机器学习

决策树

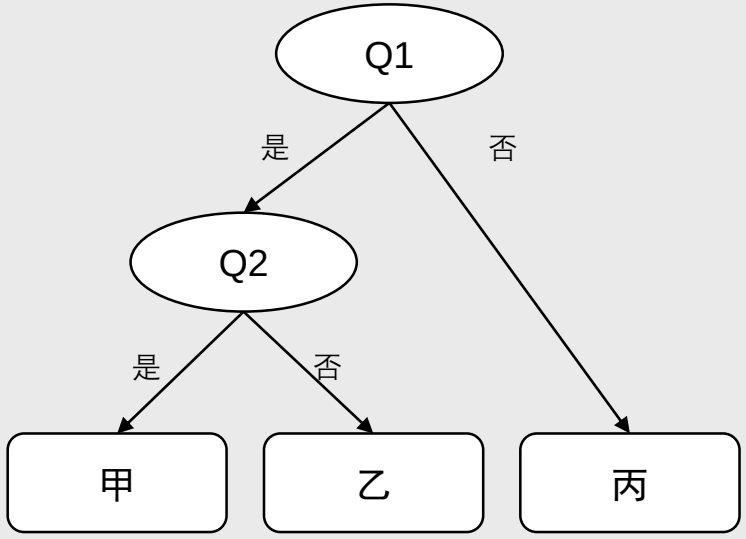
课程目录

Course catalogue

本课目标

- 学习决策树原理
- 学习不同决策树的python解决方案
- 学习使用决策树剪枝解决过拟合问题
- 学习决策树对连续值和缺失值的处理
- 学习使用python实现格搜索优化超参数

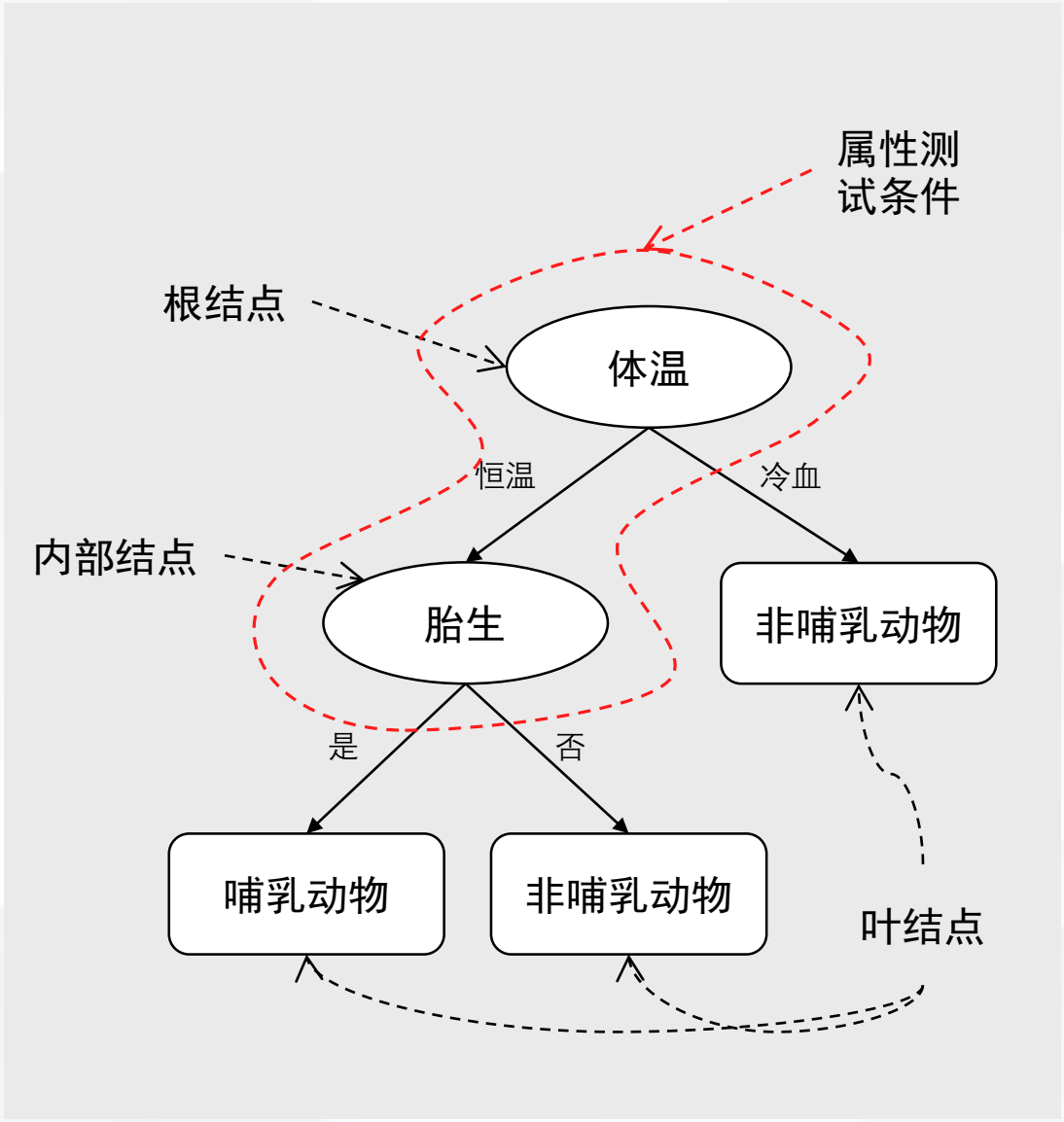
决策树原理



读心术-20个问题看透你的心

- 这个人是女的吗？ 不是
- 他是虚拟人物吗？是
- 他是学生吗？是
- 他出生在日本吗？不是
- 他是小说中的角色吗？不是
- 他属于人类吧？不是
- 他属于动物吗？是
- 他头上有角吗？是
- 他是个吃货？不是
- 他皮肤颜色深吗？不是
- 他是羊吗？是
- 他很有智慧吗？是

