第5章　决策树

刘家锋

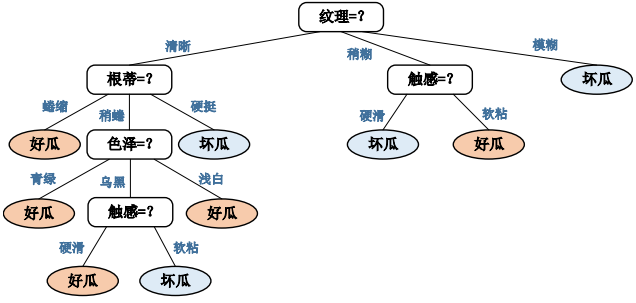
哈尔滨工业大学

第5章　决策树

5.1 基本流程

决策树

- **决策树的结构**
  - 决策树是一种常用的机器学习方法
  - 中间节点：对应某个属性，分支对应该属性的某个取值
  - 叶节点：对应一个预测结果

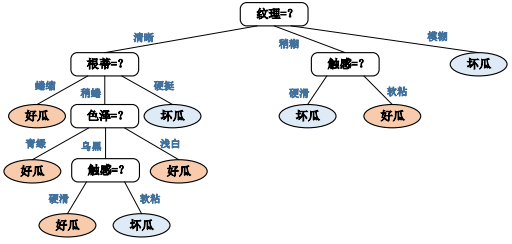


决策树

- **决策树预测**
  - 从根节点开始，测试节点的属性
  - 根据测试样例的属性值，决定下行分支
  - 在叶节点得到预测值
- **决策树学习**
  - 通过对训练集数据的分析，确定节点的划分属性，确定节点的分支，构造决策树
  - 修剪决策树，提高预测的泛化能力

决策树与规则

- **从决策树到规则**
  - 一颗决策树对应一个规则集
  - 每条从根节点到叶节点的分支路径对应一条规则



(纹理 = 清晰) ∧ (根蒂 = 蜷缩) ⟶ 好瓜

· · · · · ·

(纹理 = 清晰) ∧ (根蒂 = 硬挺) ⟶ 坏瓜

(纹理 = 稍糊) ∧ (触感 = 硬滑) ⟶ 坏瓜

(纹理 = 稍糊) ∧ (触感 = 软黏) ⟶ 好瓜

(纹理 = 模糊) ⟶ 坏瓜

5.2 决策树学习

基本算法

Algorithm 1 决策树学习算法

Input: 数据集$D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$，属性集$A = \{a_1, \cdots, a_d\}$
Output: 以node为根节点的一棵决策树

```
function TreeGenerate(D, A)
    生成节点node;
    if 满足递归停止条件 then 标记节点；return
    end if
    从A中选择最优划分属性a_*
    for a_*的每一个取值a_*^v do
        为node生成一个分支，D_v表示D中a_*取值a_*^v的样本子集
        if D_v = Φ then
            分支标记叶节点为D中样本最多类
        else
            以TreeGenerate(D_v, A \ {a_*})为分支节点
        end if
    end for
end function
```

分治递归

- **决策树的学习过程**
  - 决策树学习“分而治之”的策略，是一个从根节点到叶节点的迭代过程
  - 每一次递归，根据选择的属性值将样本集划分为若干个子集，每个子集对应节点的不同分支
  - 每个子集递归调用建树的过程
- **递归终止条件**
  1. 输入样本集合D中的样本都属于同一个类别C，标记节点为叶节点，类别为C
  2. 输入属性集A为空，或者集合D中所有样本的所有属性相同，标记节点为D中样本所属最多的类别
  3. 当前节点包含的样本集合为空，标记节点为D中样本最多的类别

划分属性

- **划分属性的选择**
  - 决策树学习的关键是如何选择“最优”的划分属性
  - 决策树根节点对应的样本集合包含所有类别的样本，而每一个叶节点只包含某一个类别的样本
  - 决策树的构建可以看作是一个使得样本集合越来越“纯净”的过程
  - 如果能够定义一个度量集合“纯度”的指标，在每个节点上应该选择使得“纯度”增加最快的属性来划分样本集合

Information Entropy

- **信息熵**
  - 信息熵是度量样本集合纯度最常用的一个指标
  - 样本集合D中第k类样本所占比例为p_k，D的信息熵定义为：

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

    其中，$\mathcal{Y}$为类别的集合，$|\mathcal{Y}|$为类别的数量

  - $\mathsf{Ent}(D)$的值越小，表示纯度越高，当所有样本属于一个类别时，取得最小值$\mathsf{Ent}(D) = 0$；(约定$p = 0$，$p \log_2 p = 0$)
  - $\mathsf{Ent}(D)$的最大值为$\log_2 |\mathcal{Y}|$

