否	优	不买
1	老	中	否	优	买

第2步计算条件属性的熵

条件属性共有4个:
年龄、收入、学生、信誉。
分别计算不同属性的信息增益。

计数	年龄	收入	学生	信誉	归类：买计算机？
64	青	高	否	良	不买
64	青	高	否	优	不买
128	中	高	否	良	买
60	老	中	否	良	买
64	老	低	是	良	买
64	老	低	是	优	不买
64	中	低	是	优	买
128	青	中	否	良	不买
64	青	低	是	良	买
132	老	中	是	良	买
64	青	中	是	优	买
32	中	中	否	优	买
32	中	高	是	良	买
63	老	中	否	优	不买
1	老	中	否	优	买

第2-1步计算年龄的熵

年龄共分三个组： 青年、中年、老年
青年买与不买比例为128/256

|D11|(买)=128
|D12|(不买)= 256
|D1|=384

P1=128/384
P2=256/384

$H(D1)=-P1\log_2 P1-P2\log_2 P2$
$=-(P1\log_2 P1+P2\log_2 P2)$
$=0.9183$

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

计数	年龄	收入	学生	信誉	归类：买计算机？
64	青	高	否	良	不买
64	青	高	否	优	不买
128	中	高	否	良	买
60	老	中	否	良	买
64	老	低	是	良	买
64	老	低	是	优	不买
64	中	低	是	优	买
128	青	中	否	良	不买
64	青	低	是	良	买
132	老	中	是	良	买
64	青	中	是	优	买
32	中	中	否	优	买
32	中	高	是	良	买
63	老	中	否	优	不买
1	老	中	否	优	买

第2-2步计算年龄的熵

年龄共分三个组： 青年、中年、老年
中年买与不买比例为256/0

|D21|(买)=256
|D22|(不买)= 0
|D2|=256

P1=256/256
P2=0/256

$H(D2)=-P1Log_2P1-P2Log_2P2$
$=-(P1Log_2P1+P2Log_2P2)$
$=0$

计数	年龄	收入	学生	信誉	归类：买计算机？
64	青	高	否	良	不买
64	青	高	否	优	不买
128	中	高	否	良	买
60	老	中	否	良	买
64	老	低	是	良	买
64	老	低	是	优	不买
64	中	低	是	优	买
128	青	中	否	良	不买
64	青	低	是	良	买
132	老	中	是	良	买
64	青	中	是	优	买
32	中	中	否	优	买
32	中	高	是	良	买
63	老	中	否	优	不买
1	老	中	否	优	买

第2-3步计算年龄的熵

年龄共分三个组： 青年、中年、老年
老年买与不买比例为125/127

|D31|(买)=125
|D32|（不买）=127
|D3|=S1+S2=252

P1=125/252
P2=127/252

$H(D3)=-P1Log_2P1-P2Log_2P2$
$=-(P1Log_2P1+P2Log_2P2)$
$=0.9157$

计数	年龄	收入	学生	信誉	归类：买计算机？
64	青	高	否	良	不买
64	青	高	否	优	不买
128	中	高	否	良	买
60	老	中	否	良	买
64	老	低	是	良	买
64	老	低	是	优	不买
64	中	低	是	优	买
128	青	中	否	良	不买
64	青	低	是	良	买
132	老	中	是	良	买
64	青	中	是	优	买
32	中	中	否	优	买
32	中	高	是	良	买
63	老	中	否	优	不买
1	老	中	否	优	买

第2-4步计算年龄的熵

年龄共分三个组： 青年、中年、老年所占比例

青年组 384/1025=0.375

中年组 256/1024=0.25

老年组 384/1024=0.375

计算年龄的平均信息期望

E（年龄）=0.375*0.9183+
0.25*0+
0.375*0.9157
=0.6877

G（年龄信息增益）
=0.9537-0.6877
=0.2660 （1）

计数	年龄	收入	学生	信誉	归类：买计算机？
64	青	高	否	良	不买
64	青	高	否	优	不买
128	中	高	否	良	买
60	老	中	否	良	买
64	老	低	是	良	买
64	老	低	是	优	不买
64	中	低	是	优	买
128	青	中	否	良	不买
64	青	低	是	良	买
132	老	中	是	良	买
64	青	中	是	优	买
32	中	中	否	优	买
32	中	高	是	良	买
63	老	中	否	优	不买
1	老	中	否	优	买

第3步 计算收入的熵

收入共分三个组：
高、中、低

E（收入）=0.9361

收入信息增益=0.9537-0.9361
=0.0176 (2)

计数	年龄	收入	学生	信誉	归类：买计算机？
64	青	高	否	良	不买
64	青	高	否	优	不买
128	中	高	否	良	买
60	老	中	否	良	买
64	老	低	是	良	买
64	老	低	是	优	不买
64	中	低	是	优	买
128	青	中	否	良	不买
64	青	低	是	良	买
132	老	中	是	良	买
64	青	中	是	优	买
32	中	中	否	优	买
32	中	高	是	良	买
63	老	中	否	优	不买
1	老	中	否	优	买

第4步计算学生的熵

学生共分二个组：
学生、非学生

E（学生）=0.7811

年龄信息增益=0.9537-0.7811
=0.1726 （3）

计数	年龄	收入	学生	信誉	归类：买计算机？
64	青	高	否	良	不买
64	青	高	否	优	不买
128	中	高	否	良	买
60	老	中	否	良	买
64	老	低	是	良	买
64	老	低	是	优	不买
64	中	低	是	优	买
128	青	中	否	良	不买
64	青	低	是	良	买
132	老	中	是	良	买
64	青	中	是	优	买
32	中	中	否	优	买
32	中	高	是	良	买
63	老	中	否	优	不买
1	老	中	否	优	买

第5步计算信誉的熵

信誉分二个组：
良好，优秀

E（信誉）= 0.9048
信誉信息增益=0.9537-0.9048
=0.0453 （4）

计数	年龄	收入	学生	信誉	归类：买计算机?
64	青	高	否	良	不买
64	青	高	否	优	不买
128	中	高	否	良	买
60	老	中	否	良	买
64	老	低	是	良	买
64	老	低	是	优	不买
64	中	低	是	优	买
128	青	中	否	良	不买
64	青	低	是	良	买
132	老	中	是	良	买
64	青	中	是	优	买
32	中	中	否	优	买
32	中	高	是	良	买
63	老	中	否	优	不买
1	老	中	否	优	买

第6步计算选择节点

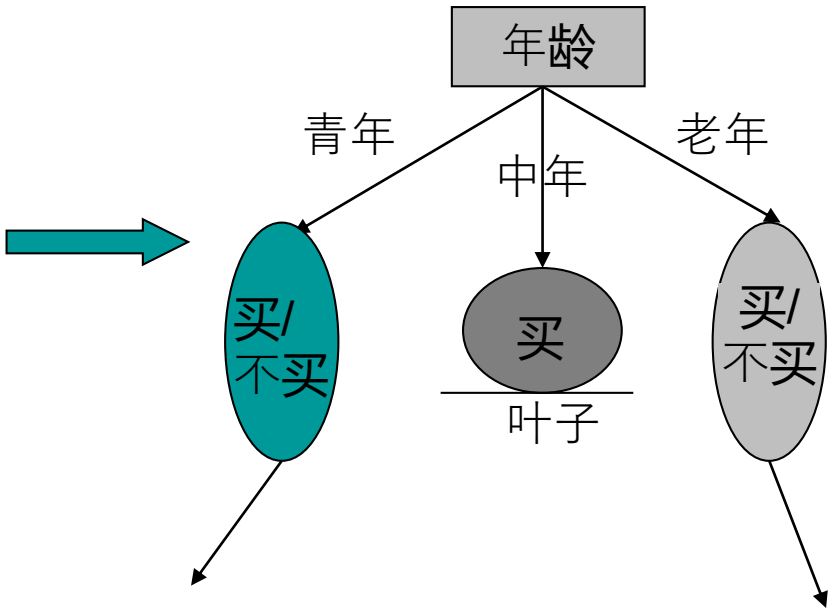
年龄信息增益=0.9537-0.6877
=0.2660 （1）

收入信息增益=0.9537-0.9361
=0.0176 （2）

年龄信息增益=0.9537-0.7811
=0.1726 （3）

信誉信息增益=0.9537-0.9048
=0.0453 （4）

计数	年龄	收入	学生	信誉	归类：买计算机？
64	青	高	否	良	不买
64	青	高	否	优	不买
128	青	中	否	良	不买
64	青	低	是	良	买
64	青	中	是	优	买