决策树原理



决策树结构

根据下列数据，构建决策树

动物名称	体温	胎生	哺乳动物
蛇	冷血	否	否
鸟	恒温	否	否
猩猩	恒温	是	是

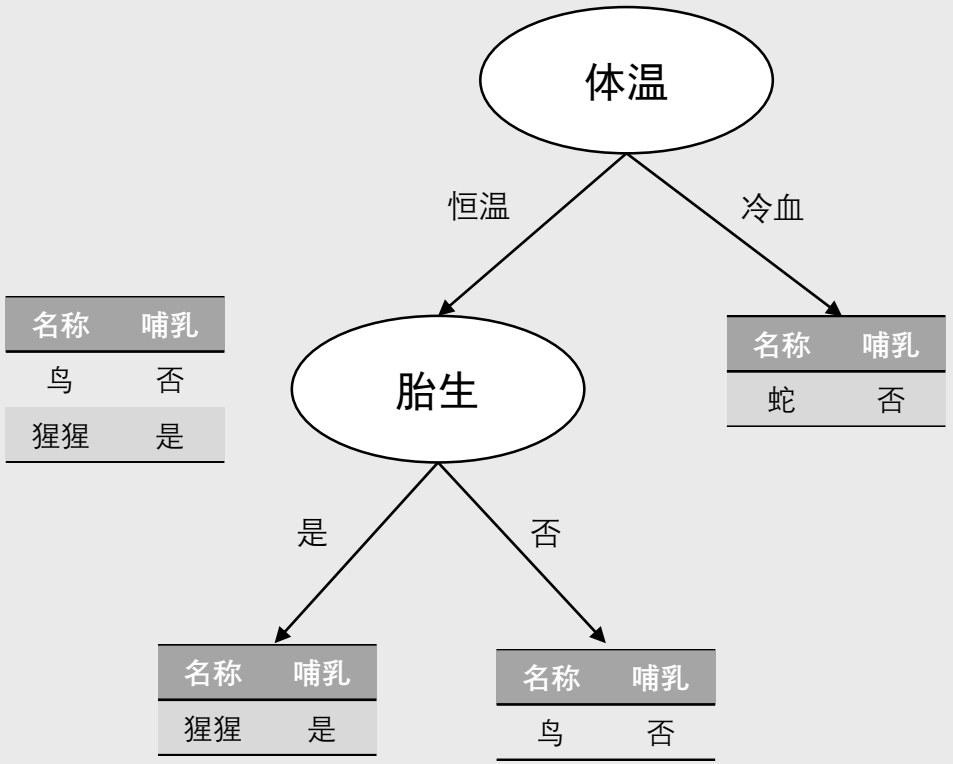
决策树是一种由结点和有向边组成的层次结构。由根结点、内部结点、叶结点组成：

- 根结点，他没有入边，但有零条或多条出边
- 内部结点，有一条入边和两条或多条出边
- 叶结点，有一条入边，但没有出边
- 每个非叶结点包含一个属性测试条件

决策树原理

【思考】给定属性集，可能的决策树有多少？

动物名称	体温	胎生	哺乳动物
蛇	冷血	否	否
鸟	恒温	否	否
猩猩	恒温	是	是



如何建立决策树

Hunt算法是许多决策树算法的基础，包括ID3、C4.5、CART等。Hunt算法通过将训练记录相继划分成较纯的子集，以递归的方式建立决策树。Hunt算法如下：

设D_t是节点t对应的训练集，而$y = \{y_1, y_2, \dots, y_c\}$是类别标号

1、如果D_t中所有记录都属于同一个类别y_t，则t是叶节点，用y_t标记。

2、如果D_t中包含属于多个类的记录，则选择一个属性测试条件，将记录划分成较小的子集。对于测试条件的每个输出，创建一个子女结点，并根据测试结果将D_t中的记录分布到子女结点中。然后，对于每个子女结点，递归地调用该算法。

【思考】该算法流程中的核心问题是什么？

决策树原理

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**
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**
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**
13: **else**
14: 以 TreeGenerate(D_v, $A \setminus \{a_*\}$)为分支结点
15: **end if**
16: **end for**
输出: 以 node 为根结点的一棵决策树

