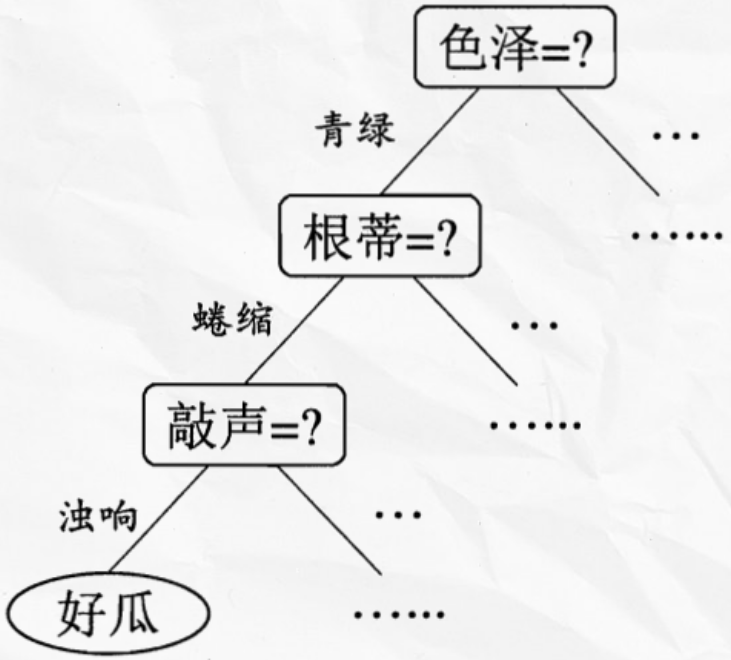
机器学习

决策树

涂文婷
tu.wenting@mail.shufe.edu.cn

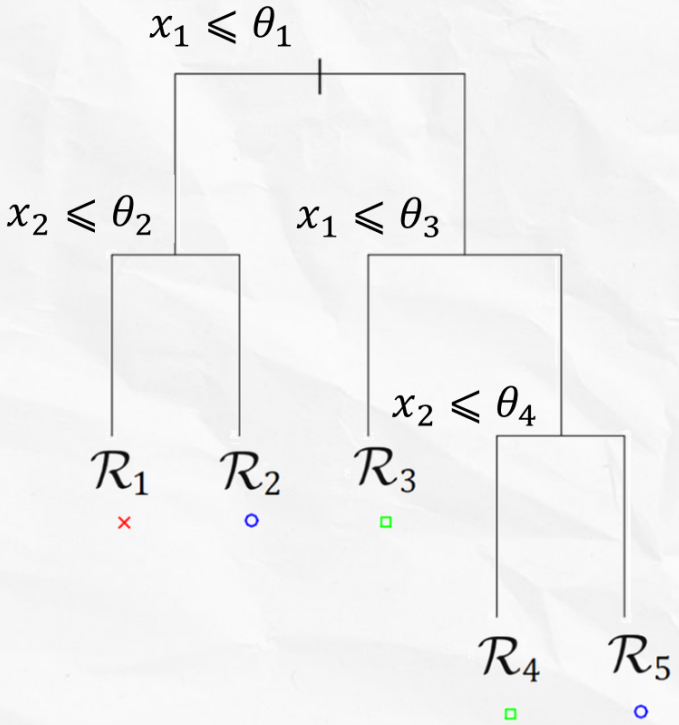
分类决策树

- 树结构的决策规则



分类决策树

- 二叉树结构的决策规则

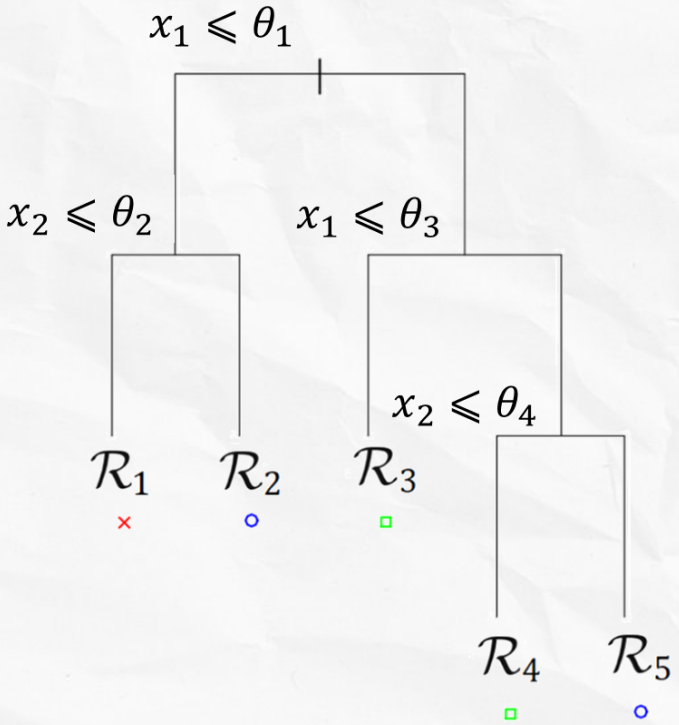


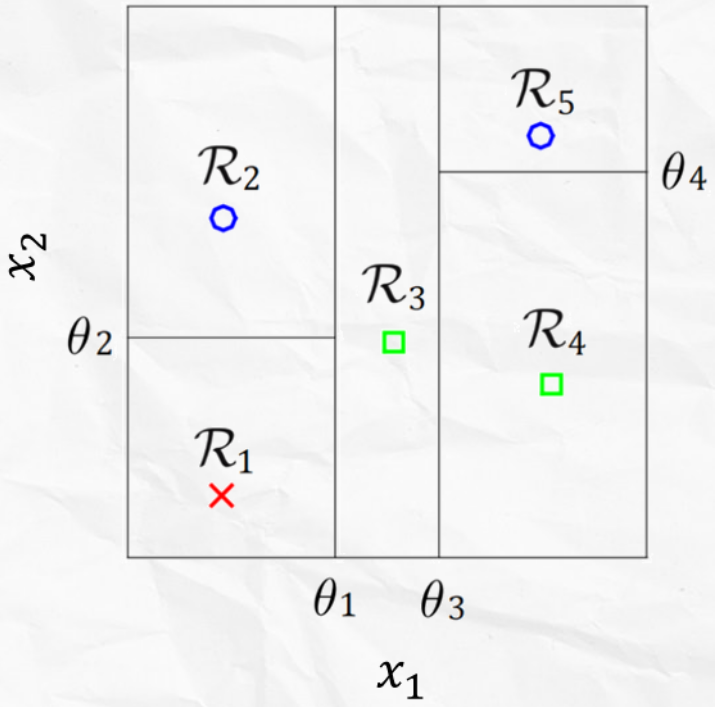
当划分依据为连续属性/特征时，
x_i为其属性本身，θ为其取值范围内的某个值

当划分依据为离散属性/特征时，
x_i为独热编码下对应于某个特定属性值的分量，
