决策树 学习笔记

--1750562 张博

决策树基础知识

定义决策树学习（decision tree learning）是学习用来作决策的树，是一种逼近离散值目标函数的方法，学习到的函数被表示为一棵决策树。

结构：

- 一棵决策树一般包含一个根节点、若干个内部结点和若干个叶子节点
  - 叶子结点对应于决策结果
  - 每个内部结点对应于一个属性测试
  - 根结点包含全部训练样本集合，叶子结点对应决策结果
  - 从根结点到每个叶子结点的路径对应一条决策规则

决策树基本算法

学习目的：决策树学习是为了构造一棵泛化能力强，即处理待测样本能力强的决策树，基本算法遵循**自顶向下、分而治之**的策略：

基本步骤为：

> 1、选择最好的属性作为测试属性并创建树的根结点
>
> 2、为测试属性每个可能的取值产生一个分支
>
> 3、训练样本划分到适当的分支形成儿子结点
>
> 4、对每个儿子结点重复上面过程，直到所有结点都是叶子结点

决策树常见问题

最佳划分的度量问题

- **决策树学习的关键是如何选择最佳划分属性**
- 随着划分过程的不断进行，我们希望决策树的分支结点所包含的样本尽可能属于同一类别，即**结点的“不纯度”越来越低**（impurity）因此需要比较划分前（父亲结点）后划分后（所有儿子结点）不纯度的降低程度，降低越多，划分效果越好。

- 若记不纯度的降低程度为 Δ ,则用来确定划分效果的度量标准可以用下面公式定义:

  $\Delta_I = I(parent) - \sum_{j=1}^{k} \frac{N(j)}{N} I(j)$

- 其中,$I(parent)$ 是父亲结点的不纯度度量, k是划分属性取值的个数， N是父亲结点上样本的总数,$N(j)$ 是第j 个儿子结点上样本数目， $I(j)$ 是第j 个儿子结点的不纯度度量

- 不纯度度量
  - 熵：$Entropy(t) = -p(i) \sum_{i=1}^{c} log_2 p(i)$
  - 基尼系数：$Gini(t) = 1 - \sum_{i=1}^{c} p(i)^2$
  - 误分类率：$Error(t) = 1 - max(pi)$
    - 其中$p(i)$表示在当前结点中,第i类样本所占有的比例
  - 由此可得三种选择最佳划分的度量标准：
  - > ✓ 熵减最大：$\Delta_{Entropy\ \mathrm{Re}duction} = Entropy(parent) - \sum_{j=1}^{k} \frac{N(j)}{N} Entropy(j)$
    >
    > ✓ 基尼指数减最大：$\Delta_{Gini\ \mathrm{Re}duction} = Gini(parent) - \sum_{j=1}^{k} \frac{N(j)}{N} Gini(j)$
    >
    > ✓ 误分类率减最大：$\Delta_{Error\ \mathrm{Re}duction} = Error(parent) - \sum_{j=1}^{k} \frac{N(j)}{N} Error(j)$

- **信息增益**：
  - **信息熵**是度量样本集合纯度的**最常用**指标。假定样本集合D中第k类样本所占比例为p_k 则D的信息熵定义为：
  - $$\mathrm{Ent}(D) = -\sum_{k=1}^{|\mathcal{Y}|} p_k \log_2 p_k \ .$$
  - > 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. 我们可根据式(4.1) 计算出 D^v 的信息熵, 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 $|D^v|/|D|$, 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 a 对样本集 D 进行划分所获得的“信息增益”(information gain)
    >
    > $$\mathrm{Gain}(D, a) = \mathrm{Ent}(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} \mathrm{Ent}(D^v) \ . \qquad (4.2)$$

处理缺失属性值问题

>

> - 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。面对缺失属性值，决策树学习会面临两个方面的问题：
> - ✓ 如何计算含缺失值属性的划分度量、并进行最佳划分的选择？
> - ✓ 选择好最佳划分后，若样本在该属性上的值缺失，如何对样本进行划分？

解决方法：

- 放弃含缺失值的样本。
- 根据此属性值已知的其他样本来估计这个缺失的属性值
  - 1、赋给它当前结点所有样本中该属性最常见的值
  - 2、赋给它当前结点同类样本中该属性最常见的值
  - 3、为含缺失值属性的每一个可能值赋予一个概率，而不是简单地将最常见的值赋给它