ID3算法

- **信息增益**
  - 离散属性a有V个可能的取值：$\{a^1, \cdots, a^V\}$
  - D^v表示D中属性$a = a^v$的样本集合，以属性a对数据集D进行划分的信息增益为：

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

  - 第1项$\mathsf{Ent}(D)$是划分前数据集的信息熵
  - 第2项是使用属性a划分之后的总信息熵，$|D^v|/|D|$是每个分支的权重，样本越多权重越大，$\mathsf{Ent}(D^v)$为其中一个子集的信息熵
  - ID3算法依据信息增益选择最优的划分属性：

$$a_* = \arg\max_{a \in A} \mathsf{Gain}(D, a)$$

例5.1 西瓜分类

包含17个样例的西瓜数据集

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
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
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
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
10	青绿	硬挺	清脆	清晰	平坦	软粘	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
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

计算信息熵

计算数据集的信息熵：$|\mathcal{Y}| = 2$，正例占比$p_1 = 8/17$，反例占比$p_2 = 9/17$

$$\mathsf{Ent}(D) = -\sum_{k=1}^{2} p_k \log_2 p_k = -\left(\frac{8}{17}\log_2\frac{8}{17} + \frac{9}{17}\log_2\frac{9}{17}\right) = 0.998$$

以属性“色泽”划分为3个子集：

$$\begin{aligned}
D^1(\text{青绿}) &= \{1,4,6,10,13,17\}, \qquad p_1 = 3/6, p_2 = 3/6 \\
D^2(\text{乌黑}) &= \{2,4,7,8,9,15\}, \qquad p_1 = 4/6, p_2 = 2/6 \\
D^3(\text{浅白}) &= \{5,11,12,14,16\}, \qquad p_1 = 1/5, p_2 = 4/5
\end{aligned}$$

计算子集的信息熵：

$$\begin{aligned}
\mathsf{Ent}(D^1) &= -\left(\frac{3}{6}\log_2\frac{3}{6} + \frac{3}{6}\log_2\frac{3}{6}\right) = 1.000 \\
\mathsf{Ent}(D^2) &= -\left(\frac{4}{6}\log_2\frac{4}{6} + \frac{2}{6}\log_2\frac{2}{6}\right) = 0.918 \\
\mathsf{Ent}(D^3) &= -\left(\frac{1}{5}\log_2\frac{1}{5} + \frac{4}{5}\log_2\frac{4}{5}\right) = 0.722
\end{aligned}$$

计算信息增益

计算属性“色泽”的信息增益：

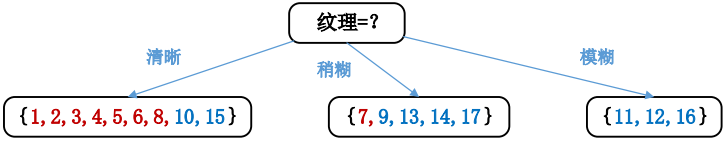
$$\begin{aligned}\mathsf{Gain}(D, \text{色泽}) &= \mathsf{Ent}(D) - \sum_{v=1}^{3} \frac{|D^v|}{|D|} \mathsf{Ent}(D^v) \\ &= 0.998 - \left(\frac{6}{17} \times 1.000 + \frac{6}{17} \times 0.918 + \frac{5}{17} \times 0.722 \right) \\ &= 0.109\end{aligned}$$

同样方法，计算其它属性的信息增益：

$$\mathsf{Gain}(D, \text{根蒂}) = 0.143 \qquad \mathsf{Gain}(D, \text{敲声}) = 0.141$$
$$\mathsf{Gain}(D, \text{纹理}) = 0.381 \qquad \mathsf{Gain}(D, \text{脐部}) = 0.289$$
$$\mathsf{Gain}(D, \text{触感}) = 0.006$$