θ为0

分类决策树

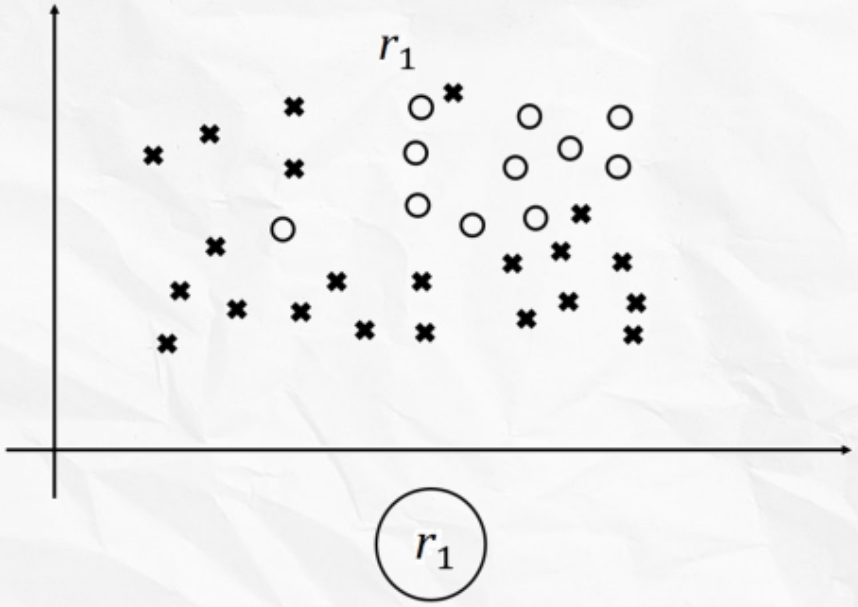
- 分类决策空间





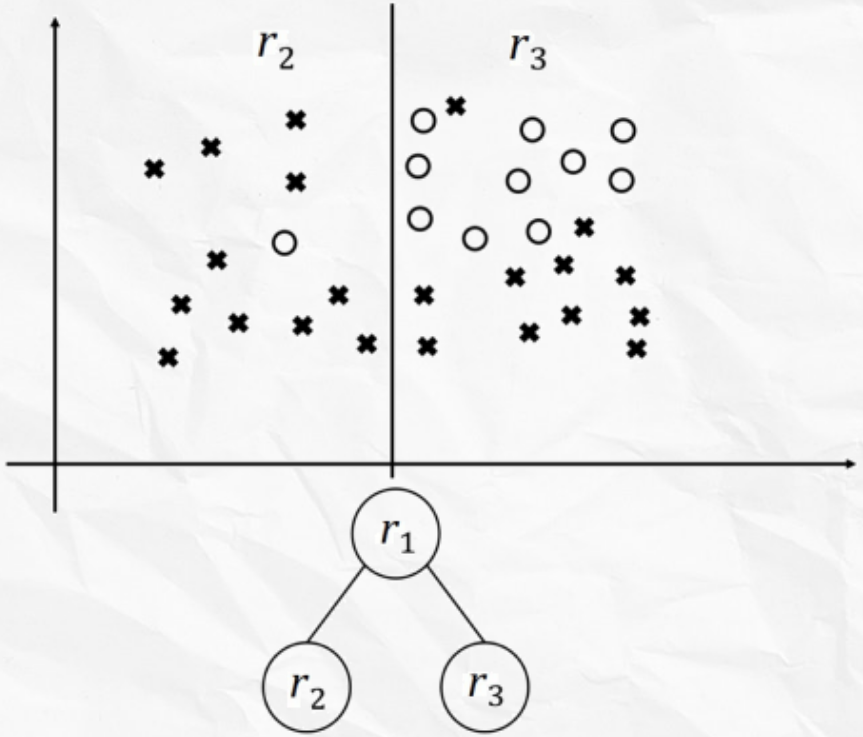
分类决策树

- “分而治之” *divide - and - conquer*



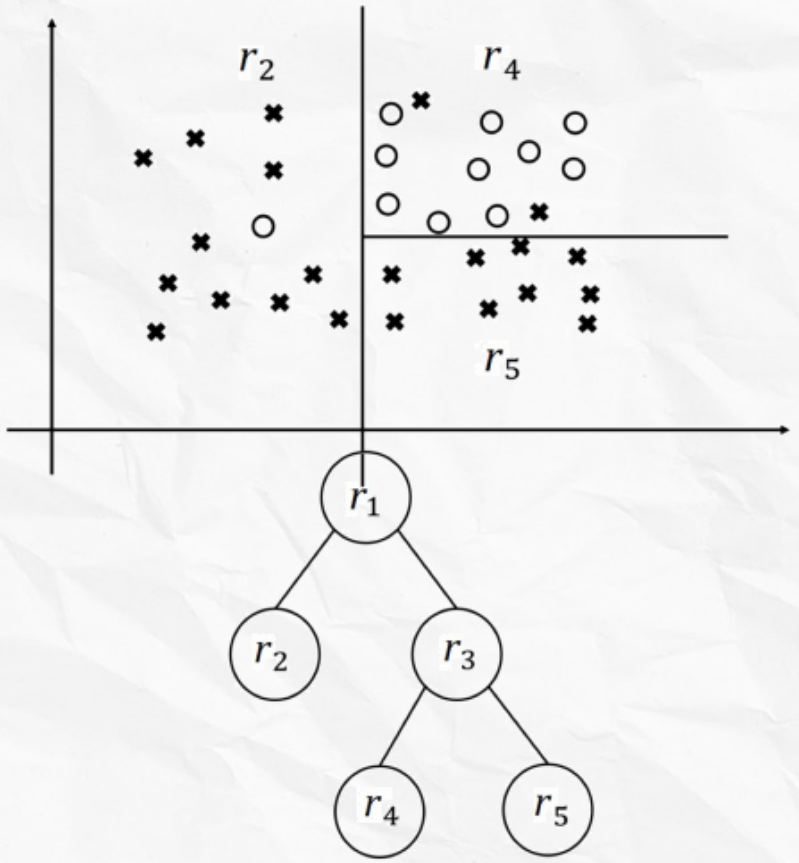
分类决策树

- “分而治之” *divide - and - conquer*



分类决策树

- “分而治之” *divide - and - conquer*



分类决策树

- “分而治之” *divide - and - conquer*