计数	年龄	收入	学生	信誉	归类：买计算机？
64	青	高	否	良	不买
64	青	高	否	优	不买
128	青	中	否	良	不买
64	青	低	是	良	买
64	青	中	是	优	买

青年买与不买比例为：128/256

|C1|(买)=128
|C2|（不买）= 256
|D|=384

P1=128/384
P2=256/384

H(D)=-P1Log2P1-P2Log2P2
=-(P1Log2P1+P2Log2P2)
=0.9183

计数	年龄	收入	学生	信誉	归类：买计算机？
64	青	高	否	良	不买
64	青	高	否	优	不买
128	青	中	否	良	不买
64	青	低	是	良	买
64	青	中	是	优	买

如果选择收入作为节点
分高、中、低

H(D1)=0
比例: 128/384=0.3333

H(D2)=0.9183
比例: 192/384=0.5

H(D3)=0
比例: 64/384=0.1667

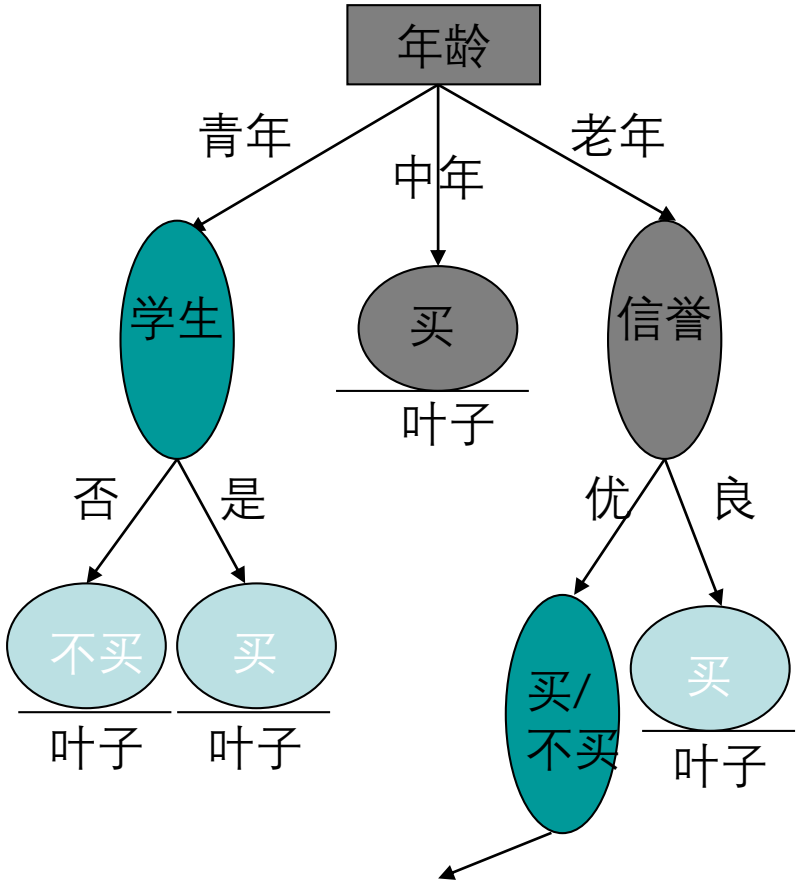
平均信息期望（加权总和）：

E(收入）= 0.3333 * 0 + 0.5 * 0.9183 + 0.1667 * 0 = 0.4592

Gain(收入) = I(128, 256) - E(收入)=0.9183 – 0.4592 = 0.4591

注意

计数	年龄	收入	学生	信誉	归类：买计算机？
64	青	高	否	良	不买
64	青	高	否	优	不买
128	中	高	否	良	买
60	老	中	否	良	买
64	老	低	是	良	买
64	老	低	是	优	不买
64	中	低	是	优	买
128	青	中	否	良	不买
64	青	低	是	良	买
132	老	中	是	良	买
64	青	中	是	优	买
32	中	中	否	优	买
32	中	高	是	良	买
63	老	中	否	优	不买
1	老	中	否	优	买



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，以构造一颗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。

CART树

- 分类回归树CART(Classification and Regression Trees)
  - 1984 << Classification and Regression Trees >>
- L.Breiman，J.Friedman，R.Olshen和C.Stone
  - http://www.stat.berkeley.edu/~breiman/
  - http://www-stat.stanford.edu/~jhf/
  - http://www-stat.stanford.edu/~olshen/

- 目标变量是类别的 --- 分类树
- 目标变量是连续的 --- 回归树

CART树

- CART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

划分选择-基尼指数

- 数据集D的纯度可用“基尼值”来度量

$$\text{Gini}(D) = \sum_{k=1}^{|\mathcal{Y}|} \sum_{k' \neq k} p_k p_{k'} = 1 - \sum_{k=1}^{|\mathcal{Y}|} p_k^2$$

$\text{Gini}(D)$ 越小，数据集 D 的纯度越高

反映了从 D 中随机抽取两个样本，其类别标记不一致的概率

- 属性 a 的基尼指数定义为：

$$\text{Gini\_index}(D, a) = \sum_{v=1}^{V} \frac{|D^v|}{|D|} \text{Gini}(D^v)$$

- 应选择那个使划分后基尼指数最小的属性作为最优划分属性，即

$$a_* = \underset{a \in A}{\text{argmin}}\ \text{Gini\_index}(D, a)$$

Exercise

- Exercise: Can decision trees solve non-linear classification problems? Why?

Send your answer to sysucsers@163.com

今天中午12点之前把答案发送到邮箱: sysucsers@163.com
邮件标题和文件名称：学号-姓名-Lecture 3(decision tree)，
例如: 01111-张三- Lecture 3(decision tree)

3.4 Pruning

决策树面临的问题

- 理想的决策树有三种:
  - (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。
- 找到这种最优的决策树是NP难题。决策树优化的目的就是要找到尽可能趋向于最优的决策树。

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渡拟合训练样例 (过拟合：overfitting).

