智能软件开发
方向基础

第五章 决策树
decision tree

第3部分 决策树剪枝

张朝晖

2022~2023学年第二学期

河北师范大学软件学院
Software College of Hebei Normal University

序号	内容
1	概述
2	机器学习的基本概念
3	模型的选择与性能评价
4	数据的获取、探索与准备
5	近邻模型------分类、回归
6	决策树模型------分类、回归
7	集成学习------分类、回归
8	(朴素)贝叶斯模型------分类
9	聚类
10	特征降维及低维可视化(PCA, t-SNE)
11	总复习

本课件主要内容及有关例子，主要参考了

1. 周志华，《机器学习》
2. 李　航，《统计学习方法》

特此感谢！

思考题

1. 什么是决策树？
   决策树模型的叶子节点与特征空间、训练样本集存在什么对应关系？
2. 如何利用到达决策树某节点处的训练集度量该节点的不纯度？(三种典型的节点不纯度度量方式)
3. ID3,C4.5,CART三种典型决策树的算法实现步骤？
4. 三种决策树模型中，非叶子节点所用的特征是采用何种规则进行选择的？给出具体的选择方式.以根节点处特征选择为例，描述原理。
5. 哪种决策树模型还可用于实值函数回归？若用于回归，如何生成预测结果？
6. 给定一棵初步构建的决策树，如何对其进行剪枝？

河北师范大学软件学院
Software College of Hebei Normal University

主要内容

决策树

基于树形结构的决策模型—决策树

包括：决策树构建方法；决策树的剪枝；决策树的使用

1 非度量特征(*nonmetric features*)

2 初步认识决策树

3.决策树的构建

3.1 面向分类问题的决策树特征选择

3.2 分类树的构建

3.3 回归树的构建

4.过学习与决策树的剪枝

(1)模型的过拟合(*overfitting*)和欠拟合(*underfitting*)

- 分类模型的误差：

  训练误差，是训练样本集内错分样本占比

  泛化误差，是模型关于未知样本分类的期望误差

  训练误差越低，模型的学习能力越好；

  泛化误差越低，模型的推广能力越强

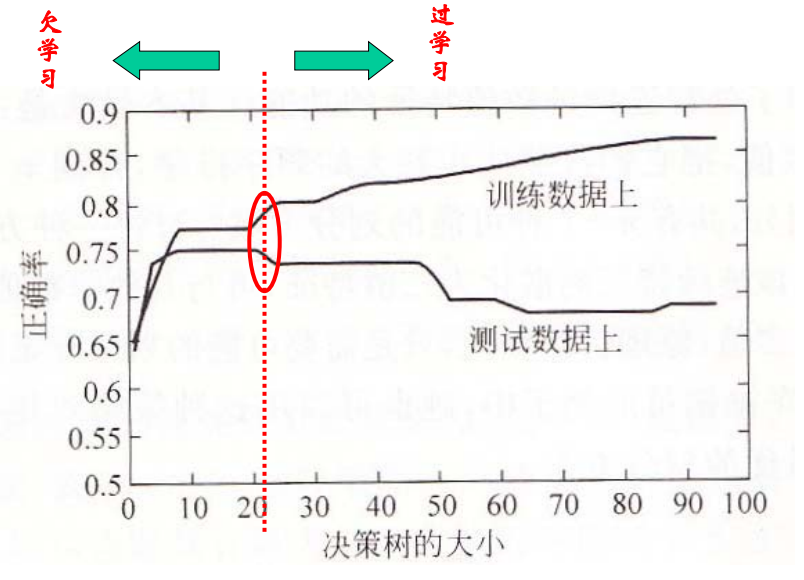
- 好的分类模型应具有低训练误差和低泛化误差。

- 具有较低训练误差的模型，其泛化误差可能高于具有较高训练误差的模型，这种情况称为模型过拟合(过学习)

- 决策树规模很小时，训练和检验误差都很大，这种情况为模型的欠拟合(欠学习)，原因是模型尚未学习到数据的真实结构。

- 随着决策树节点数的增加，模型的训练误差和检验误差都会随之下降。当树的规模变得太大时，即使训练误差还在继续降低，但是检验误差开始增大，导致模型过拟合(过学习),其原因在于过分关注采样偶然性或噪声等因素影响。

