

# 9 Decision Tree

## 决策树

数据纯度越高，不确定度越低，信息熵越小



热力学两个基本定理是整个宇宙的基本规律：1. 宇宙能量守恒；2. 宇宙的熵不断增大。

*The fundamental laws of the universe which correspond to the two fundamental theorems of the mechanical theory of heat.*

*1. The energy of the universe is constant.*

*2. The entropy of the universe tends to a maximum.*

—— 鲁道夫·克劳修斯 (Rudolf Clausius) | 德国物理学家 | 1822 ~ 1888



```
◀ matplotlib.pyplot.contour() 绘制等高线图
◀ matplotlib.pyplot.contourf() 绘制填充等高线图
◀ numpy.meshgrid() 创建网格化数据
◀ seaborn.scatterplot() 绘制散点图
◀ sklearn.datasets.load_iris() 加载鸢尾花数据集
◀ sklearn.tree.DecisionTreeClassifier 决策树分类函数
◀ sklearn.tree.plot_tree 绘制决策树树形
```

## 9.1 决策树：可以分类，也可以回归

### 决策树结构

**决策树** (decision tree) 类似《数学要素》第 20 章介绍的**二叉树** (binomial tree)。如图 1 所示，决策树结构主要由**结点** (node) 和**子树** (branch) 构成；结点又分为**根结点** (root node)、**内部结点** (internal node) 和**叶结点** (leaf node)。其中，内部结点又叫**母节点** (parent node)，叶节点又叫**子节点** (child node)。

每一个根节点和内部结点可以生长出一层二叉树，其中包括**左子树** (left branch) 和**右子树** (right branch)；构造子树的过程也是将结点数据划分为两个子集的过程。

图 1 所示树形结构有 4 个叶节点。请大家格外注意叶节点数目；决策树算法可以输入**最大叶节点数量** (maximum leaf nodes)，控制决策树大小，也称**剪枝** (pruning)。

此外，**深度** (depth) 也可以控制树形大小，所谓深度就是二叉树的层数。比如，图 1 二叉树有两层，所以深度为 2。深度也是决策树函数用户输入量之一。

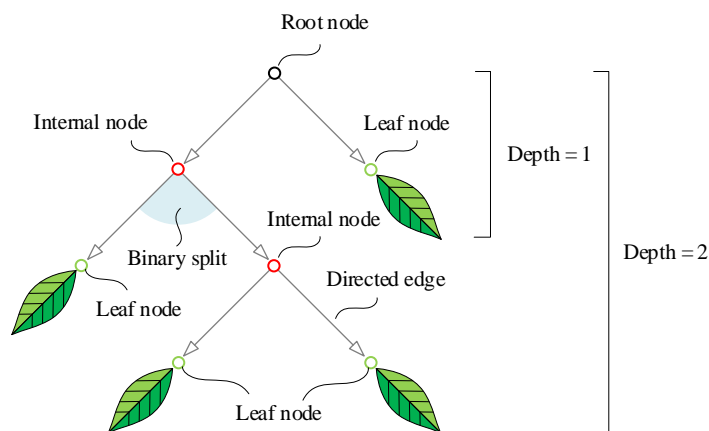


图 1. 决策树树形结构

### 如何用决策树分类

下面展开讲解决策树如何分类。

图 2 展示的决策树第一步划分：样本数据中  $x_1 \geq a$ ，被划分到右子树；样本数据中  $x_1 < a$ ，被划分到左子树。经过第一步二叉树划分，原始数据被划分为 A 和 B 两个区域。A 区域以红色 ● ( $C_1$ ) 为主，B 区域以蓝色 ● ( $C_2$ ) 为主。

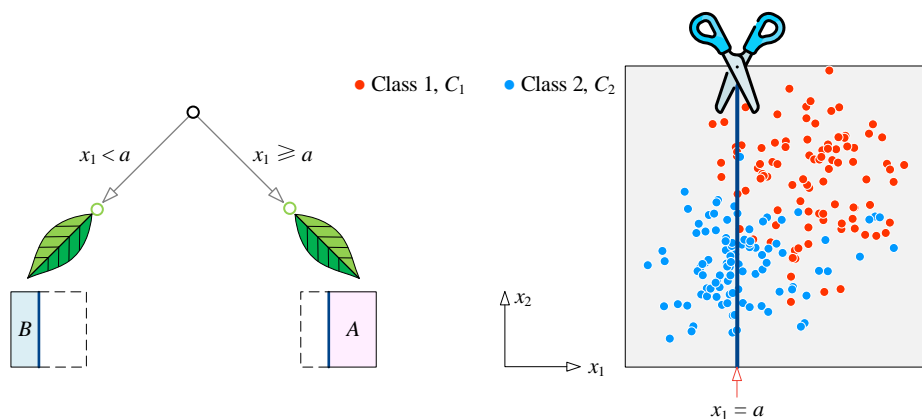


图 2. 决策树第一步划分

图 3 所示为图 2 右子树内部结点生长出一个新的二叉树。样本数据中  $x_2 \geq b$ ，被划分到右子树；样本数据中  $x_2 < b$ ，被划分到左子树。经过第二步二叉树划分， $A$  被划分为  $C$  和  $D$  两个区域。 $C$  区域以红色  $\bullet$  ( $C_1$ ) 为主， $D$  区域以蓝色  $\bullet$  ( $C_2$ ) 为主。