递归地进行以下步骤：

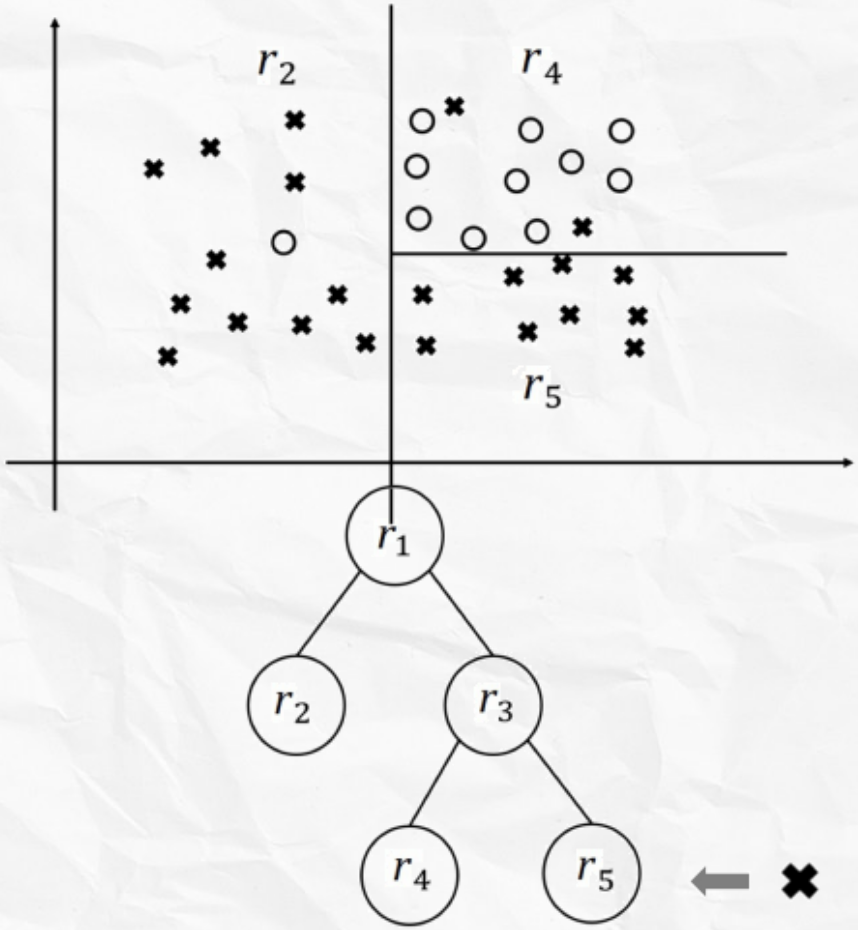
- 选择一个划分数据的特征作为当前节点
- 将划分的分支刻画出来
- 对划分的分支进行停止条件的检查：

  如果停止条件满足，则生成叶子节点

  否则继续选择特征继续划分

停止条件：

- 样本全属于同一类别
- 样本的特征向量完全一样



分类决策树

- “分而治之” *divide - and - conquer*

递归地进行以下步骤：

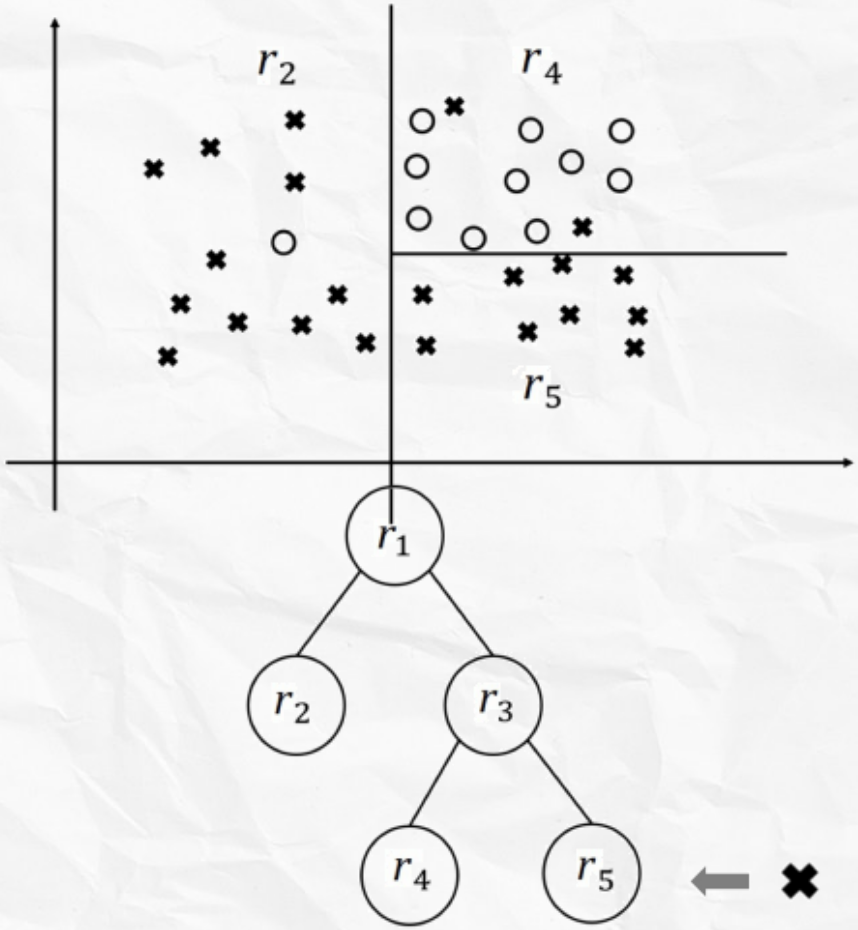
- 选择一个划分数据的特征作为当前节点
- 将划分的分支刻画出来
- 对划分的分支进行停止条件的检查：