如何建立决策树

Hunt算法伪代码如左图所示。

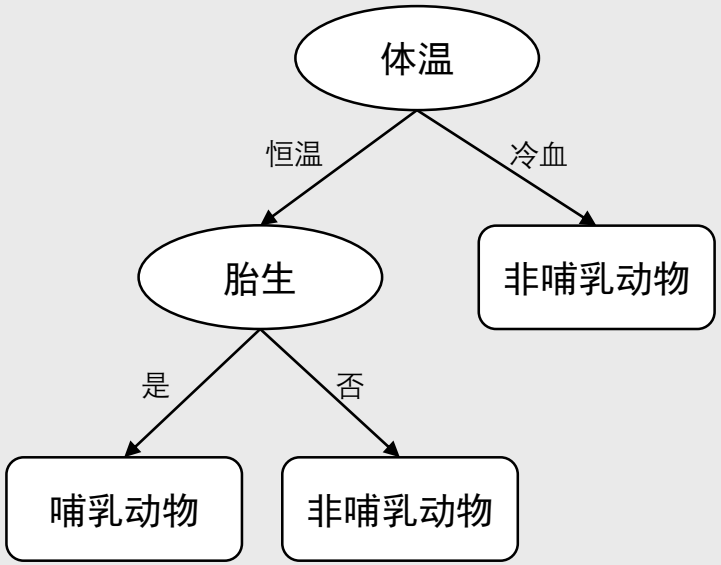
以下三种情况会导致递归返回，即停止分裂：

(1)当前结点包含的样本全部为同一类别，无需划分

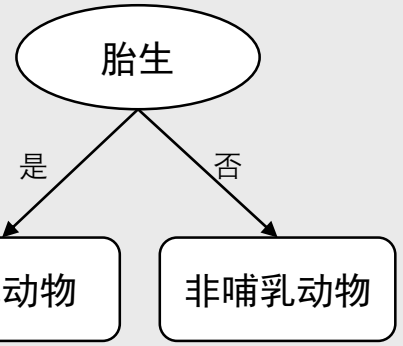
(2)当前属性集为空，或是所有样本在所有属性上取值相同，无法划分

(3)当前结点包含的样本集合为空，不能划分

决策树原理



1



2

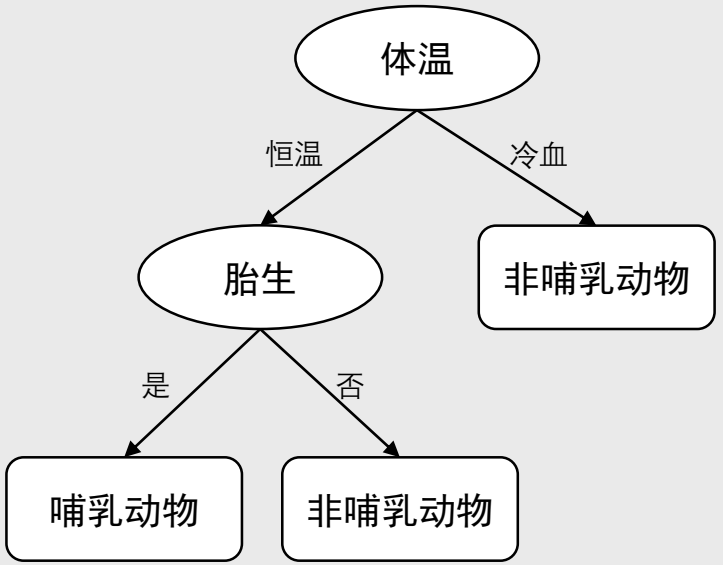
核心问题

选择一个属性测试条件，将记录划分成较小的子集

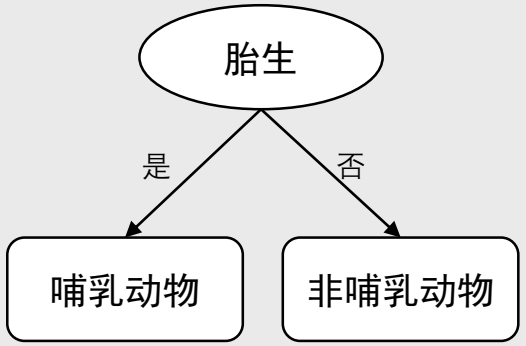
【问题】对如下数据，左图中的两棵决策树哪颗更好？为什么？

动物名称	体温	胎生	哺乳动物
蛇	冷血	否	否
鸟	恒温	否	否
猩猩	恒温	是	是

决策树原理



1



2

核心问题

选择一个属性测试条件，将记录划分成较小的子集

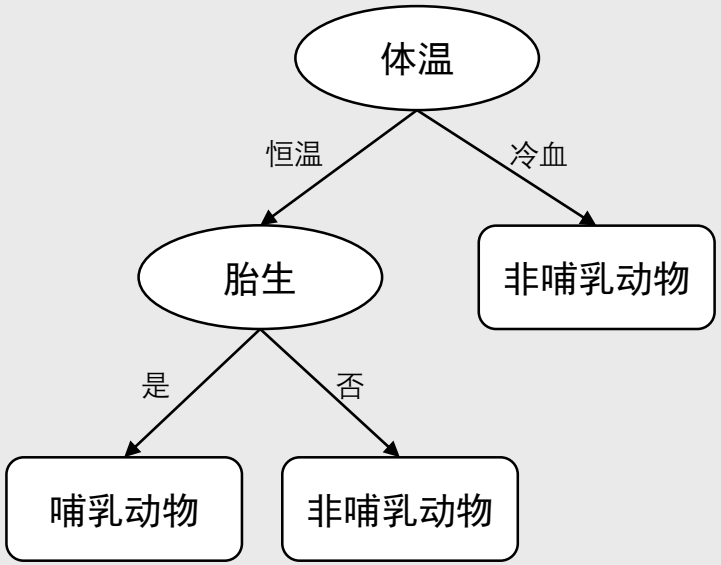
【问题】对如下数据，左图中的两棵决策树哪颗更好？为什么？

动物名称	体温	胎生	哺乳动物
蛇	冷血	否	否
鸟	恒温	否	否
猩猩	恒温	是	是

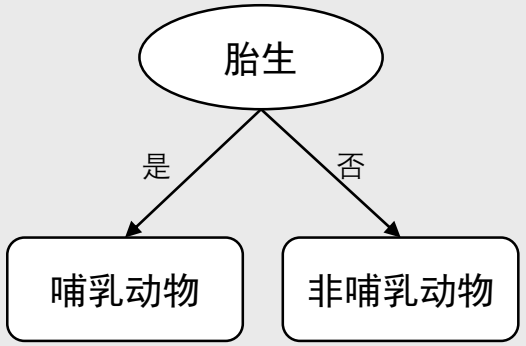
第二棵树更“短”，这意味着算法更快

【思考】更短的原因是什么？

决策树原理



1



2

核心问题

选择一个属性测试条件，将记录划分成较小的子集

【问题】对如下数据，左图中的两棵决策树哪颗更好？为什么？

动物名称	体温	胎生	哺乳动物
蛇	冷血	否	否
鸟	恒温	否	否
猩猩	恒温	是	是

第二棵树更“短”，这意味着算法更快

【思考】更短的原因是什么？

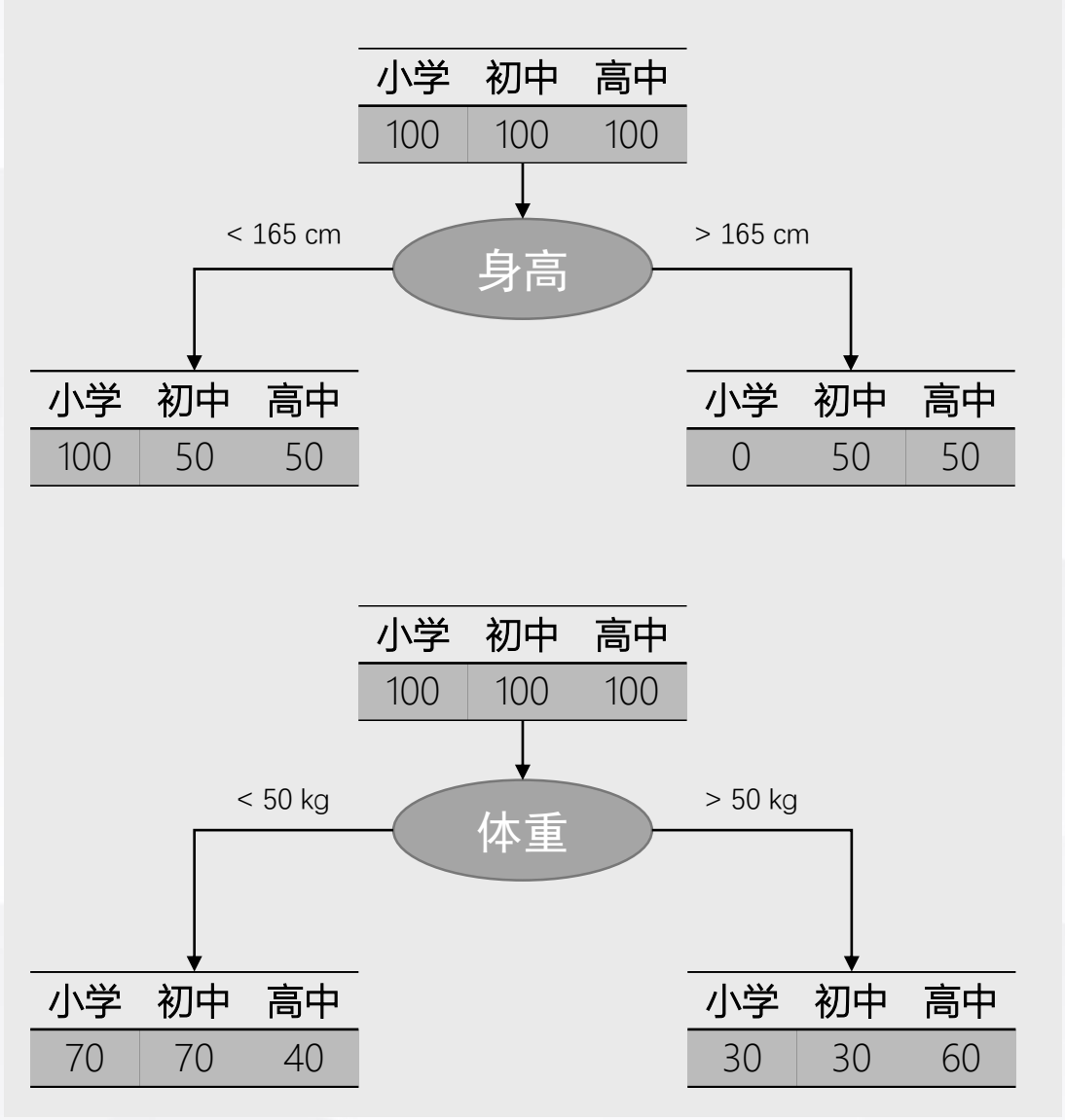
属性测试条件划分的子集更“纯”

