第4章：决策树

2025年7月

本讲关键问题

（1） 什么是决策树？其与OLS有何联系与区别？

（2） 如何基于决策树估计条件期望？

（3） 如何基于信息准则生成一颗决策树？

（4） 如何避免决策树中的过拟合问题？

关键词：决策树、样本空间、回归树、分类树、熵、信息增益、信息增益率、基尼系数、预剪枝、后剪枝。

一、回顾：线性回归

- 在经济学研究中，研究者期望通过一系列协变量X来预测因变量Y。基于均方损失（Mean Square Loss）法则，其最优的预测值$g(x)$是因变量Y给定协变量X的条件期望：

$$g(x) = E(Y|X = x) = argmin_g E[(Y - g)^2|X = x]$$

- 在之前的章节中，我们介绍了如何基于线性拟合来估计上述条件期望，即$g(x) = x'\beta$ （4.1）
- 基于模型(4.1)，我们可以采用最小二乘（OLS）来进行估计。当协变量X的维数较高时，我们也可以利用LASSO等模型选择方法加以估计。
- 然而，线性拟合方法的合理性依赖于模型设定是否正确，即当其设定背离真实的模型时，我们得到的估计/拟合结果不可靠，从而要寻求更为稳健的估计方法。

二、决策树

为了得到更为稳健的估计结果，我们舍弃模型(4.1)的设定，直接基于可观测数据来拟合条件期望$g(X)$。决策树是非线性拟合中的一种经典方法。其主要思想是将协变量X所属的空间$\mathcal{X}$划分成若干（广义）长方形，然后在每一个（广义）长方形中分别进行预测。

- 被解释变量连续——回归树
- 被解释变量离散——分类树

二、决策树：分割特征空间

我们通过一个简单的例子加以说明。假设Y表示企业的产出，协变量X表示影响企业产出的因素。我们考虑两个维度：$(X1, X2)$，分别表示企业资本以及劳动力，其构成的空间为$\mathcal{X}: [0, a] \times [0, b], a, b>0$。决策树方法是通过一系列递归方式将$\mathcal{X}$划分成若干子空间$\mathcal{X}_1, \cdots, \mathcal{X}_K$,子空间$\mathcal{X}_1, \cdots, \mathcal{X}_K$满足$\mathcal{X} = \mathcal{X}_1 \cup \cdots \cup \mathcal{X}_K$以及$\emptyset = \mathcal{X}_1 \cap \cdots \cap \mathcal{X}_K$ 。图1展示了一种决策树对于空间$\mathcal{X}$的划分方式·

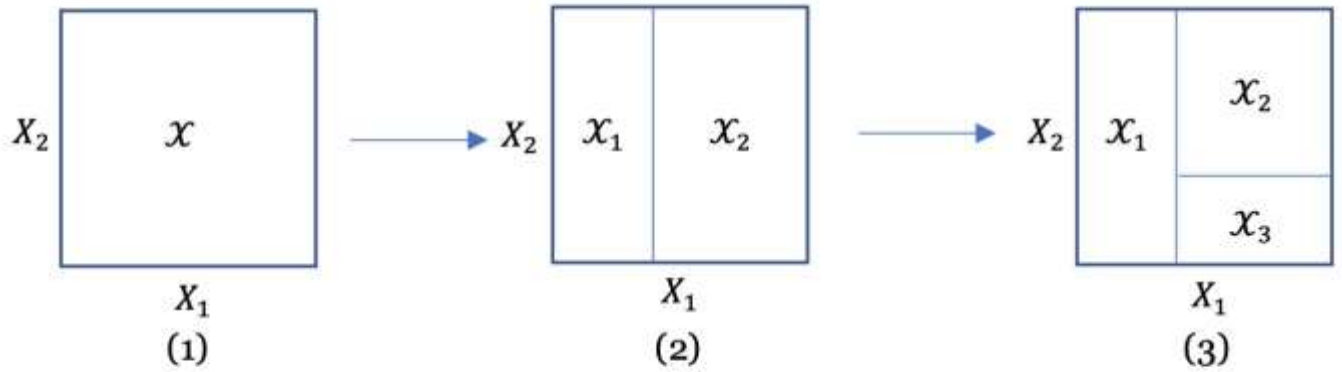


图 1:决策树样本空间划分步骤

二、决策树：估计

- 基于划分后的空间$\mathcal{X}$，我们可以在每一个子空间$\mathcal{X}_k$中给出一个关于Y的预测值，记为β_k。最后，我们可以将条件期望表示成

$$g(x) = \sum_{k=1}^{K} \beta_k 1\{x \in \mathcal{X}_k\}. (4.2)$$

- 模型(4.2)中的参数β_k可以等价表示为以下最优化问题的解：

$$\beta = (\beta_1, \beta_2, \cdots, \beta_K) = argmin_{b_1, b_2, \cdots, b_K} E\left[Y - \sum_{k=1}^{K} b_k 1\{X \in \mathcal{X}_k\}\right]^2 . (4.3)$$