- 若训练数据缺乏具有代表性的样本，并且样本规模较小，模型也会产生过拟合。



ID3决策树的过拟合现象

(2)决策树的剪枝(*pruning*)

目的：控制决策树规模，防止模型的过拟合

策略1：先剪枝(*pre-pruning*，预剪枝)

实质—控制决策树的生长

在**完全拟合整个训练集**之前就停止决策树的生长。

决策树生长过程中，对每个节点在划分前先进行估计。

若当前节点的划分不能导致决策树泛化性能的提升，则停止划分，并将该节点标记为叶节点。

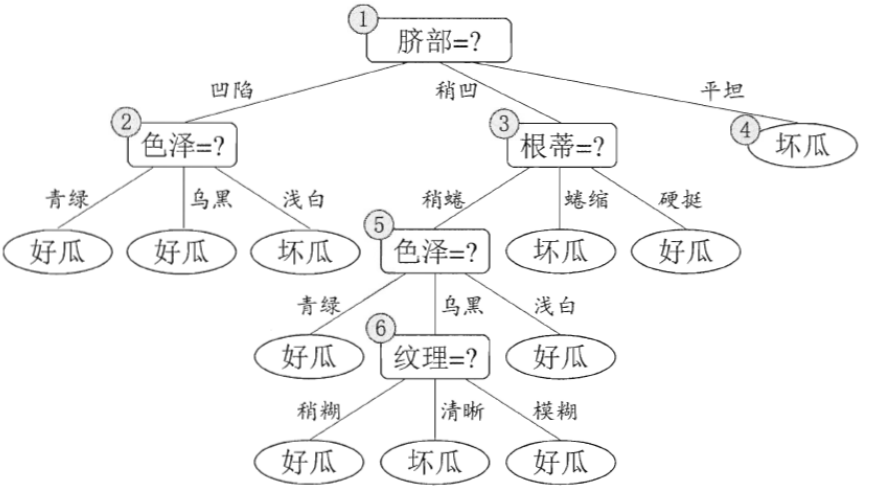
用于决策树生长的训练集

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

用于决策树预剪枝的验证集

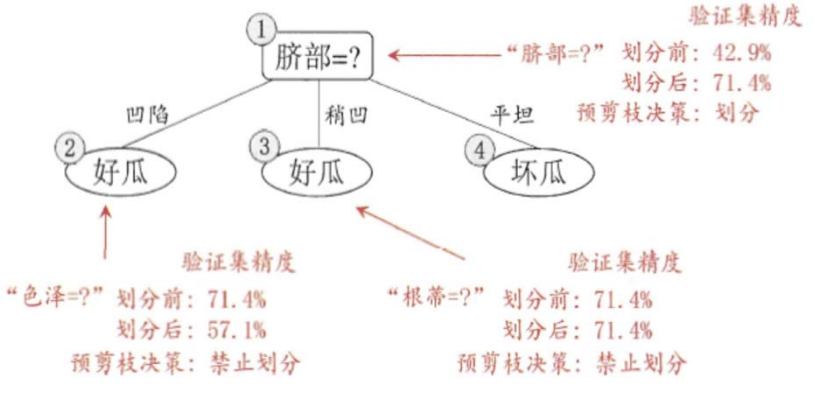
编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否

河北师范大学软件学院
Software College of Hebei Normal University



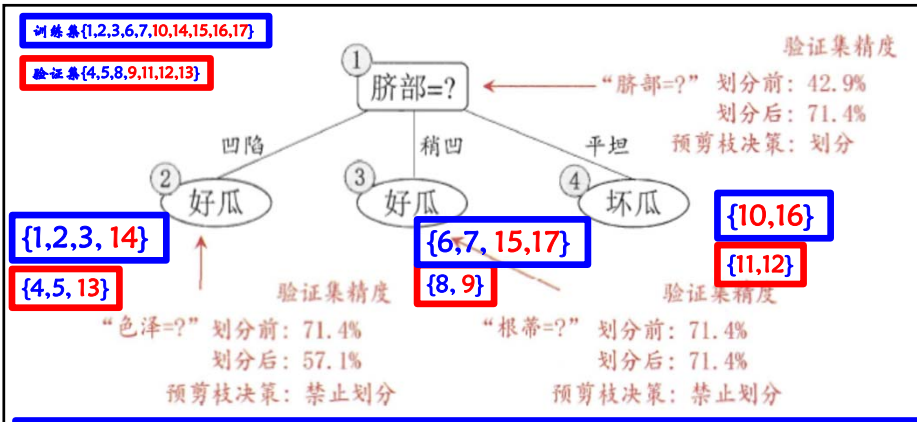
基于训练集完全生长的决策树(未剪枝)