  如果停止条件满足，则生成叶子节点

  否则继续选择特征继续划分

停止条件：

- 样本全属于同一类别
- 样本的特征向量完全一样



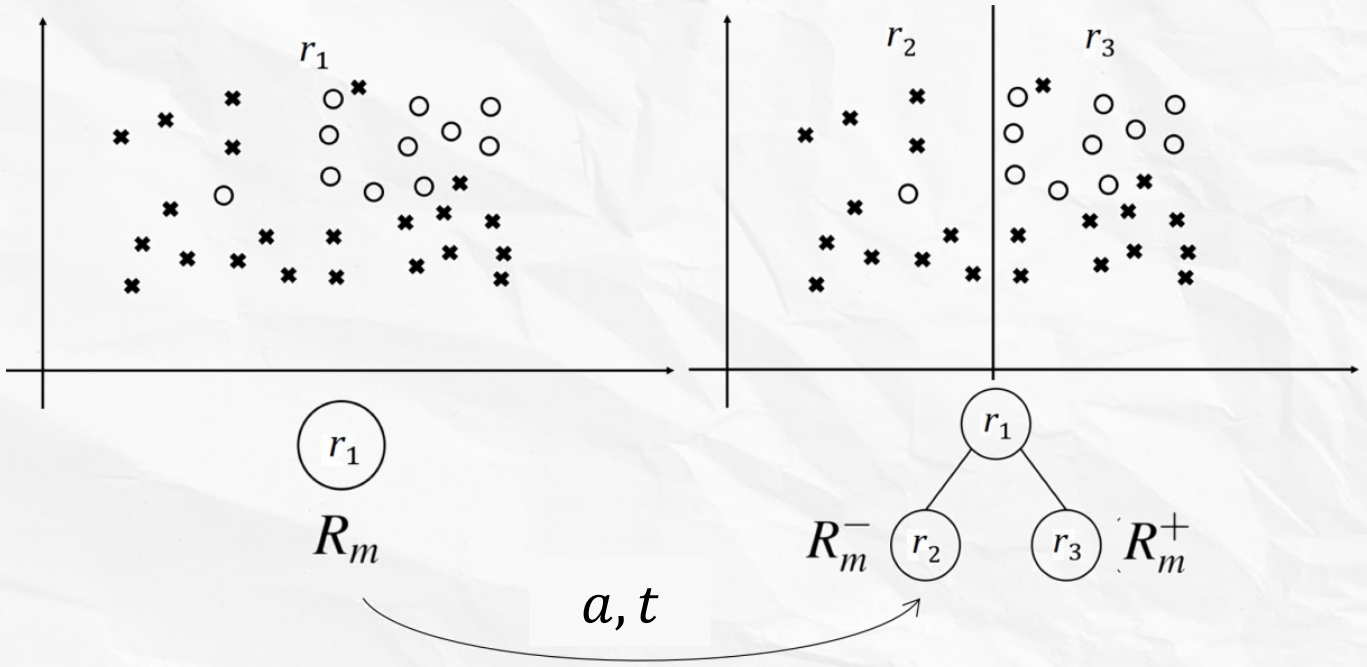
分类决策树

- 选择最优划分

$$\text{Score}(D,a,t) = \max_{t\in T_a} \text{Risk}(D) - \sum_{\lambda\in\{-,+\}} \frac{|D_t^\lambda|}{|D|}\text{Risk}(D_t^\lambda)$$

$$\text{Split}(D) = \max_{a\in A, t\in T_a} \text{Score}(D,a,t)$$

$T_a = \left\{ \frac{a^i+a^{i+1}}{2} \middle| 1 \leqslant i \leqslant n-1 \right\}$，如果$a$是连续属性/特征

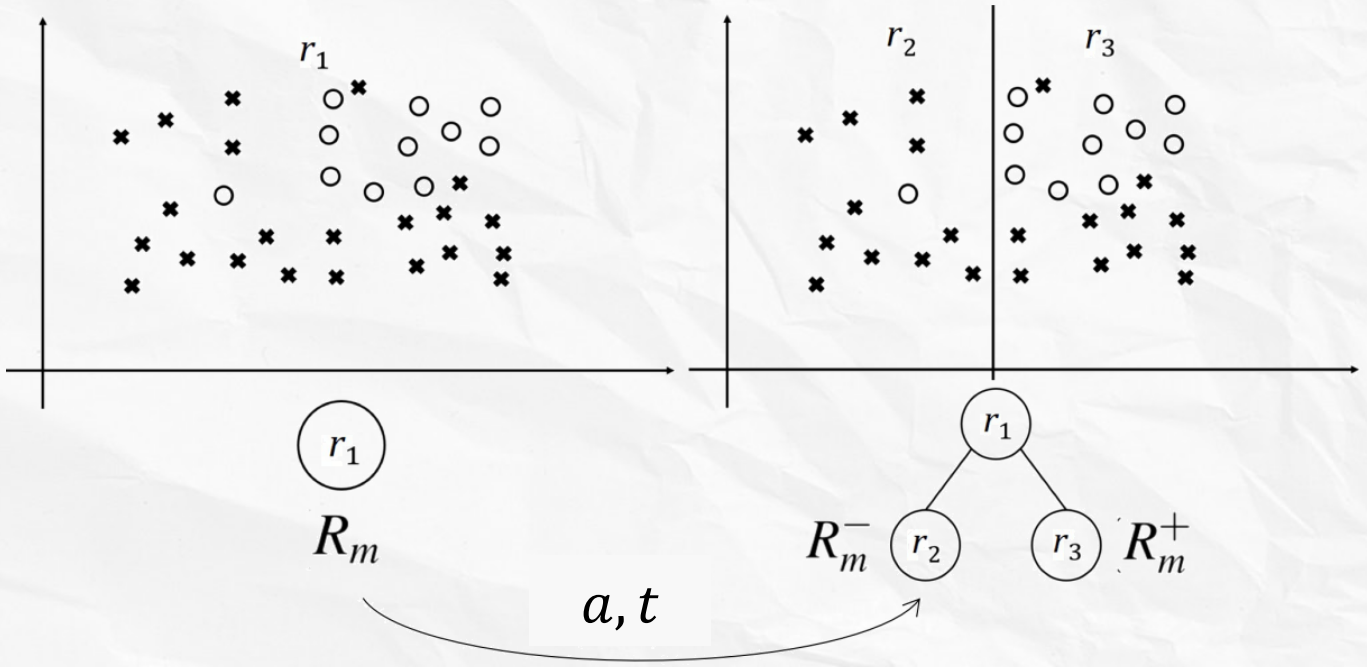


分类决策树

选择最优划分

随着划分过程不断进行，分类决策树的分支结点所包含的样本尽可能属于同一类别，即结点所包含的样本集的“纯度” *purity* 越来越高.