“纯度”是区分不同决策树的重要依据

课程目录

Course catalogue

最优划分属性



纯度（不纯度）

某数据集包含小学、初中、高中学生各100条样本，属性包括学生的身高和体重，依据属性建立决策树分类器，预测新样本的标签。

【问题】左图中两棵树中，哪棵树更好？为什么？

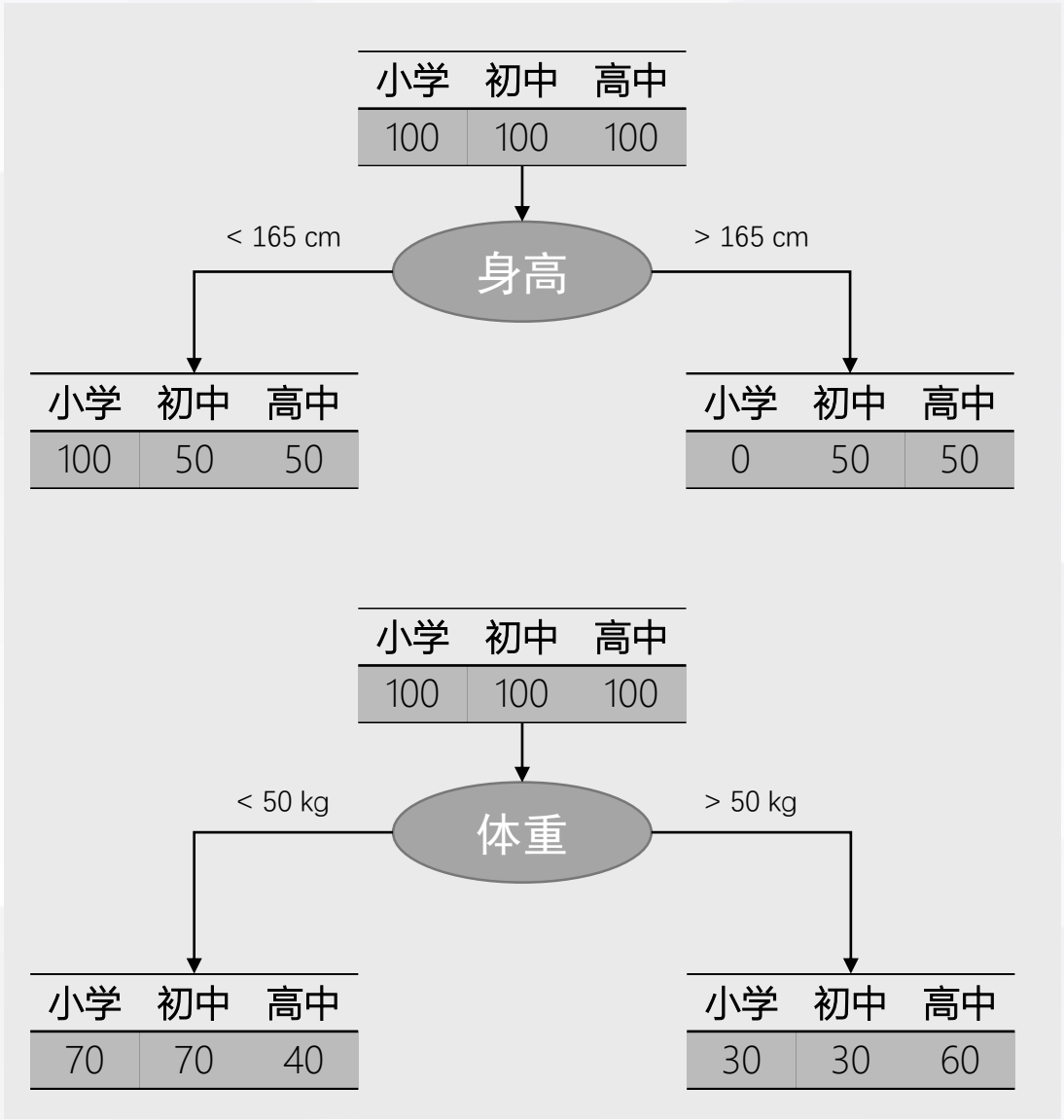
为了解决这个问题，需要引入纯度（不纯度）度量指标。常用度量指标包括：{**信息增益**（information gain），**增益率**（gain ratio），**基尼指数**（Gini index）}