预剪枝得到的决策树只存在一层节点划分，这样的决策树称为“决策树桩(decision stump)”



基于验证集的预剪枝的决策树

问题：上述“预剪枝”结果是如何得到的？



验证集{4,5,8,9,11,12,13}

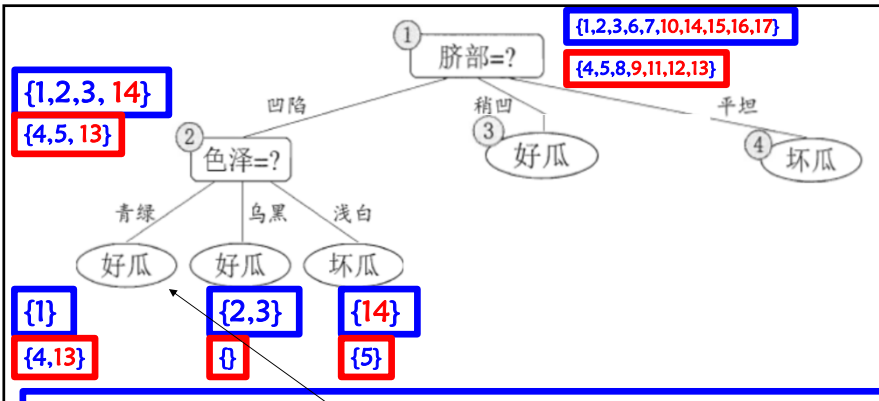
节点①：划分前，训练集{1,2,3,6,7,10,14,15,16,17} ➔ 节点①(好瓜)

验证集预测正确率＝3/7＝42.9%

划分后，得到三个子节点②(好瓜)、③(好瓜)、④(坏瓜)

验证集预测正确率＝(2+1+2)/7＝71.4%

71.4%>42.9%，因此，节点①应继续划分，得到三个子节点②③④



验证集{4,5,8,9,11,12,13}

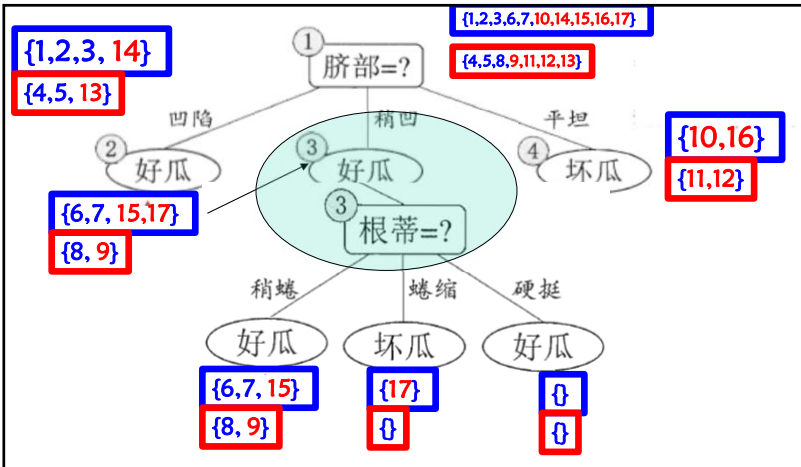
节点②：划分前，训练集{1,2,3, 14} ➔ 节点②(好瓜)