决策树面临的问题

- 对学习算法是否成功的真正测试是看它对于训练中**未见到的数据的执行性能**。训练过程应该包含训练样本和验证样本。验证样本用于测试训练后的性能。**如果验证结果差，则需要考虑采用不同的结构重新进行训练**，例如使用更大的样本集，或者改变从连续值到离散值得数据转换等。

- 通常应该建立一个验证过程，在训练最终完成后用来检测训练结果的泛化能力。

决策树的剪枝

- 为什么剪枝
  - “剪枝”是决策树学习算法对付“过拟合”的主要手段
  - 可通过“剪枝”来一定程度避免因决策分支过多，以致于把训练集自身的一些特点当做所有数据都具有的一般性质而导致的过拟合
- 剪枝的基本策略
  - 预剪枝
  - 后剪枝
- 判断决策树泛化性能是否提升的方法
  - 留出法：预留一部分数据用作“验证集”以进行性能评估

决策树的剪枝-预剪枝

- 决策树生成过程中，对每个结点在划分前先进行估计，若当前结点的划分不能带来决策树泛化性能提升，则停止划分并将当前结点记为叶结点，其类别标记为训练样例数最多的类别；

决策树的剪枝-预剪枝

验证集

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否

结点1：若不划分，则将其标记为叶结点，类别标记为训练样例中最多的类别，即好瓜。验证集中，$\{4,5,8\}$被分类正确，得到验证集精度为

$$\frac{3}{7} \times 100\% = 42.9\%$$

验证集精度

划分前：42.9%

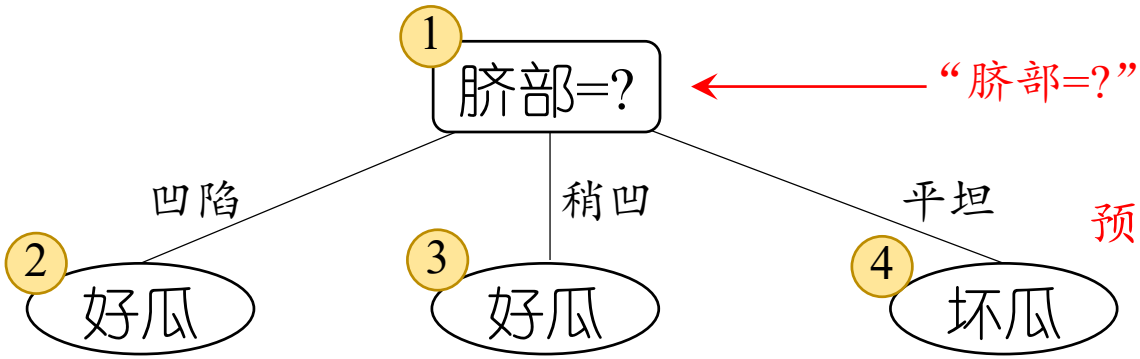
决策树的剪枝-预剪枝

验证集

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否

结点1：若划分，根据结点 ②, ③ , ④ 的训练样例，将这 3个结点分别标记为“好瓜”、“好瓜”、“坏瓜”。此时，验证集中编号为 $\{4, 5, 8, 11, 12\}$ 的样例被划分正确，验证集精度为