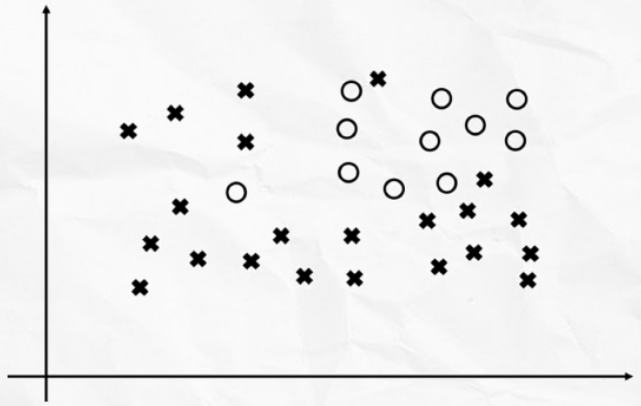
$$\text{Risk}(D) = -\text{Purity}(D)$$



分类决策树

- 选择最优划分
  - 信息熵

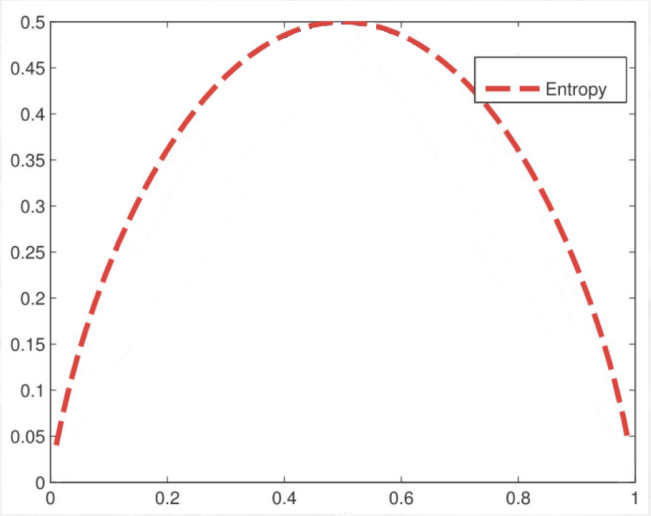
$$\mathrm{Ent}(D) = -\sum_{k=1}^{|\mathcal{Y}|} p_k \log_2 p_k$$



  - 信息增益

$$\mathrm{Gain}(D, a) = \mathrm{Ent}(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} \mathrm{Ent}(D^v)$$

Risk (D)=Ent(D)

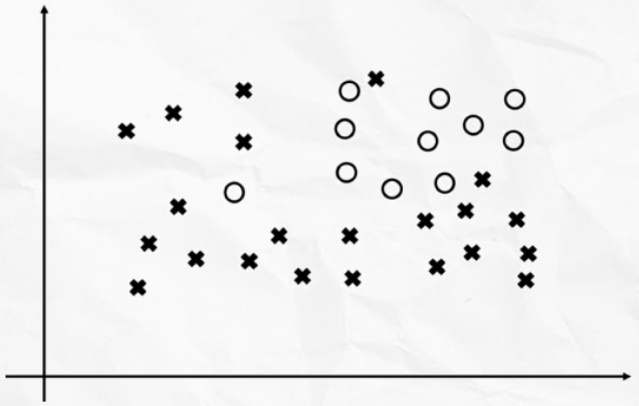


p_1

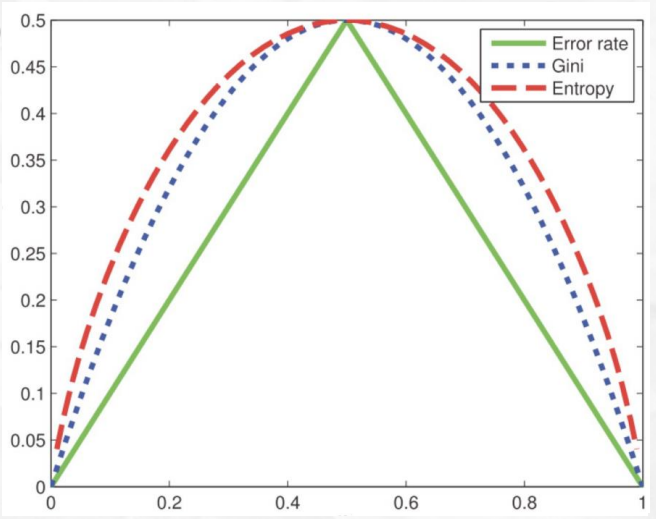
分类决策树

- 选择最优划分
  - 基尼指数

$$\mathrm{Gini}(D) = \sum_{k=1}^{|\mathcal{Y}|} \sum_{k' \neq k} p_k p_{k'} = 1 - \sum_{k=1}^{|\mathcal{Y}|} p_k^2$$



Risk (D)=Gini(D)



p_1

回归决策树

- 分类决策函数

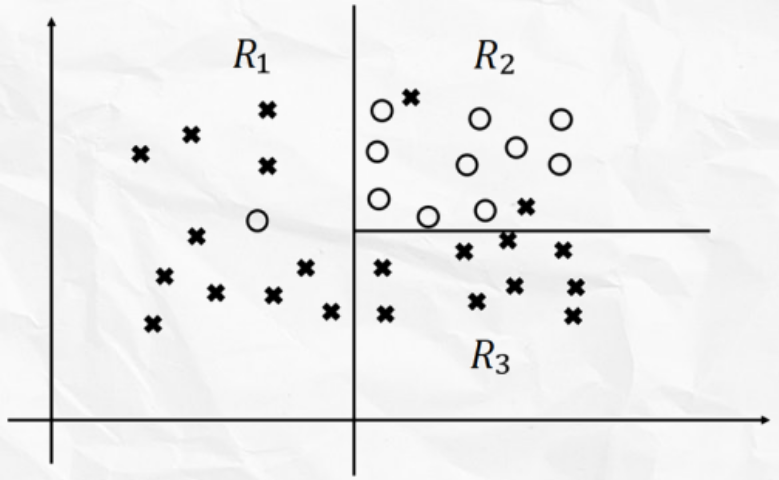
$$f(\mathbf{x}_i) = \sum_{m=1}^{M} c_m \cdot \mathbb{I}\{\mathbf{x}_i \in R_m\}$$

$$\hat{p}(k \mid \mathbf{x}_i) = \sum_{m=1}^{M} p_k^m \cdot \mathbb{I}\{\mathbf{x}_i \in R_m\}$$

分类决策树 $\longrightarrow \{R_1, \dots, R_M\}, \{c_1, \dots, c_M\} / \{p_k^1, \dots, p_k^2\}, \forall k \in |\mathcal{Y}|$

$$c_m = \max_k \sum_{\mathbf{x}_i \in R_m} \mathbb{I}\{y_i = k\}$$

$$p_k^m = \frac{1}{|R_m|} \sum_{\mathbf{x}_i \in R_m} \mathbb{I}\{y_i = k\}$$



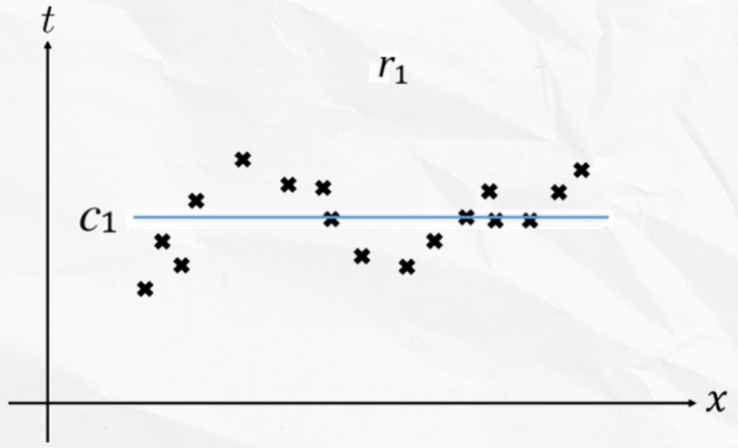
回归决策树

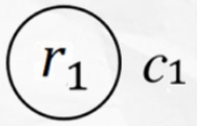
- 回归决策函数

$f(x_i) = \sum_{m=1}^{M} c_m \cdot \mathbb{I}\{\mathbf{x}_i \in R_m\}$

回归决策树 $\longrightarrow \{R_1, \ldots, R_M\}, \{c_1, \ldots, c_M\}$

$$c_m = \text{ave}(y_i \mid \mathbf{x}_i \in R_m) = \arg\min_{c} \sum_{\mathbf{x}_i \in R_m} (y_i - c)^2$$