验证集总体预测正确率＝(2+1+2)/7＝71.4%

划分后，得到三个子节点，预测类别见图

验证集总体预测正确率＝(1+1+2)/7＝57.1%

71.4%>57.1%，因此，应禁止对节点②的划分



验证集{4,5,8,9,11,12,13}

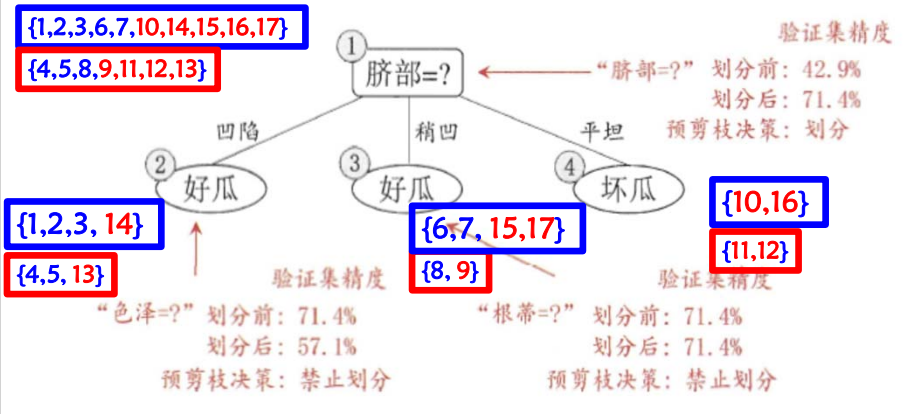
节点③：划分前，训练集{6,7,15,17}➔节点③(好瓜)

验证集总体预测正确率=(2+1+2)/7=71.4%

划分后，得到三个子节点，预测类别见图

验证集总体预测正确率=(2+1+2)/7=71.4%

划分前后验证集总体预测正确率一致，因此，应禁止对节点③的划分



验证集{4,5,8,9,11,12,13}

节点④：到达该节点的训练样本子集{10,16}，该节点为纯节点，没必要划分。

策略2：后剪枝(*post-pruning*)

实质： 决策树生长后处理，合并分枝

初始阶段--决策树按照最大规模生长。

剪枝阶段--修剪完全增长的决策树。

自底向上，对非叶子节点进行考察。

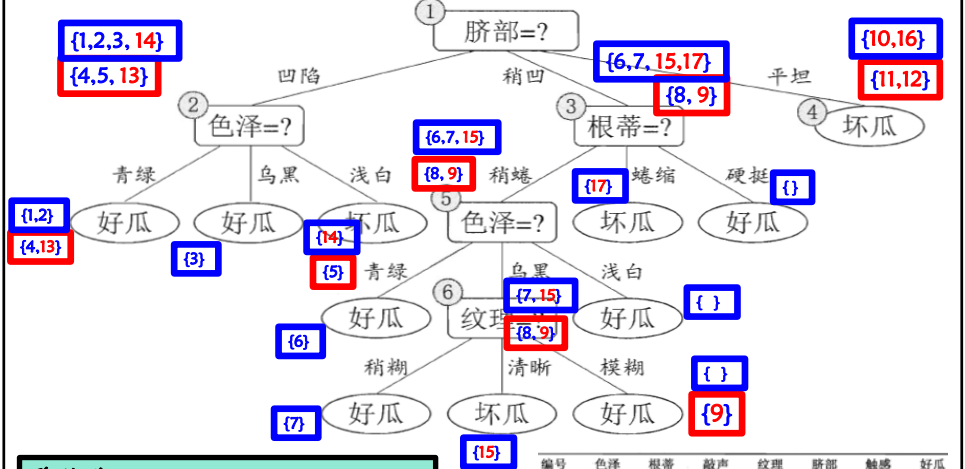
若将该节点的子树替换为叶子节点能带来决策树泛化性能的提升，则砍掉该子树，以该节点作为叶子节点。