$$\frac{5}{7} \times 100\% = 71.4\%$$



验证集精度

划分前：42.9%

划分后：71.4%

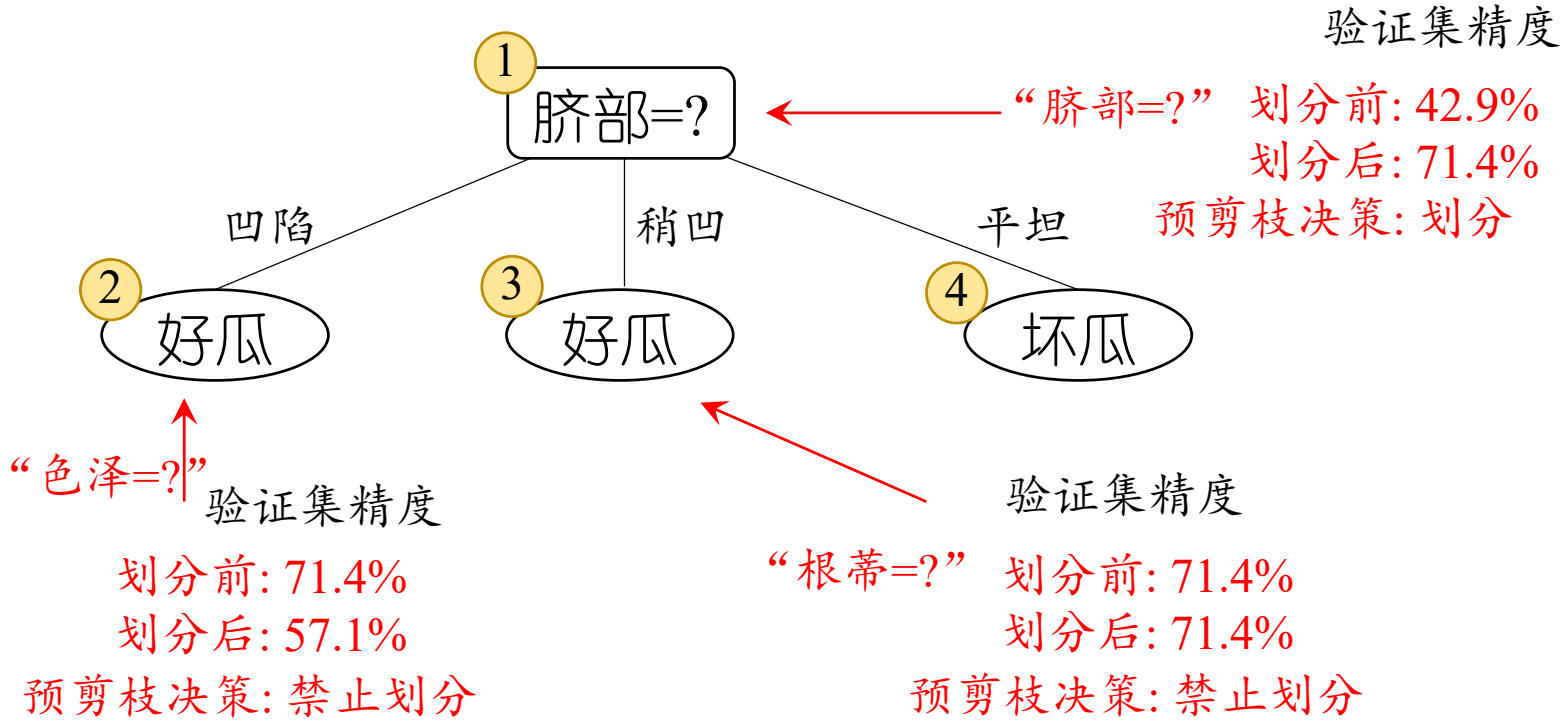
预剪枝决策：划分

决策树的剪枝-预剪枝

验证集

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否

对结点②，③，④分别进行剪枝判断，结点②，③都禁止划分，结点④本身为叶子结点。最终得到仅有一层划分的决策树，称为**“决策树桩”**



决策树的剪枝-预剪枝

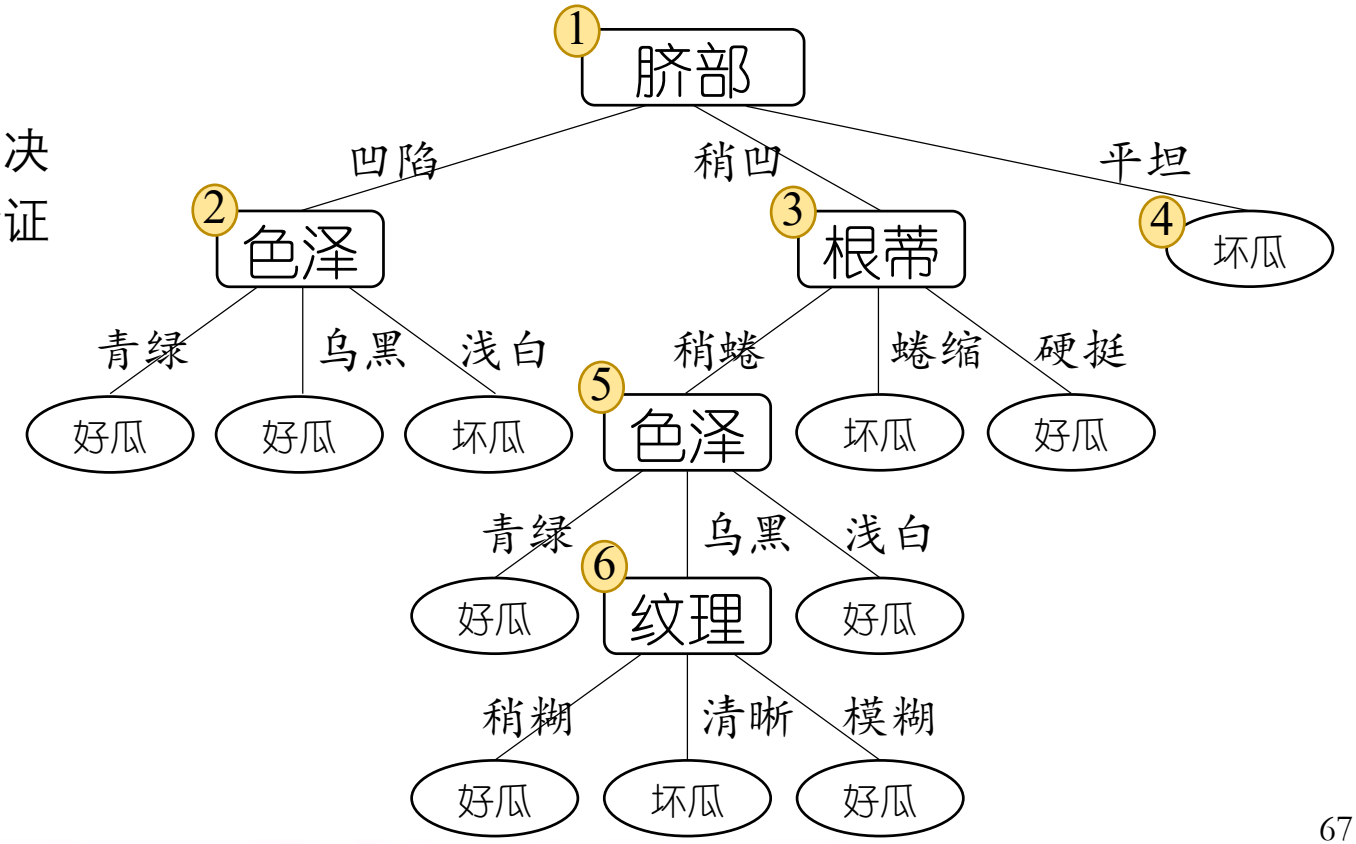
预剪枝的优缺点

- 优点
  - 降低过拟合风险
  - 显著减少训练时间和测试时间开销
- 缺点
  - 欠拟合风险：有些分支的当前划分虽然不能提升泛化性能，但在其基础上进行的后续划分却有可能导致性能显著提高。预剪枝基于“贪心”本质禁止这些分支展开，带来了欠拟合风险

决策树的剪枝-后剪枝

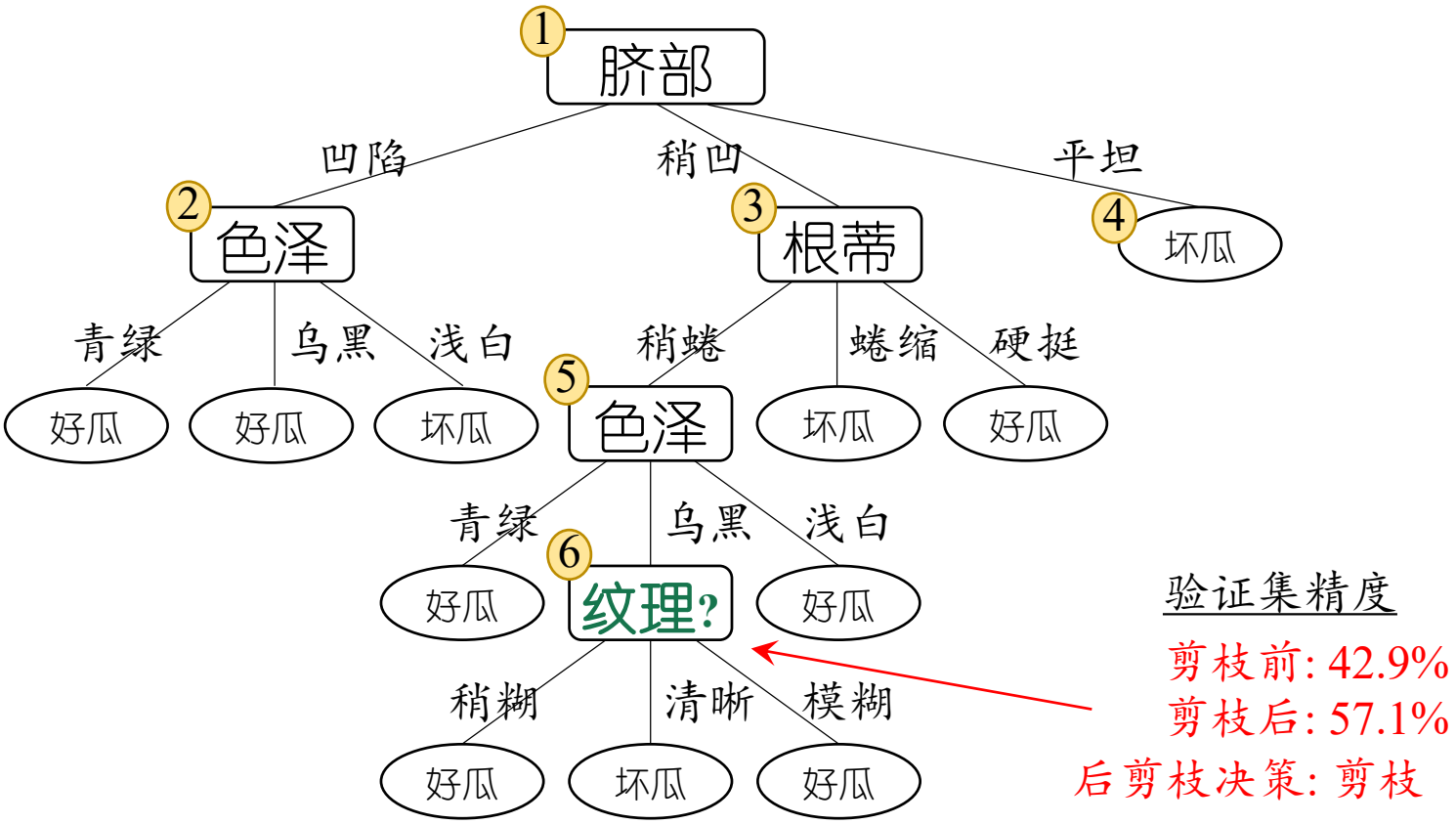
- 先从训练集生成一棵完整的决策树，然后自底向上地对非叶结点进行考察，若将该结点对应的子树替换为叶结点能带来决策树泛化性能提升，则将该子树替换为叶结点