分别对应了三种决策树算法：

{**信息增益**：ID3，**增益率**：C4.5，**基尼指数**：CART}

显然，哪棵树更"纯"，首先选择哪个属性测试条件。

最优划分属性



信息熵、信息增益与ID3

学习信息增益，首先要了解什么是信息熵，假定当前样本集合D中第k类样本所占的比例为$p_k(k = 1,2,\dots,|Y|)$，则D的信息熵定义为：

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

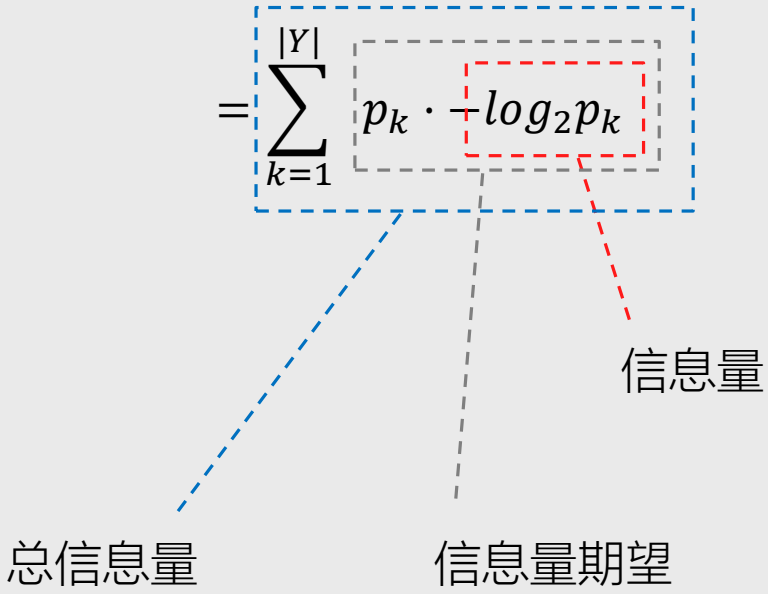
信息熵越小，则D的纯度越高，信息熵越大，包含的信息越大，不纯度越高。

【思考】信息熵为什么这么定义？

最优划分属性

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

$$= \sum_{k=1}^{|Y|} p_k \cdot -log_2 p_k$$



信息熵、信息增益与ID3

熵在信息论中代表随机变量不确定度的度量，简单地说，信息熵就是衡量信息量的度量。

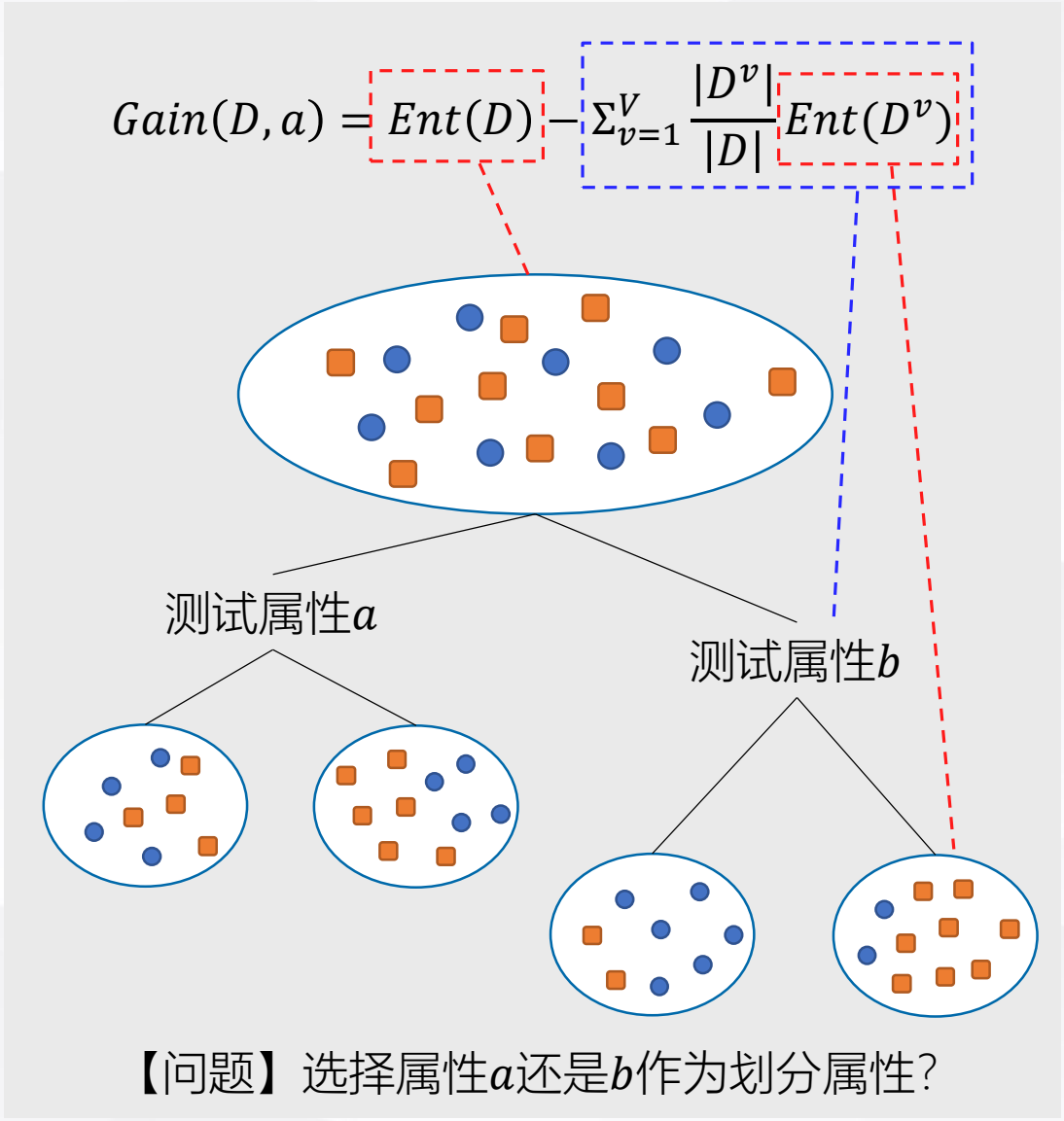
【思考】想一想，信息量用数学形式怎么表示？

信息量的三个性质：

- 非负性，一个事件包含的信息不能是负的
- 单调性，概率越大，越确定的事件，信息越少
- 可加性，独立随机事件包含的信息可加

现在再来看看信息熵的定义，有没有觉得很漂亮？

最优划分属性



信息熵、信息增益与ID3

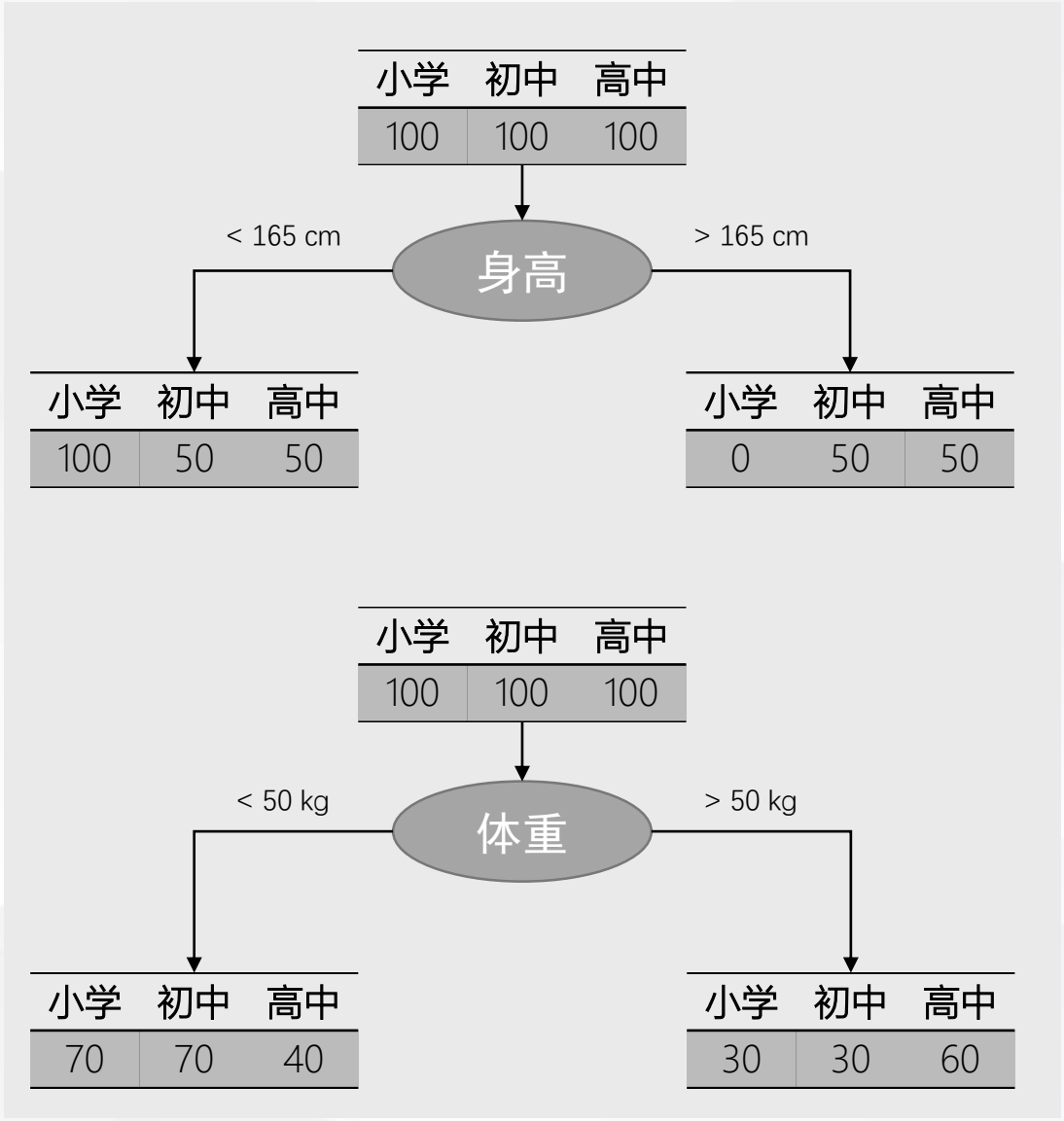
假定离散属性a有V个可能的取值$\{a^1, a^2, \ldots, a^V\}$，若使用$a$来对样本集$D$进行划分，则会产生$V$个分支结点，其中第$v$个分支结点包含了$D$中所有在属性$a$上取值为$a^v$的样本，记为$D^v$。则属性$a$对样本集$D$进行划分所得的"信息增益"为：

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

通常来说，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。

著名的ID3(Iterative Dichotomiser)决策树算法，就是以信息增益为准则来选择划分标准。

最优划分属性



信息熵、信息增益与ID3

【问题】计算**体重**的信息增益

$$Ent(D) = -\sum_{k=1}^{3} \frac{1}{3} \log_2 \frac{1}{3} = 1.58$$

$$Ent(D^{<50}) = -\left(\frac{70}{180}\log_2\frac{70}{180} + \frac{70}{180}\log_2\frac{70}{180} + \frac{40}{180}\log_2\frac{40}{180}\right) = 1.54$$

$$Ent(D^{>50}) = -\left(\frac{30}{120}\log_2\frac{30}{120} + \frac{30}{120}\log_2\frac{30}{120} + \frac{60}{120}\log_2\frac{60}{120}\right) = 1.5$$