c_1

回归决策树

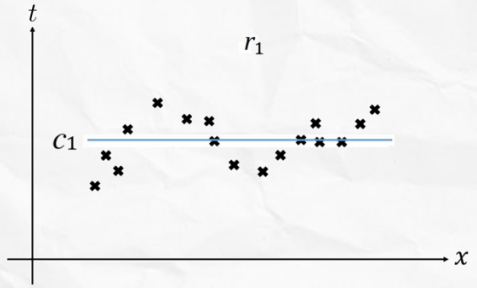
- 选择最优划分

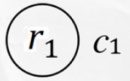
$$\text{Risk}(D) = \sum_{\mathbf{x}_i \in R_m} (y_i - c_m)^2 = \sum_{\mathbf{x}_i \in R_m} \left(y_i - \text{ave}(y_i \mid \mathbf{x}_i \in R_m)\right)^2$$

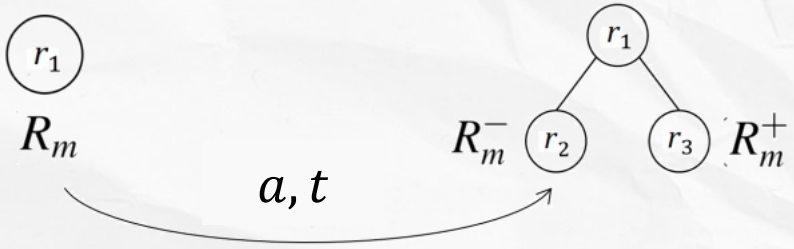
$$\text{Score}(D, a, t) = \max_{t \in T_a} \text{Risk}(D) - \sum_{\lambda \in \{-,+\}} \frac{|D_t^\lambda|}{|D|} \text{Risk}(D_t^\lambda)$$

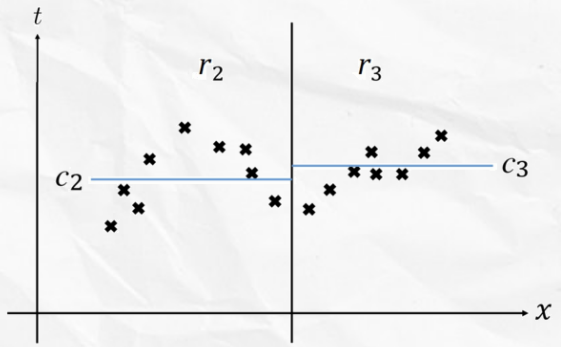
$$\text{Split}(D) = \max_{a \in A, t \in T_a} \text{Score}(D, a, t)$$

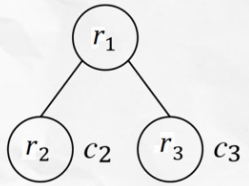
$T_a = \left\{ \frac{a^i + a^{i+1}}{2} \,\middle|\, 1 \leqslant i \leqslant n-1 \right\}$，如果$a$是连续属性/特征











决策树算法

- 算法
  - ID3

ID3是一种分类决策树算法，主要特点是在选择特征的时候，采用信息增益的方法。

  - C4.5

C4.5是ID3的改进，但依旧提供的是分类决策树算法，特点是采用信息增益率进行特征选择。减轻了ID3偏向原则特征值多的特征。另外，采用单点离散化的思想处理连续值，用信息增益率来进行连续值特征的属性值选择。

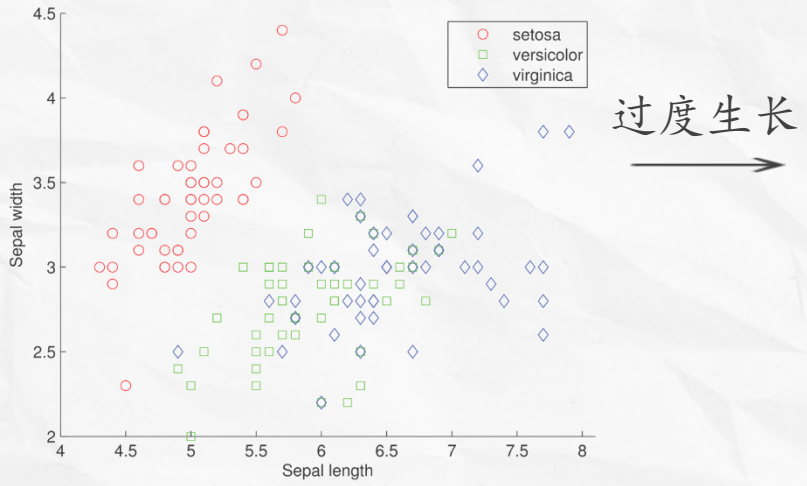
  - CART（*Classification And Regression Tree* 分类回归树）

CART算法同时提供了分类决策树与回归决策树的实现，构建分类决策树时用基尼系数选择划分的特征和分裂点；构建回归决策树时用累积平方误差选择划分的特征和分列点。 CART的一大特点是采用二分递归分割的技术将当前样本集分为两个子样本集，使得生成的每个非叶子节点都有两个分支。因此CART算法生成的决策树是结构简洁的二叉树。