首先生成一棵完整的决策树，该决策树的验证集精度为 42.9%



决策树的剪枝-后剪枝

- 首先考虑结点⑥，若将其替换为叶结点，根据落在其上的训练样本{7,15}将其标记为“好瓜”，得到验证集精度提高至57.1%，则决定剪枝



验证集精度

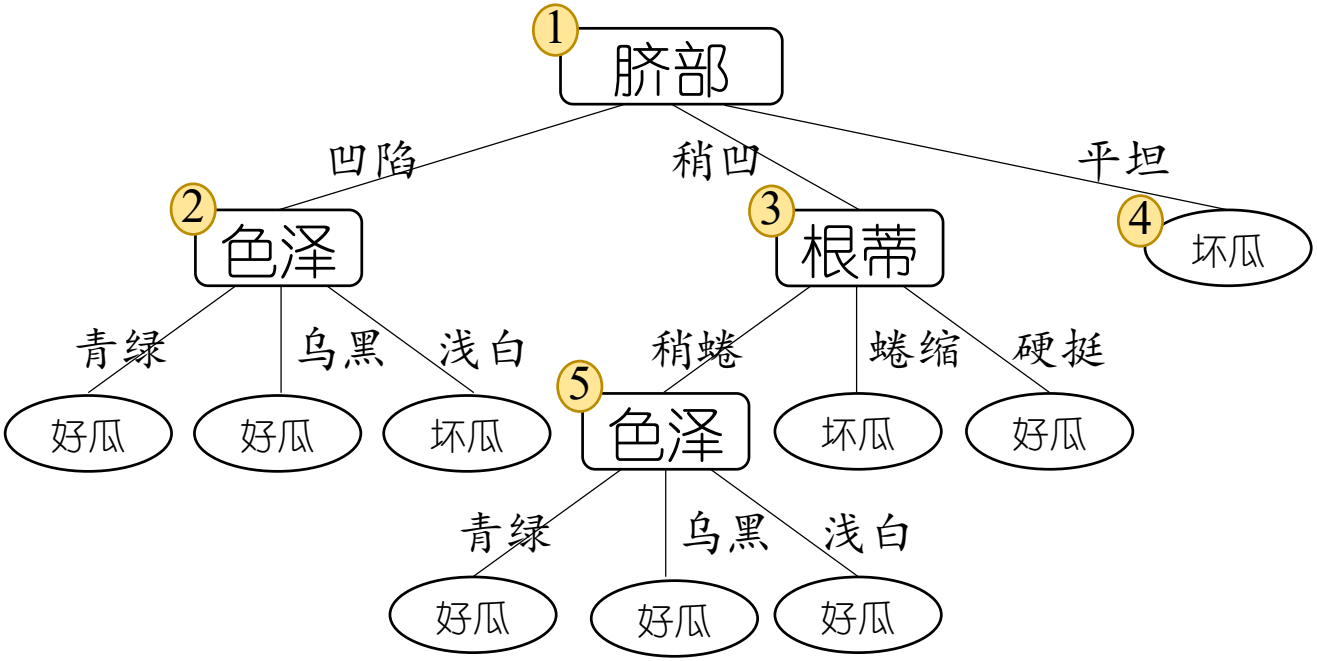
剪枝前: 42.9%

剪枝后: 57.1%

后剪枝决策: 剪枝

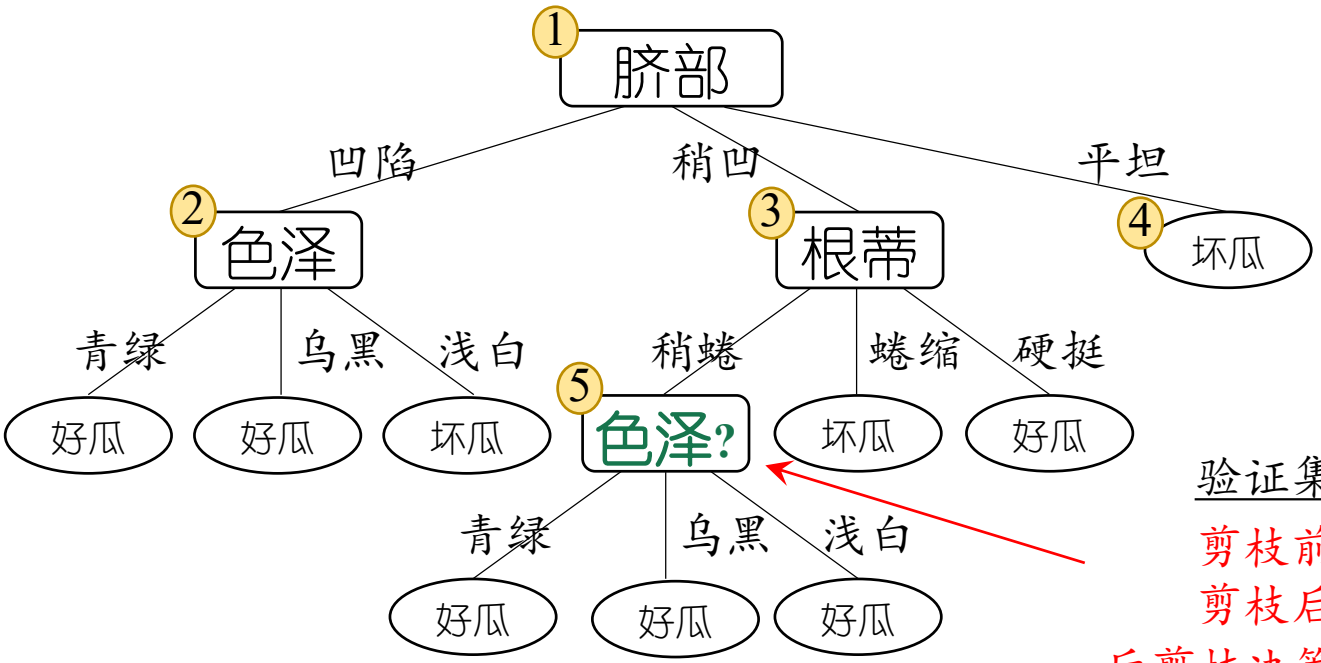
决策树的剪枝-后剪枝

- 首先考虑结点⑥，若将其替换为叶结点，根据落在其上的训练样本{7,15}将其标记为“好瓜”，得到验证集精度提高至57.1%，则决定剪枝



决策树的剪枝-后剪枝

- 然后考虑结点⑤，若将其替换为叶结点，根据落在其上的训练样本{6, 7, 15}将其标记为“好瓜”，得到验证集精度仍为57.1%，可以不进行剪枝



验证集精度

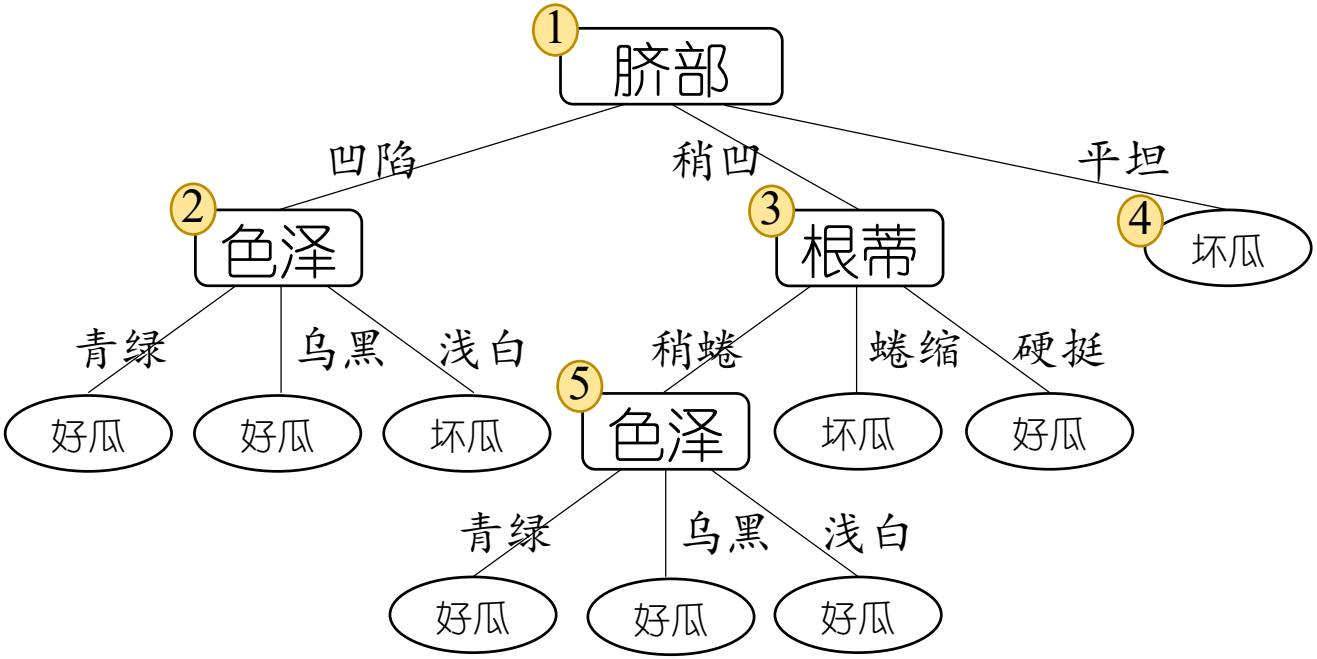
剪枝前: 57.1 %

剪枝后: 57.1%

后剪枝决策: 不剪枝

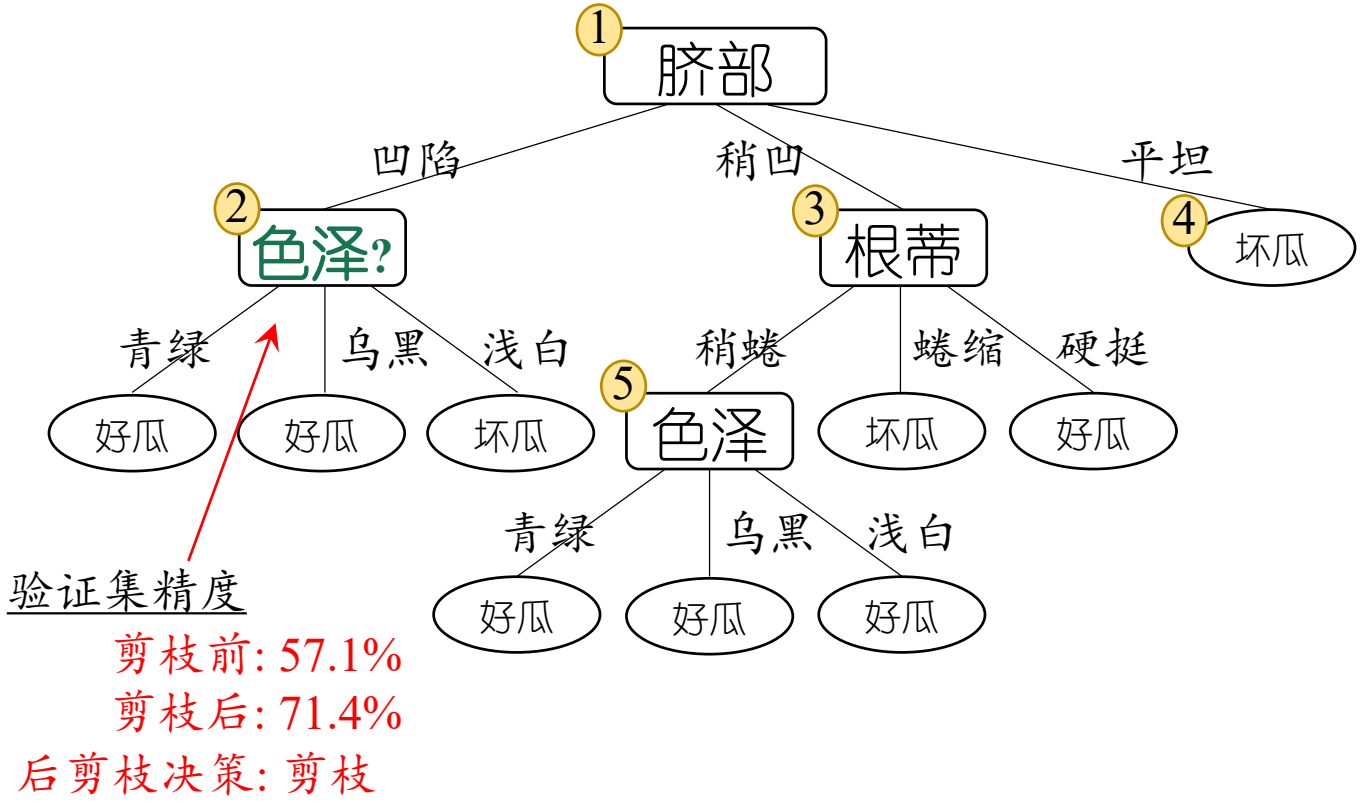
决策树的剪枝-后剪枝

- 然后考虑结点⑤，若将其替换为叶结点，根据落在其上的训练样本$\{6,7,15\}$将其标记为“好瓜”，得到验证集精度仍为57.1%，可以不进行剪枝



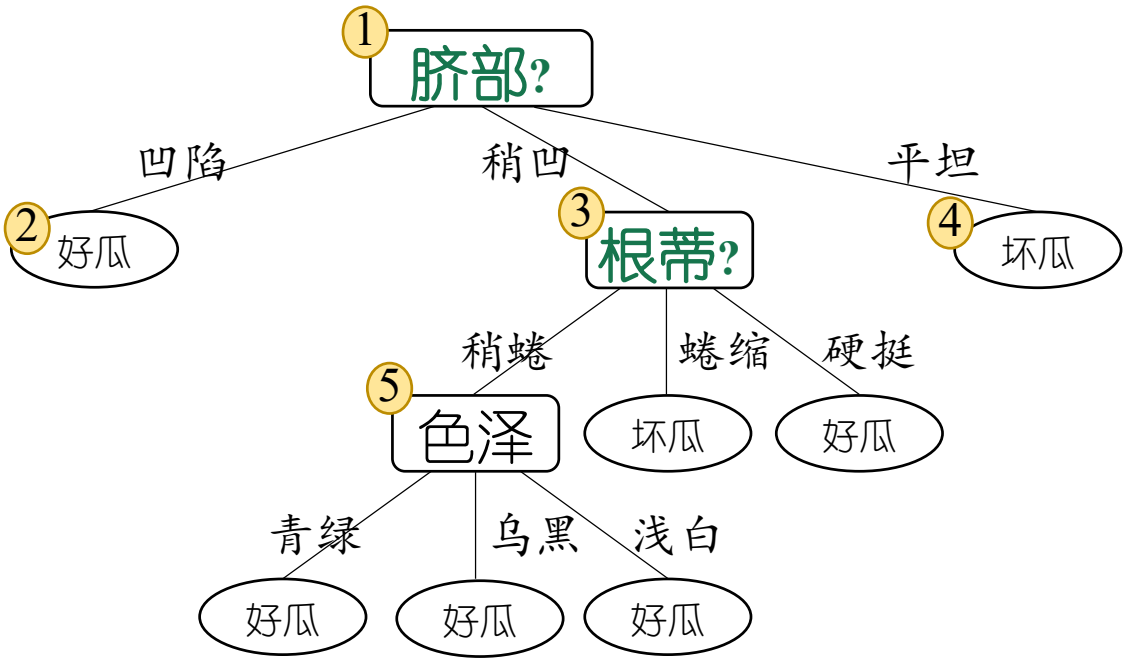
决策树的剪枝-后剪枝

- 对结点②，若将其替换为叶结点，根据落在其上的训练样本$\{1, 2, 3, 14\}$，将其标记为“好瓜”，得到验证集精度提升至71.4%，则决定剪枝



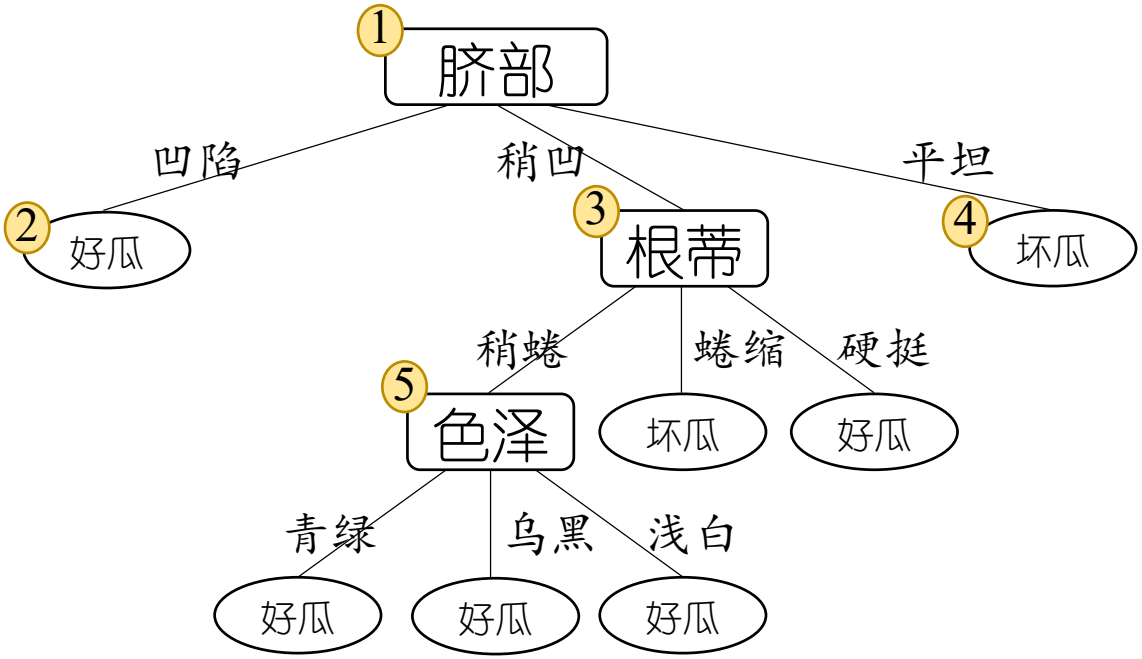
决策树的剪枝-后剪枝

- 对结点③和①，先后替换为叶结点，验证集精度均未提升，则分支得到保留



决策树的剪枝-后剪枝

- 最终基于后剪枝策略得到的决策树如图所示



决策树的剪枝-后剪枝

后剪枝的优缺点

- 优点
  - 后剪枝比预剪枝保留了更多的分支，欠拟合风险小，泛化性能往往优于预剪枝决策树
- 缺点
  - 训练时间开销大：后剪枝过程是在生成完全决策树之后进行的，需要自底向上对所有非叶结点逐一考察

3.5 连续与缺失值

连续与缺失值 – 连续值处理

- 连续属性离散化(二分法)
  - 第一步：假定连续属性a在样本集D上出现n个不同的取值，从小到大排列，记为 $a^1, a^2, ...a^n$，基于划分点t，可将D分为子集D_t^-和D_t^+，其中D_t^-包含那些在属性a上取值不大于t的样本，D_t^+包含那些在属性a上取值大于t的样本。考虑包含$n-1$个元素的候选划分点集合

