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

人工智能

第7讲：机器学习及有监督学习-III

张晶

2023年春季

- 参考教材： 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

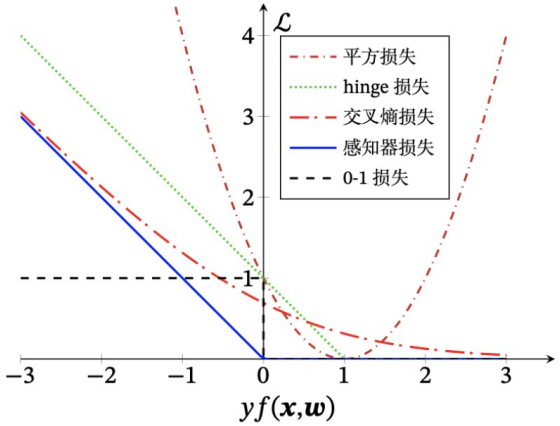
北京航空航天大學 软件学院

提纲

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

北京航空航天大学 软件学院

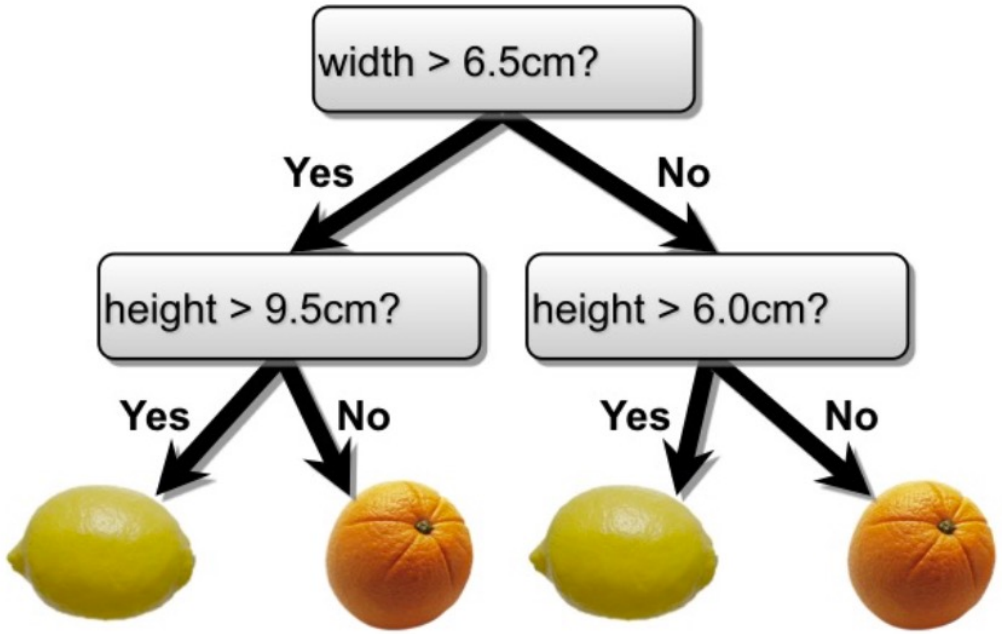
回顾：线性分类模型



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
		0-1损失	$\begin{cases}1, f(\boldsymbol{x}_i) \neq y_i \\ 0, f(\boldsymbol{x}_i) = y_i\end{cases}$	
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	$sign(\boldsymbol{w}^T\boldsymbol{x})$	感知器准则	$\max(0, -y_i\boldsymbol{w}^T\boldsymbol{x}_i)$	随机梯度下降
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

决策树 – 另一种分类思想

- 决策树是一种通过树形结构来进行分类的方法。
- 在决策树中，树形结构中每个非叶子节点表示对分类目标在某个属性上的一个判断，每个分支代表基于该属性做出的一个判断，最后树形结构中每个叶子节点代表一种分类结果，
- 所以决策树可以看作是一系列以叶子节点为输出的决策规则（Decision Rules）[Quinlan 1987]。



