北京航空航天大學
COLLEGE OF SOFTWARE BEIHANG UNIVERSITY 软件学院

# 人工智能

## 第5讲：机器学习-有监督学习III

### 决策树与集成学习

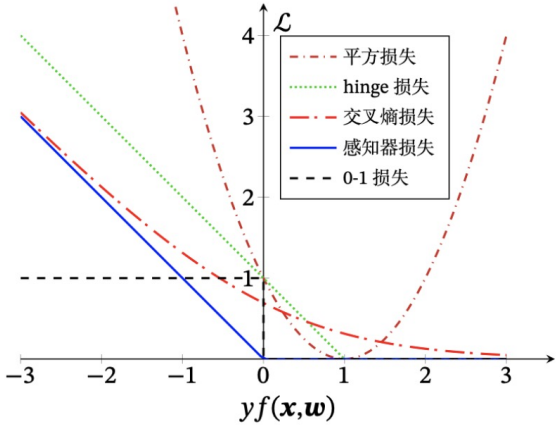
张晶

2025年春季

- 参考资料： 吴飞，《人工智能导论：模型与算法》，高等教育出版社
- 在线课程： https://www.icourse163.org/course/ZJU-1003377027?from=searchPage

北京航空航天大學 软件学院

# 回顾：线性分类模型




| 线性模型 | 激活函数 | 损失/目标函数 | 损失/目标函数定义 | 优化方法 |
|---|---|---|---|---|
| 线性回归 | - | 平方损失 | $(y_i - \boldsymbol{w}^T\boldsymbol{x_i})^2$ | 最小二乘、梯度下降 |
| 对数几率回归 | $sigmoid(\boldsymbol{w}^T\boldsymbol{x})$ | 二值交叉熵损失 | $-y_i \log\sigma(\boldsymbol{w}^T\boldsymbol{x_i}) + (1-y_i)\log(1-\sigma(\boldsymbol{w}^T\boldsymbol{x_i}))$ | 梯度下降 |
| Softmax分类 | $softmax(\boldsymbol{W}^T\boldsymbol{x})$ | 交叉熵损失 | $-y_i \log softmax(\boldsymbol{w}^T\boldsymbol{x_i})$ | 梯度下降 |
| 线性判别分析 | - | Fisher准则 | $\frac{\|m_2 - m_1\|_2^2}{s_1^2 + s_2^2}$ | 广义特征值分解 |
| 感知器 | $sign(\boldsymbol{w}^T\boldsymbol{x})$ | 0-1损失 | $\begin{cases}0, y\cdot f(\boldsymbol{x_i}) \geq 0\\ 1, y\cdot f(\boldsymbol{x_i}) < 0\end{cases}$ | 迭代优化 |
| 支持向量机 | $sign(\boldsymbol{w}^T\boldsymbol{x})$ | Hinge损失 | $\max(0, 1 - y_i\boldsymbol{w}^T\boldsymbol{x_i})$ | 二次规划、SMO等 |

北京航空航天大學 软件学院

# 提纲


一、机器学习基本概念

二、线性回归与线性分类

三、线性判别分析

四、支持向量机

五、决策树

六、集成学习

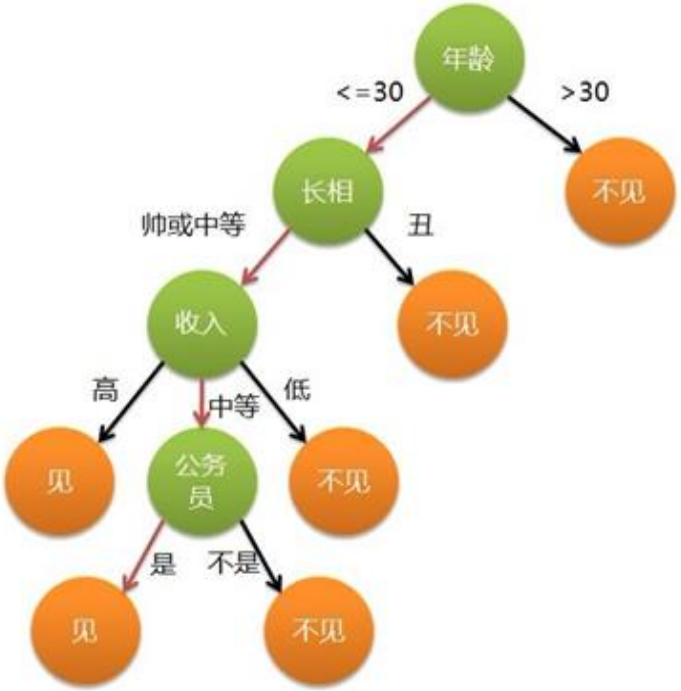
北京航空航天大學
COLLEGE OF SOFTWARE BEIHANG UNIVERSITY 软件学院