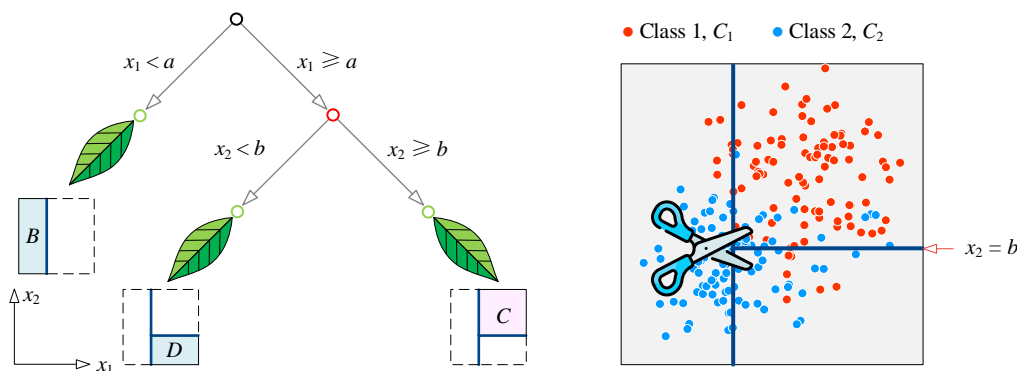


图 3. 决策树第二步划分

决策树分类算法有自己独特的优势。决策树的每个节点可以生长成一颗二叉树，这种基于某一特征的二分法很容易解释。此外，得到的决策树很方便可视化，本章后续将介绍如何可视化决策树树形结构。

如老子所言，“一生二，二生三，三生万物”，根据数据的复杂程度，决策树树形可以不断生长。数据结构越复杂，对应树形结构也就越复杂。但是，过于复杂的树形会导致过度拟合，模型泛化能力变弱。这种情况需要控制叶节点数量或者最大深度来控制树形规模，从而避免过度拟合。

有读者可能会问，依据什么标准选择划分的位置呢？比如图 2 中， $a$  应该选在什么位置？图 3 中的  $b$  又该选择什么位置？这就是下几节要回答的问题。

## 9.2 信息熵：不确定性度量

为了解决在决策树在哪划分节点的问题，需要介绍几个新概念：**信息熵** (information entropy)、**信息增益** (information gain) 和**基尼指数** (Gini index)。本节首先介绍信息熵。

### 熵

**熵** (entropy) 是物理系统混乱程度的度量。系统越混乱，熵越大；系统越有序，熵越小。熵这个概念起源热力学。1854 年，德国物理学家**鲁道夫·克劳修斯** (Rudolf Clausius) 引入熵这一概念。

**维纳过程** (Wiener process) 的提出者——**诺伯特·维纳** (Norbert Wiener)，认为随着熵的增加，宇宙以及宇宙中所有封闭系统都会自然地退化，并失去其独特性。

### 信息熵

在**信息论** (information theory) 中，信息的作用是降低不确定性。**信息熵** (information entropy) 可以用来表示随机变量的不确定性度量。信息熵越大，不确定性越大。1948 年，**香农** (Claude Shannon) 提出信息熵这一概念，因此信息熵也常被称作**香农熵** (Shannon entropy)。



**克劳德·香农** (Claude Shannon)

美国数学家、工程师、密码学家 | 1916 ~ 2001  
信息论创始人。丛书关键词：● 信息熵 ● 信息增益



样本数据集  $\Omega$  的信息熵定义为：

$$\text{Ent}(\Omega) = -\sum_{k=1}^n p_k \log_2 p_k \quad (1)$$

其中， $p_k$  为  $\Omega$  中第  $k$  类样本所占比例，即概率值。由于  $\log_2 0$  不存在，特别指定  $0 \times \log_2 0 = 0$ 。

### 举个例子

当样本数据集  $\Omega$  只有两类  $K=2$ ，类别序数  $k=1, 2$ 。

令

$$p_1 = p, \quad p_2 = 1 - p \quad (2)$$

其中,  $p$  取值范围  $[0, 1]$ 。

这种情况下,  $\Omega$  的信息熵  $\text{Ent}(\Omega)$  为:

$$\begin{aligned} \text{Ent}(\Omega) &= -\sum_{k=1}^2 p_k \log_2 p_k = -(p_1 \log_2 p_1 + p_2 \log_2 p_2) \\ &= -p \log_2 p - (1-p) \log_2 (1-p) \end{aligned} \quad (3)$$

其中,  $p_1 = p, \quad p_2 = 1 - p$ 。

观察 (3), 可以发现  $\text{Ent}(\Omega)$  是以  $p$  为变量的函数。

图 6 告诉我们, 在  $A$  和  $C$  点, 当样本只属于某一特定类别时 ( $p = 0$  或  $p = 1$ ), 也就是数据纯度最高, 不确定性最低, 信息熵  $\text{Ent}(\Omega)$  最小。

在  $B$  点, 两类样本数据各占一半 ( $p = 0.5$ ), 这时数据纯度最低, 不确定性最高, 信息熵  $\text{Ent}(\Omega)$  最大。

