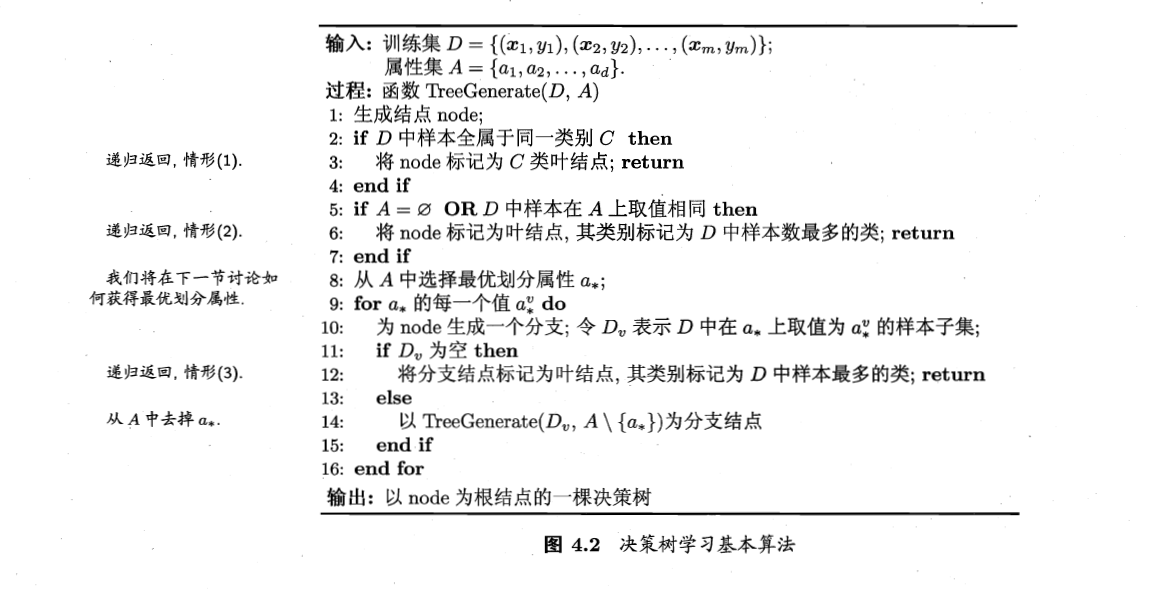
# 4.1基本流程

决策树是基于**树结构**进行决策的，决策过程的最终结论对应了判定结果。每个叶结点对应一个决策结果，其他每个结点对应一个属性测试。

伪代码如下：



决策树的搭建是一个**递归**的过程，当发生以下情形，会导致递归返回：

1）当前结点包含的样本**完全属于一个类别**无法划分。

2）当前的**属性集为空**，或者所有样本在所有属性上**取值相同**，无法划分。

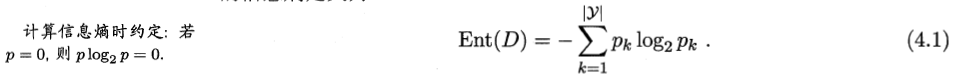
3）当前结点包含的**样本集合为空**，不能划分。

# 4.2划分选择

上述伪代码第8行“**最优化分属性的选择**”是决策树学习的关键。我们希望决策树分支结点所包含的样本尽可能属于同一类别，即“**纯度**”越高越好。

## 4.2.1信息增益

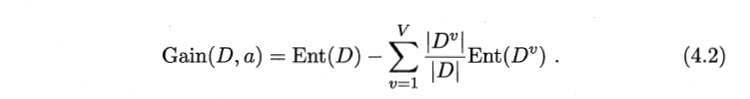
“信息熵”是度量样本集合纯度最常用的指标，假定当前样本集合D中第k类样本所占的比例为pk，则D的信息熵定义如下：