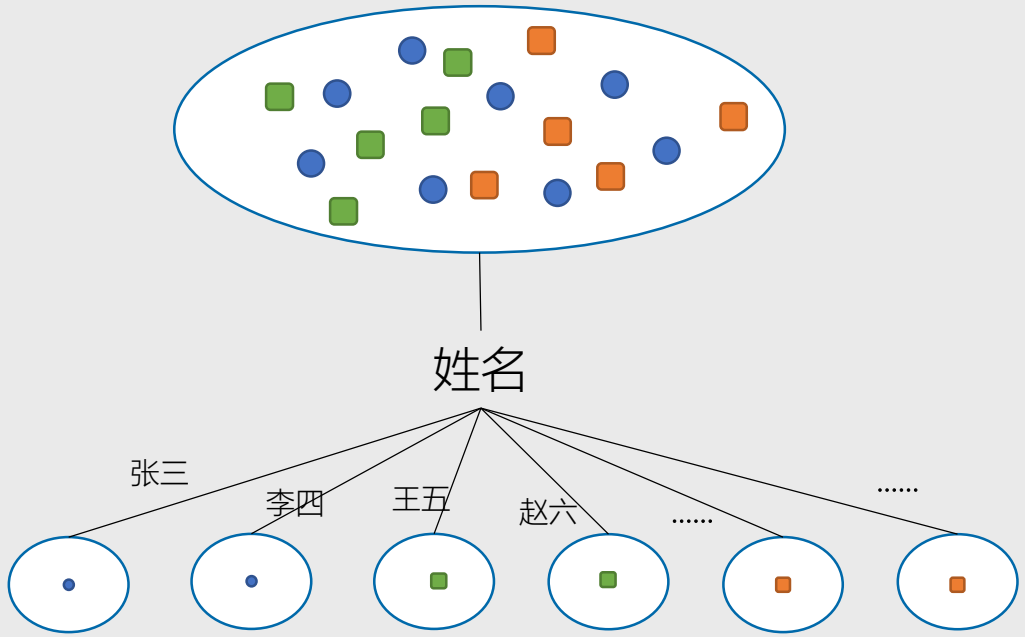
$$Gain(D,w) = Ent(D) - \Sigma_{v=1}^{V}\frac{|D^v|}{|D|}Ent(D^v)$$

$$Gain(D,w) = 1.58 - \left(\frac{180}{300}1.54 + \frac{120}{300}1.5\right) = 0.056$$

【问题】尝试计算身高的信息增益，看看会遇到什么问题，比较结果并说明谁是最优划分属性。

最优划分属性

姓名	身高	体重	类别
张三	…	…	小学
李四	…	…	初中
王五	…	…	高中
赵六	…	…	小学
…	…	…	…



增益率与C4.5

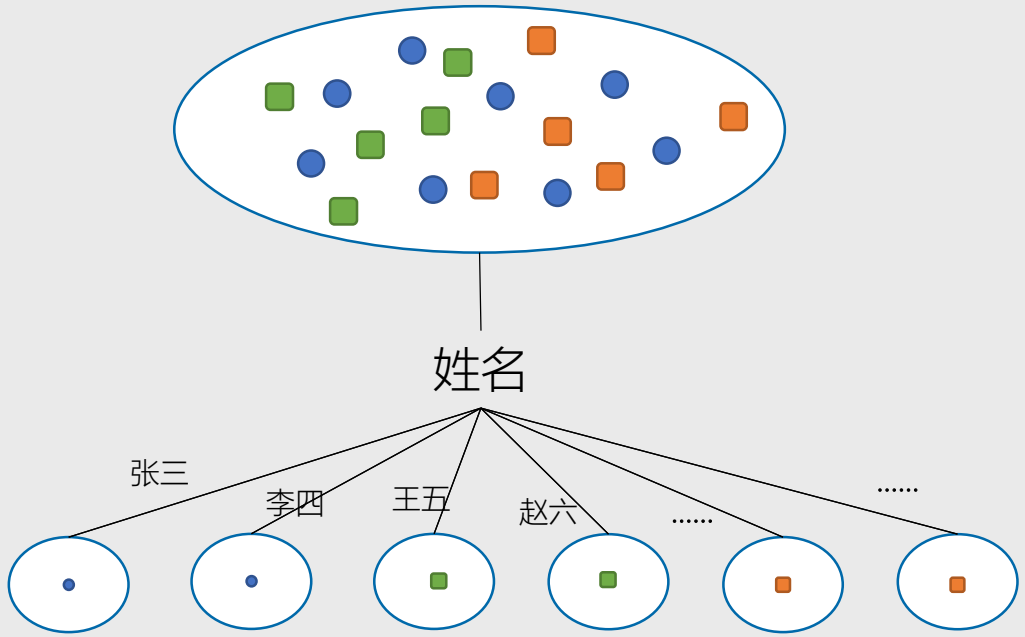
【思考】在ID3数据中，我们使用了身高、体重作为属性，考虑如果数据包含"姓名"属性，并作为候选划分属性，会出现什么结果？

姓名的信息增益远大于身高和体重，因为姓名产生了300个分支结点，每个分支结点仅包含一个样本，纯度以达到最大。显然，这样的决策树不具备泛化能力，无法对新样本进行有效预测。

C4.5算法使用"增益率（gain ratio）"来减少这种不利影响。

最优划分属性

姓名	身高	体重	类别
张三	…	…	小学
李四	…	…	初中
王五	…	…	高中
赵六	…	…	小学
…	…	…	…



增益率与C4.5

增益率定义为：

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

其中：

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

称为属性a的"固有值"。属性a的可能取值数目越多，即V越大，则$IV(a)$的值通常会越大。

C4.5算法不是直接选择增益率最大的属性作为最优划分属性，而是先从候选划分属性中，找出信息增益高于平均水平的属性，再从中选择增益率最高的属性。

最优划分属性

基尼指数与CART

除了信息熵，还可以从概率的角度去衡量“不纯度”。

【思考】从一筐水果中随机抽取两个水果，其水果种类不一致的概率是多少？

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

CART决策树使用“基尼指数”（Gini index）来选择划分属性，$Gini(D)$越小，则数据集D的纯度越高。

时间顺序上，1984年提出的CART，1986年提出的ID3，1993年提出的C4.5。

最优划分属性

```
import pandas as pd
df = pd.read_csv('datas/iris.csv',index_col=0)
X = df.iloc[:,:-1]
y = df.iloc[:,-1]
df.sample(3)
```

	sepal_l	sepal_w	petal_l	petal_w	classes
145	6.7	3.0	5.2	2.3	2.0
33	5.5	4.2	1.4	0.2	0.0
57	4.9	2.4	3.3	1.0	1.0

```
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier().fit(X,y)
y_pred = model.predict(df.iloc[:,:-1])
```

```
from sklearn.model_selection import cross_val_score
cross_val_score(model,X,y,scoring='accuracy',cv=5).mean()
```

```
0.9666666666666668
```

决策树的python解决方案

读取数据

pandas.read_csv()