二、决策树：估计

假设我们有一组观测值$\{Y_i, X_i\}_{i=1}^n$。根据方程(4.3)，参数β所对应的估计量$\hat{\beta}$可以表示成以下最优化问题的解：

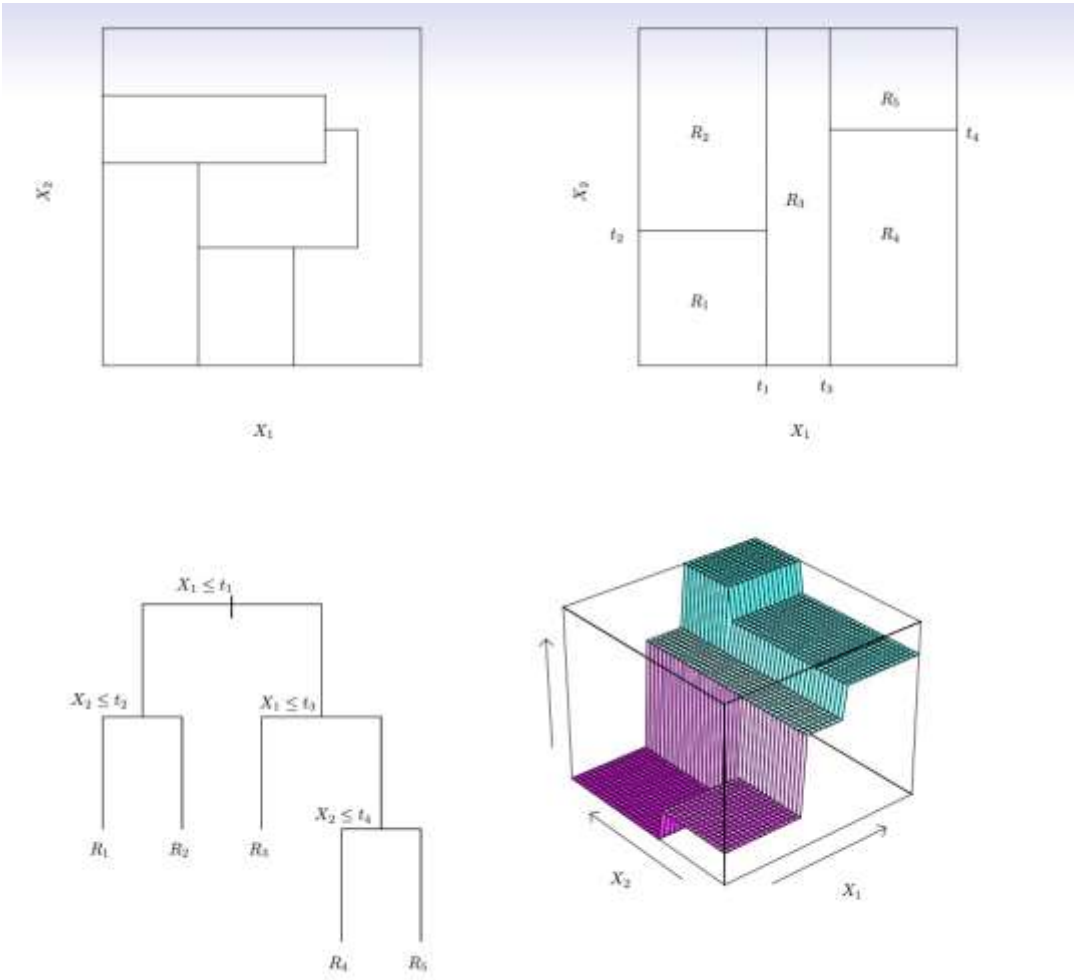
$$\hat{\beta} = argmin_{b_1,b_2,\cdots,b_K} \sum_{i=1}^{n} \left[Y_i - \sum_{k=1}^{K} b_k 1\{X_i \in \mathcal{X}_k\} \right]^2 .$$