Ent(D)的**值越小**，则D的**纯度越高**。

**信息增益计算方法**：

假定离散属性a有V个可能的取值，若使用a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上有该取值的样本。我们可以根据式（4.1）计算该分支上的样本集合的信息熵。考虑到样本数越多的分支结点影响越大，给分支结点赋予权重，于是计算出用属性a对样本集D进行划分所获得的信息增益计算如下：



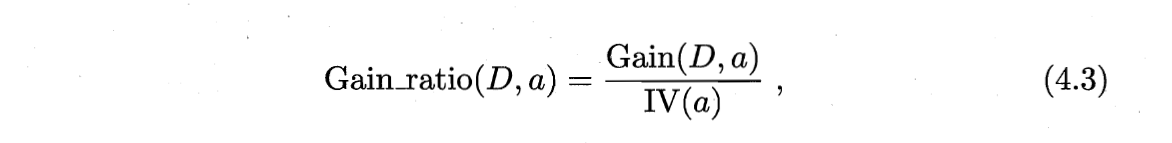
Dv代表a属性取值为v的样本集合。

一般而言，**信息增益越大**，意味着按照a属性来划分所获得的**纯度提升越大**。生成决策树的过程中，每当需要**进一步下分**（执行到第八行），可以依次计算**每个属性的信息增益**，选择**最大者**作为下一步的划分依据。

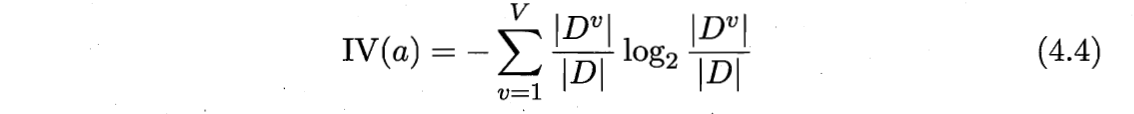
## 4.2.2增益率

但是有的情况下**并非信息增益越大越好**，比如将**样品序号**作为属性，每个样品**各自被识别成一类**，此时**信息增益非常大**，**纯度非常高**，但是并不是我们想要的结果。

产生这种现象的**本质原因**：信息增益对于**可取值较多的属性**有所**偏好**，进而可能产生不利影响。因此引入增益率也作为最优属性划分的选择标准。



其中：

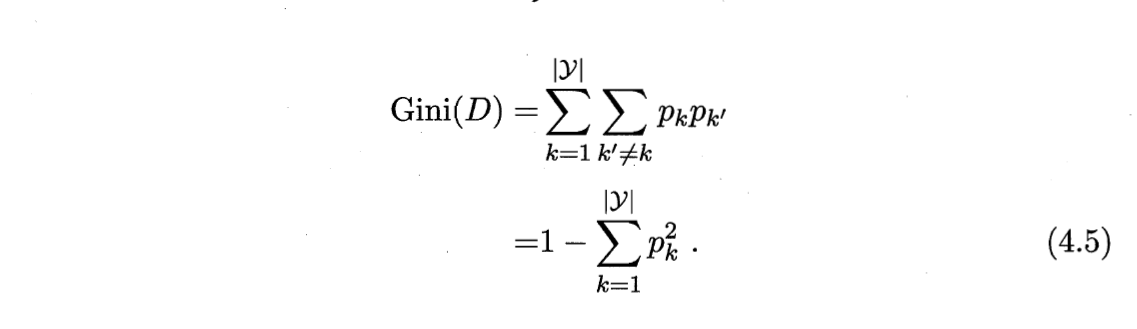


称为属性a的“固有值”。a的可取值数目越多，则IV(a)越大。

需要注意的是**增益率**对**可取值较少的属性**有所**偏好**，因此常见的做法是使用**启发式**：先从候选划分属性中找出**信息增益高出平均水平的属性**，再从中选择**增益率最高的**。

