

# 研究生算法课课堂笔记

上课日期：2016.09.29

第(2)节课

组长学号及姓名：黄凯鹏 1601214543

组员学号及姓名：付钰雯 1601214540

卢苇 1601214553

## 一. 内容概要

本节课所学的内容包括以下几点：

1. 决策树需要解决过拟合的原因，及解决决策树过拟合的几种算法；
2. 第二次算法作业第4题解析。

## 二. 详细内容

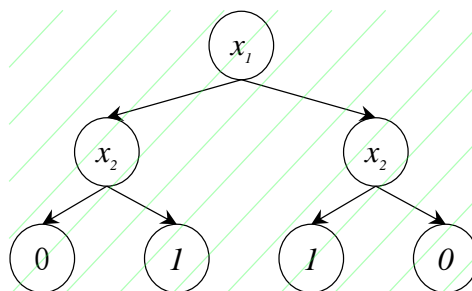
1. 决策树的算法是贪心的，贪心法是否一定能找到正确划分且高度最短的树？

答：贪心法不一定是最优解，但只有极少情况（如异或）不是最优解。

反例：

$x_1$	$x_2$	$x_3$	...	$x_{10}$	$y = x_1 \oplus x_2$
$\begin{matrix} 0 \\ \vdots \\ 0 \end{matrix}$	$\begin{matrix} 0 \\ \vdots \\ 0 \end{matrix}$				$\begin{matrix} 0 \\ \vdots \\ 0 \end{matrix}$
$\begin{matrix} 0 \\ \vdots \\ 0 \end{matrix}$	$\begin{matrix} 1 \\ \vdots \\ 1 \end{matrix}$				$\begin{matrix} 1 \\ \vdots \\ 1 \end{matrix}$
$\begin{matrix} 1 \\ \vdots \\ 1 \end{matrix}$	$\begin{matrix} 0 \\ \vdots \\ 0 \end{matrix}$				$\begin{matrix} 1 \\ \vdots \\ 1 \end{matrix}$
$\begin{matrix} 1 \\ \vdots \\ 1 \end{matrix}$	$\begin{matrix} 1 \\ \vdots \\ 1 \end{matrix}$				$\begin{matrix} 0 \\ \vdots \\ 0 \end{matrix}$

最优解法：



但是贪心法构造不出这样的决策树。

贪心法 (ID3):  $Gain(Y, x_1) = 1 - 1 = 0$ ;  $Gain(Y, x_2) = 1 - 1 = 0$





基于 $x_1$ 或 $x_2$ 为节点划分后，在每个子节点中 $y$ 还是均匀分布的，所以每个子节点的熵仍为1，加权平均后也为1，故其信息增益为0；而 $x_3$ 到 $x_{10}$ 的划分有可能不会引起 $y$ 的均匀分布，其信息增益不为0。由于贪心算法会选择info gain大的作为根节点，所以会选择 $x_3$ 到 $x_{10}$ 作为根节点，从第二层开始才有可能选择 $x_1$ 或 $x_2$ 。

虽然贪心法不一定能得到最优解，但是在实际中得不到最优解的情况是比较少见的。

**Occam's Razor (剃刀原理):** 如果有几种解释都成立，那么最简单的解释往往是正确的。

为什么：简单的假设数目比较少，能够划分正确是偶然性的几率也较小。  
 争议：以数量少分优劣不合理。

## 2. 决策树构建过程算法：

3.  Recursion	对于不同子集自顶向下递归，直到终止条件	√
 Greedy	每次都是找信息增益最大的属性作为节点	√
 Dynamic programming	未使用	×
 Divide and conquer	每次对于整个空间分裂成子集进行构建	√

**fitting**（过拟合）：对于  $h \in H$  来说，如果存在另一个  $h' \in H$  在训练集上效果较差，但在测试集上效果就较好，那么称  $h$  是过拟合的。

普遍情况：在训练集上训练如果追求过好的训练效果，会出现过拟合，此时模型在测试集效果较差，泛化能力低。例如：“第十名现象”。

大数据发展现况：模型的表达能力是足够的，在不限制复杂度的情况下，一定能找到解释问题的模型，主要问题是避免 **overfitting**。

## 4. 决策树避免过拟合的思想：限制每个叶节点的最小样本数，即使分类不纯也停止分裂。或者分类后子节点的样本数过少了也停止分裂。

决策树避免过拟合的两种方法：

	算法	优点	缺点
<b>Pre-prune</b> （预剪枝）	数据的分裂在统计上无意义，即分裂前后的熵相差较小，信息增益不够大，此时停止分裂	效率高	可能剪掉有用的枝：再分裂几步信息增益可能又增大 例如：异或情况下，单个属性区别度低，多个属性联合区分度高
<b>Post-prune</b> （后剪枝）	先长好全部的节点，从后往前剪枝，若剪枝后在验证集上的效果变好，则剪掉该枝。 （实际中使用更多）	结果好	效率低

