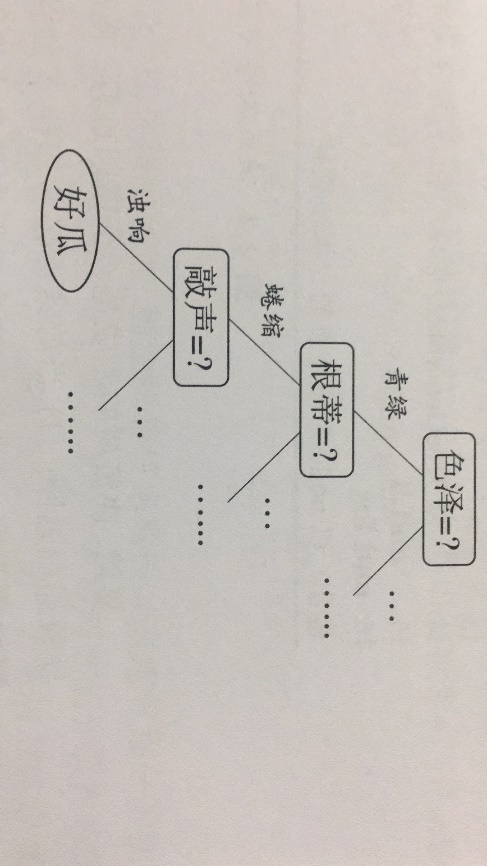
决策树

* 1. 基本流程

决策树（Decision Tree）是一类常见的机器学习方法。以二分类任务为例：我们希望从给定训练数据集学得一个模型用以对新样本进行分类。顾名思义，决策树是基于树结构来进行决策的。给定训练数据集如下：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **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 |
| 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 |

例如，对一个样本，进行决策时，通常会进行一系列的判断（子决策）：先看是什么颜色；如果是青绿色，则再看根蒂是什么形态；如果是蜷缩，再看敲起来的声音是什么声音；最后得出最终决策：这是个好瓜。决策过程如下图所示：



显然，决策过程的最终结论对应了我们所希望的判定结果（类别）；决策过程中提出的每个判定问题都是对某个属性的测试；每个测试的结果导出最终结论（或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。

一般的，一颗决策树包含一个根结点、若干个内部结点和若干个叶结点。

* 叶子结点：对应于决策结果
* 其他每个结点则对应于一个属性测试

根节点包含样本全集；每个结点包含的样本集合：根据属性测试的结果被划分到子结点中。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颗泛化能力强，即处理未见样本能力强的决策树，其基本流程遵循“分而治之”的策略，如下所示：

|  |
| --- |
| **输入**：训练集 |
| 属性集 |
| **过程**：函数 |
| 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： |
| 6： |
| 7： |
| 8：从A中选择最优划分属性； |
| 9： |
| 10： |
| 11： |
| 12： |
| 13： |
| 14： |
| 15： |
| 16： |
| **输出**：以node为根节点的一颗决策树 |

显然，决策树的生成是一个递归过程。在决策树基本算法中，有三种情形会导致递归返回：

（1）

（2）

（3）

* 1. 划分选择

决策树学习的关键是第8行，即如何选择最优划分属性。一般而言，随着划分过程不断进行，我们希望决策树的分支结点所包含的样本（子集）尽可能属于同一类别，即结点的“纯度”越来越高。（本质是将大集合按照最优的属性划分成纯度很高的子集）