从  $A$  到  $B$ , 信息熵不断增大; 从  $B$  到  $C$ , 信息熵不断减小。

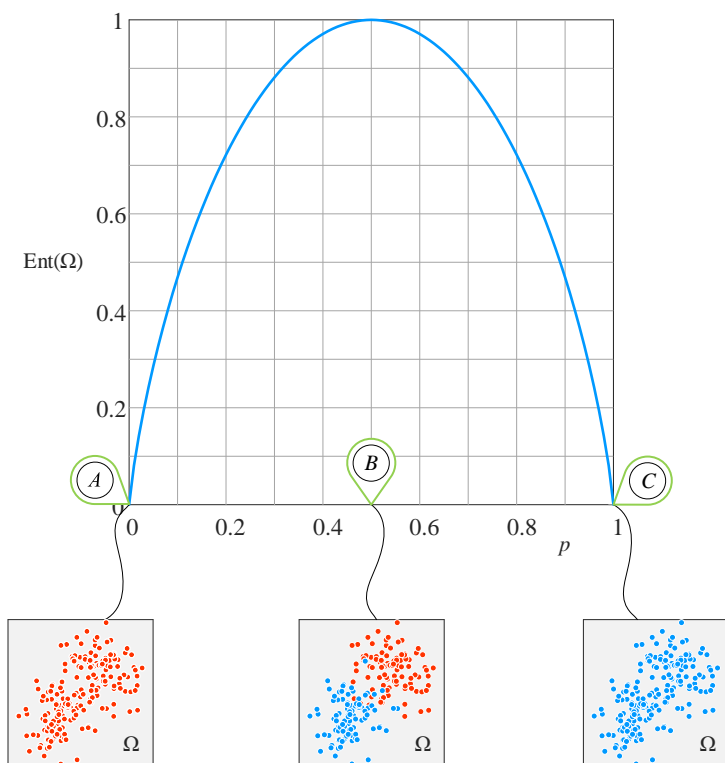


图 4. 信息熵  $\text{Ent}(\Omega)$  随  $p$  变化趋势

## K 类标签

如果样本数据集  $\Omega$  分为  $K$  类时，即  $\Omega = \{C_1, C_2, \dots, C_K\}$ ，各类标签样本数量之和等于  $\Omega$  中所有样本总数，即下式：

$$\sum_{k=1}^K \text{count}(C_k) = \text{count}(\Omega) \quad (4)$$

其中， $\text{count}(C_k)$  计算  $C_k$  类样本数量。

$C_k$  类样本概率  $p_k$  可以通过下式计算获得：

$$p_k = \frac{\text{count}(C_k)}{\text{count}(\Omega)} \quad (5)$$

将 (5) 代入 (1)，得到样本数据集  $\Omega$  的信息熵为：

$$\text{Ent}(\Omega) = -\sum_{k=1}^K p_k \log_2 p_k = -\sum_{k=1}^K \left\{ \frac{\text{count}(C_k)}{\text{count}(\Omega)} \log_2 \left( \frac{\text{count}(C_k)}{\text{count}(\Omega)} \right) \right\} \quad (6)$$

## 9.3 信息增益：通过划分，提高确定度

假设存在某个特征  $a$  将  $\Omega$  划分为  $m$  个子集，即：

$$\Omega = \{\Omega_1, \Omega_2, \dots, \Omega_m\} \quad (7)$$

而子集  $\Omega_j (j = 1, 2, \dots, m)$  中属于  $C_k$  类样本集合为  $\Omega_{j,k}$ ：

$$\Omega_{j,k} = \Omega_j \cap C_k \quad (8)$$

类别  $C_k$  元素在  $\Omega_j$  中占比为：

$$p_{j,k} = \frac{\text{count}(\Omega_{j,k})}{\text{count}(\Omega_j)} \quad (9)$$

计算子集  $\Omega_j$  信息熵：

$$\text{Ent}(\Omega_j) = -\sum_{k=1}^K \left\{ \frac{\text{count}(\Omega_{j,k})}{\text{count}(\Omega_j)} \log_2 \left( \frac{\text{count}(\Omega_{j,k})}{\text{count}(\Omega_j)} \right) \right\} \quad (10)$$

而经过特征  $a$  划分后的集合  $\Omega$  的信息熵为， $m$  个子集  $\Omega_j$  信息熵的加权和：

$$\underbrace{\text{Ent}(\Omega|a)}_{\text{Weighted sum of entropy after split}} = \sum_{j=1}^m \left\{ \frac{\text{count}(\Omega_j)}{\text{count}(\Omega)} \text{Ent}(\Omega_j) \right\} \quad (11)$$

将 (10) 代入 (11)，得到：

$$\text{Ent}(\Omega|a) = - \sum_{j=1}^m \left\{ \frac{\text{count}(\Omega_j)}{\text{count}(\Omega)} \sum_{k=1}^K \left\{ \frac{\text{count}(\Omega_{j,k})}{\text{count}(\Omega_j)} \log_2 \left( \frac{\text{count}(\Omega_{j,k})}{\text{count}(\Omega_j)} \right) \right\} \right\} \quad (12)$$

经过特征  $a$  划分后的  $\Omega$  信息熵减小，确定度提高。

### 举个例子

图 5 所示数据集  $\Omega$  有两个标签， $C_1$  和  $C_2$ 。特征  $a$  将数据集  $\Omega$  划分为 2 个子集—— $\Omega_1$ 、 $\Omega_2$ 。

根据 (9) 类别  $C_1$  元素在  $\Omega_1$  中占比为：

$$p_{1,1} = \frac{\text{count}(\Omega_{1,1})}{\text{count}(\Omega_1)} \quad (13)$$