## 5. Reduced-Error Prune：使用验证集来剪枝，剪掉后在验证集中性能上升的剪掉，多个节点，剪掉性能上升最大的。

Training set	用于训练模型；决定跟属性直接相关的普通参数，例如决策平面法向量 $w$
Validation set	控制复杂度；决定超参数（不是跟属性直接相关的参数），例如决策树的高度
Test set	用于测试模型

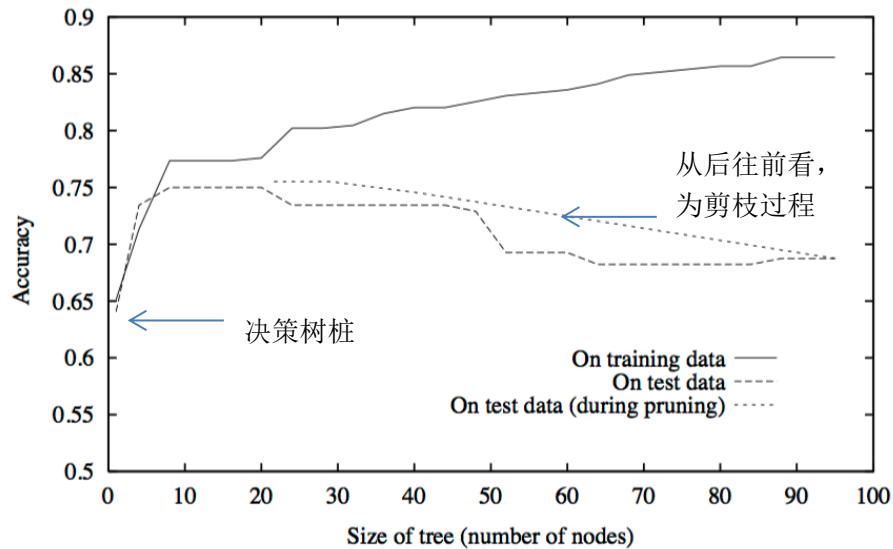
验证集训练过多也可能出现过拟合。

## 6. 对于 64 页图的讲解：

**Decision stump**（决策树桩）：节点数为 1 的决策树，此时 **bias** 很高。

训练集性能随节点增多而不断提升，测试集性能随节点增多先提升后减退，即出现过拟合的情况。

## 7. 对于 67 页图的讲解



**剪枝过程测试集性能变化：**剪枝过程，测试集性能不断上升，到某处停止，此时再剪枝有害。

8. 决策树生长过程中，有没有可能从根节点到叶节点的过程中，同一个属性用了两次？

答：若使用两次，子节点的区分度为 0，信息增益为 0。所以必然不会重复选择同一个属性。

9. **Rule post-pruning:** 形式不是树，而是转化为 rules，用逻辑表示。

好处：剪枝时可以不从叶节点开始，从后往前按顺序进行。可以剪掉 rule 中任一属性（条件）。

10. **第二次作业第 4 题讲解：**

这道题目就是求逆序数，目前在  $O(n \lg n)$  的复杂度下求逆序数的算法一般是利用树状数组或归并排序。老师在课上简单讲解了一下归并排序的思想，其中主要思想是将两个子串合并时，分别比较第一串中的数和第二串中的数，从而得到逆序数。

$$\begin{array}{c}
 \begin{array}{ccccc}
 & i & & & \text{last} \\
 & | & & & | \\
 A = \{1, & \boxed{4, 6, 7, 9} & \}
 \end{array} \\
 \begin{array}{c}
 j \\
 | \\
 B = \{\boxed{2}, 3, 5, 10, 13, 21\}
 \end{array}
 \end{array}$$

例如：现在将要  $A$  和  $B$  合并，有两个 index:  $i$  和  $j$ ，当  $i$  指向 4,  $j$  指向 2 时， $A[i]$  比  $B[j]$  要大，说明 4 后面的数都要比 2 大，此时 2 的逆序数要加上  $A$  中被圈中的数的个数。当  $i$  和  $j$  继续向后移动时， $i$  和  $j$  之前的数在算逆序数时不再重复考虑。