阶段1. 基于训练集的完整决策树生长。

训练集{1,2,3,6,7,10,14,15,16,17}

验证集{4,5,8,9,11,12,13}

A



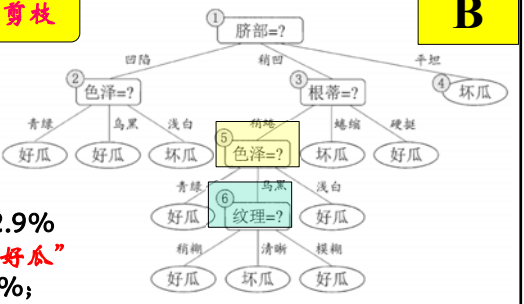
剪枝前：
验证集预测正确率=3/7=42.9%

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

阶段2. 基于验证集的完整决策树后剪枝

B

自底向上，逐层剪枝

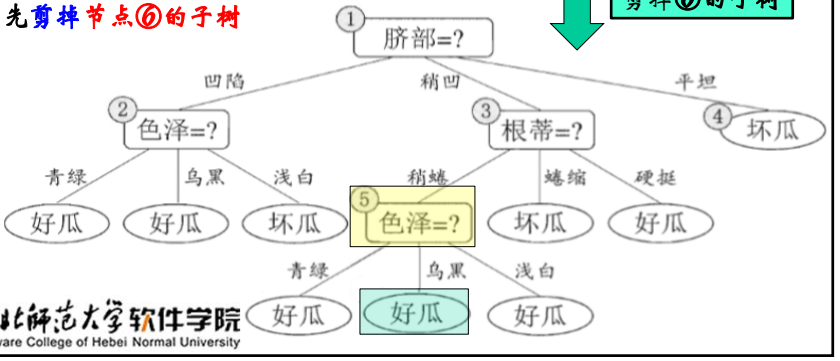


首先，考查是否剪掉节点⑥的子树

- 若不剪
  验证集总体预测正确率=3/7=42.9%
- 若剪枝，则节点⑥成为叶节点“好瓜”
  验证集总体预测正确=4/7=57.1%；

剪掉⑥的子树

因此，首先剪掉节点⑥的子树



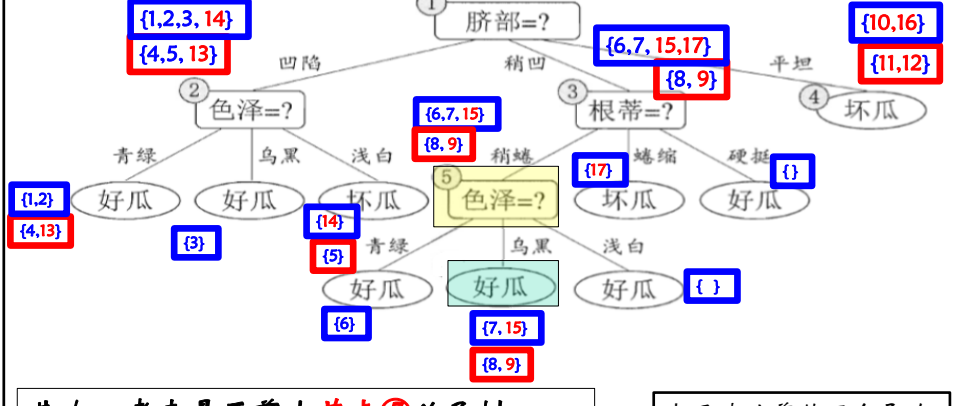
河北师范大学软件学院
Software College of Hebei Normal University

阶段2. 基于验证集的完整决策树后剪枝。

训练集{1,2,3,6,7,10,14,15,16,17}

验证集{4,5,8,9,11,12,13}

C



其次，考查是否剪去节点⑤的子树

- 若不剪枝，则验证集总体预测正确=4/7=57.1%
- 若剪枝，则节点⑤成为新的叶子节点（“好瓜”）
  验证集总体预测正确率=4/7=57.1%

由于此处剪枝不会导致验证集预测正确率下降，基于奥克姆剃刀准则，应剪去节点⑤的子树

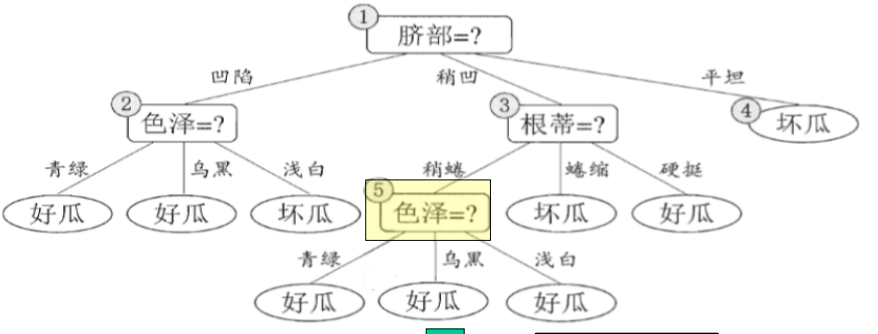
河北师范大学软件学院
Software College of Hebei Normal University