子集  $\Omega_1$  信息熵为：

$$\text{Ent}(\Omega_1) = - \frac{\text{count}(\Omega_{1,1})}{\text{count}(\Omega_1)} \log_2 \left( \frac{\text{count}(\Omega_{1,1})}{\text{count}(\Omega_1)} \right) - \frac{\text{count}(\Omega_{1,2})}{\text{count}(\Omega_1)} \log_2 \left( \frac{\text{count}(\Omega_{1,2})}{\text{count}(\Omega_1)} \right) \quad (14)$$

同理，可以计算得到  $\Omega_2$  子集信息熵。

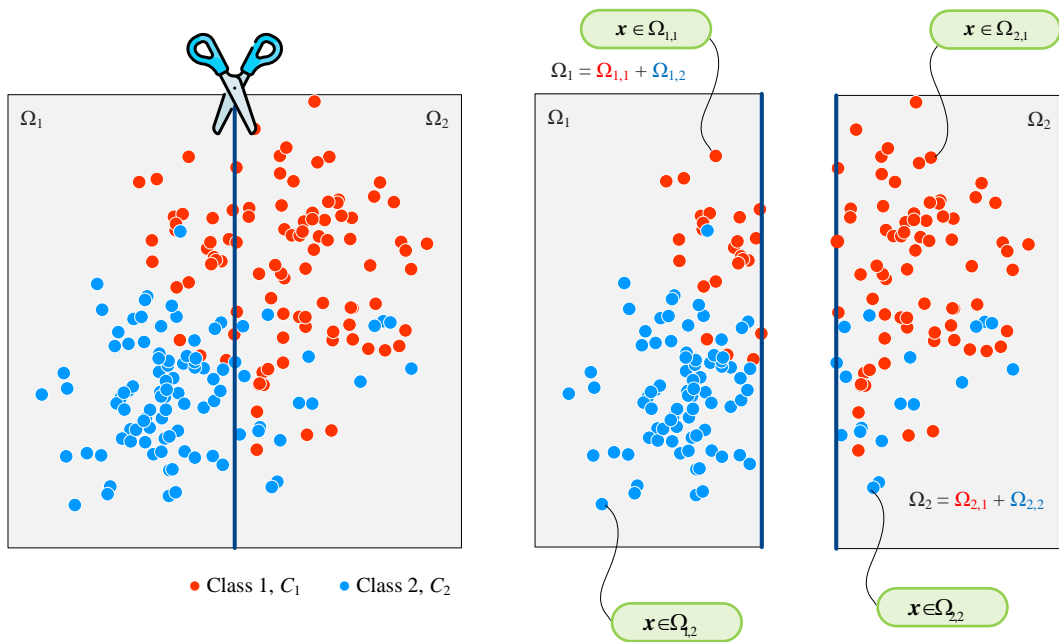


图 5. 数据集  $\Omega$  划分为 2 个子集

### 信息增益

**信息增益** (information gain) 量化划分前后信息熵变化：

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger：<https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：[jiang.visualize.ml@gmail.com](mailto:jiang.visualize.ml@gmail.com)

$$\text{Gain}(D, a) = \underbrace{\text{Ent}(D)}_{\text{Entropy before split}} - \underbrace{\text{Ent}(D|a)}_{\text{Weighted sum of entropy after split}} \quad (15)$$

最佳划分  $a$  位置对应最大化信息增益：

$$\arg \max_a \text{Gain}(D, a) \quad (16)$$

## 9.4 基尼指数：指数越大，不确定性越高

类似信息熵，**基尼指数** (Gini index) 也可以用来表征样本数据集合  $\Omega$  纯度。注意，这个基尼指数不同于衡量国家或地区收入差距的基尼指数。

基尼指数  $\text{Gini}(\Omega)$  定义如下：

$$\text{Gini}(\Omega) = \sum_{i=1}^n p_i (1 - p_i) = \sum_{i=1}^n p_i - \sum_{i=1}^n p_i^2 = 1 - \sum_{i=1}^n p_i^2 \quad (17)$$

类似上节，当样本数据集合  $\Omega$  只有两类  $K=2$ 。这种情况下， $p_1 = p$ ,  $p_2 = 1 - p$ 。  $\Omega$  的信息熵  $\text{Gini}(\Omega)$  为。

$$\begin{aligned} \text{Ent}(D) &= 1 - \sum_{i=1}^n p_i^2 = 1 - p_1^2 - p_2^2 \\ &= 1 - p^2 - (1 - p)^2 = -2p^2 + 2p \end{aligned} \quad (18)$$

如图 6 (a) 所示， $\text{Gini}(\Omega)$  越大，不确定性越高，数据纯度越低。 $\text{Gini}(\Omega)$  最大值为  $1/2$ ，对应图中  $p = 0.5$ ，也就是两类标签样本数据各占一半。图 6 (b) 比较  $2 \times \text{Gini}(\Omega)$  和  $\text{Ent}(\Omega)$  两图形关系。