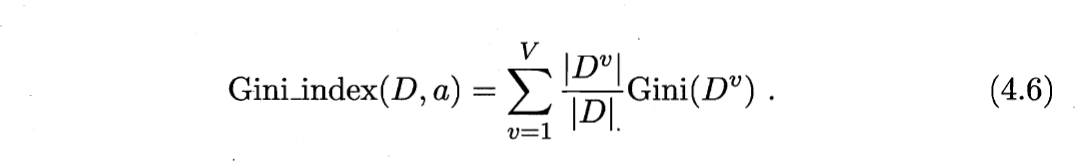
## 4.2.3基尼指数

基尼指数是**CART决策树（一种著名的决策树算法的简称）**选择划分属性的依据。数据集D的纯度可以用基尼值来度量：



直观来说，基尼指数反映了从数据集D中随机抽取两个样本，其**类别不一致的概率**。基尼值**越小**，纯度**越高**。

采用与（4.2）相同的符号表示，属性a的基尼指数：



每轮选择基尼指数最小的属性作为最优划分属性。

# 4.3剪枝处理

过拟合在决策树中表现为分支过多，需要通过剪枝降低拟合风险。

剪枝的策略有“**预剪枝**”和“**后剪枝**”。

**预剪枝思路**：在决策树生成过程中对每个结点**在划分前进行评估**，若当前结点的划分不能带来决策树泛化性能的提升，则停止划分并将当前结点标记为叶子结点。

**预剪枝特点**：降低了过拟合风险，并且显著缩短训练时间，但是由于是一种**基于贪心**的策略，为学习带来了**欠拟合的风险**。当某一步划分可能不能提升泛化性，甚至会降低泛化性，但是以它为基础的后续划分可以显著提高泛化性，这样的情况被忽略了。