# 决策树
# Decision Tree

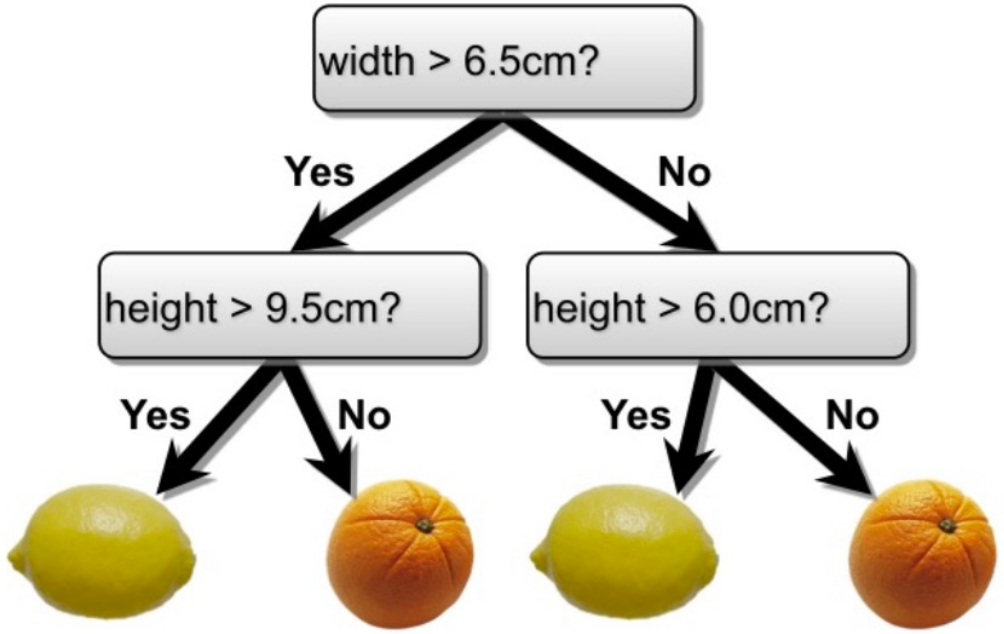
北京航空航天大學 软件学院

# 引例

## 例1：相亲




## 例2：柠檬和橙子的分类

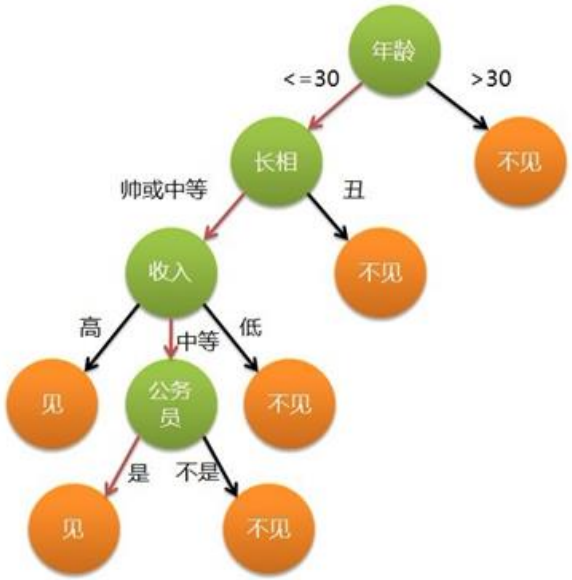


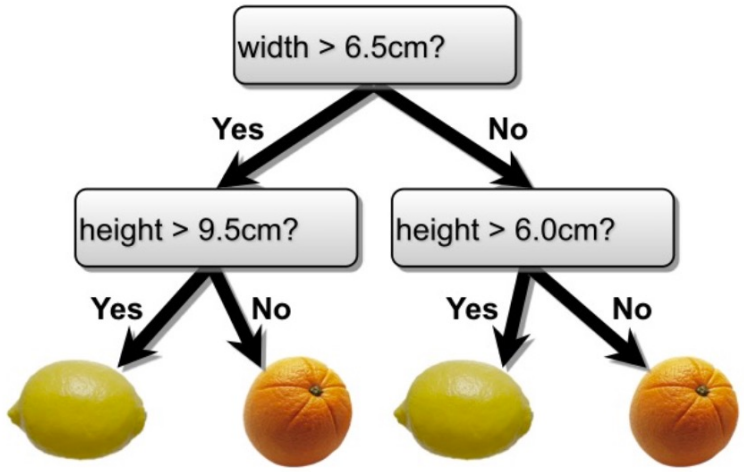
# 填空题 4分

设置

在“相亲”的例子中，如果转化为分类问题，输入特征是 [填空1]？输出类别是 [填空2]？

在柠檬橙子分类的例子中，输入特征是 [填空3]？ 输出类别是 [填空4]？



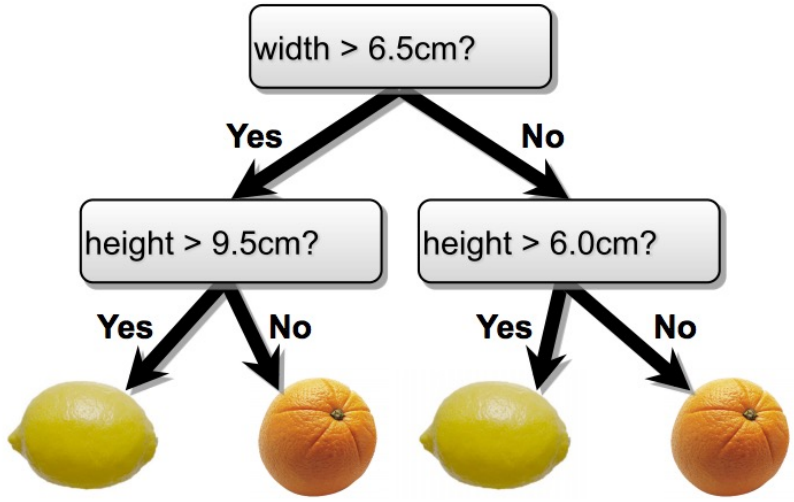




作答

# 决策树：符号定义

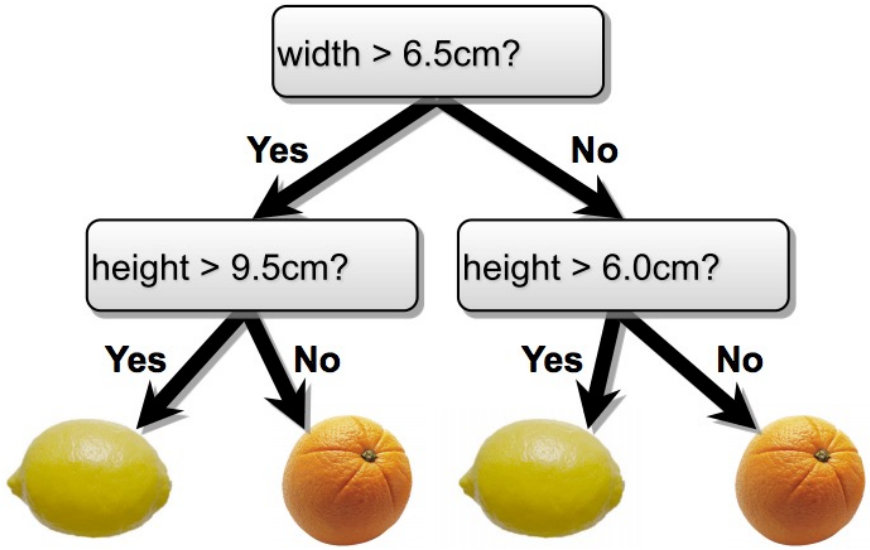
- 训练集: $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$
- 属性集合: $A = \{a_1, a_2, \ldots, a_d\}$,其中 $a_i \in \{a_i^1, \ldots, a_i^V\}$
- 标签集: $y_k \in \{y_1, y_2, \ldots, y_K\}$

- $a_1$：宽度
  - $a_1^1 > 6.5,\ a_1^2 \leq 6.5$
- $a_2$：高度
  - $a_2^1 > 9.5,\ a_2^2 \leq 9.5,\ a_2^3 > 6.0, a_2^4 \leq 6.0$



# 决策树（Decision Tree）

- 决策树定义
  - 决策树是一种树型结构，由节点和有向边组成
  - 节点
    - 内部节点表示一个属性或特征
    - 叶节点代表一种类别
  - 有向边/分支
    - 分支代表一个判断预测输出