选择属性

属性“纹理”的信息增益最大，选择作为划分属性



D^3全部为反例，满足终止条件，标注为“坏瓜”，D^1, D^2递归构造决策树
子集D^1分别计算剩余属性的信息增益：

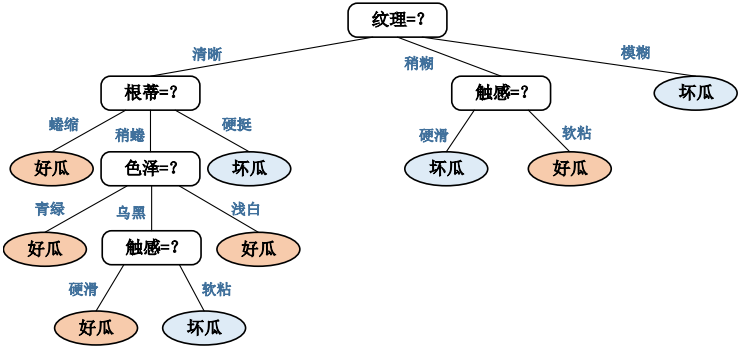
$$\text{Gain}(D, \text{色泽}) = 0.043 \qquad \text{Gain}(D, \text{根蒂}) = 0.458$$
$$\text{Gain}(D, \text{敲声}) = 0.331 \qquad \text{Gain}(D, \text{脐部}) = 0.458$$
$$\text{Gain}(D, \text{触感}) = 0.458$$

根蒂、脐部、触感的信息增益均取得最大增益，可以任选一个作为划分属性

… …

构建决策树

每个节点继续递归构造决策树，直到节点的样例集合满足终止条件为止



C4.5算法

- **增益率**
  - 信息增益倾向于选择取值比较多的属性，例如例5.1中如果计算“编号”的信息增益，可以分成17个子集，每个子集1样例，信息熵为0，增益最大
  - 定义增益率：

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

其中

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

属性a取值越多，$\text{IV}(a)$通常越大，例5.1中：

$$\text{IV}(\text{触感})=0.874,\quad \text{IV}(\text{色泽})=1.580,\quad \text{IV}(\text{编号})=4.088$$

  - C4.5算法在信息增益高于平均值的属性中，选择增益率最大的作为最优属性

CART算法

- **基尼指数**
  - 基尼值是数据集D纯度的另外一种度量：

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

  - 基尼值反映了从数据集D中随机抽取两个样例，其类别不一致的概率，值越小纯度越高，值越大纯度越低
  - CART算法选择基尼指数最小的属性作为划分属性：

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

5.3 剪枝

剪枝

- **决策树是容易过拟合的模型**
  - 一般来说，决策树可以完美地预测训练数据，但对测试数据可能效果会很差
  - 决策树的分支过多，从根节点到叶节点的路径过长都是过拟合的表现
- **剪枝处理**
  - 剪枝是控制决策树过拟合的主要手段
  - 预剪枝：决策树生成过程中，估计当前节点的划分是否能够带来泛化性能的提升，决定是否停止划分节点
  - 后剪枝：先生成完整的决策树，然后自底向上考察将非叶节点替换为叶节点，能否带来泛化性能的提升

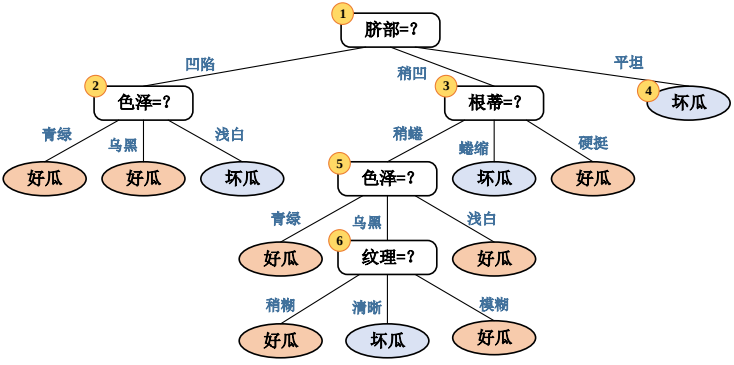
例5.2 剪枝

西瓜数据随机划分出一部分作为验证集，用于评估决策树的泛化性能：

训练集	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
验证集	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

完整决策树

不剪枝，使用训练集构造的完整决策树



预剪枝

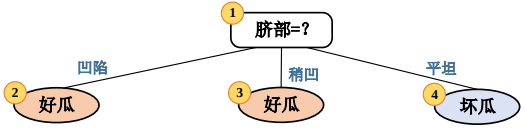
验证集	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否

- **节点1**
  - 不划分，标注为“好瓜”，验证集正确率：$\frac{3}{7} = 42.9\%$

预剪枝

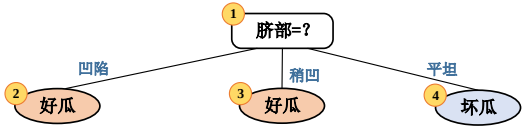
验证集	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否