$$T_a = \left\{ \frac{a^i + a^{i+1}}{2} \mid 1 \leq i \leq n-1 \right\}$$

即把区间 $[a^i, a^{i-1})$ 的中位点 $\frac{a^i + a^{i+1}}{2}$ 作为候选划分点

连续与缺失值 – 连续值处理

- 连续属性离散化(二分法)
  - 第二步：采用离散属性值方法，考察这些划分点，选取最优的划分点进行样本集合的划分

$$\begin{aligned}\mathrm{Gain}(D,a) &= \max_{t\in T_a} \mathrm{Gain}(D,a,t) \\ &= \max_{t\in T_a} \mathrm{Ent}(D) - \sum_{\lambda\in\{-,+\}} \frac{|D_t^\lambda|}{|D|}\mathrm{Ent}(D_t^\lambda)\end{aligned}$$

其中 $\mathrm{Gain}(D,a,t)$ 是样本集 D 基于划分点 t 二分后的信息增益，于是，就可选择使 $\mathrm{Gain}(D,a,t)$ 最大化的划分点

连续与缺失值 – 连续值处理

连续值处理实例

编号	色泽	根蒂	敲声	纹理	脐部	触感	密度	含糖率	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	0.697	0.460	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	0.774	0.376	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	0.634	0.264	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	0.608	0.318	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	0.556	0.215	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	0.403	0.237	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	0.481	0.149	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	0.437	0.211	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	0.666	0.091	否