求解上述最优化问题可得

$$\hat{\beta}_k = \frac{1}{n_k} \sum_{X_i \in \mathcal{X}_k} Y_i , (4.4)$$

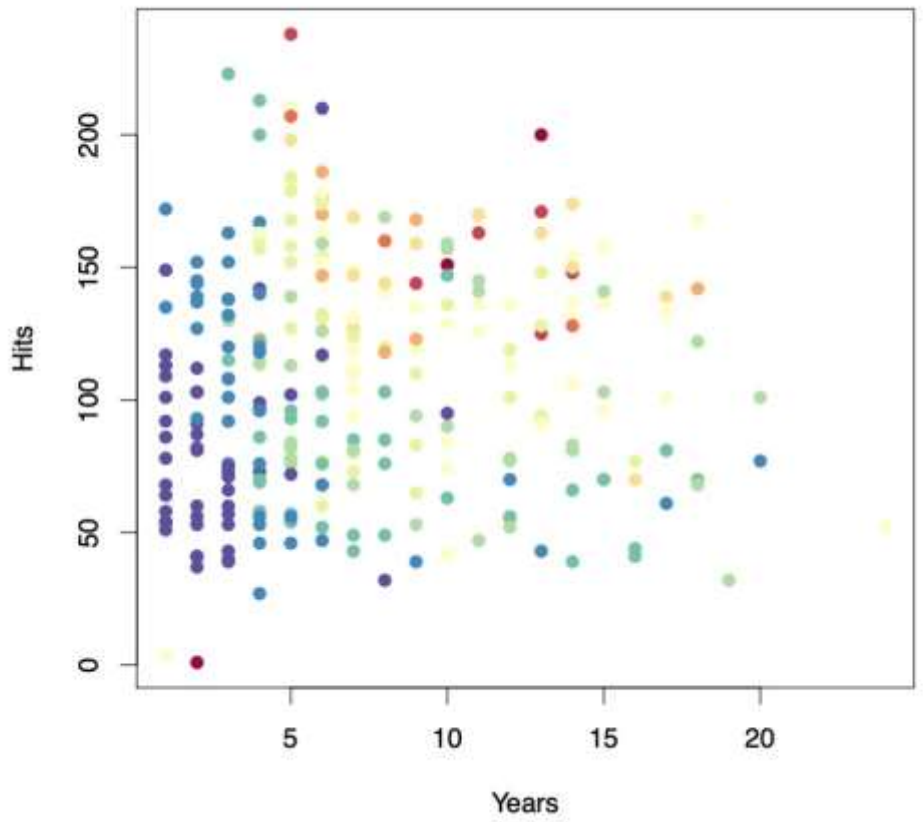
其中n_k表示集合$\{i: X_i \in \mathcal{X}_k\}$中元素的个数。等式(4.4)表明了决策树的预测准则，即在每一个子空间中，其返回的预测值为该子空间中因变量的平均值。