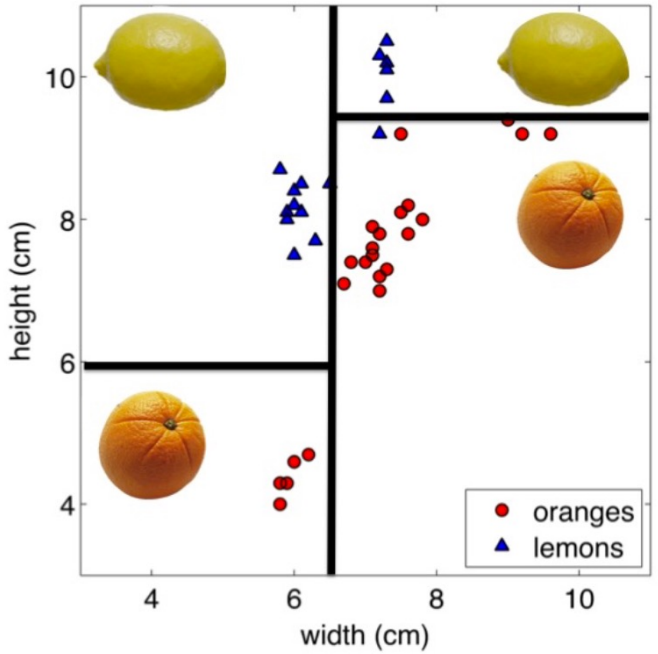
北京航空航天大学 软件学院

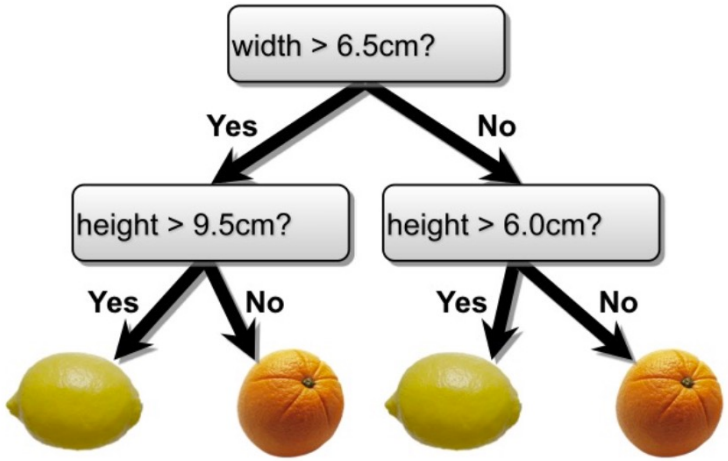
# 决策树（Decision Tree）简史

- 第一个决策树算法：CLS (Concept Learning System)
  - [E. B. Hunt, J. Marin, and P. T. Stone's book "*Experiments in Induction*" published by Academic Press in 1966]
- 使决策树受到关注、成为机器学习主流技术的算法：ID3
  - [J. R. Quinlan's paper in a book "*Expert Systems in the Micro Electronic Age*" edited by D. Michie, published by Edinburgh University Press in 1979]
- 最常用的决策树算法：C4.5
  - [J. R. Quinlan's book "*C4.5: Programs for Machine Learning*" published by Morgan Kaufmann in 1993]
- 可以用于回归任务的决策树算法：CART (Classification and Regression Tree)
  - [L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone's book "Classification and Regression Trees" published by Wadsworth in 1984]
- 基于决策树的最强大算法：RF (Random Forest)
  - [L. Breiman's MLJ'01 paper "Random Forest"]

# 决策树（Decision Tree）算法

- 基本思想：采用自顶向下的递归方法，以信息熵为度量构造一颗熵值下降最快的树，到叶子节点处的熵值为零，此时每个叶节点中的实例都属于同一类
  - 决策树可以看做是一个if-then的规则集合
  - 一个决策树将特征空间划分为不相交的单元或区域



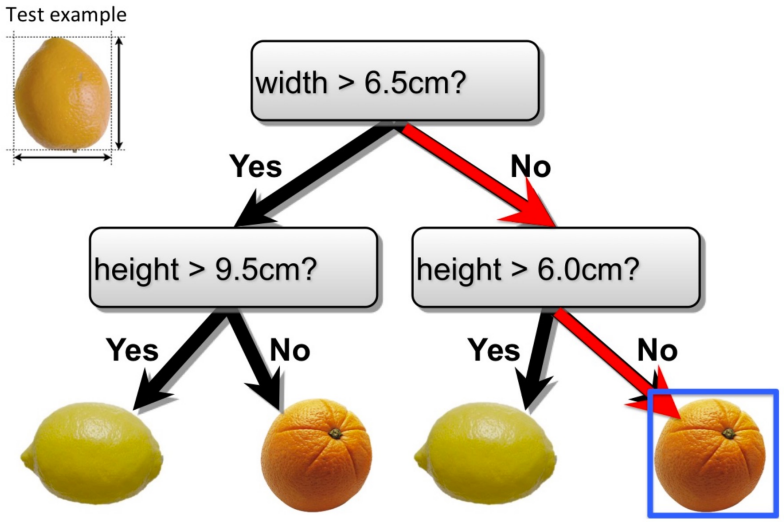



# 决策树（Decision Tree）算法

- 算法流程分为两步
  - **第一步：训练，从数据中获取知识进行学习**

    利用训练集构建并精化一颗决策树，构建决策树模型
  - **第二步：测试，利用生成的模型对输入数据进行分类**

    对测试样本，从根节点依次测试记录的属性值，直至到达某个叶节点，找到该样本所在类别。



北京航空航天大學 软件学院

# 决策树：分类训练

- 训练过程基本流程
  - 步骤1：选取一个属性作为决策树的根节点，然后就这个属性所有的取值创建树的分支
  - 步骤2：用这棵树来对训练数据进行分类
    - 如果一个叶节点的所有实例都属于同一类，则以该类为标记表示此叶节点
    - 如果所有的叶节点都有类标记，则算法终止
  - 步骤3：否则，选取一个从该节点到根路径中没有出现过的属性作为标记标识该节点，然后就这个属性的所有取值继续创建树的分支；重复算法步骤2。

# 决策树：节点划分

- 离散属性：按属性类别划分（如“颜色=红/蓝/绿”）。
- 连续属性：选择最优分割点（如“年龄≤30”vs “年龄>30”）。
  - 对特征值排序，取相邻值的中间点作为候选分割点。
  - 遍历所有候选点，选择使目标函数最大的分割点。

# 决策树：分类训练

策略：“分而治之”(divide-and-conquer)

自根至叶的递归过程

在每个中间节点寻找一个“划分”(split or test)属性

三种停止条件：

(1) 当前节点包含的样本全属于同一类别，无需划分;

(2) 当前属性集为空, 或是所有样本在所有属性上取值相同，无法划分;

(3) 当前节点包含的样本集合为空，不能划分.

北京航空航天大学 软件学院

# 决策树：最优划分属性选择

- 建立决策树的关键，即在当前状态下选择哪个属性作为分类依据，即寻找最优划分方式
  - 示例：高？富？帅？会C++？
- 目标：每个分支节点的样本尽可能属于同一类别
  - 即节点的“纯度”（Purity）越来越高
- 不同算法采用不同度量指标
  - ID3: 信息增益（Information Gain）
  - C4.5: 信息增益率（Gain Ratio）
  - CART: 基尼系数（Gini Index）

# 决策树：属性选择

- 例：假设有$a_1$和$a_2$两个属性的6个训练样本，请构建最佳决策树

| | $a_1$ | $a_2$ | Y |
|---|---|---|---|
| $x_1$ | T | T | T |
| $x_2$ | T | F | T |
| $x_3$ | T | T | T |
| $x_4$ | T | F | T |
| $x_5$ | F | T | T |
| $x_6$ | F | F | F |

# 填空题 4分

设置

根据以下定义，回答该问题中：

$m =$ [填空1] ， $d =$ [填空2] ，$a_i \in$[填空3] ， $y_k \in$[填空4]

- 训练集: $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$
- 属性集合: $A = \{a_1, a_2, \ldots, a_d\}$,其中 $a_i \in \{a_i^1, \ldots, a_i^V\}$
- 标签集: $y_k \in \{y_1, y_2, \ldots, y_K\}$

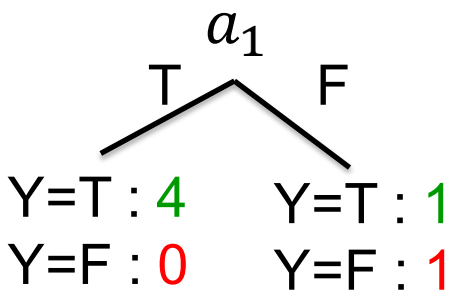
| | $a_1$ | $a_2$ | Y |
|---|---|---|---|
| $x_1$ | T | T | T |
| $x_2$ | T | F | T |
| $x_3$ | T | T | T |
| $x_4$ | T | F | T |
| $x_5$ | F | T | T |
| $x_6$ | F | F | F |