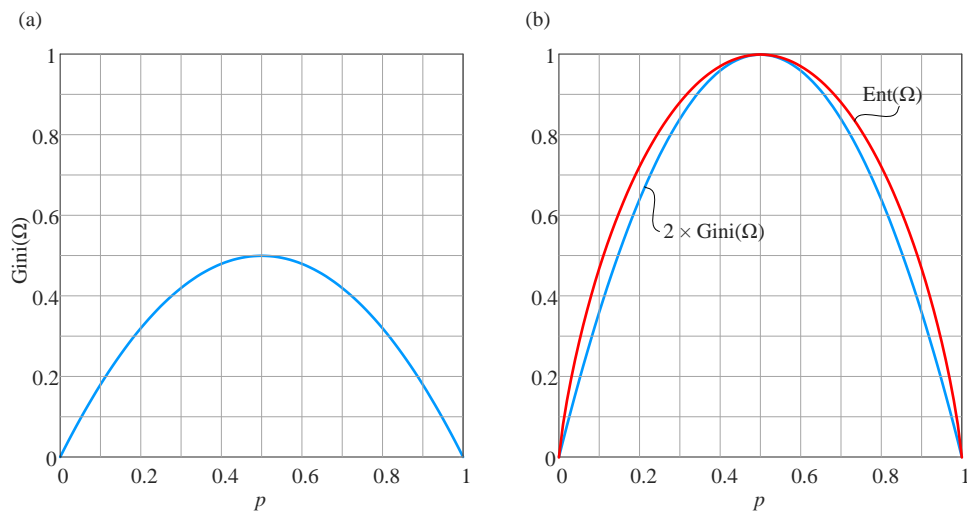


图 6. 比较信息熵和 Gini 指数图像



Scikit-learn 中决策树分类函数 `DecisionTreeClassifier`，就是默认采用 Gini 指数最大化作为分割依据。

## 9.5 最大叶节点：影响决策边界

本节利用决策树算法分类鸢尾花样本数据，并着重展示最大叶节点数分类影响。Scikit-learn 工具包决策树分类函数为 `sklearn.tree.DecisionTreeClassifier`；该函数可以用最大叶节点数 `max_leaf_nodes` 控制决策树树形大小。

同时，本节和下一节利用 `sklearn.tree.plot_tree` 绘制决策树。

### 最大叶节点数为 2

图 7 所示为当最大叶节点数  $L$  为  $L = 2$  时，鸢尾花数据分类情况。图 7 (a) 所示，根据花萼长度  $x_1$  这一特征，特征平面被划分为两个区域——A 和 B。

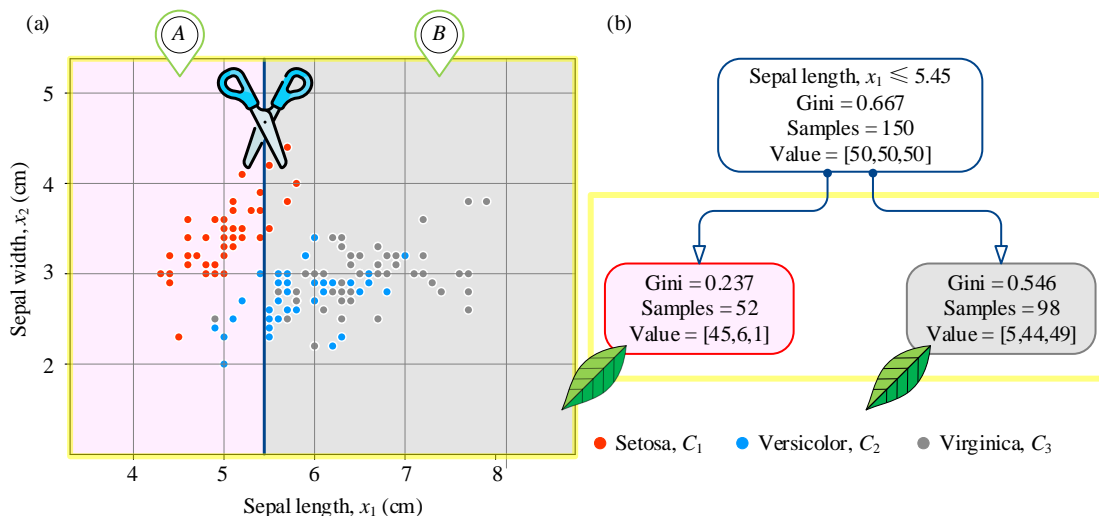


图 7. 最大叶节点数量为 2，（画出百分比，饼图）

图 7 (b) 树形图有大量重要信息。150 个样本数据 Gini 指数为 0.667。根据 Gini 指数最大化原则，找到划分花萼长度  $x_1$  最佳位置为， $x_1 = 5.45$ 。 $x_1 \leq 5.45$  为区域 A； $x_1 > 5.45$  为区域 B。

区域 A 中，样本数据为 52；其中，● ( $C_1, y = 0$ ) 为 45 个，● ( $C_2, y = 1$ ) 为 6 个，● ( $C_3, y = 2$ ) 为 1 个。显然，区域 A 预测分类为  $C_1$ 。区域 A 的 Gini 指数为 0.237。

区域 B 中，样本数据为 98；其中，● ( $C_1, y = 0$ ) 为 5 个，● ( $C_2, y = 1$ ) 为 44 个，● ( $C_3, y = 2$ ) 为 49 个。显然，区域 A 预测分类为  $C_3$ 。区域 B 的 Gini 指数为 0.546。

根据 (11)，可以计算得到特征  $x_1$  划分后信息熵：

$$\underbrace{\text{Ent}(\Omega|x_1 = 5.45)}_{\text{Weighted sum of entropy after split}} = \frac{52}{150} \times 0.237 + \frac{98}{150} \times 0.546 = 0.4389 \quad (19)$$

根据 (15) 信息增益为：

$$\text{Gain}(D, a) = \underbrace{\text{Ent}(D)}_{\text{Entropy before split}} - \underbrace{\text{Ent}(\Omega|x_1 = 5.45)}_{\text{Weighted sum of entropy after split}} = 0.667 - 0.4389 = 0.228 \quad (20)$$