二、决策树：估计



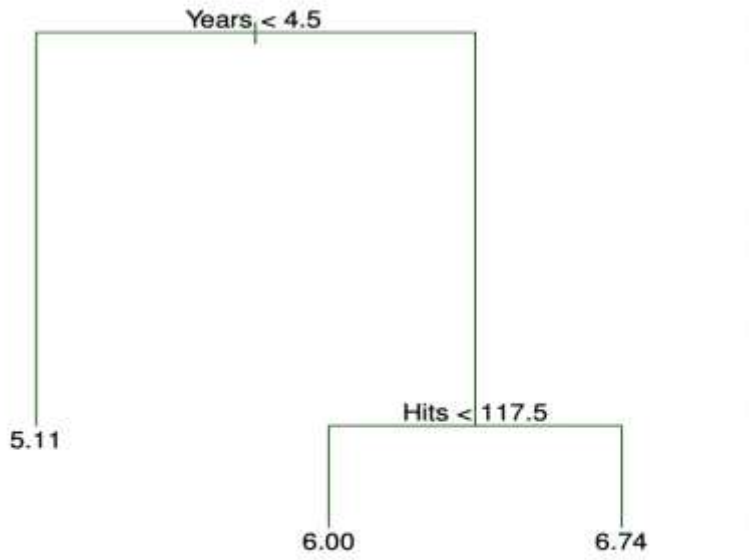
思考：图片中特征空间的划分方式，哪一种是由决策树生成的？

二、决策树：实例



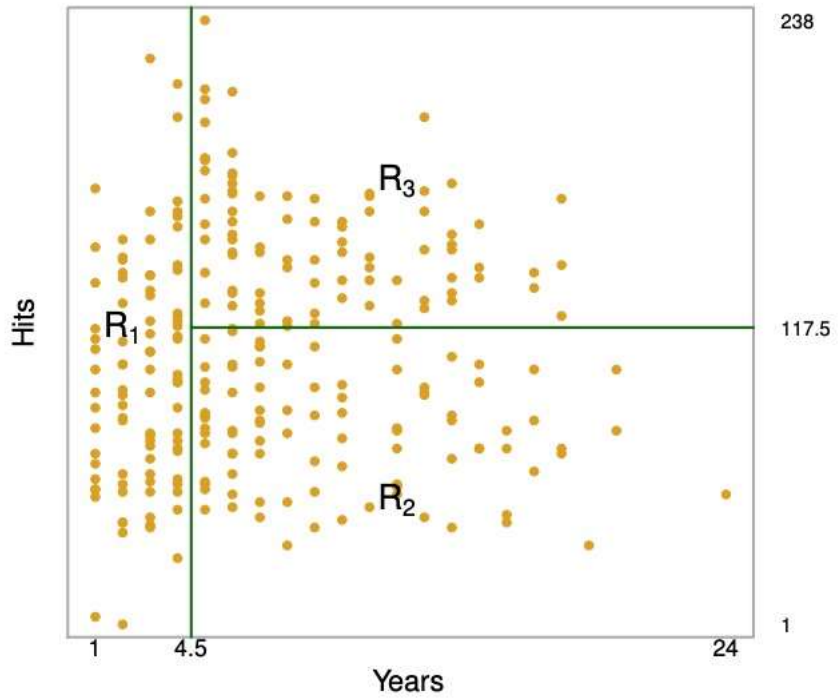
这是一个棒球选手收入影响因素的实例。横坐标为选手进入大联盟的年数，纵坐标为选手每年的平均击球数，选手的收入由颜色表示，暗色（蓝，绿）表示低收入，亮色（黄，红）表示高收入。

二、决策树：实例



- 上图是通过决策树得到的估计结果。
- 思考：如何解释这一结果？

二、决策树：估计



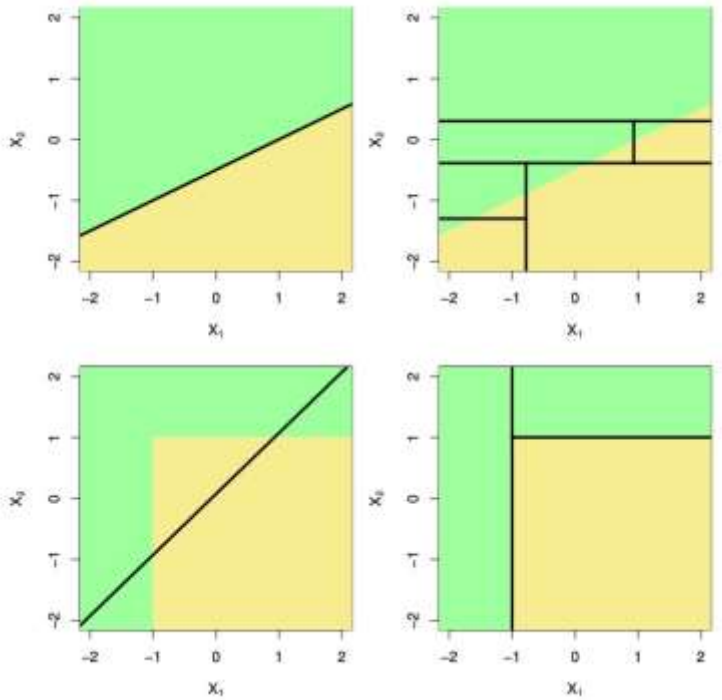
决策树将特征空间划分成$R1, R2, R3$三个预测空间，这三个空间两两不相交，且在每一个空间内的预测结果相同。

二、决策树：简单算法

- 算法1：

1. 在协变量X中选择其中一个分量，记为$X_j, j \in \{1,2,\cdots,n\}$。
2. 将X_j的定义域分成互不相交的两部分。
3. 根据步骤2对空间$\mathcal{X}$进行划分。
4. 设定正整数K。重复步骤1～3共计K次。可得K个子空间，其满足$\mathcal{X} = \mathcal{X}_1 \cup \cdots \cup \mathcal{X}_K$以及$\emptyset = \mathcal{X}_1 \cap \cdots \cap \mathcal{X}_K$。
5. 估计条件期望，其估计量$\hat{g}(x) = \sum_{k=1}^{K} \hat{\beta}_k 1\{x \in \mathcal{X}_k\}$，$\hat{\beta}_k$的形式为等式(4.4)。