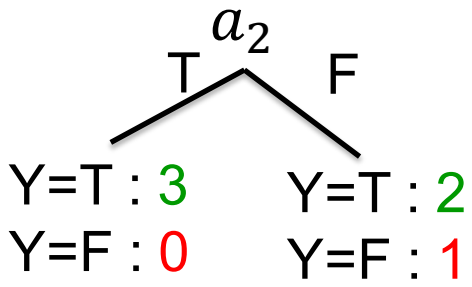
作答

# 填空题 1分

设置

为了实现最优划分，直觉上，你认为先选择 $a_1$ 还是 $a_2$？[填空1]



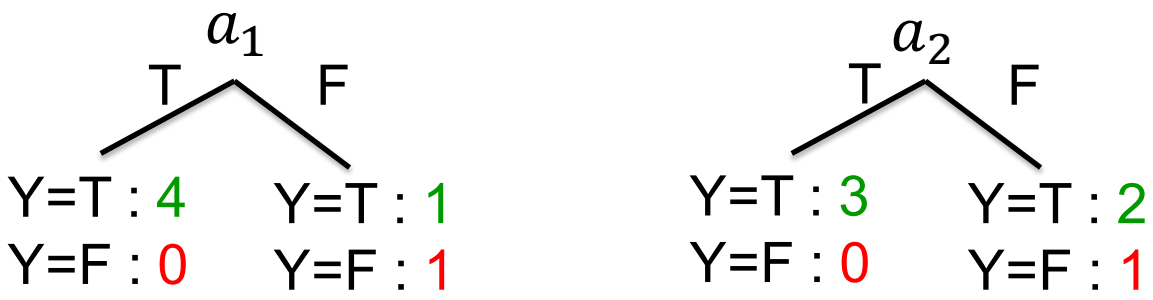




| | $a_1$ | $a_2$ | Y |
|---|---|---|---|
| $x_1$ | T | T | T |
| $x_2$ | T | F | T |
| $x_3$ | T | T | T |
| $x_4$ | T | F | T |
| $x_5$ | F | T | T |
| $x_6$ | F | F | F |

作答

# 决策树：属性选择

- 为了实现最优划分，先选择 $a_1$ 还是 $a_2$？




| | $a_1$ | $a_2$ | Y |
|---|---|---|---|
| $x_1$ | T | T | T |
| $x_2$ | T | F | T |
| $x_3$ | T | T | T |
| $x_4$ | T | F | T |
| $x_5$ | F | T | T |
| $x_6$ | F | F | F |

核心思想：利用叶子节点正确和错误分类个数来定义概率分布，以衡量不确定性。

# 决策树：属性选择

- 划分后分类结果更加确定，则优先选择该属性
  - 比如，2分类问题中，如果叶节点类别分布如下

| | |
|---|---|
| P(Y=T) = 0 | P(Y=F) = 1 |

| | |
|---|---|
| P(Y=T) = 1/4 | P(Y=F) = 3/4 |

| | |
|---|---|
| P(Y=T) = 1/2 | P(Y=F) = 1/2 |

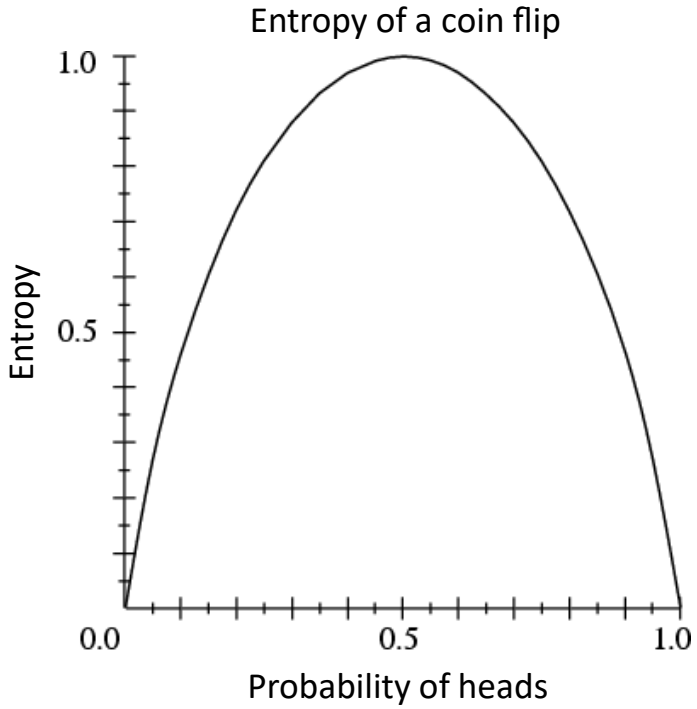
# 决策树：信息熵

- 令离散随机变量$Y$的概率分布为$P(Y = y_k)$，$Y$的信息熵 $H(Y)$定义：

$$H(Y) = -\sum_{k=1}^{K} P(Y = y_k) \log_2 P(Y = y_k)$$

**信息熵越大，不确定性越大！**

**信息论：** 信息熵$H(Y)$是信息论的基本概念。描述信息源各可能事件发生的不确定性。



# 决策树：信息熵

$$H(Y) = -\sum_{k=1}^{K} P(Y = y_k) \log_2 P(Y = y_k)$$

| | |
|---|---|
| P(Y=T) = 0 | P(Y=F) = 1 |

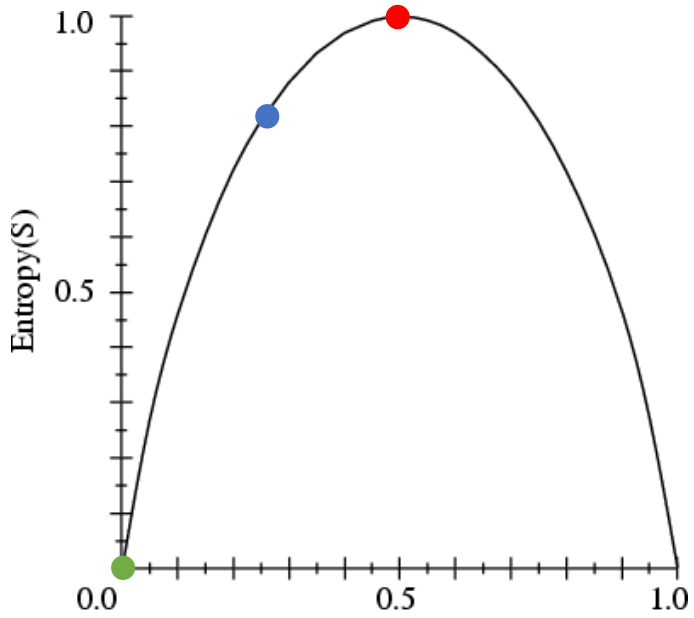
$H(Y)$=0

| | |
|---|---|
| P(Y=T) = 1/4 | P(Y=F) = 3/4 |

$H(Y)$=0.81

| | |
|---|---|
| P(Y=T) = 1/2 | P(Y=F) = 1/2 |

$H(Y)$=1



# 决策树：经验信息熵

- 经验（信息）熵：
  - 假设当前样本集合$D$中第$k$ $(k = 1, 2, \ldots, K)$类样本所占比例为$P(Y = y_k)$，则$D$的经验信息熵(简称经验熵)定义为:

$$H(Y) = -\sum_{k=1}^{K} P(Y = y_k) \log_2 P(Y = y_k)$$

  - 由 $P(Y = T) = \frac{5}{6}$ $P(Y = F) = \frac{1}{6}$
  - 得到 $H(Y) = -\frac{5}{6}\log_2\frac{5}{6} - \frac{1}{6}\log_2\frac{1}{6} = 0.65$

| | $a_1$ | $a_2$ | Y |
|---|---|---|---|
| $x_1$ | T | T | T |
| $x_2$ | T | F | T |
| $x_3$ | T | T | T |
| $x_4$ | T | F | T |
| $x_5$ | F | T | T |
| $x_6$ | F | F | F |