- **节点1**
  - 不划分，标注为“好瓜”，验证集正确率：$\frac{3}{7} = 42.9\%$
  - 划分3个分支，验证集正确率：$\frac{5}{7} = 71.4\%$
  - 正确率提高，不剪枝，划分



预剪枝

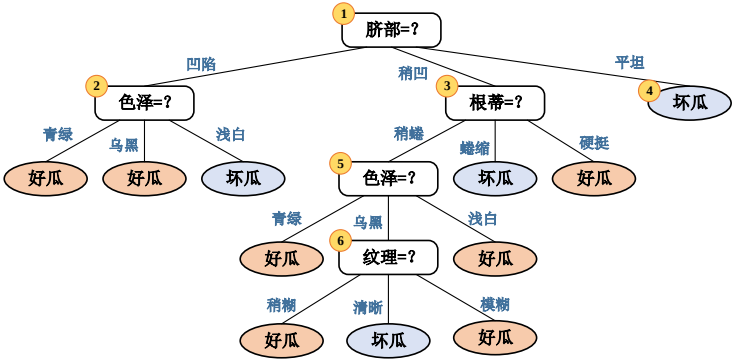
- **节点2**
  - 划分前：$\frac{5}{7} = 71.4\%$
  - 划分后：$\frac{4}{7} = 57.1\%$
  - 正确率下降，剪枝，不划分
- **节点3**
  - 划分前：$\frac{5}{7} = 71.4\%$
  - 划分后：$\frac{5}{7} = 71.4\%$
  - 正确率不变，剪枝，不划分