载入决策树

sklearn.tree.DecisionTreeClassifier

参数：criterion:{‘gini’, ‘entropy’}分别对应基尼系数和信息增益，即CART和ID3决策树

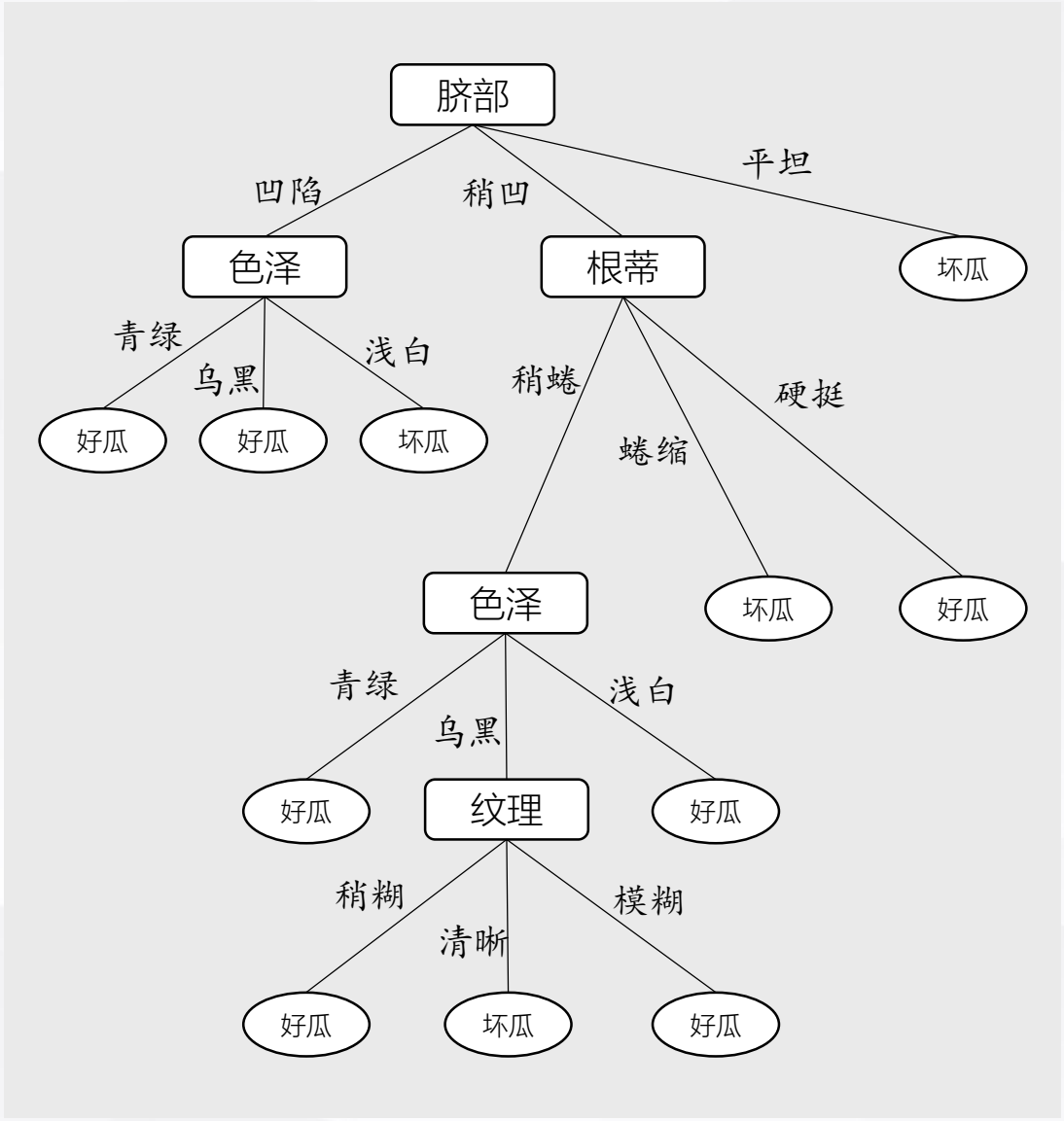
模型评估

sklearn.model_selection.cross_val_score

课程目录

Course catalogue

过拟合与剪枝



周志华的西瓜

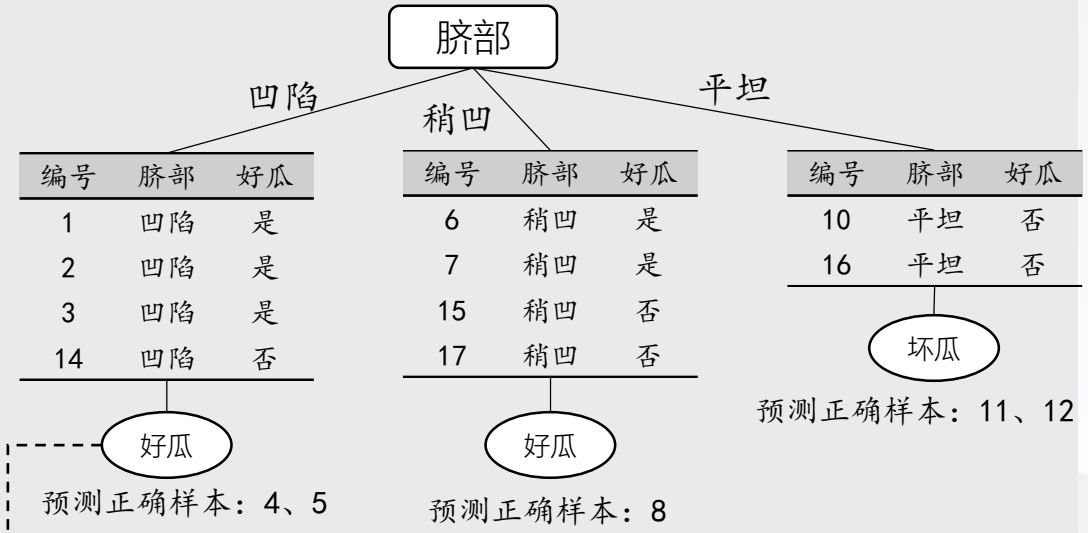
西瓜数据集2.0的训练集（上）和测试集（下）

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

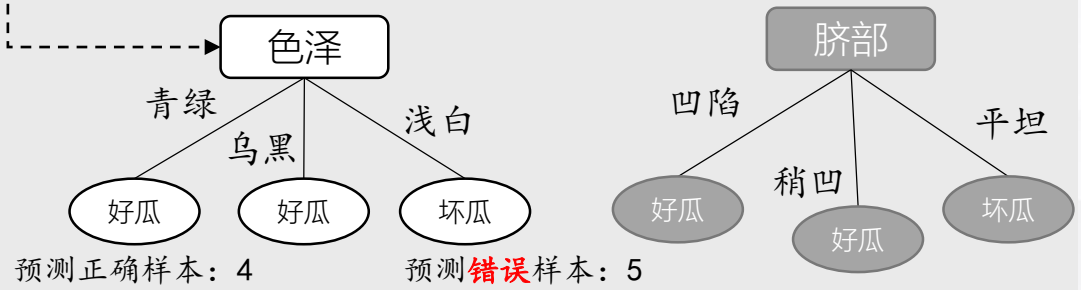
【思考】训练集准确率100%的决策树好不好？

过拟合与剪枝

不进行划分，验证集精度=$\frac{3}{7}\times 100\% = 42.9\%$



“脐部”划分后，验证集精度=$\frac{5}{7}\times 100\% = 71.4\% > 42.9\%$，结点**保留**。



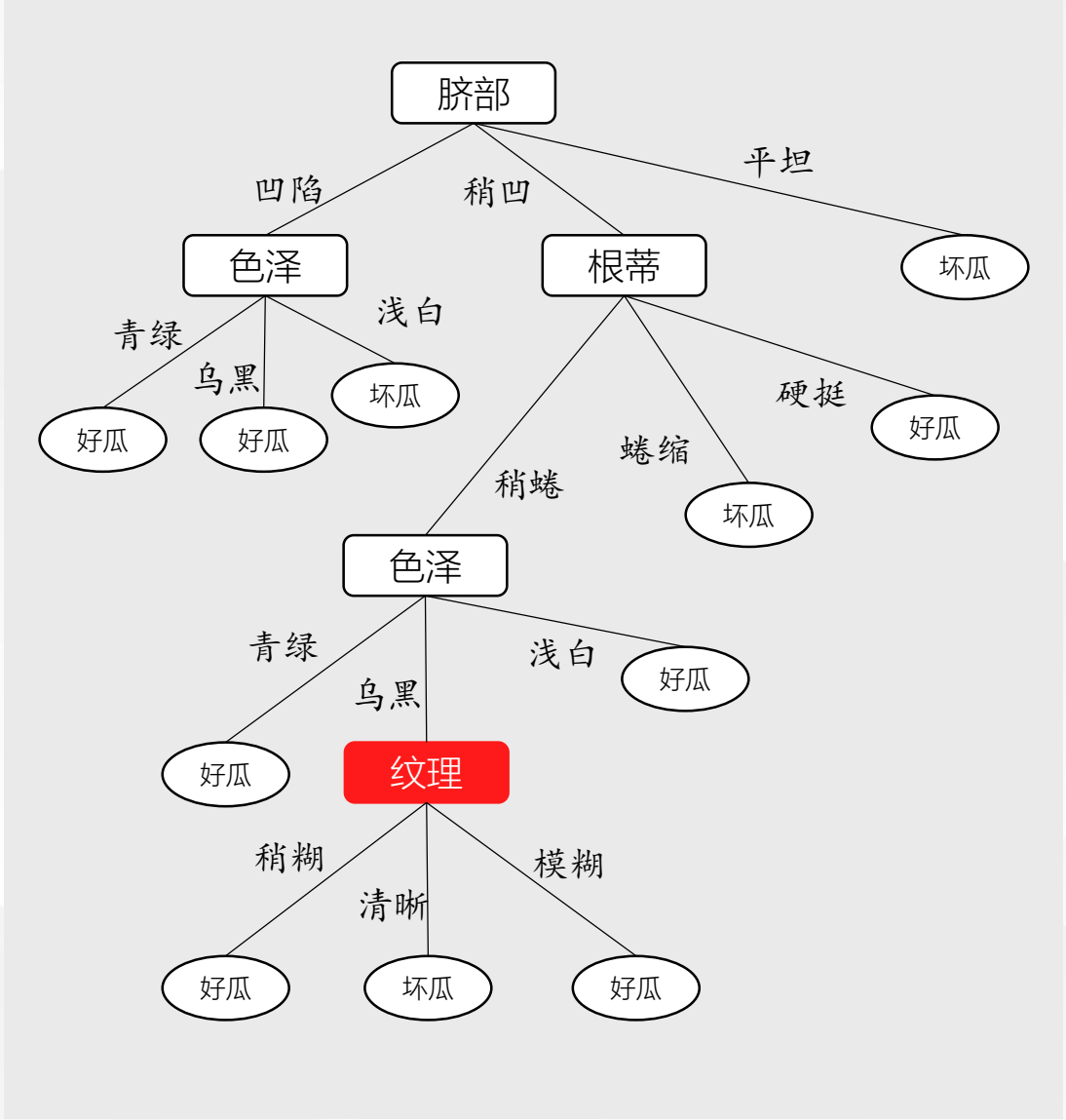
“色泽”划分后，验证集精度=$\frac{4}{7}\times 100\% = 57.1\% < 71.4\%$，结点**剪除**。

周志华的西瓜与预剪枝

预剪枝：一边生成决策树，一边比较划分前后的泛化性能

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

过拟合与剪枝



周志华的西瓜与后剪枝

后剪枝：先生成一棵完整的决策树，然后依次从底部判断是否剪除结点

容易计算完整决策树的验证集精度为42.9%。

考察结点“纹理”，剪除该结点后精度提升至57.1%，因此决定剪除该结点。

同理依次对其他结点进行判断，最终得到后剪枝决策树。

课程目录

Course catalogue

连续值、缺失值处理

连续属性a的候选划分点集合为：

$$T_a = \{\frac{a^i + a^{i+1}}{2}\}$$

信息增益可以改造为：

$$Gain(D, a) = \max_{t \in T_a} Gain(D, a, t)$$

$$= \max_{t \in T_a} Ent(D) - \sum_{\lambda \in \{-,+\}} \frac{|D_t^\lambda|}{|D|} Ent(D_t^\lambda)$$

连续值处理

决策树中处理连续属性，需要对连续属性进行离散化。通常来说，样本数量是有限的，因此尽管连续变量取值是连续的，但其样本值是分布在数轴上的点，因此也可以看作是“离散”的。

连续值处理，最简单的策略就是采用二分法进行处理。

给定样本集D和连续属性a，假定a在D上出现了n个不同的取值，从小到大排序，分别记为$\{a^1, a^2, \ldots, a^n\}$。基于划分点$t$将$D$分为子集$D_t^-$和$D_t^+$。

连续值、缺失值处理

推广信息增益：

$$Gain(D,a) = \rho\, Gain(\tilde{D},a)$$

【思考】为什么不是

$$Gain(D,a) = \frac{Gain(\tilde{D},a)}{\rho}$$

缺失值处理

如果数据集足够大，可以假定缺失值分布和非缺失值分布一致，因此缺失值的信息，可以计算非缺失值的数据信息，然后按照缺失值比例进行加权。

给定训练集D和属性a，令$\tilde{D}$表示D中在属性a上没有缺失值的样本子集，假定我们为每个样本$\boldsymbol{x}$赋予一个权重$w_{\boldsymbol{x}}$，并定义：