阶段2. 基于验证集的完整决策树后剪枝。

训练集{1,2,3,6,7,10,14,15,16,17}

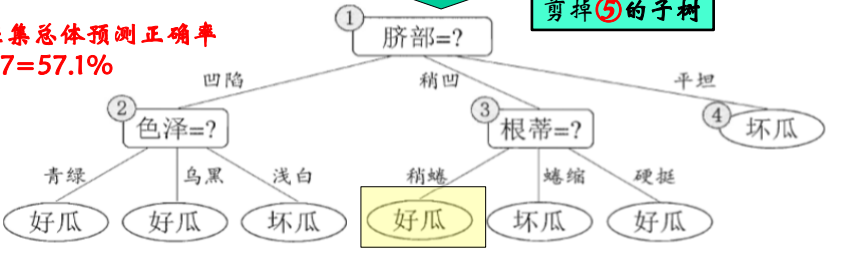
验证集{4,5,8,9,11,12,13}

D



剪掉⑤的子树

验证集总体预测正确率
=4/7=57.1%

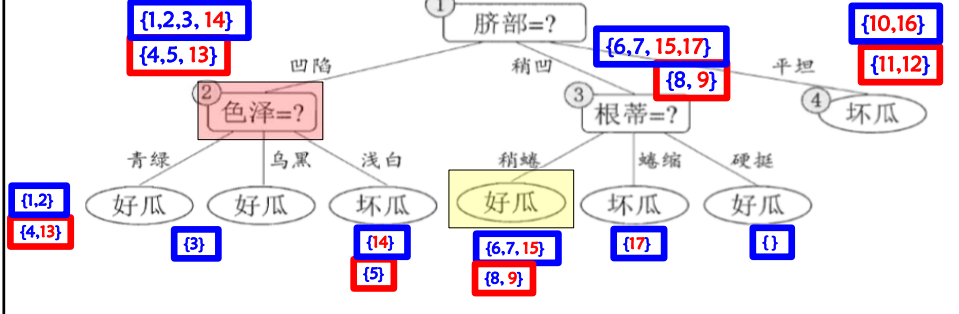


阶段2. 基于验证集的完整决策树后剪枝。

训练集{1,2,3,6,7,10,14,15,16,17}

验证集{4,5,8,9,11,12,13}

E



第三，考查是否剪去节点②的子树

- 若不剪枝，则验证集总体预测正确=4/7=57.1%
- 若剪枝，则节点② 成为新的叶子节点（“好瓜”）
  验证集总体预测正确率=5/7=71.4%

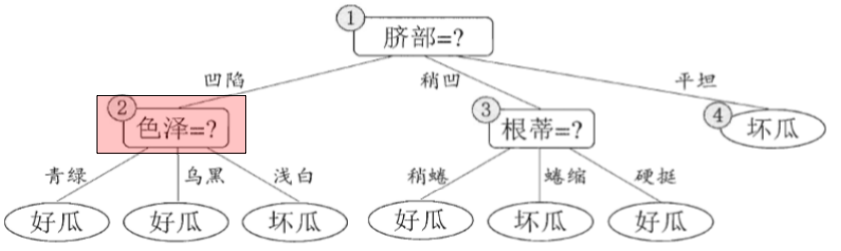
由于此处剪枝后，验证集预测正确率将会上升，应剪去节点②的子树

F

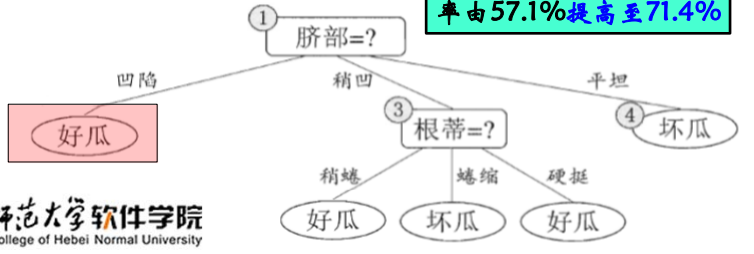
阶段2. 基于验证集的完整决策树后剪枝。

训练集{1,2,3,6,7,10,14,15,16,17}

验证集{4,5,8,9,11,12,13}



剪掉节点②的子树，使验证集的总体预测正确率由57.1%提高至71.4%



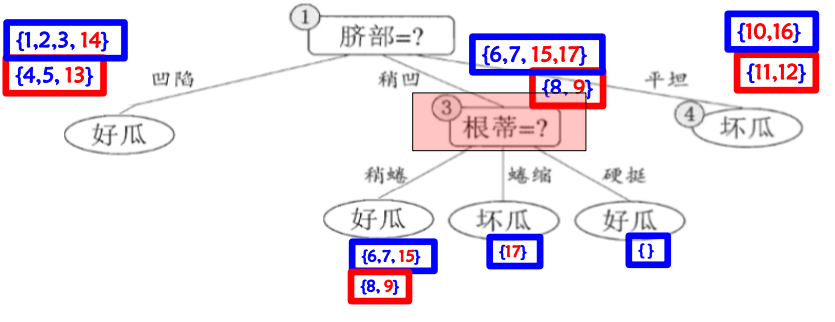
河北师范大学软件学院
Software College of Hebei Normal University

G

阶段2. 基于验证集的完整决策树后剪枝。