预剪枝之后的决策树

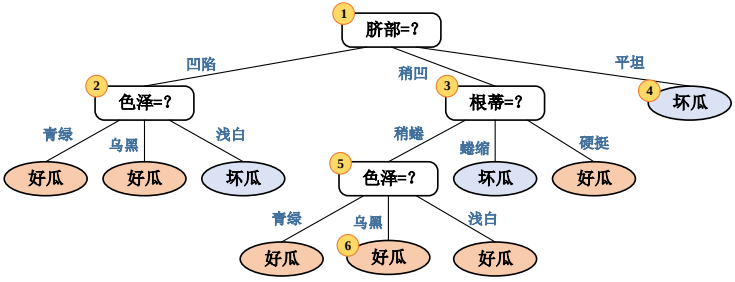
后剪枝

使用训练集构造的完整决策树，验证集上的正确率：42.9%



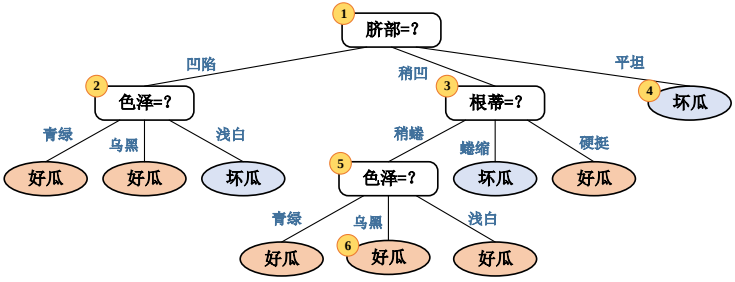
后剪枝

考察节点6，剪枝前后正确率：$42.9\% \rightarrow 57.1\%$，剪枝



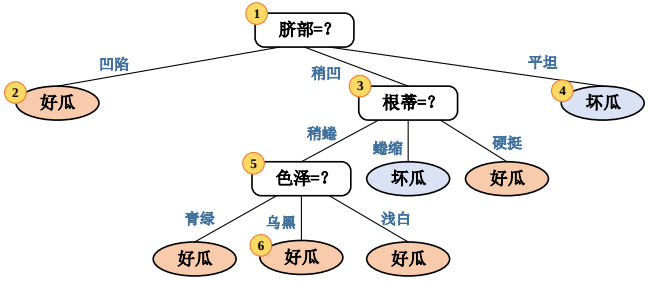
后剪枝

考察节点5，剪枝前后正确率：57.1% → 57.1%，不剪枝



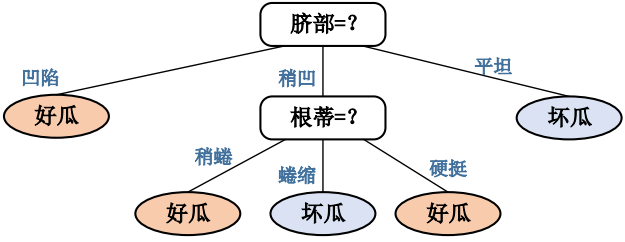
后剪枝

考察节点2，剪枝前后正确率：57.1% → 71.4%，剪枝



后剪枝

节点3和1的剪枝都不会提高正确率，应被保留；节点4本身就是叶节点，不需要剪枝；
节点5的子节点都是“好瓜”，也可以精简；



后剪枝之后的决策树

剪枝处理

- **时间开销**
  - 预剪枝：降低训练时间开销，降低测试时间开销
  - 后剪枝：增加训练时间开销，降低测试时间开销
- **过/欠拟合风险**
  - 预剪枝：降低过拟合风险，增加欠拟合风险
  - 后剪枝：降低过拟合风险，欠拟合风险基本不变
- **泛化性能**
  - 后剪枝通常优于预剪枝

5.4 连续属性

连续值属性

- **连续值的处理**
  - 基本思路：连续属性离散化
  - 二分法：设定划分点t将样本集D划分为两个子集，连续属性$a \geq t$的样本作为D_t^+，$a < t$的样本作为D_t^-
  - 属性a在D中有n个取值，由小到大排列为$\{a^1, \cdots, a^n\}$
  - $t \in [a^i, a^{i+1})$的任意取值，划分效果相同，包含$n-1$个候选划分点的集合：

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

连续值属性

- **连续值属性的信息增益**
  - 属性a以t为划分点的信息增益：

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

  - 选择最优的划分点t来划分，属性a的信息增益：

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

- **连续值属性的选择**
  - 每个节点按照$\mathsf{Gain}(D,a)$来选择划分属性
  - 与离散属性不同，划分属性为连续属性，该属性仍可作为后代节点的划分属性

例5.3 连续属性决策树

连续属性西瓜数据集

编号	密度	含糖量	好瓜	编号	密度	含糖量	好瓜
1	0.697	0.460	是	9	0.666	0.091	否
2	0.774	0.376	是	10	0.243	0.267	否
3	0.634	0.264	是	11	0.245	0.057	否
4	0.608	0.318	是	12	0.343	0.099	否
5	0.556	0.215	是	13	0.639	0.161	否
6	0.403	0.237	是	14	0.657	0.198	否
7	0.481	0.149	是	15	0.360	0.370	否
8	0.437	0.211	是	16	0.593	0.042	否
				17	0.719	0.103	否

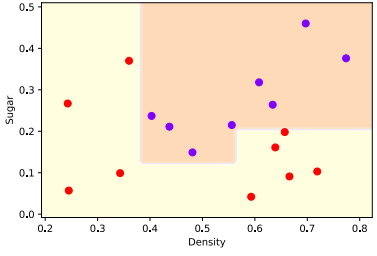
连续属性决策树

```
import numpy as np
from sklearn.tree import DecisionTreeClassifier

X=np.array([[0.697,0.460],[0.774,0.376],[0.634,0.264],[0.608,0.318],[0.556,0.215],
            [0.403,0.237],[0.481,0.149],[0.437,0.211],[0.666,0.091],[0.243,0.267],
            [0.245,0.057],[0.343,0.099],[0.639,0.161],[0.657,0.198],[0.360,0.370],
            [0.593,0.042],[0.719,0.103]])
y = np.array(['yes','yes','yes','yes','yes','yes','yes','yes','no',
              'no','no','no','no','no','no','no','no'])

tree = DecisionTreeClassifier(max_depth=5,random_state=0,criterion='entropy')
tree.fit(X, y)
print("Predict on training set:", tree.predict(X))
print("Accuracy on training set:", tree.score(X, y))
```

```
Predict on training set: ['yes' 'yes' 'yes' 'yes' 'yes' 'yes' 'yes' 'yes' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'no']
Accuracy on training set: 1.0
```

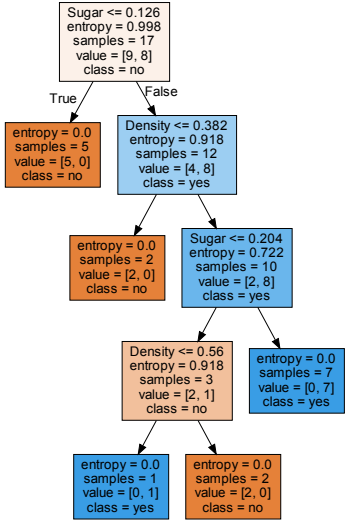
显示分类边界

```
import matplotlib.pyplot as plt
from plot_decision_boundary
       import plot_decision_boundary

y[y=='yes'] = 0; y[y=='no'] = 1

eps = 0.05
x_min,x_max = X[:,0].min()-eps, X[:,0].max()+eps
y_min,y_max = X[:,1].min()-eps, X[:,1].max()+eps

plot_decision_boundary(tree,axis=[x_min,\
                    x_max,y_min,y_max])
plt.scatter(X[:,0], X[:,1], c=y.reshape(-1,1),\
            s=50, cmap='rainbow')
plt.xlabel("Density"); plt.ylabel("Sugar")

plt.show()
```