二、决策树与线性回归



比较两种DGP下决策树与线性回归的拟合效果。

没有哪一种算法是在任何情况下严格占优于另一种算法！具体问题具体分析。

二、决策树：最优特征选择

- 在决策树的构建过程中，最关键的步骤之一是选择合适的特征X进行数据集划分，以便递归地生成树的各个节点。为了简单起见，我们假设特征X为离散变量。
- 对于回归树（被解释变量连续）而言，最优特征选择及特征空间的划分标准比较清晰，可以采用RSS（残差平方和）作为划分依据。
- 在分类树（被解释变量离散）中，RSS不再是一个合适的划分依据。

- 思考：如何衡量离散变量的预测精准度？

二、决策树：分类错误率

- 在离散情形中，RSS的一个替代是分类错误率，其定义为

$$CE = 1 - \max_k \hat{p}_{mk},$$

其中$\hat{p}_{mk}$表示将第k类样本分类至第m类的比例。

- 从定义来看，CE衡量了所有组中最差的分类错误率。
- 分类错误率只考虑了最差的组，无法衡量其他组的分类结果，即数据集的纯度。

- 思考：如何改进分类错误率？

二、决策树：信息增益

信息增益的核心思想是通过某个特征的划分，使得划分后的子集的熵比原始数据集的熵减少更多。熵用于衡量数据集的纯度，熵越低，数据集的纯度越高。

对于给定数据集D，其熵的定义为：

$$E(D) = -\sum_{j=1}^{k} p_j \, log_2 p_j,$$

其中k表示因变量Y所含类别个数，p_j表示第j类样本所占总样本的比例。根据定义可知，$E(D)$值越小，数据集D的纯度越高。

计算信息熵时候约定：若$p = 0$，则$plog_2 p = 0$。

二、决策树：信息增益

对于某一特征X，数据集D基于特征X的划分可以形成若干子集。基于信息熵，信息增益定义为：

$$Gain(D,X) = E(D) - \sum_{v \in value(V)} \frac{|D_v|}{|D|} E(D_v),$$

其中D_v表示数据集中特征X取值为v的子集，$\frac{|D_v|}{|D|}$表示该子集占总样本的比例。信息增益的思想是选择那个能最大化纯度提升（即减少熵最多）的特征来进行划分。特征的熵越低，表示该特征越能将数据“纯化”，因此信息增益较大。

二、决策树：信息增益

编号	X_1	X_2	X_3	Y
1	1	0	0	1
2	1	1	0	1
3	0	1	1	0
4	0	1	0	1
5	0	0	0	0
6	1	1	1	0
7	1	1	1	1
8	1	0	0	1

课堂练习：计算总体样本的信息熵。

进一步计算三种特征所对应的信息增益。

二、决策树：信息增益率

尽管信息增益方法在很多场景下有效，但它偏向于选择具有更多取值的特征。为了解决这一问题，研究者提出**信息增益率**，作为对信息增益的改进。信息增益率在信息增益的基础上引入了固有值，其定义为：

$$GainRatio(D,X) = \frac{Gain(D,X)}{IV(X)},$$

其中固有值$IV(X)$的定义为：

$$IV(X) = -\sum_{v \in value(V)} \frac{|D_v|}{|D|} log_2 \frac{|D_v|}{|D|}.$$

特征X取值数目越多时，其固有值$IV(X)$通常会越大。信息增益率旨在平衡属性取值多寡的影响，使得划分不再偏向于多取值属性。

二、决策树：信息增益率

编号	X_1	X_2	X_3	Y
1	1	0	0	1
2	1	1	0	1
3	0	1	1	0
4	0	1	0	1
5	0	0	0	0
6	1	1	1	0
7	1	1	1	1
8	1	0	0	1

- 课堂练习：计算三种特征所对应的固有值。

二、决策树：基尼系数

- 最后介绍一种度量数据纯度的标准，称为基尼系数。其定义为：

$$Gini(D) = 1 - \sum_{j=1}^{k} p_j^2 .$$

- 直观来说，$Gini(D)$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。因此，基尼系数的值越小，数据集的纯度越高。

二、决策树：基尼系数

- 进一步地，特征X的基尼系数可以定义为：

$$Gini(D,X)=\sum_{v\in value(V)}\frac{|D_v|}{|D|}Gini(D_v).$$

- 划分后的基尼系数越小，说明特征X对数据集纯化的效果越好。

最优划分特征的选择是决策树生成过程中至关重要的一步。不同的划分准则会影响决策树的结构和性能。信息增益、信息增益率和基尼系数各有优缺点，应该根据具体场景和需求选择合适的准则。