北京航空航天大學 软件学院

# 决策树：条件熵

- 对随机变量$(Y, a)$，联合分布为 $p(Y = y_k, a = a^v)$
- 条件熵$H(Y|a)$表示在已知随机变量$a$的条件下，随机变量$Y$的不确定性
- $H(Y|a)$定义为在$a$给定条件下$Y$的条件概率分布的熵对$a$的数学期望：

$$H(Y|a) = -\sum_{v=1}^{V} P(a = a^v) \sum_{k=1}^{K} P(Y = y_k|a = a^v) \log_2 P(Y = y_k|a = a^v)$$

- $(Y, a)$发生所包含的信息熵，减去$Y$单独发生包含的信息熵
- 也就是在$Y$发生前提下，$a$发生“新”带来的信息熵

# 决策树：条件熵

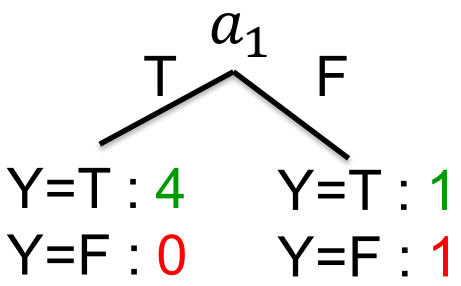
- 随机变量$Y$以随机变量$a$ 为条件的条件熵$H(Y|a)$定义为

$$H(Y|a)=-\sum_{v=1}^{V}P(a=a^v)\sum_{k=1}^{K}P(Y=y_k|a=a^v)\log_2 P(Y=y_k|a=a^v)$$

例如：

$$P(a_1=T)=\frac{4}{6}$$

$$P(a_1=F)=\frac{2}{6}$$




$$H(Y|a_1)=-\frac{4}{6}(1\log_2 1+0\log_2 0)-\frac{2}{6}\left(\frac{1}{2}\log_2\frac{1}{2}+\frac{1}{2}\log_2\frac{1}{2}\right)=0.33$$

| | $a_1$ | $a_2$ | Y |
|---|---|---|---|
| $x_1$ | T | T | T |
| $x_2$ | T | F | T |
| $x_3$ | T | T | T |
| $x_4$ | T | F | T |
| $x_5$ | F | T | T |
| $x_6$ | F | F | F |

# 决策树：条件熵

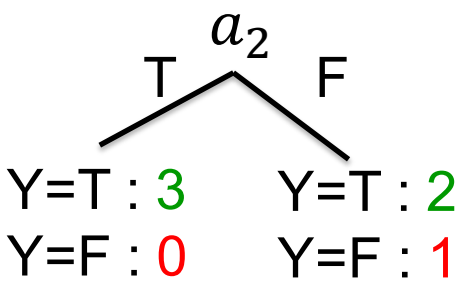
- 随机变量$Y$以随机变量$a$ 为条件的条件熵$H(Y|a)$定义为

$$H(Y|a)=-\sum_{v=1}^{V}P(a=a^v)\sum_{k=1}^{K}P(Y=y_k|a=a^v)\log_2 P(Y=y_k|a=a^v)$$

例如：

$$P(a_2=T)=\frac{3}{6}$$

$$P(a_2=F)=\frac{3}{6}$$




$$H(Y|a_2)=-\frac{3}{6}(1\log_2 1+0\log_2 0)-\frac{3}{6}\left(\frac{2}{3}\log_2\frac{2}{3}+\frac{1}{3}\log_2\frac{1}{3}\right)=0.46$$

| | $a_1$ | $a_2$ | Y |
|---|---|---|---|
| $x_1$ | T | T | T |
| $x_2$ | T | F | T |
| $x_3$ | T | T | T |
| $x_4$ | T | F | T |
| $x_5$ | F | T | T |
| $x_6$ | F | F | F |

# 决策树：信息增益(ID3)

- 信息增益(Information Gain)：划分后熵的变化：

$$IG(D,a) = H(Y) - H(Y|a)$$

划分前的经验熵　　划分后的条件熵

在本例子中：

$$IG(D,a_1) = H(Y) - H(Y|a_1)$$
$$= 0.65 - 0.33 = 0.32$$

$$IG(D,a_2) = H(Y) - H(Y|a_2)$$
$$= 0.65 - 0.46 = 0.19$$

$IG(D,a_1) > IG(D,a_2) > 0$

因此，倾向于选择$a_1$进行划分

| | $a_1$ | $a_2$ | Y |
|---|---|---|---|
| $x_1$ | T | T | T |
| $x_2$ | T | F | T |
| $x_3$ | T | T | T |
| $x_4$ | T | F | T |
| $x_5$ | F | T | T |
| $x_6$ | F | F | F |

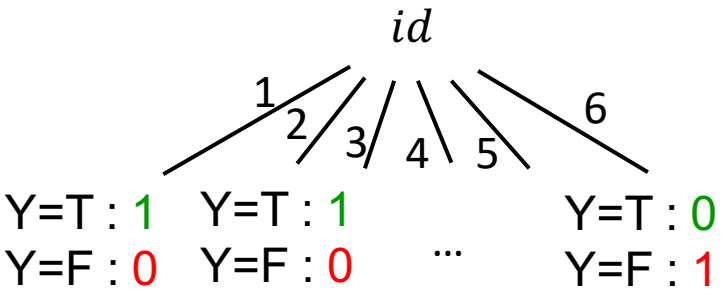
# 主观题 10分

设置

请思考，什么样的属性特征信息增益通常较大？分支多的还是分支少的？

作答

# 决策树：信息增益(ID3)




- 如果以数据$id$作为属性，划分后熵的变化：

$$H(Y|a)=-\sum_{v=1}^{V}P(a=a^v)\sum_{k=1}^{K}P(Y=y_k|a=a^v)\log_2 P(Y=y_k|a=a^v)$$

$$H(Y)=-\sum_{k=1}^{K}P(Y=y_k)\log_2 P(Y=y_k)=0.65$$

$$IG(D,a)=H(Y)-H(Y|a)$$

$$IG(D,id)=H(Y)-H(Y|id)=0.65-0$$

| | $id$ | $a_1$ | $a_2$ | Y |
|---|---|---|---|---|
| $x_1$ | 1 | T | T | T |
| $x_2$ | 2 | T | F | T |
| $x_3$ | 3 | T | T | T |
| $x_4$ | 4 | T | F | T |
| $x_5$ | 5 | F | T | T |
| $x_6$ | 6 | F | F | F |