10	青绿	硬挺	清脆	清晰	平坦	软粘	0.243	0.267	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	0.245	0.057	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	0.343	0.099	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	0.639	0.161	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	0.657	0.198	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	0.360	0.370	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	0.593	0.042	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	0.719	0.103	否

对属性“密度”，其候选划分点集合包含16 个候选值：

$$T_{\text{密度}} = \{0.244, 0.294, 0.351, 0.381, 0.420, 0.459, 0.518, 0.574, 0.600, 0.621, 0.636, 0.648, 0.661, 0.681, 0.708, 0.746\}$$

可计算其信息增益为0.262，对应划分点为0.381

对属性“含糖量”进行同样处理

与离散属性不同，若当前结点划分属性为连续属性，该属性还可作为其后代结点的划分属性

连续与缺失值 – 缺失值处理

- 不完整样本，即样本的属性值缺失
- 仅使用无缺失的样本进行学习？

对数据信息极大的浪费

- 使用有缺失值的样本，需要解决哪些问题？

Q1：如何在属性缺失的情况下进行划分属性选择？

Q2：给定划分属性,若样本在该属性上的值缺失，如何对样本进行划分？

连续与缺失值 – 缺失值处理

- $\tilde{D}$ 表示 D 中在属性 a 上没有缺失值的样本子集，$\tilde{D}^v$ 表示 $\tilde{D}$ 中在属性 a 上取值为 a^v 的样本子集，$\tilde{D}_k$ 表示 $\tilde{D}$ 中属于第 k 类的样本子集