二、决策树：剪枝

- 通过不断扩展决策树，我们可以完美地拟合训练数据，反而会导致模型过拟合。生成过于复杂的树模型，导致对新数据的泛化能力下降。因此，剪枝技术被引入，以简化决策树，避免过拟合问题。
- 预剪枝
- 后剪枝

二、决策树：预剪枝

- 预剪枝是在树构建过程中通过提前停止分裂节点来简化树结构。当某些条件满足时（如节点包含的数据少于设定的阈值、信息增益不足等），决策树会提前停止继续划分，这称为预剪枝。
- 预剪枝的一个优点是减少了计算量。
- 缺点是可能会过早停止，从而导致模型欠拟合。

二、决策树：后剪枝

- 后剪枝是在决策树完全构建之后进行的。此时，我们将已经生成的复杂树进行简化，通过剪掉不必要的子树来减少模型的复杂度。
- 后剪枝通常会从节点开始，逐步向上合并，剪掉对模型贡献较小的分支。相比预剪枝，后剪枝能够保证树的充分生长，再进行优化，因此泛化能力往往较好。
- 通过剪枝，决策树的模型复杂度降低，泛化能力提升。未剪枝的树容易过拟合，而经过剪枝后的树能够更好地应对新的数据。剪枝的过程如同修剪一棵繁茂的树木，去掉多余的枝叶，保持树的整体形状紧凑且有用。

二、决策树：预剪枝vs后剪枝

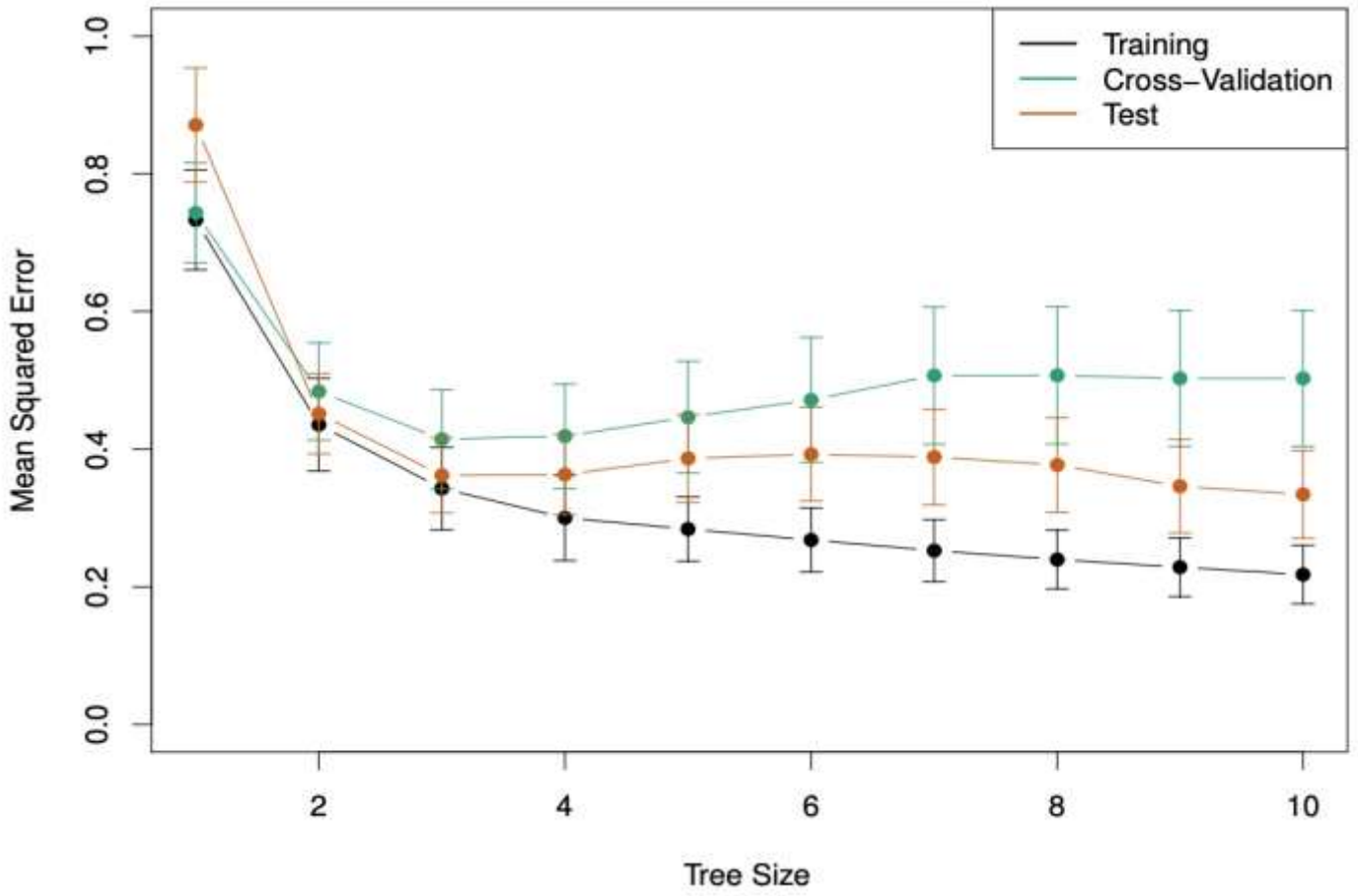
- 预剪枝适合对计算资源要求高、训练时间有限的场景。
- 而后剪枝通常在有足够计算资源的情况下能提供更好的模型效果。
- 在实际应用中，我们可以通过交叉验证来选择最优的剪枝方法。
- 考虑Y是连续型随机变量的情形。我们的目标是最小化