### 最大叶节点数为 3

当最大叶节点数量  $L$  继续提高到  $L = 3$  时，图 7 (b) 某一叶节点将会在某一特征基础上继续划分。图 8 所示为  $L = 3$ ，决策树分类鸢尾花结果。

观察图 8 (a)，可以发现图 7 (a) 中  $B$  区域沿着  $x_1$  方向进一步被划分为  $C$  和  $D$ 。划分的位置为  $x_1 = 6.15$ 。

区域  $C$  中，样本数据为 43；其中，● ( $C_1, y = 0$ ) 为 5 个，● ( $C_2, y = 1$ ) 为 28 个，● ( $C_3, y = 2$ ) 为 10 个。显然，区域  $C$  预测分类为  $C_2$ 。区域  $C$  的 Gini 指数为 0.508。

区域  $D$  中，样本数据为 55；其中，● ( $C_1, y = 0$ ) 为 0 个，● ( $C_2, y = 1$ ) 为 16 个，● ( $C_3, y = 2$ ) 为 39 个。显然，区域  $A$  预测分类为  $C_3$ 。区域  $D$  的 Gini 指数为 0.413。

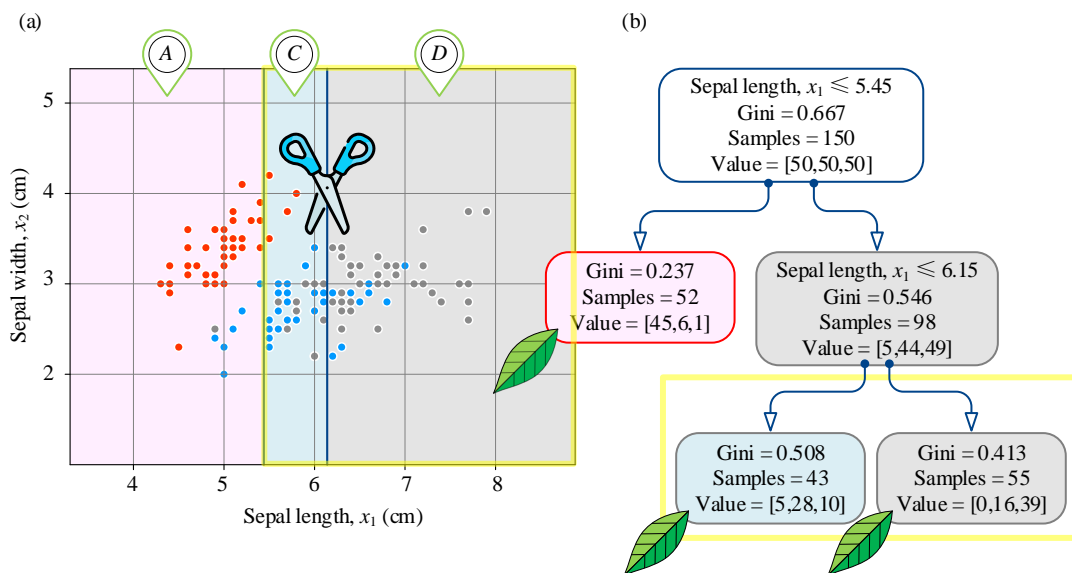


图 8. 最大叶节点数量为 3，（画出百分比，饼图）

### 最大叶节点数为 4

图 9 所示为最大叶节点数量  $L = 4$  时，决策树分类结果和树形结构。可以发现图 8 中，A 区沿  $x_2$  方向被进一步划分为两个区域；其中一个区域 44 个 ● ( $C_1, y = 0$ )，1 个 ● ( $C_2, y = 1$ )，Gini 指数进一步降低到 0.043。请读者自行计算 Gini 指数变化。

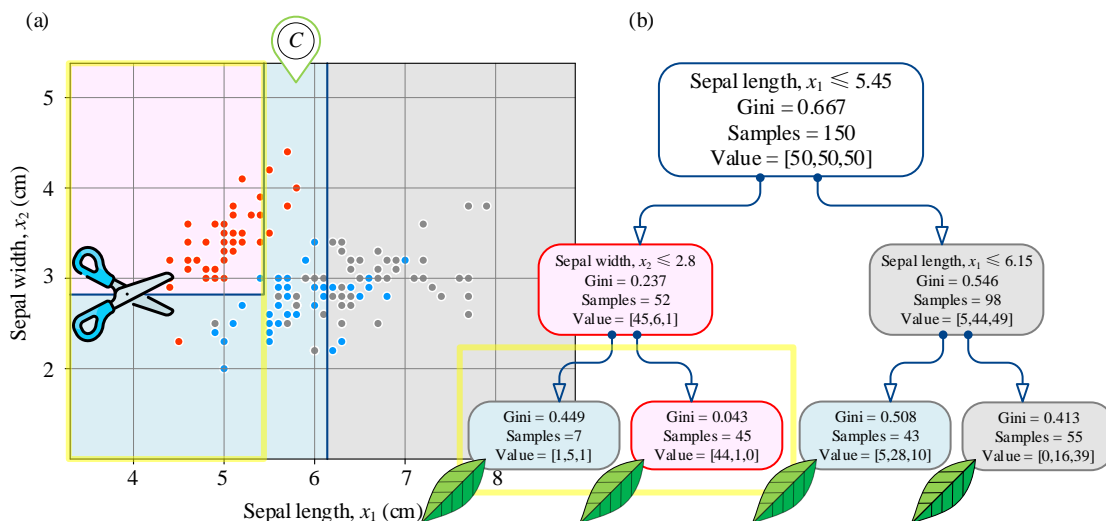


图 9. 最大叶节点数量为 4，（画出百分比，饼图）

### 最大叶节点数为 5

图 10 所示为最大叶节点数量  $L = 5$  时，决策树分类结果和树形结构。比较图 10 和图 9，C 区沿  $x_2$  方向被进一步划分为两个区域，得到的一个区域全部样本数据为 ● ( $C_1, y = 0$ )；因此，该区域的 Gini 指数为 0，纯度最高。