**后剪枝思路**：先从训练集生成一个**完整的决策树**，然后**自底向上**地对非叶子节点进行考察，若将该结点对应的子树替换为叶子结点能带来泛化性能提升，则将该子树替换为叶结点。

**后剪枝特点**：相比预剪枝一般能**保留更多分支**，**欠拟合风险很小**，泛化性能更优，但是训练时间较长。

Q：如何评估泛化性能？

A：预留一部分数据作为验证集进行评估。

# 4.4连续与缺失值

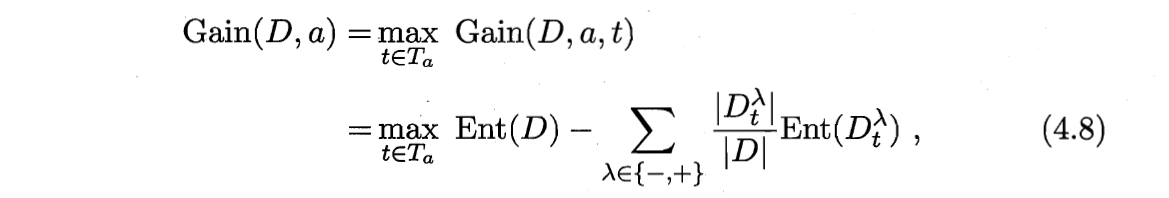
## 4.4.1连续值处理

之前处理的属性值均为离散值，容易划分，但是连续的属性值不能照搬之前的操作吗，需要先进行**离散化处理**。

最简单的处理方案：**二分法**

将区间的中位点作为分界点，两侧区间内连续的点各自抽象为离散的点进行处理。

对（4.2）**信息增益**计算公式改造如下：



Ta是对连续属性a上n-1个候选划分点的集合（一共有n个样本）。

实例详见P85

## 4.4.2缺失值处理

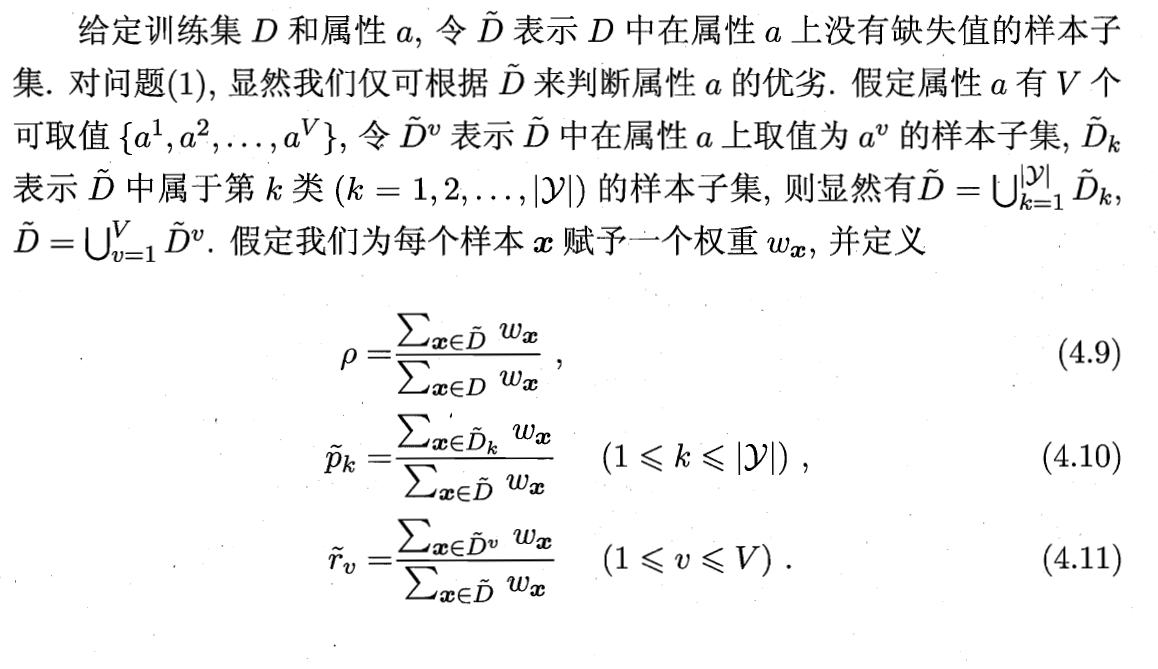