# 决策树：信息增益率(C4.5)

- 信息增益的缺陷：变量较多的属性更容易被选择。
- 信息增益率（Gain ratio）：

$$Gain\_ratio(D,a) = \frac{H(Y) - H(Y|a)}{IV(a)}$$

其中，$IV(a)$称为属性$a$的“固有值” (Intrinsic Value)：

$$IV(a) = -\sum_{v=1}^{V} P(a = a^v)\log_2 P(a = a^v)$$

属性$a$的可能取值数目越多，则 $IV(a)$ 的值通常会越大

- 注意：为了避免对取值数目较少的属性有所偏好，算法通常先找出信息增益较高的属性，再从中选择增益率最高的。

# 决策树：基尼系数(CART)

- 另一种衡量数据纯度（不确定性）的方法：

基尼值：$Gini(D) = -\sum_{k=1}^{K}\sum_{k' \neq k} p_k p_{k'} = 1 - \sum_{k=1}^{K} p_k^2$

反映了从数据集 D 中随机抽取两个样本，其类别标记不一致的概率。

属性$a$的基尼指数：

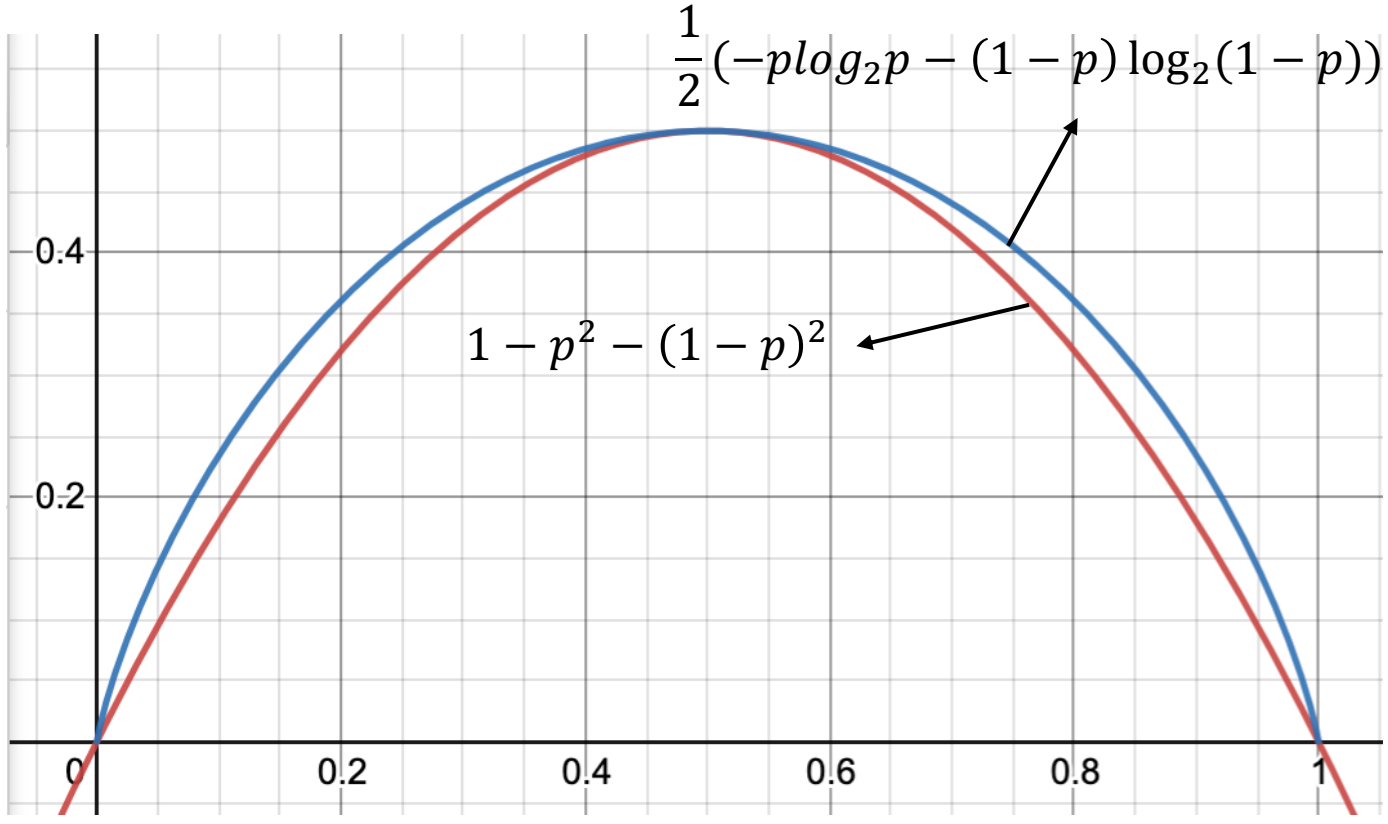
$$Gini\_index(D,a) = -\sum_{v=1}^{V} P(a = a^v) Gini(a = a^v)$$

# 决策树：基尼系数和信息熵的关系

- 在二分类问题中，

$$Gini(D) = 1 - p^2 - (1-p)^2$$

$$H(D) = -plog_2p - (1-p)\log_2(1-p)$$



北京航空航天大學 软件学院

# 决策树：训练流程总结

- 从空决策树开始
- 选择下一个最佳属性进行划分，选择依据：
  - 信息增益

$$a_* = \arg\max_{a \in A} IG(D, a) = \arg\max_{a \in A} H(Y) - H(Y|a)$$

  - 或信息增益率

$$a_* = \arg\min_{a \in \mathrm{A}} Gain\_ratio(D, a) = \arg\max_{a \in A} \frac{H(Y) - H(Y|a)}{IV(a)}$$

  - 或基尼系数

$$a_* = \arg\min_{a \in \mathrm{A}} Gini\_index(D, a)$$

- 迭代

# 决策树：基本算法

**输入：** 训练集 $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$;
属性集 $A = \{a_1, a_2, \ldots, a_d\}$.

**过程：** 函数 TreeGenerate($D$, $A$)

```
1: 生成结点 node;
2: if D 中样本全属于同一类别 C then
3:     将 node 标记为 C 类叶结点; return
4: end if
5: if A = ∅ OR D 中样本在 A 上取值相同 then
6:     将 node 标记为叶结点, 其类别标记为 D 中样本数最多的类; return
7: end if
8: 从 A 中选择最优划分属性 a_*;
9: for a_* 的每一个值 a_*^v do
10:    为 node 生成一个分支; 令 D_v 表示 D 中在 a_* 上取值为 a_*^v 的样本子集;
11:    if D_v 为空 then
12:        将分支结点标记为叶结点, 其类别标记为 D 中样本最多的类; return
13:    else
14:        以 TreeGenerate(D_v, A \ {a_*})为分支结点
15:    end if
16: end for
```

**输出：** 以 node 为根结点的一棵决策树

递归返回，情形(1)

递归返回，情形(2)

利用当前节点的后验分布

递归返回，情形(3)

将父节点的样本分布作为当前节点的先验分布

决策树算法的核心

# 划分选择 vs. 剪枝

- 研究表明: 划分选择的各种准则虽然对决策树的尺寸有较大影响，但对泛化性能的影响很有限
- 例如信息增益与基尼指数产生的结果，仅在约 2% 的情况下不同

- 剪枝方法和程度对决策树泛化性能的影响更为显著
  - 在数据带噪时甚至可能将泛化性能提升 25%

剪枝是决策树对付“过拟合”的主要手段！

# 决策树：剪枝

- 为了尽可能正确分类训练样本，有可能造成分支过多 → 过拟合

可通过主动去掉一些分支来降低过拟合的风险

- 基本策略：
  - 预剪枝 (pre-pruning): 提前终止某些分支的生长
  - 后剪枝 (post-pruning): 生成一棵完全树，再“回头”剪枝