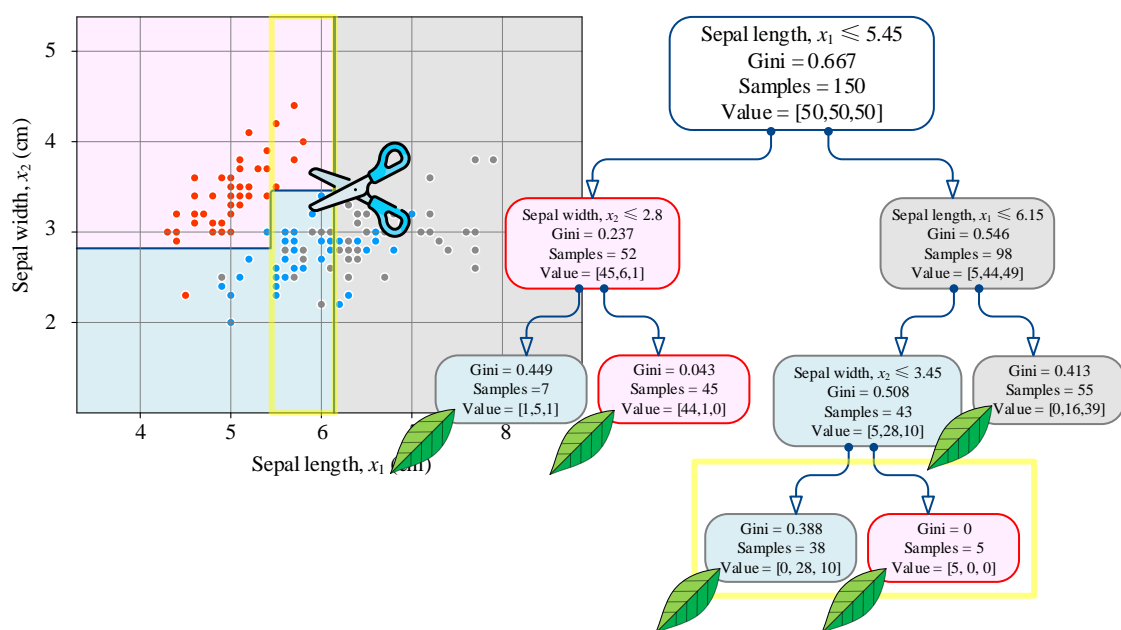


图 10. 最大叶节点数量为 5，（画出百分比，饼图）

下一节提供获得本节图像代码，代码中最大叶节点数量包括 10、15 和 20 等更大数值。请大家自行设定最大叶节点数量，比较决策边界和树形结构变化。

## 9.6 最大深度：控制树形大小

类似最大叶节点数量，最大深度从二叉树层数角度控制树形大小。`sklearn.tree.DecisionTreeClassifier` 函数用 `max_depth` 改变最大深度。

图 11 所示为最大深度为 1 时，鸢尾花的分类结果和树形图。可以发现，图 11 和图 7 结果完全一致。图 12 所示为最大深度为 2 时，鸢尾花的分类结果和树形图。可以发现，图 12 和图 9 结果完全一致。

图 13 所示为最大深度为 3 时，鸢尾花分类结果。图 14 所示树形结构有 3 层二叉树。注意，当最大深度不断增大时，如果某一区域样本数据为单一样本；则该区域 Gini 指数为 0，无法进一步划分。图 14 中 8 个叶节点中，有 4 个纯度已经达到最高；

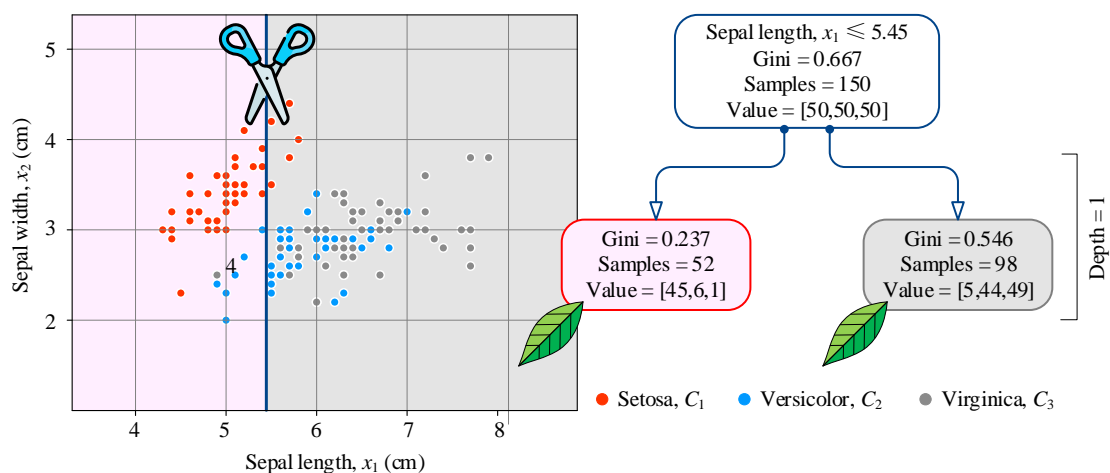


图 11. 最大深度为 1, (画出百分比, 饼图)