缺失值是一个较为现实的问题，采集的样本数据可能会有属性值的缺失。

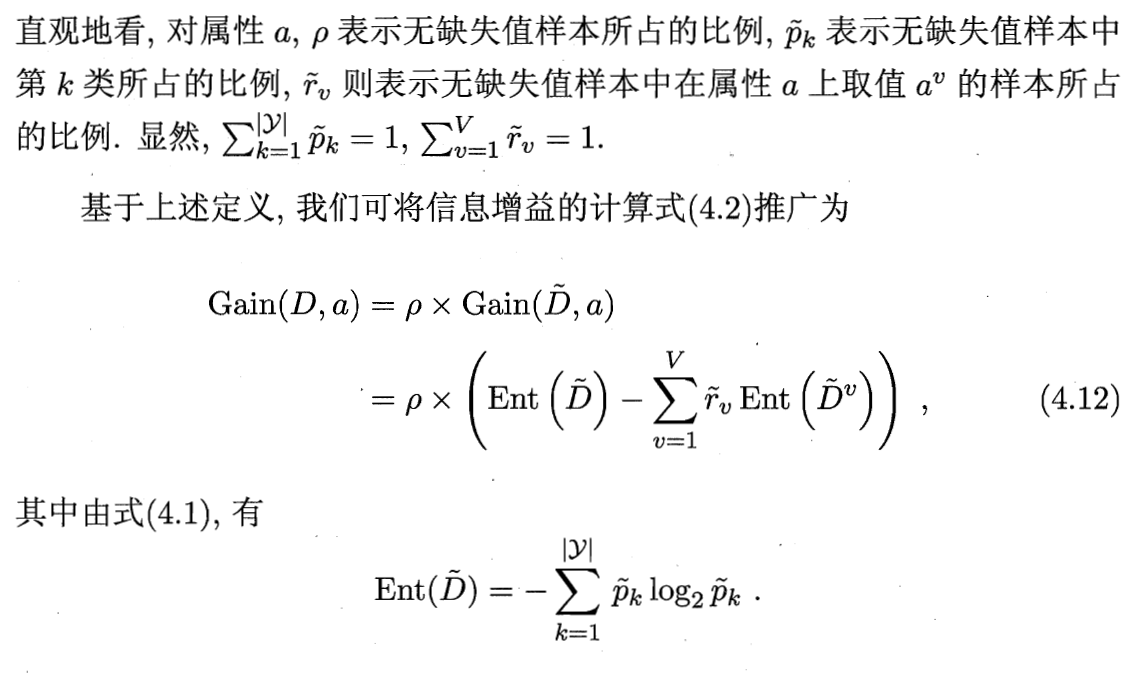
带来的问题：

1）如何在属性值缺失的情况下进行划分属性选择？（问题核心在于**如何计算信息增益**）

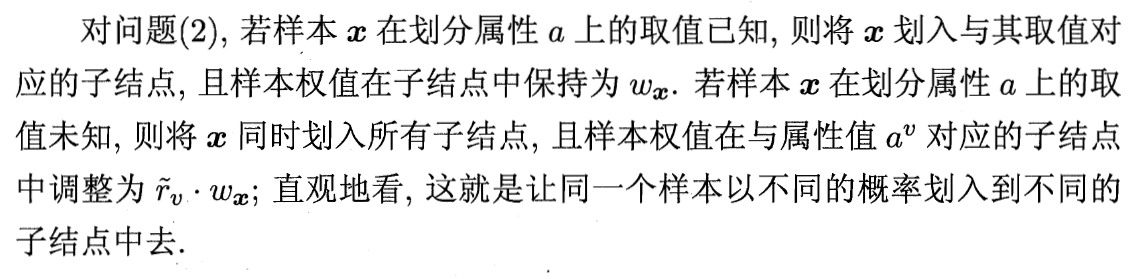
2）给定划分属性，若样本该属性缺失，如何划分？

问题一的解决方案：



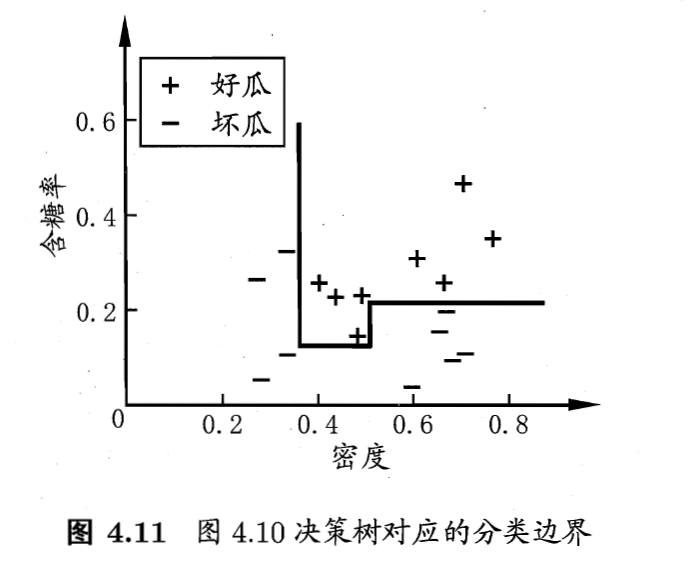


问题二的解决方案：



# 4.5多变量决策树

决策树形成的分类边界在属性空间中由**若干个与坐标轴平行的分段**组成。



实际的分类任务中边界往往比较复杂，需要**很多段**这样的小线段进行拟合才能真正勾勒分类边界，因此**决策树会相当复杂**，**预测时间开销非常大**。

希望能够用**斜线**来勾勒分类边界，简化决策树，因此提出多变量决策树。

在此类决策树中，**非叶子结点**是对**属性的线性组合**进行的测试，而非针对某个属性。

例子见P91