# 决策树：剪枝

- 剪枝方法：
  - 剪掉某节点为根节点的子树
  - 将该节点设置为叶节点
- 此节点的纯度不一定高
  - 其类别标记为训练集中样本最多的类
- 剪枝将导致训练误差增大，
  - 何时停止剪枝？--利用验证集
  - 哪些节点被剪掉？--剪掉提升验证集性能的节点

北京航空航天大学 软件学院

# 提纲


一、机器学习基本概念

二、线性回归与线性分类

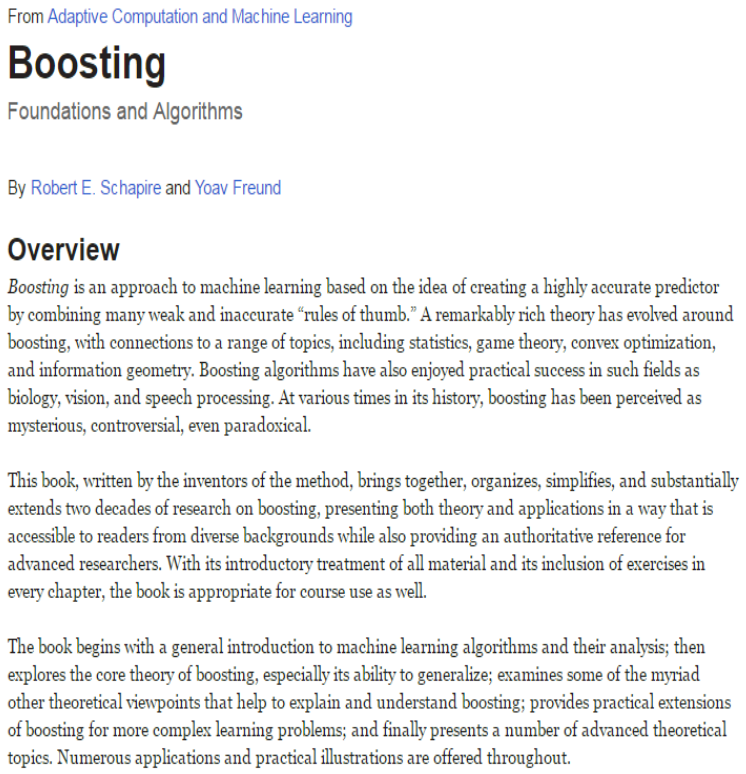
三、线性判别分析

四、支持向量机

五、决策树

六、集成学习

# 集成学习



From Adaptive Computation and Machine Learning

## Boosting

Foundations and Algorithms

By Robert E. Schapire and Yoav Freund

### Overview

*Boosting* is an approach to machine learning based on the idea of creating a highly accurate predictor by combining many weak and inaccurate "rules of thumb." A remarkably rich theory has evolved around boosting, with connections to a range of topics, including statistics, game theory, convex optimization, and information geometry. Boosting algorithms have also enjoyed practical success in such fields as biology, vision, and speech processing. At various times in its history, boosting has been perceived as mysterious, controversial, even paradoxical.

This book, written by the inventors of the method, brings together, organizes, simplifies, and substantially extends two decades of research on boosting, presenting both theory and applications in a way that is accessible to readers from diverse backgrounds while also providing an authoritative reference for advanced researchers. With its introductory treatment of all material and its inclusion of exercises in every chapter, the book is appropriate for course use as well.

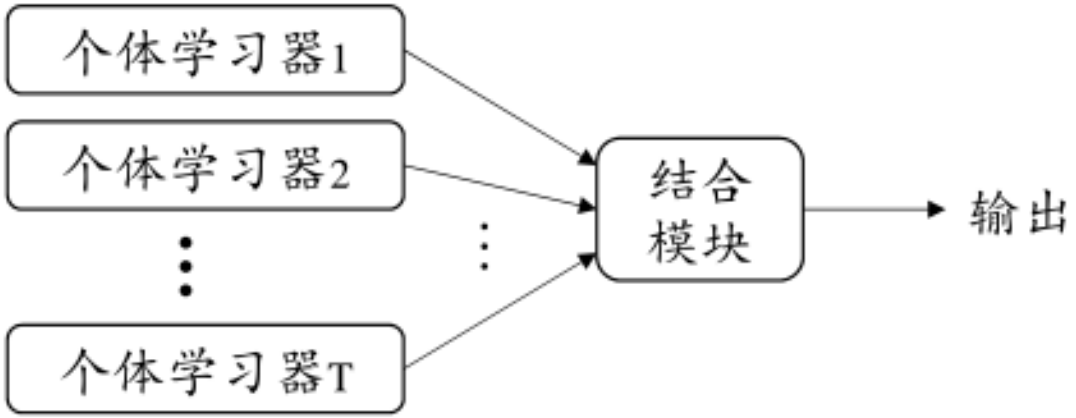
The book begins with a general introduction to machine learning algorithms and their analysis; then explores the core theory of boosting, especially its ability to generalize; examines some of the myriad other theoretical viewpoints that help to explain and understand boosting; provides practical extensions of boosting for more complex learning problems; and finally presents a number of advanced theoretical topics. Numerous applications and practical illustrations are offered throughout.

- 对于一个复杂的分类任务，可以将其分解为若干子任务，然后将若干子任务完成方法综合，最终完成该复杂任务。
- 将若干个弱分类器(weak classifiers)组合起来，形成一个强分类器(strong classifier)。
- 能用众力，则无敌于天下矣；**能用众智，则无畏于圣人矣**(语出《三国志·吴志·孙权传》)

Freund, Yoav; Schapire, Robert E (1997), A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences (original paper of Yoav Freund and Robert E.Schapire where AdaBoost is first introduced.)

# 集成学习：个体与集成

- 集成学习(ensemble learning)通过构建并结合多个学习器来提升性能



# 主观题 10分

设置

想要集结多个弱分类器，形成强分类器，请思考，这些弱分类应具备什么样的条件?

作答

北京航空航天大学 软件学院

# 集成学习：个体与集成

- 考虑一个简单的例子，在二分类问题中，假定3个分类器在三个样本中的表现如下图所示，其中√ 表示分类正确，× 号表示分类错误，集成的结果通过投票产生。

| | 测试例1 | 测试例2 | 测试例3 |
|---|---|---|---|
| $h_1$ | √ | √ | × |
| $h_2$ | × | √ | √ |
| $h_3$ | √ | × | √ |
| 集群 | √ | √ | √ |

(a) 集群提升性能

| | 测试例1 | 测试例2 | 测试例3 |
|---|---|---|---|
| $h_1$ | √ | √ | × |
| $h_2$ | √ | √ | × |
| $h_3$ | √ | √ | × |
| 集群 | √ | √ | × |

(b) 集群不起作用

| | 测试例1 | 测试例2 | 测试例3 |
|---|---|---|---|
| $h_1$ | √ | × | × |
| $h_2$ | × | √ | × |
| $h_3$ | × | × | √ |
| 集群 | × | × | × |