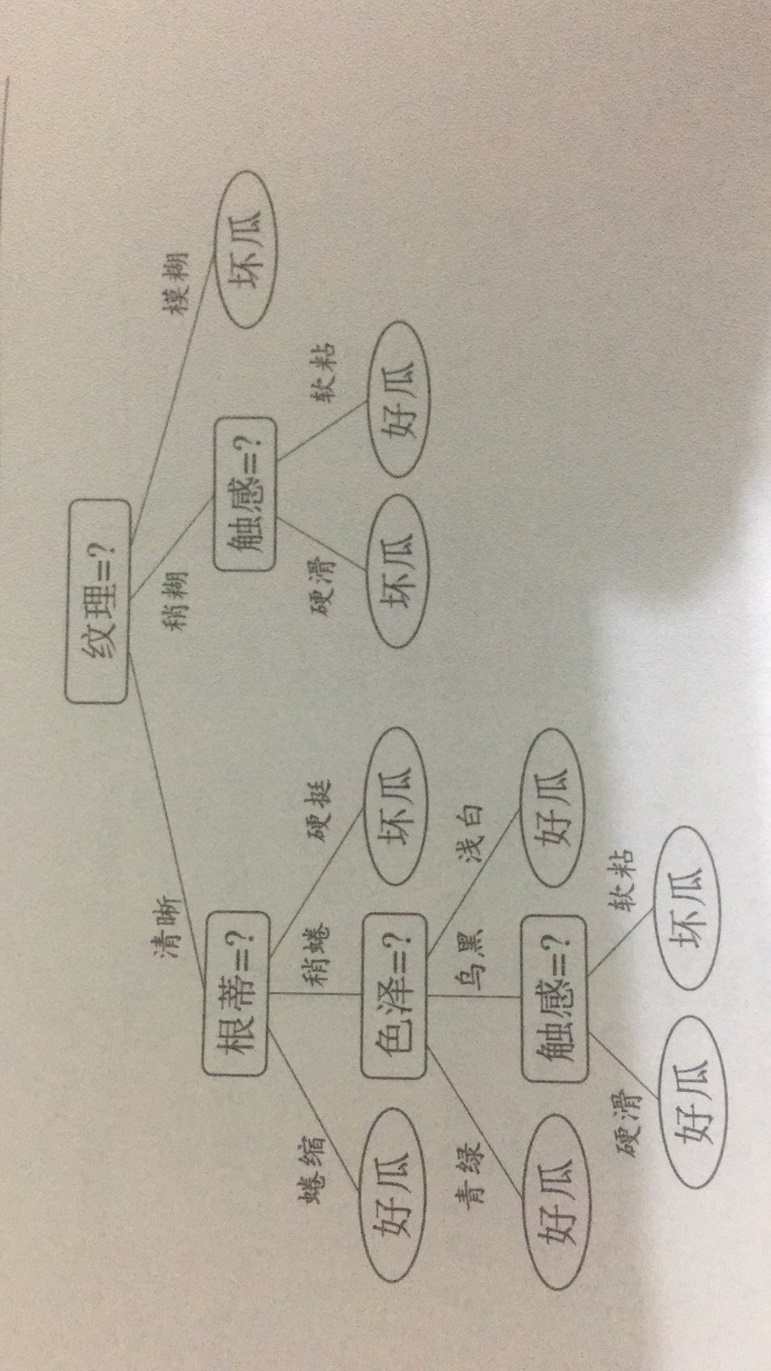
1.2.1、信息增益（ID3）

“信息熵”（information entropy）是度量样本集合纯度最常用的一种指标。假设当前样本集合中第类样本所占的比例为，则的信息熵定义为

的值越小，则的纯度越高（可作为终止条件，只要纯度不高，就可以继续分裂）。

假定离散属性有个可能的取值，若使用对样本集进行划分，则会产生个分支结点，其中第个分支结点包含了中所有在属性上取值为的样本（即为的子集），记为。可以根据上式计算出的信息熵（子集的信息熵计算），再考虑到不同的分支结点所包含的样本数不同，给分支结点赋予权重，即样本数越多的分支结点的影响越大，于是可计算出用属性对样本集进行划分所获得的“信息增益”

一般而言，信息增益越大，则意味着使用属性来进行划分所获得的“纯度提升”越大。因此，我们可用信息增益来进行决策树的划分属性选择，即选择属性。



1.2.2、增益率（C4.5）

在上面的介绍中，有意忽略了最左边的“编号”这一列。若把“编号”也作为一个候选划分属性，则可计算出它的信息增益为0.998，远大于其他候选划分属性。这是因为“编号”将产生17个分支，每个分支结点仅包含一个样本，这些分支结点的纯度已达到最大。然而，这样的决策树显然不具有泛化能力，无法对新样本进行有效预测。

实际上，信息增益准则对可取值数目较多的属性有所偏好，为减少这种偏好可能带来的不利影响，著名的C4.5决策树算法不直接使用信息增益，而是使用“增益率”来选择最优划分属性。增益率定义为：

其中，

称为属性的“固有值”。属性的可能取值数目越多（即V越大），则的值通常会越大。例如：IV（感触）=0.874（V=2）、IV（色泽）=1.580（V=3）、IV（编号）=4.088（V=17）。

需要注意的是，增益率准则对可取数目较少的属性有所偏好，因此C4.5算法并不是直接选择增益率最大的候选划分属性，而是使用了一个启发式：先从候选划分属性中找出信息增益高于平均水平的属性，再从中选择增益率最高的。