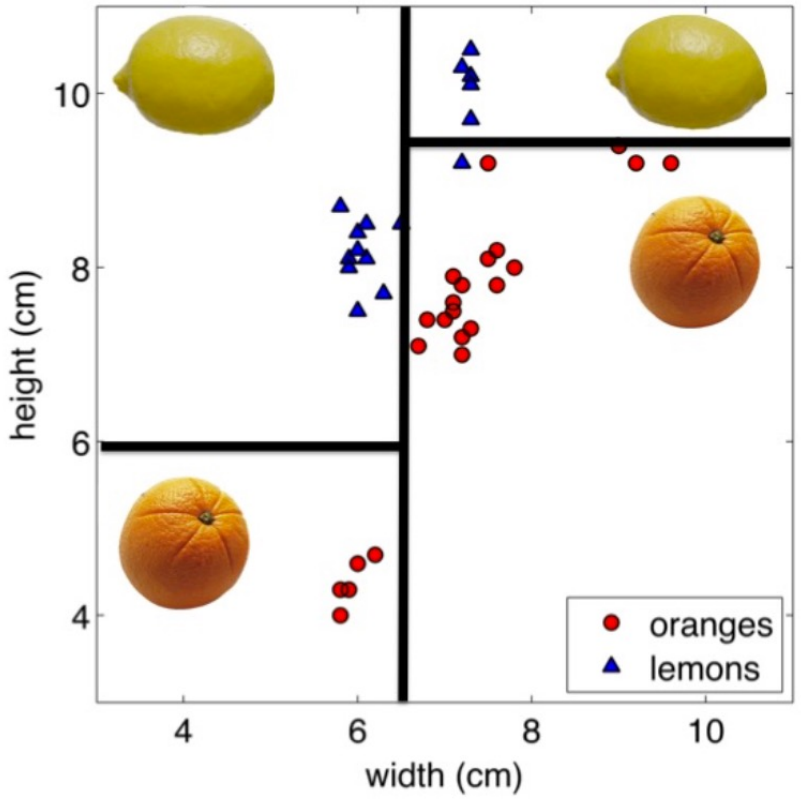
北京航空航天大學 软件学院

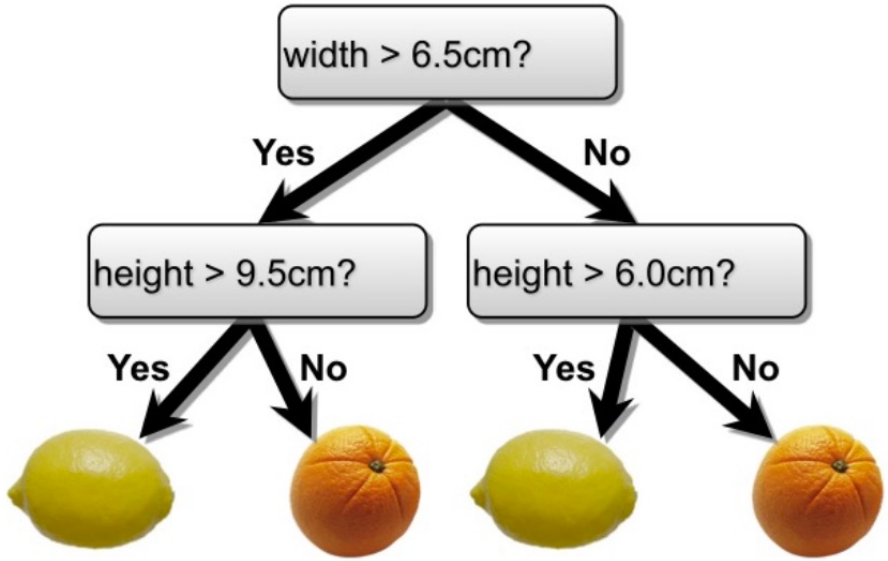
决策树简史

- 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"]