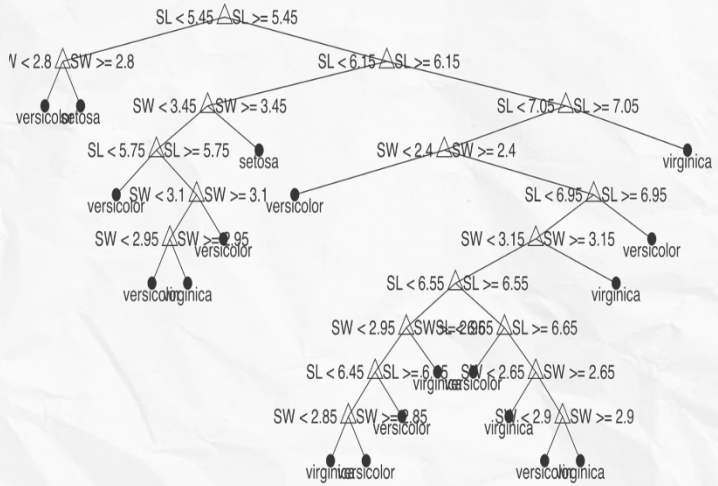
决策树算法

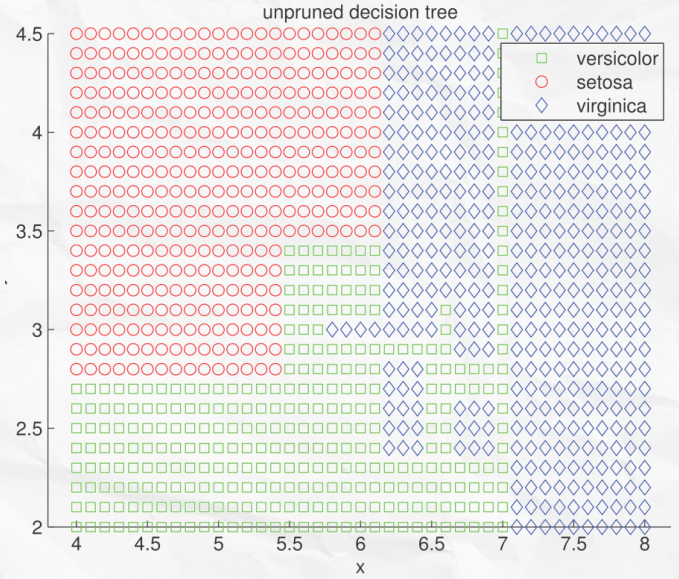
- 剪枝

过度生长可能会导致过拟合



过度生长

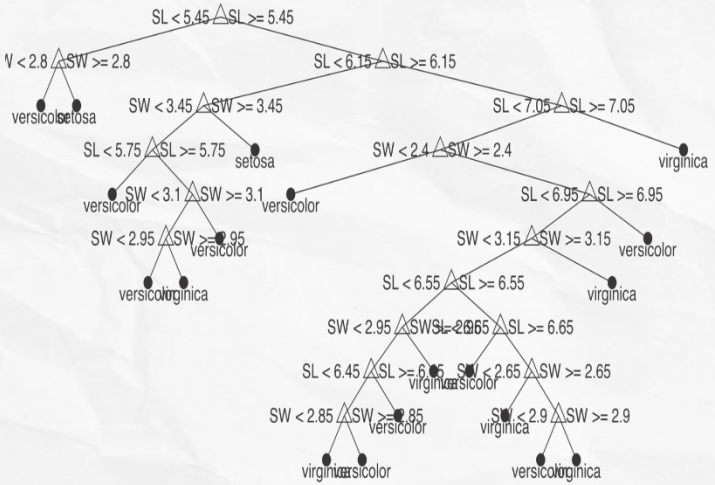




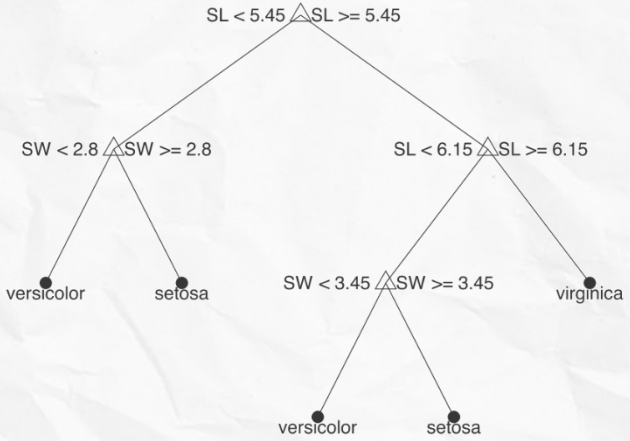
决策树算法

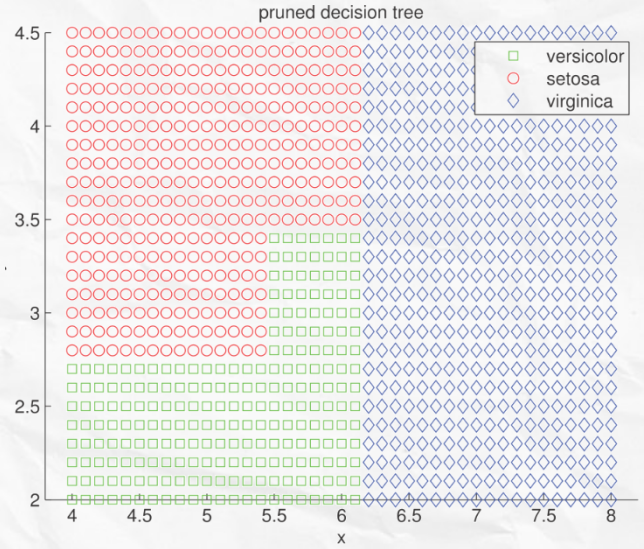
- 剪枝

利用剪枝操作避免过拟合



剪枝操作

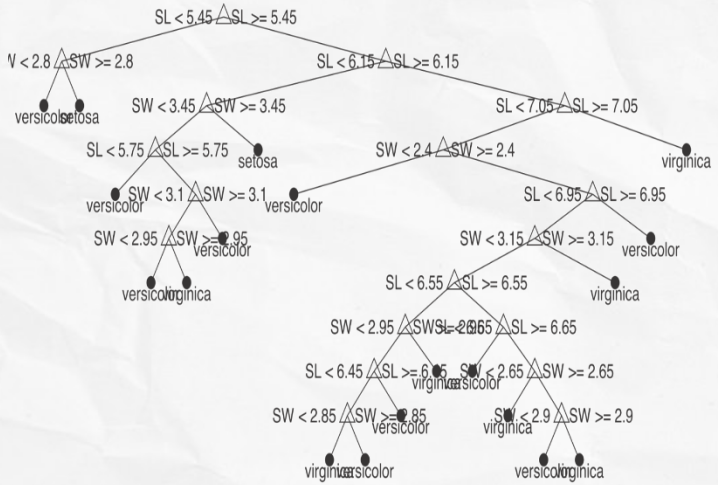




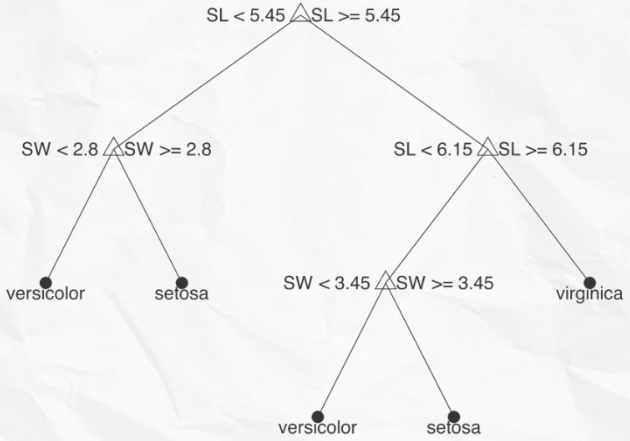
决策树算法

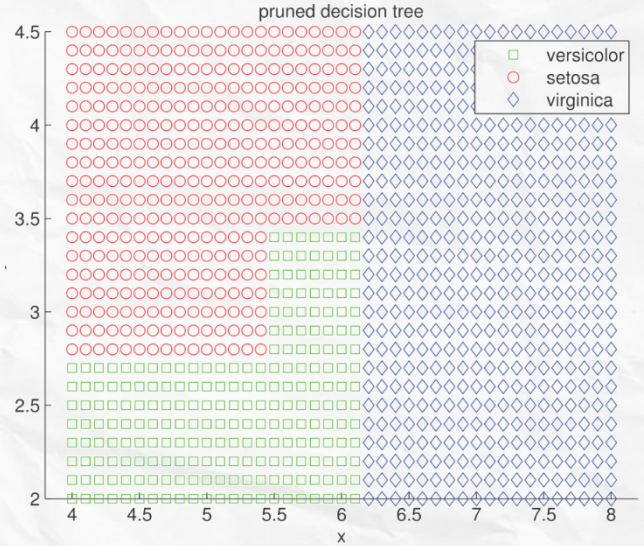
- 剪枝