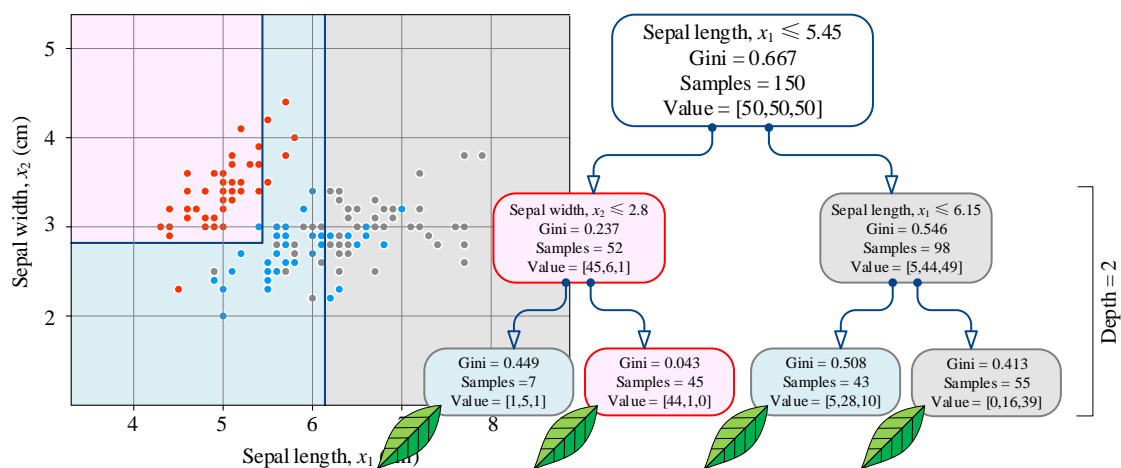


图 12. 最大深度为 2, (画出百分比, 饼图)

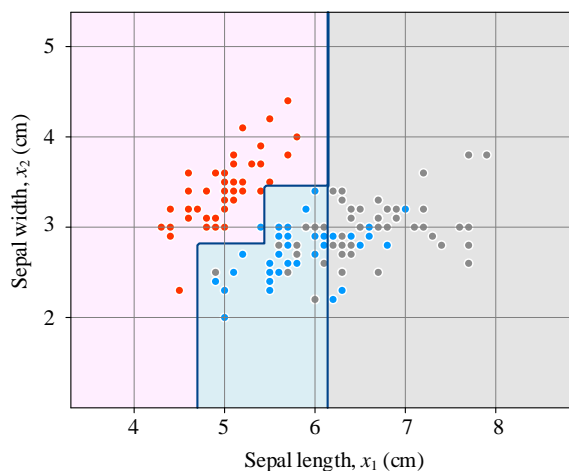


图 13. 最大深度为 3, 分类结果

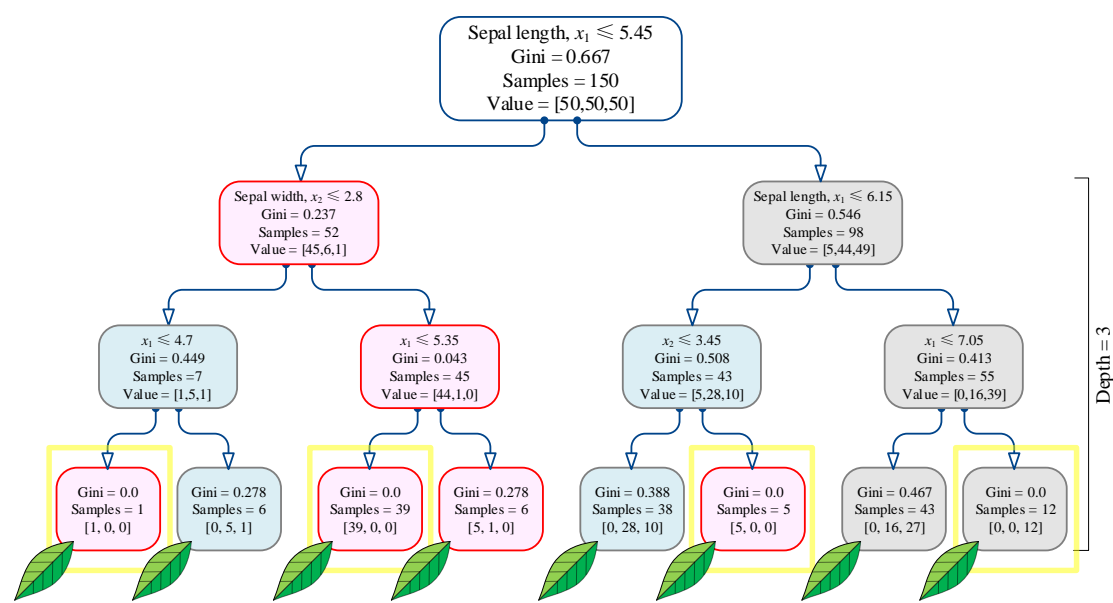


图 14. 最大深度为 3，树形结构，（画出百分比，饼图）



代码 Bk7\_Ch09\_01.py 利用决策树方法分类鸢尾花数据，并绘制本节和上一节图像。