处理连续属性值问题

> 由于连续属性的可取值数目不再有限，因此不能直接根据连续属性的可取值来对结点进行划分，此时连续属性离散化技术派上用场，最简单的策略是用二分法对连续属性进行处理

- 无监督离散化
  - 等深分箱法：每个分箱中样本数目一致
  - 等宽分箱法：每个分箱中的取值范围一致，把一个连续取值的区间分成若干段，每段赋一个离散值
- 有监督离散化
  - 二分法（bi-partition）：将连续取值属性按照选定的阈值分割成布尔类型。
    - 按照某个连续属性A排列训练样本，找出类标记不同的相邻样本
    - 计算类标记不同的相邻样本的属性A的取值的中间值，产生一组候选阈值，可以证明产生最大信息增益的阈值一定在这样的边界中
    - 计算与每个候选阈值关联的信息增益，选择具有最大信息增益的阈值来离散化连续属性A
  - 二分法的拓展是：最小描述长度法,简称：MDL,将连续取值的属性分割成多个区间，而不是单一阈值的两个区间

叶子结点的判定问题

- 判定当前结点为叶子结点的条件：
  - 空叶子：当前结点中样本集合为空
  - 最佳划分度量值为0的结点：
    - 纯叶子：当前结点中所有样本全部属于同一类别
    - 属性被测试完的叶子：当前结点中所有样本属性取值相同

怎样解决过拟合问题

- 剪枝（pruning）是解决过拟合问题的主要手段，基本策略有：预剪枝（prepruning）和后剪枝（post pruning）

- 预剪枝：在算法完美划分训练数据之前就停止树生长
  - 在决策树生成过程中，对每个结点在划分前进行估计，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
  - 预留一部分数据用作“验证集”来对划分前后的泛化性能进行性能评估
  - 优点：降低过拟合的风险，显著减少了决策树的训练时间开销和测试时间开销
  - 缺点：预剪枝基于贪心策略禁止某些可能在后续划分中显著提高性能的分支的展开，带来欠拟合的风险
- 后剪枝：允许树过度拟合训练数据，然后对树进行后剪枝
  - 先从训练集生成一棵完整的决策树，然后自底向上地对非叶子结点进行考察，若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
  - 优点：后剪枝决策树通常比预剪枝决策树保留了更多的分支，一般情形下后剪枝决策树欠拟合风险很小，泛化能力优于预剪枝决策树
  - 缺点：训练时间开销比预剪枝决策树、未剪枝决策树大的多
  - 示例为课本81-82
  -

表 4.2 西瓜数据集 2.0 划分出的训练集(双线上部)与验证集(双线下部)

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

未剪枝结果：

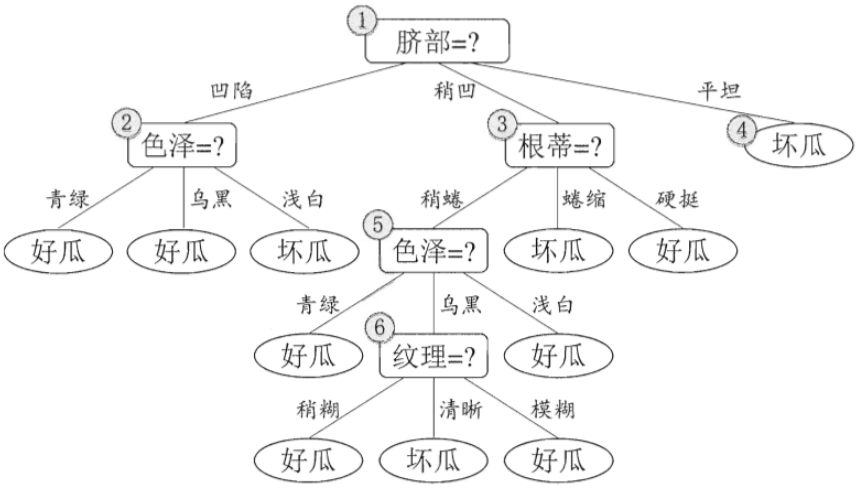
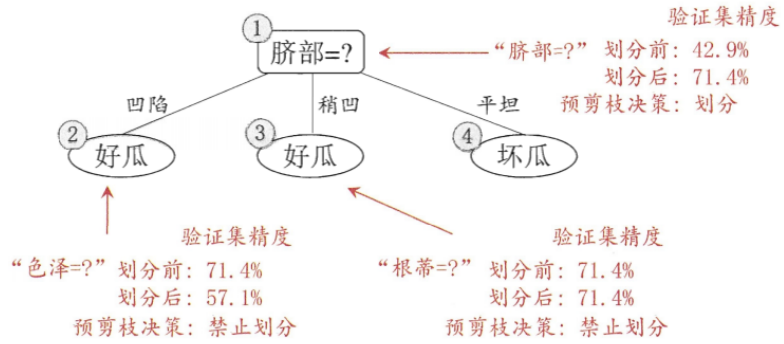
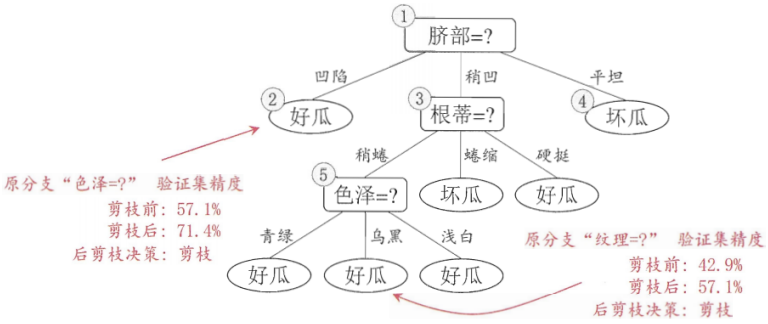