利用剪枝操作避免过拟合， 在决策树学习中将已生成的树进行简化的过程称为剪枝。具体地，剪枝从已生成的树上裁掉一些子树或叶结点， 并将其根结点或父结点作为新的叶结点，从而简化树模型。



剪枝操作

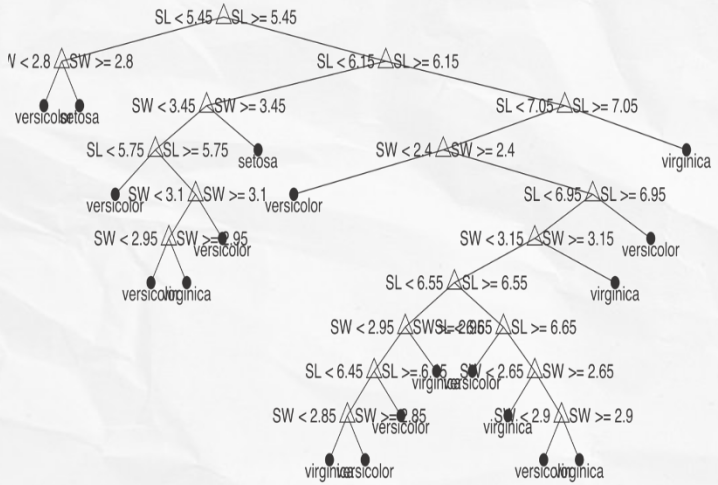




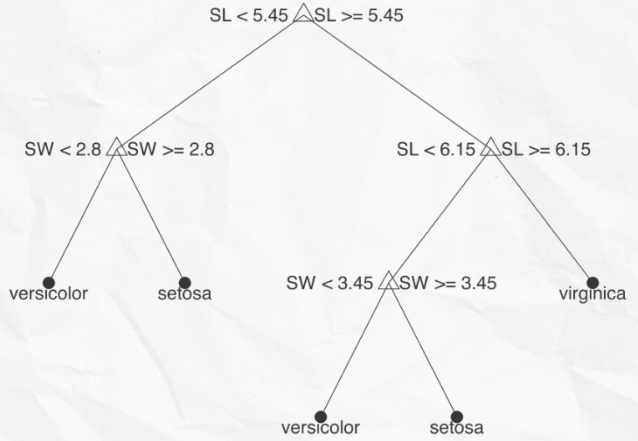
决策树算法

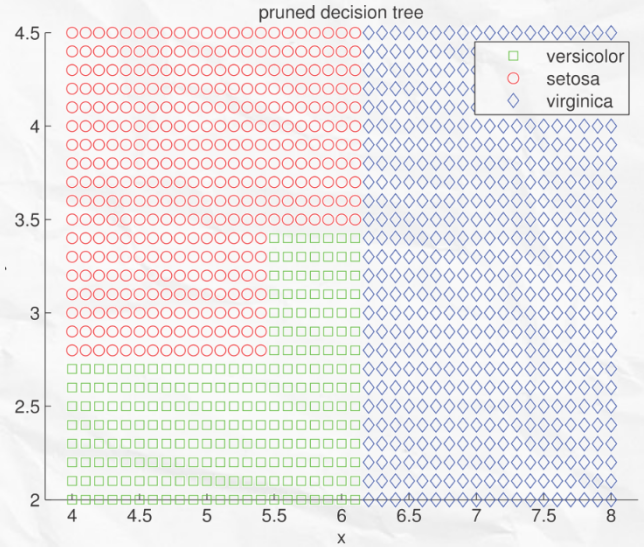
- 剪枝
  - 预剪枝

在决策树生成过程中，对每个节点在划分前先进行估计若当前节点的划分不能带来决策树泛化性能的提升，则停止划分并将当前节点标记为叶节点。



剪枝操作

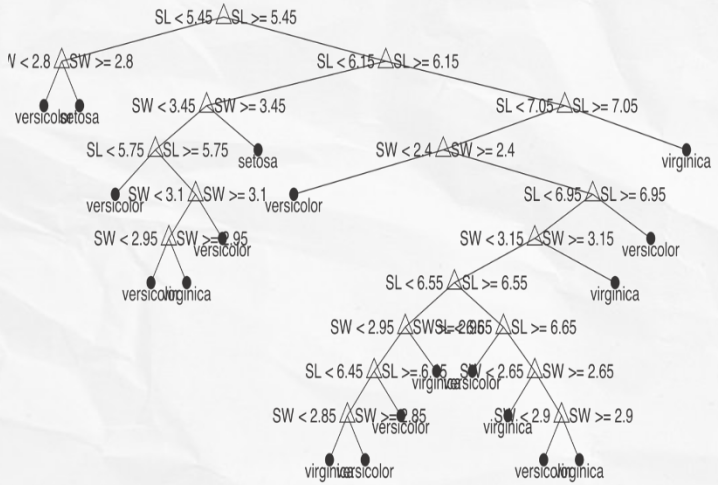




决策树算法

- 剪枝
  - 后剪枝

先从训练集生成一颗完整的决策树，然后自底向上地对非叶节点进行考察。若将该节点对应的子树替换成叶节点能够带来决策树泛化性能的提升，则将该子树替换为叶节点



剪枝操作