训练集{1,2,3,6,7,10,14,15,16,17}

验证集{4,5,8,9,11,12,13}



第四，考查是否剪去节点③的子树

- 若不剪枝，则验证集总体预测正确=5/7=71.4%
- 若剪枝，则节点③成为新的叶子节点（“好瓜”）
  验证集总体预测正确率=5/7=71.4%，不变

由于此处剪枝不会导致验证集预测正确率下降，基于奥克姆剃刀准则，应剪去节点③的子树

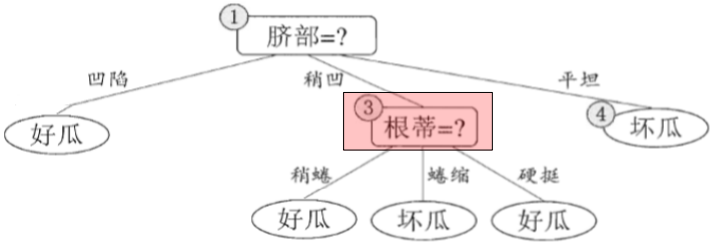
河北师范大学软件学院
Software College of Hebei Normal University

H

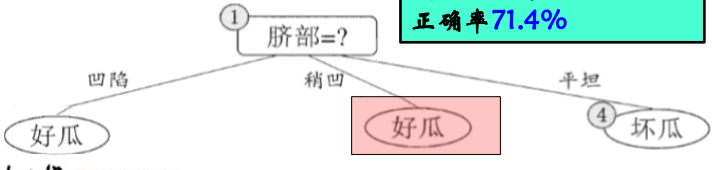
阶段2. 基于验证集的完整决策树后剪枝。

训练集{1,2,3,6,7,10,14,15,16,17}

验证集{4,5,8,9,11,12,13}



剪掉节点③的子树，在简化模型的同时，不会损失验证集的总体预测正确率71.4%



I

阶段2. 基于验证集的完整决策树后剪枝。

训练集{1,2,3,6,7,10,14,15,16,17}

验证集{4,5,8,9,11,12,13}

{1,2,3,14}
{4,5,13}

{6,7,15,17}
{8,9}

{10,16}
{11,12}

验证集{4,5,8,9,11,12,13}

第五，考查是否剪掉节点①领衔的树

剪掉之前：验证集预测正确率=(2+1+2)/7=71.4%

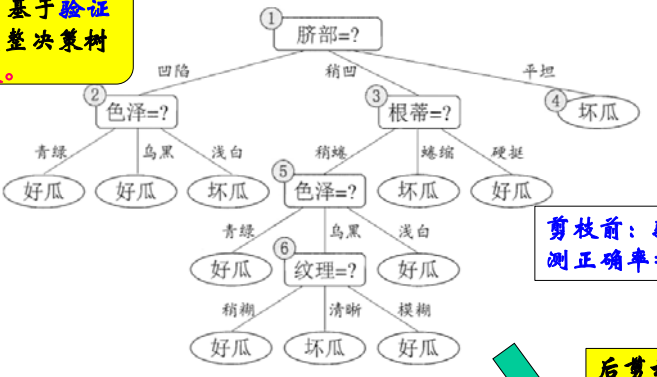
剪掉之后：决策树将成为单节点树，预测类别为“坏瓜”