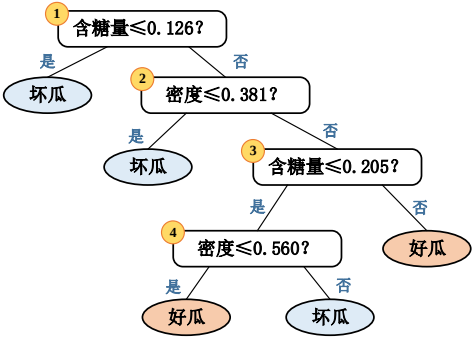


显示决策树

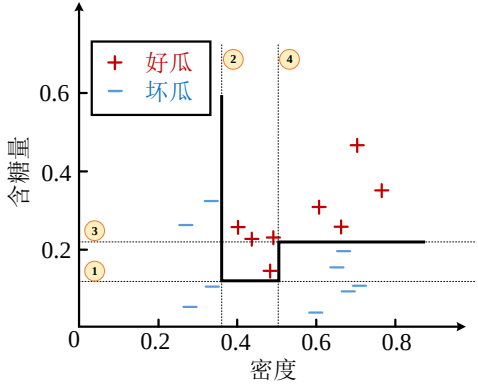
```
from sklearn.tree import export_graphviz
import graphviz

export_graphviz(tree, out_file="tree.dot",\
    class_names=['no','yes'],\
    feature_names=np.array(['Density','Sugar']),\
    impurity=True, filled=True)

with open("tree.dot") as f:
    dot_graph = f.read()
graphviz.Source(dot_graph)
```



决策树与分类边界



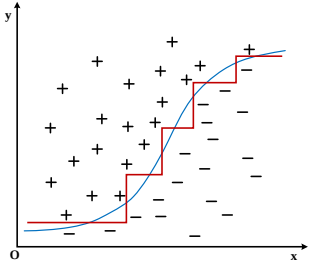
生成的决策树



对应的分类边界

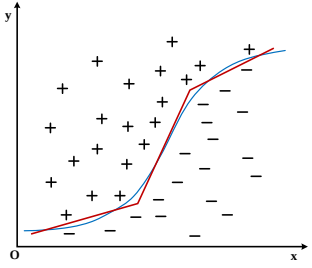
多变量决策树

- **轴平行分类边界**
  - 每个节点选择一个连续属性划分，得到的分类边界是一个平行于坐标轴的分类面
  - 决策树的分类边界是由若干轴平行分类面分段组成的
  - 真实分类边界比较复杂时，需要复杂的决策树用很多个分段才能很好地近似

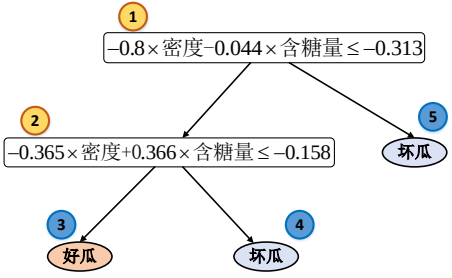


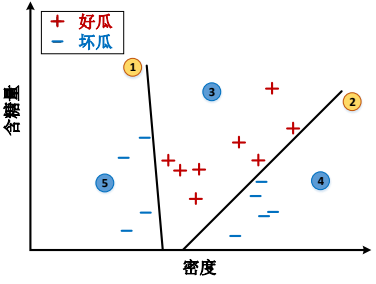
多变量决策树

- **斜线分类边界**
  - 每个节点采用多个属性的线性组合来划分，用一个线性分类器来判别不同分支
  - 可以用一个简单的决策树和对应的斜线分类面近似真实分类边界



多变量决策树


1
−0.8×密度−0.044×含糖量≤−0.313
2
−0.365×密度+0.366×含糖量≤−0.158
5
坏瓜
3
好瓜
4
坏瓜


好瓜
坏瓜
含糖量
密度
1
2
3
4
5