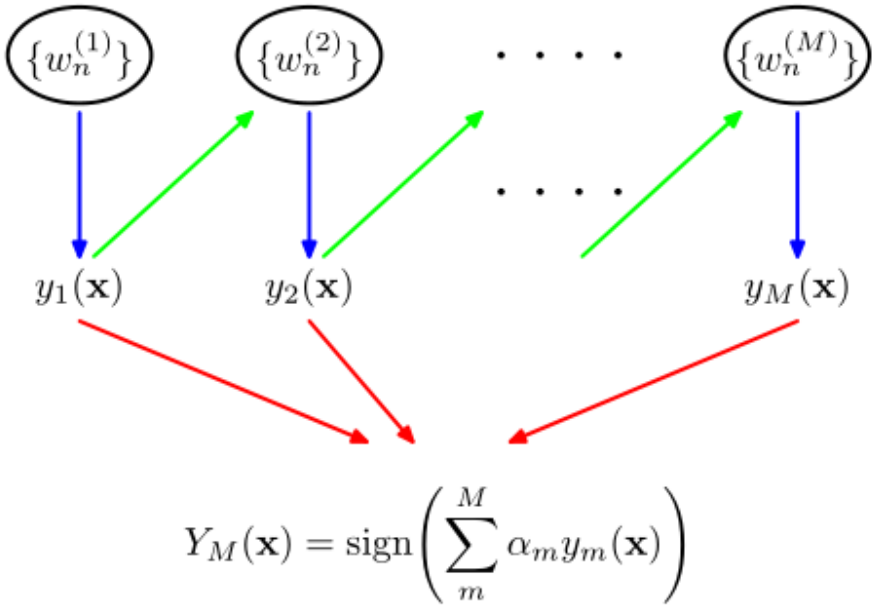
(c) 集群起负作用

- 集成个体应该：好而不同

# 集成学习：提升法 Boosting

- 提升法 Boosting
  - 个体学习器存在强依赖关系
  - 串行生成
  - 每次调整训练数据的样本分布

  - Boosting族算法最著名的代表是AdaBoost



# 主观题 10分

设置

为了满足“好而不同”，请思考，给定训练集，串行学习个体学习器，也就是弱分类器时，有哪些思路可以保证这些弱分类器之间好而不同。

作答

# 自适应提升 Adaboost：思路描述

- Adaboost 迭代算法有三步：
  1. 初始化训练样本的权值分布，每个样本具有相同权重；
  2. 训练弱分类器，如果样本分类正确，则在构造下一个训练集中，它的权值就会被降低；反之提高。用更新过的样本集去训练下一个分类器；
  3. 将所有弱分类组合成强分类器，各个弱分类器的训练过程结束后，加大分类误差率小的弱分类器的权重，降低分类误差率大的弱分类器的权重。

# 自适应提升 Adaboost：思路描述

- Adaboost算法中两个核心问题：
  1. 在每个弱分类器学习过程中，如何改变训练数据的权重：提高在上一轮中分类错误样本的权重。
  2. 如何将一系列弱分类器组合成强分类器：通过加权多数表决方法来提高分类误差小的弱分类器的权重，让其在最终分类中起到更大作用。同时减少分类误差大的弱分类器的权重，让其在最终分类中仅起到较小作用。

# 自适应提升 Adaboost：思路描述

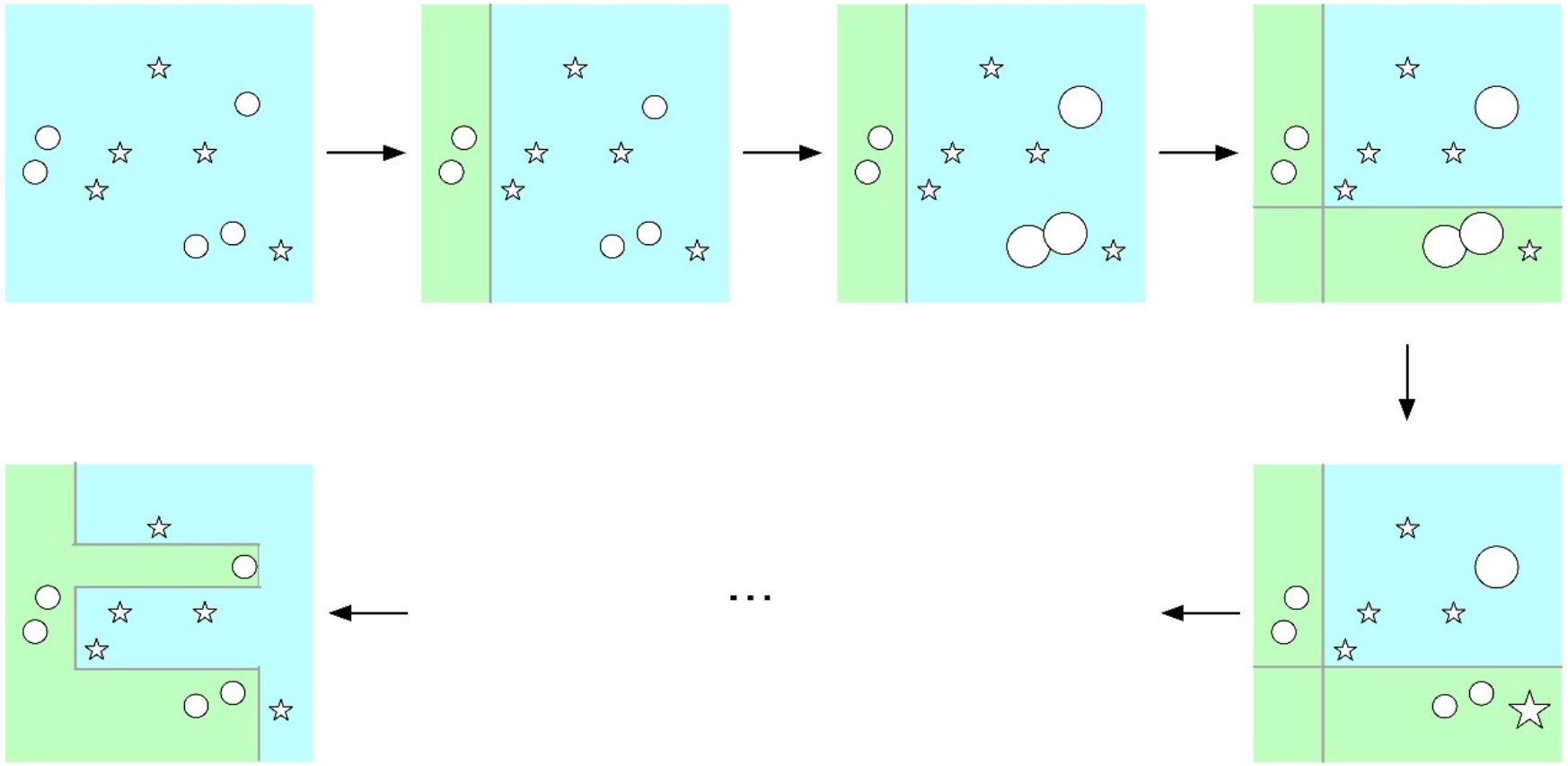


图4.9 Adaboost算法学习过程示意图

北京航空航天大學 软件学院

# 总结

## 一、机器学习基本概念

## 二、线性回归与线性分类

## 三、线性判别分析

## 四、支持向量机

## 五、决策树

## 六、集成学习