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

直观地看，对属性a，ρ表示无缺失值样本所占的比例。基于此，信息增益计算方式推广为左图。

课程目录

Course catalogue

超参数与格搜索

python中决策树的超参数：

```
# 分裂准则
criterion='gini'
# 最大树深度
max_depth=None
# 每次分裂所需最小样本数量
min_samples_split=2
# 叶结点所需包含的最小样本数量
min_samples_leaf=1
```

超参数

不同于参数（$\boldsymbol{w}, b$），超参数不是在分类估计器中直接学习的参数，而是传递给估计器的，用于构造分类估计器的参数。

例如指定损失函数中正则项为l_1还是l_2，正则项系数λ（python对应参数C），梯度下降中学习率α等等。

显然，不同的超参数，对模型有巨大的影响。

【思考】如何选择恰当的的超参数?

超参数与格搜索

```
# 读取数据
import pandas as pd
df = pd.read_csv('datas/iris.csv',index_col=0)
df.head(2)
```

	sepal_l	sepal_w	petal_l	petal_w	classes
0	5.1	3.5	1.4	0.2	0.0
1	4.9	3.0	1.4	0.2	0.0

```
# 载入决策树及格搜索库，并设置超参数空间
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import GridSearchCV
params = {'criterion':['gini','entropy'],
          'max_depth':[5,10,None],
          'min_samples_split':list(range(2,5,1)),
          'min_samples_leaf':list(range(1,4,1))}
```

```
# 建立决策树模型，并使用格搜索寻找恰当的超参数组合
dtc = DecisionTreeClassifier()
gs = GridSearchCV(dtc,param_grid=params,cv=5,scoring='accuracy',n_jobs=1,verbose=0)
gs.fit(df.iloc[:,:-1],df.iloc[:,-1])
gs.best_params_,gs.best_score_
```

```
({'criterion': 'gini',
  'max_depth': 5,
  'min_samples_leaf': 1,
  'min_samples_split': 2},
 0.9666666666666667)
```

```
# 使用“最优”模型进行预测
gs.best_estimator_.predict(df.iloc[:,:-1])[:5]
```

```
array([0., 0., 0., 0., 0.])
```

格搜索 Grid Search

格搜索通过遍历指定的超参数空间组合，根据指定的评估参数，来找到“恰当”的超参数组合。

【问题】对于如下超参数空间，使用格搜索找到最“恰当”的超参数组合。

{criterion:['gini','entropy'],

max_depth:[5,10,None],

min_samples_split:list(range(2,5,1)),

min_samples_leaf:list(range(1,4,1))

}