图 4.5 基于表 4.2 生成的未剪枝决策树

预剪枝结果：



后剪枝结果：



- 通过剪枝得到最终正确树的方法：
  - 使用与训练样例截然不同的一套分离的样例来评估剪枝的效果
  - 使用所有可用数据进行训练，但进行统计生长或修剪一个特定的结点是否可能改善在训练集以外的样例上的性能
  - 使用一个明确的标准来衡量训练样例和决策树的复杂度，当编码的长度最小时停止树增长

待测样本的分类问题

- 分类待测样本的方法：
  - 从决策树根节点开始测试这个结点指定的划分属性,然后按照待测样本的属性值对应的数值向下移动
  - 将上述过程再在新结点的子树上重复，直到把待测样本分到某个叶子结点为止

- 根据该叶子结点上的训练样本集计算其后验概率，最后把具有最大后验概率的类赋给待测样本
- 在计算后验概率的过程经常会采用一些常用的概率估计方 法：基于频率的极大似然估计、拉普拉斯估计、基于相似 度（距离）加权的拉普拉斯估计、m-估计，朴素贝叶斯估 计等等。

多变量决策树：

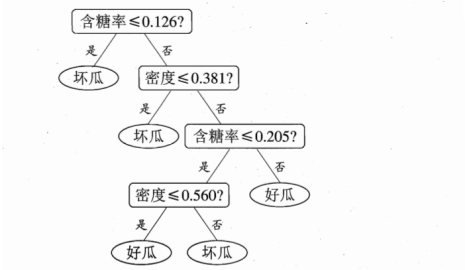
- 决策树形成的分类边界有一个明显的特点：轴平行（axis-parallel）即它的分类边界由若干个与坐标轴平性的分段组成。在学习任务的真实分类边界比较复杂时，必须使用很多段划分才能获得比较好的近似
- 在多变量决策树中，非也结点不再是针对某单个属性，而是对属性的线性组合进行测试。每一个非叶结点是一个线性分类器
- 多变量决策树的学习过程是试图为每个非叶结点建立一个合适的线性分类器
- 用决策树求解复杂边界问题：
- 

图 4.10 在西瓜数据集 3.0α 上生成的决策树

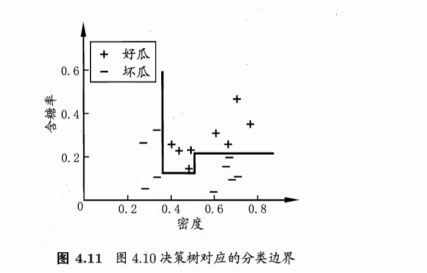


图 4.11 图 4.10 决策树对应的分类边界

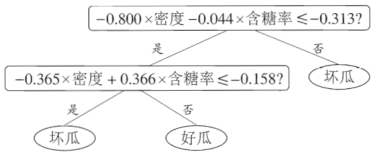
- 用多变量决策树求解：
- 

图 4.13 在西瓜数据集 3.0α 上生成的多变量决策树

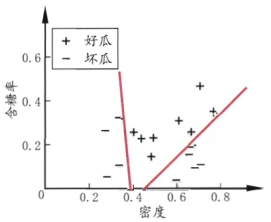


图 4.14 图 4.13 多变量决策树对应的分类边界

决策树学习理解：

- 决策树学习是以样本为基础的归纳学习方法，采用自顶向下的递归方式产生树
- 决策树生住过程是一个熵减低、信息增益、从混沌到有序的过程
- 决策树优点：
  - 对噪声数据具有很好的鲁棒性
  - 很方便地转化为分类规则
  - 训练开销小
- 决策树缺点：
  - 样本类别较多时产生的树结构很复杂
  - 决策树很容易过拟合，泛化能力差