  为每个样本 $\boldsymbol{x}$ 赋予一个权重 w_x，并定义：

  - 无缺失值样本所占的比例

$$\rho = \frac{\sum_{x \in \tilde{D}} w_x}{\sum_{x \in D} w_x}$$

  - 无缺失值样本中第 k 类所占比例

$$\tilde{p}_k = \frac{\sum_{x \in \tilde{D}_k} w_x}{\sum_{x \in \tilde{D}} w_x} \quad (1 \leq k \leq |\mathcal{Y}|)$$

  - 无缺失值样本中在属性 a 上取值 a^v 的样本所占比例

$$\tilde{r}_v = \frac{\sum_{x \in \tilde{D}^v} w_x}{\sum_{x \in \tilde{D}} w_x} \quad (1 \leq v \leq V)$$

Q1：如何在属性缺失的情况下进行划分属性选择？

连续与缺失值 – 缺失值处理

- 基于上述定义，可得

$$\begin{aligned}\mathrm{Gain}(D,a) &= \rho \times \mathrm{Gain}(\tilde{D},a) \\ &= \rho \times \left(\mathrm{Ent}(\tilde{D}) - \sum_{v=1}^{V} \tilde{r}_v \mathrm{Ent}(\tilde{D}^v)\right)\end{aligned}$$

$$\text{其中}\quad \mathrm{Ent}(\tilde{D}) = -\sum_{k=1}^{|\mathcal{Y}|} \tilde{p}_k \log_2 \tilde{p}_k$$

- 对于Q2
  - 若样本 $\boldsymbol{x}$ 在划分属性 a 上的取值已知，则将 $\boldsymbol{x}$ 划入与其取值对应的子结点，且样本权值在子结点中保持为 w_x
  - 若样本 $\boldsymbol{x}$ 在划分属性 a 上的取值未知，则将 $\boldsymbol{x}$ 同时划入所有子结点，且样本权值在与属性值 a^v 对应的子结点中调整为 $\tilde{r}_v \cdot w_x$ (直观来看，相当于让同一个样本以不同概率划入不同的子结点中去)

连续与缺失值 – 缺失值处理

缺失值处理实例

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	–	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	–	是
3	乌黑	蜷缩	–	清晰	凹陷	硬滑	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	–	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	–	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
8	乌黑	稍蜷	浊响	–	稍凹	硬滑	是
9	乌黑	–	沉闷	稍糊	稍凹	硬滑	否
10	青绿	硬挺	清脆	–	平坦	软粘	否
11	浅白	硬挺	清脆	模糊	平坦	–	否
12	浅白	蜷缩	–	模糊	平坦	软粘	否
13	–	稍蜷	浊响	稍糊	凹陷	硬滑	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	–	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	–	沉闷	稍糊	稍凹	硬滑	否

- 学习开始时，根结点包含样本集 D 中全部17个样例，各样例的权值均为1；
- 以属性“色泽”为例，该属性上无缺失值的样例子集 $\tilde{D}$ 包含14个样例，$\tilde{D}$ 的信息熵为

$$\mathrm{Ent}(\tilde{D}) = -\sum_{k=1}^{2} \tilde{p}_k \log_2 \tilde{p}_k$$

$$= -\left(\frac{6}{14}\log_2\frac{6}{14} + \frac{8}{14}\log_2\frac{8}{14}\right) = 0.985$$

连续与缺失值 – 缺失值处理

- 令 $\tilde{D}^1$, $\tilde{D}^2$, $\tilde{D}^3$ 分别表示在属性“色泽”上取值为“青绿”“乌黑”以及“浅白”的样本子集，有

$$\mathrm{Ent}(\tilde{D}^1) = -\left(\tfrac{2}{4}\log_2\tfrac{2}{4} + \tfrac{2}{4}\log_2\tfrac{2}{4}\right) = 1.000 \quad \mathrm{Ent}(\tilde{D}^2) = -\left(\tfrac{4}{6}\log_2\tfrac{4}{6} + \tfrac{2}{6}\log_2\tfrac{2}{6}\right) = 0.918$$

$$\mathrm{Ent}(\tilde{D}^3) = -\left(\tfrac{0}{4}\log_2\tfrac{0}{4} + \tfrac{4}{4}\log_2\tfrac{4}{4}\right) = 0.000$$

- 因此，样本子集 $\tilde{D}$ 上属性“色泽”的信息增益为

$$\begin{aligned}\mathrm{Gain}(\tilde{D}, \text{色泽}) &= \mathrm{Ent}(\tilde{D}) - \sum_{v=1}^{3} \tilde{r}_v \mathrm{Ent}(\tilde{D}^v) \\ &= 0.985 - \left(\tfrac{4}{14} \times 1.000 + \tfrac{6}{14} \times 0.918 + \tfrac{4}{14} \times 0.000\right) \\ &= 0.306\end{aligned}$$

- 于是，样本集 D 上属性“色泽”的信息增益为

$$\mathrm{Gain}(D, \text{色泽}) = \rho \times \mathrm{Gain}(\tilde{D}, \text{色泽}) = \tfrac{14}{17} \times 0.306 = 0.252$$

连续与缺失值 – 缺失值处理

- 类似地可计算出所有属性在数据集上的信息增益

$$\text{Gain}(D, \text{色泽}) = 0.252 \qquad \text{Gain}(D, \text{根蒂}) = 0.171$$

$$\text{Gain}(D, \text{敲声}) = 0.145 \qquad \text{Gain}(D, \text{纹理}) = 0.424$$

$$\text{Gain}(D, \text{脐部}) = 0.289 \qquad \text{Gain}(D, \text{触感}) = 0.006$$

进入“纹理=清晰”分支

进入“纹理=稍糊”分支

进入“纹理=模糊”分支

样本权重在各子结点仍为1

在属性“纹理”上出现缺失值，样本8和10同时进入3个分支，调整8和10在3分支权值分别为7/15，5/15，3/15

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	–	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	–	是
3	乌黑	蜷缩	–	清晰	凹陷	硬滑	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	–	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	–	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
8	乌黑	稍蜷	浊响	–	稍凹	硬滑	是
9	乌黑	–	沉闷	稍糊	稍凹	硬滑	否
10	青绿	硬挺	清脆	–	平坦	软粘	否
11	浅白	硬挺	清脆	模糊	平坦	–	否
12	浅白	蜷缩	–	模糊	平坦	软粘	否
13	–	稍蜷	浊响	稍糊	凹陷	硬滑	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	–	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	–	沉闷	稍糊	稍凹	硬滑	否

总结

1）决策树采用的是一种简单且直观的“分而治之”策略

2）决策树根据数据集的总体纯度来衡量不同属性划分的优劣

3）当属性选择过于频繁会导致模型过拟合，需要使用剪枝算法来进行处理

4）剪枝算法的判断标准是剪枝前后模型的性能指标是否提高

5）当遇到连续值的属性时，可以将其拆解成两块将数据转换为连续值

6）有缺失值属性时，通过在属性选择算法中加入缺失值数据的影响来适应

成熟的决策树软件包

- ID3,C4.5,C5.0
  http://www.rulequest.com/Personal/

思考题

1.简述决策树的原理

2.简述决策树的构建过程

3.信息增益率有什么优缺点

4.如何对决策树进行剪枝

5.为什么决策树需要进行剪枝

6.决策树对缺失值如何处理

7.如果决策树属性用完了仍未对决策树完成划分应该怎么办

8.如何避免决策树的过拟合

9.常用的决策树一定是二叉树吗

10.你认为在一棵决策树构建过程中较为耗时的步骤是什么

11.在选择特征进行分类时一个特征被选择过后，之后还会选择到这个特征吗

12.和其他模型比，决策树有哪些优点和缺点

Thank you!

权小军 中山大学数据科学与计算机学院