1.2.3、基尼指数（CART）

CART决策树使用“基尼指数”来选择划分属性。此时，数据集的纯度可用基尼值来度量：

直观来说，反映了从数据集中随机抽取两个样本，其类别标记不一致的概率。因此越小，则数据集的纯度越高。

则，属性的基尼指数定义为：

于是，我们在候选属性集合中，选择那个使得划分后基尼指数最小的属性作为最优划分属性，

* 1. 裁枝处理

剪枝是决策树学习算法对付“过拟合”的主要手段。在决策树学习中，为了尽可能正确分类训练样本，结点划分过程将不断重复，有时会造成决策树分支过多，这时就可能因训练样本学得“太好”了，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过拟合。因此，可通过主动去掉一些分支来降低过拟合的风险。

决策树剪枝的基本策略有“预剪枝”和“后剪枝”。

* 预剪枝：在决策树生成过程中，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标记为叶结点。
* 后剪枝：先从训练集生成一颗完整的决策树，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。

如何判断决策树泛化性能是否提升呢？采用留出法，即预留一部分数据用作“验证集”以进行性能评估。

**（训练集）**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **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 |
| 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）**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **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 |

* 1. 连续与缺失值

1.4.1、连续值处理

到目前为止，我们仅讨论了基于离散属性来生成决策树。现在学习任务中，常常会遇到连续属性，需要讨论如何在决策树学习中使用连续属性。

此时，连续属性离散化技术非常有用。最简单的策略是采用二分法对连续属性进行处理，这正是C4.5决策树算法中采用的机制。

给定样本集D和连续属性，假定在D上出现了n个不同的取值，将这些值从小到大进行排序，记为。基于划分点t可将D分为两个子集和，其中包含那些属性上取值不大于t的样本，而则包含那些在属性上取值大于t的样本。显然，对相邻的属性取值与来说，t在区间,中任意值所产生的划分结果相同。因此，对连续属性，我们可考察包含n-1个元素的候选划分集合：

然后，就可像离散属性值一样来考察这些划分点，选择最优的划分点进行样本集合的划分（即获得该属性的信息增益）。

信息增益公式：

其中，是样本集D基于划分点t二分后的信息增益。于是，我们就可选择使最大化的划分点。

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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 |

需要注意的是，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？？？？？？。

1.4.2、缺失值处理

现实任务中常会遇到不完整样本，即样本的某些属性值缺失。尤其是在属性数目较多的情况下，往往会有大量样本出现缺失值。如果简单地放弃不完整样本，仅使用无缺失值的样本来进行学习，显然是对数据信息极大的浪费。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **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 |

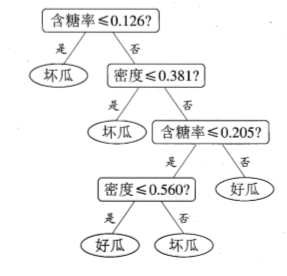
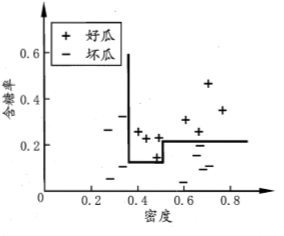
我们需要解决两个问题：

1. 如何在属性值缺失的情况下进行划分属性选择？
2. 给定划分属性，若样本在该属性值上的值缺失，如何对样本进行划分。
   1. 多变量决策树

若我们把每个属性视为坐标空间中的一个坐标轴，则d个属性描述的样本就对应了d维空间中的一个数据点，对样本分类则意味着在这个坐标空间中寻找不同类样本之间的分类边界。决策树所形成的分类边界有一个明显的特点：轴平行，即它的分类边界由若干个与坐标轴平行的分段组成。

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

显然，分类边界的每一段都是与坐标轴平行的。这样的分类边界使得学习结果有较好的可解释性，因为每一段划分都直接对应了某个属性取值。但是学习任务的真实分类边界比较复杂时，必须使用很多段划分才能获得较好的近似。

若能使用斜的划分边界（图中红色线段），则决策树模型将大为简化。“多变量决策树”就是能实现这样的“斜划分”甚至更复杂划分的决策树。以实现斜划分的多变量决策树为例，在此类决策树中，非叶子结点不再是仅对某个属性，而是对属性的线性组合进行测试；换言之，每个非叶子结点是一个形如的线性分类器，其中是属性的权重，和t可在该结点所含的样本集合属性集上学得。于是，与传统的“单变量决策树”不同，在多变量决策树的学习过程中，不是为每个非叶结点寻找一个最优划分属性，而是试图建立一个合适的线性分类器。