$$\sum_{m=1}^{|T|}\sum_{i:X_i\in\mathcal{X}_m}\left(Y_i-\hat{Y}_{\mathcal{X}_m}\right)^2+\alpha\,|T|.$$

- α表示调节变量，控制了决策树的规模。

- 思考：一个最优的α如何平衡偏误与方差？如何估计最优的α？

二、决策树：规模与均方误差



《面向机器学习的计量经济学》

二、决策树：工业增加值预测

再次研究企业生产函数这一问题。本章中我们使用决策树模型以及2004年工业企业数据库中的数据来估计企业的生产函数：

$$lnY_i = g(lnL_i, lnK_i, X_i, e_i),$$

其中Y_i表示第i个企业的工业增加值；

L_i表示第i个企业的从业人员规模；

K_i表示第i个企业的固定资产；

X_i表示第i个企业的一些其他控制变量；

e_i表示第i个企业的不可观测扰动项。

与Lasso一章的设定不同，本章中函数g为非线形设定。

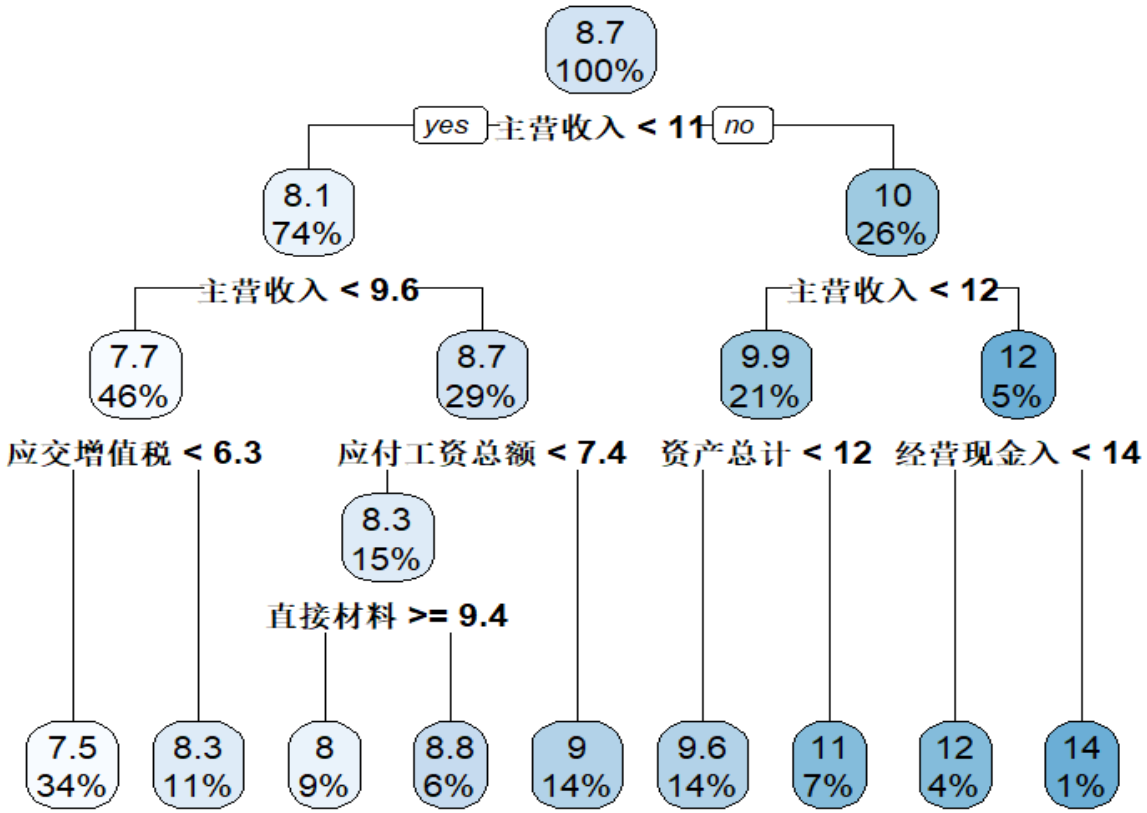
之后我们可以用2005-2007年的数据来拟合，考察决策树模型在样本外的拟合效果，并与Lasso，Ridge和Elastic net方法进行比较。