验证集预测正确率=3/7=42.9%

由于71.4%>42.9%，因此，节点①应继续划分，得到三个子节点。

阶段2. 基于验证集的完整决策树后剪枝。

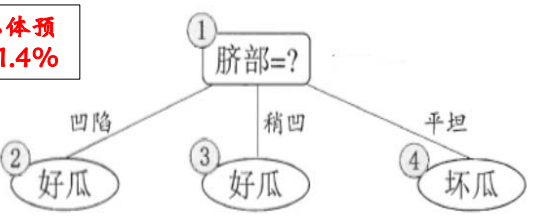
J



剪枝前：验证集总体预测正确率=4/7=57.1%

后剪枝

剪枝后：验证集总体预测正确率=5/7=71.4%



策略2：后剪枝(*post-pruning*)

剪枝规则

例：一种后剪枝算法--最小代价与复杂性的折中：

平衡“错误率的增加”与“模型复杂程度的降低”

决策树的后剪枝(1)

设决策树T的叶节点数目为$|T|$，叶节点序号$t=1,\ldots,|T|$

训练样本集到达叶节点t的样本数为N_t，

其中第k类的样本数为$N_{tk}, k=1,\ldots,K$.

叶节点t的**不纯度**$H_t(T)$ $\quad H_t(T)=-\sum_{k=1}^{K}\frac{N_{tk}}{N_t}\log\frac{N_{tk}}{N_t}$

决策树T关于训练样本集的拟合误差

(训练集划分的综合不纯度):

$$C(T)=\sum_{t=1}^{|T|}N_tH_t(T)=-\sum_{t=1}^{|T|}N_t\left(\sum_{k=1}^{K}\frac{N_{tk}}{N_t}\log\frac{N_{tk}}{N_t}\right)$$

$$=-\sum_{t=1}^{|T|}\sum_{k=1}^{K}N_{tk}\log\frac{N_{tk}}{N_t}$$

决策树的后剪枝(2)

决策树T关于训练样本的拟合误差:

$$C(T)=\sum_{t=1}^{|T|}N_tH_t(T)=-\sum_{t=1}^{|T|}N_t\sum_{k=1}^{K}\frac{N_{tk}}{N_t}\log\frac{N_{tk}}{N_t}=-\sum_{t=1}^{|T|}\sum_{k=1}^{K}N_{tk}\log\frac{N_{tk}}{N_t}$$

模型的损失函数=模型关于训练样本的拟合误差+模型的复杂度

模型的损失函数: $\quad C_\alpha(T)=C(T)+\alpha|T|$

决策树的后剪枝，就是在给定$\alpha\geq 0$ 的前提下，

选择具有最小$C_\alpha(T)$的子树。

决策树T的后剪枝算法

输入： 生成算法产生的整棵树T，控制参数α

输出： 对树T修剪，得到的子树T_α

步骤：

***STEP*1.**基于训练集计算每个节点(不只是叶节点)的经验熵。

***STEP*2.**递归地从树的叶节点向上回溯。

设**一组叶节点**回溯到其父节点**之前**、**之后**的**整体树**分别为T_{Before}和T_{After};对应的损失函数值分别为

$$C_\alpha(T_{Before}) = C(T_{Before}) + \alpha|T_{Before}|$$

$$C_\alpha(T_{After}) = C(T_{After}) + \alpha|T_{After}|$$

若$C_\alpha(T_{After}) \le C_\alpha(T_{Before})$，则剪枝，将**叶节点**的父节点作为新的叶节点

STEP*3.**返回STEP*2，直到不能继续为止，得到损失函数最小的子树T_α

(3)关于“剪枝”的讨论：

- **预剪枝**可能过早终止决策树的生长
  存在欠拟合风险.

- **后剪枝**倾向于产生更好的结果
  根据完全生长的决策树作出剪枝决策，需要更多时间开销.
  欠拟合的风险小,泛化性能优于预剪枝决策树.

- “预剪枝”与“后剪枝”结合