二、决策树：工业增加值预测

表 5.1 2004 年数据的描述性统计

变量	均值	标准差	最大值	最小值
log(工业增加值+1)	8.657	1.4109	16.033	1.609
log(固定资产合计+1)	8.209	1.9788	18.116	0
Log(从业人员总数+1)	214.6	1.0249	9.599	6
log(主营收入+1)	10.006	1.2983	17.559	6.897
log(应交增值税+1)	6.037	2.2782	14.587	0
log(资产总计+1)	9.930	1.4509	18.180	6.006
log(实收资本+1)	7.822	2.3933	15.790	0
log(直接材料+1)	9.329	1.5484	17.380	0
log(应付工资总额+1)	7.317	1.2462	13.509	4.111
log(管理费用+1)	7.201	1.6886	13.820	0

二、决策树：工业增加值预测



二、决策树：工业增加值预测

表 2 不同估计方法得到的均方误差（MSE）

	决策树（预剪枝）	决策树（后剪枝）	Lasso	Ridge	Elastic net（$\alpha = 0.5$）
2005 年	0.5329	0.5291	0.3683	0.2869	0.3696
2006 年	0.4357	0.4349	0.2682	0.2800	0.2677
2007 年	0.5937	0.5698	0.3911	0.4294	0.3953

二、决策树的优势和劣势

优势

- 非线性建模能力，OLS 假设自变量与因变量之间是线性关系，而决策树能够捕捉复杂的非线性和交互效应，使其在处理具有非线性结构的数据时更具优势。
- 自动变量选择和交互项捕捉。决策树通过递归分割数据，自动筛选出最具预测力的变量，并能够有效捕捉变量之间的交互作用，而 OLS需要手动指定交互项。
- 对异常值和数据预处理的鲁棒性OLS 受异常值影响较大，而决策树相对不易受异常值干扰，因为它依赖于数据的分裂规则，而不是参数估计。
- 处理高维数据的能力。当自变量维度较高时，OLS可能面临多重共线性问题，而决策树方法能够较好地应对高维数据，尤其是在降维或特征选择方面具有优势。
- 可解释性。决策树的结构可以直观地展现变量对因变量的影响路径，使其在可视化和政策分析中更具可解释性。

二、决策树的优势和劣势

- 劣势：
- 易于过拟合。决策树容易对训练数据拟合过度，导致泛化能力较差，需要剪枝或集成方法(如随机森林、梯度提升树)来缓解。
- 对数据噪声敏感由于决策树方法基于递归划分，小的扰动可能会显著改变树的结构，而 OLS 依赖于整体优化，因此对噪声的鲁棒性较强。
- 不适用于平滑效应估计和因果推断OLS 提供了全局回归系数，可以直观地解释变量的边际影响，而决策树更关注局部划分，不易得到平滑的因果效应估计。此外，传统决策树方法不易直接应用于因果推断，而 OLS 在匹配合适的工具变量或固定效应时更适用于因果分析。
- 4.难以进行推断和假设检验OLS 允许进行统计推断，如t检验、F检验和置信区间估计，而标准决策树模型不提供类似的推断框架，使得在政策评估等应用中较难进行显著性分析。
- 5.计算复杂度较高，在大规模数据集上需要递归分裂，计算复杂度相对较高

二、决策树的应用场景

- OLS适用于: 线性关系明确、关注变量边际效应、需要进行统计推断的场景，如政策评估、因果推断、经济理论检验等
- 决策树适用于: 数据关系复杂、包含非线性和交互效应、不要求严格统计推断的应用，如信用评分、市场细分、机器学习预测任务等。
- 如果目标是政策评估或因果推断，OLS或基于回归的因果方法(如工具变量、双重差分等)通常更优;如果目标是数据驱动的预测或分类，决策树及其集成方法(如随机森林、XGBoost)可能更合适。

谢 谢!