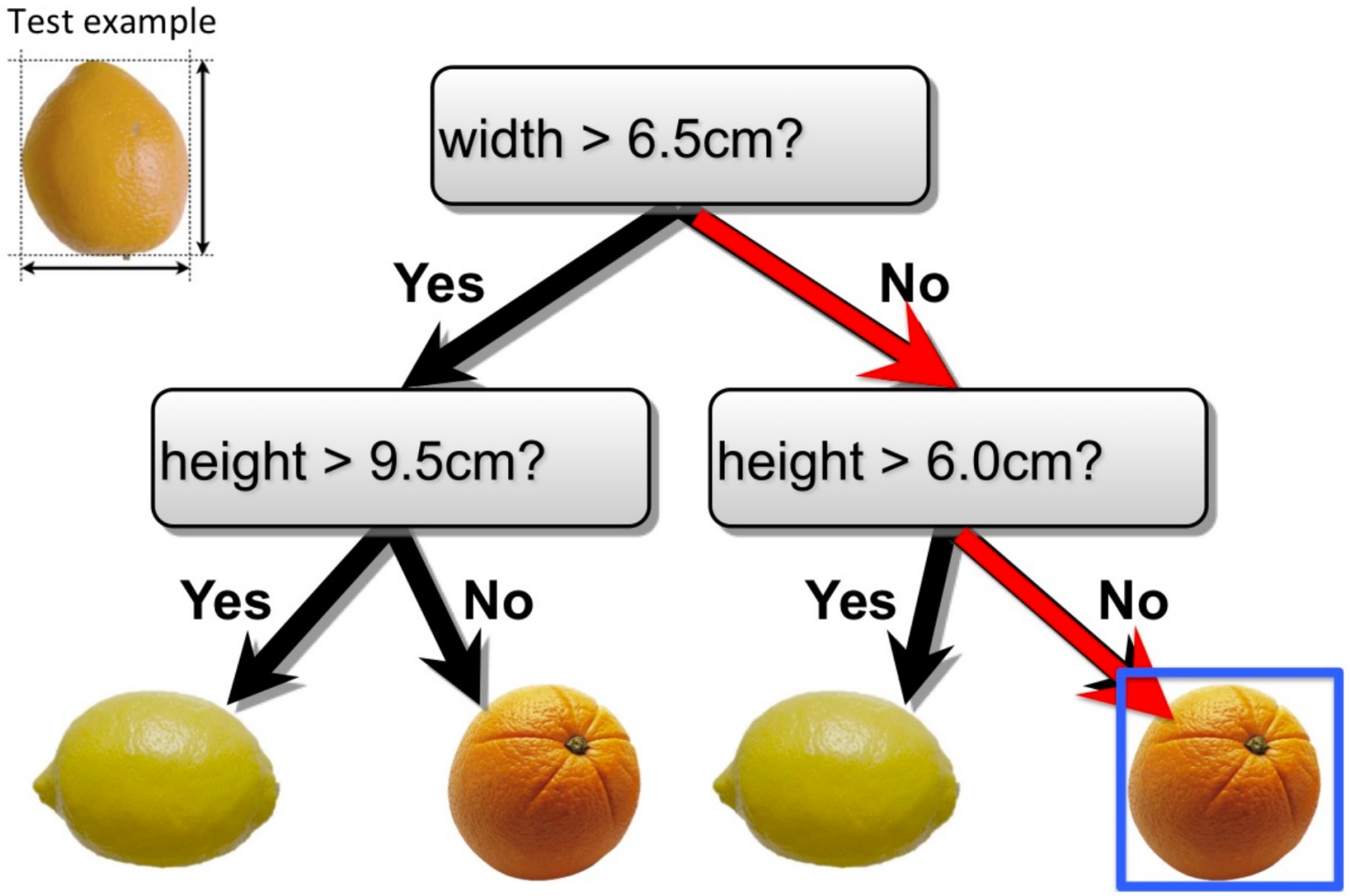
决策树：分类训练

- 得到跟特征向量平行的分类器





决策树：分类测试


Test example
width > 6.5cm?
Yes
No
height > 9.5cm?
height > 6.0cm?
Yes
No
Yes
No

决策树：基本流程

策略：“分而治之”(divide-and-conquer)

自根至叶的递归过程

在每个中间结点寻找一个“划分”(split or test)属性

三种停止条件：

(1) 当前结点包含的样本全属于同一类别，无需划分;

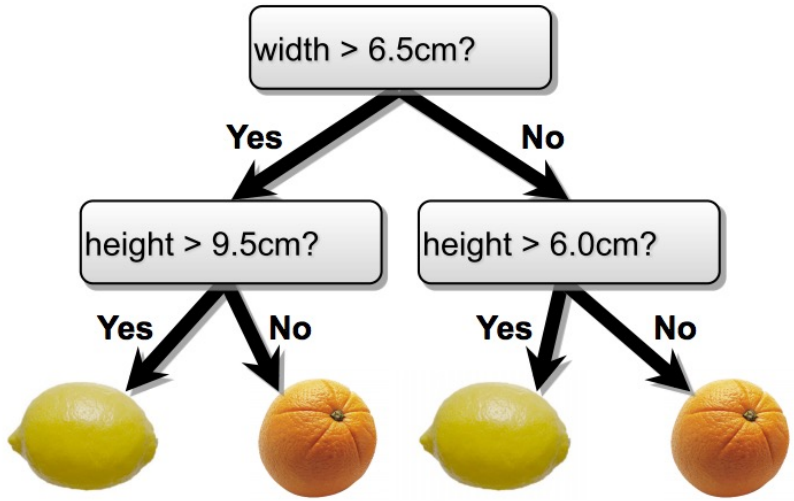
(2) 当前属性集为空, 或是所有样本在所有属性上取值相同，无法划分;

(3) 当前结点包含的样本集合为空，不能划分.

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: $y_k \in \{y_1, y_2, \ldots, y_K\}$
- 训练数据样本子集D_v: D中 a_i取值 $a_i = a_i^v$的样本集合

- 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：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)

利用当前结点的后验分布

递归返回，情形(3)

将父结点的样本分布作为当前结点的先验分布

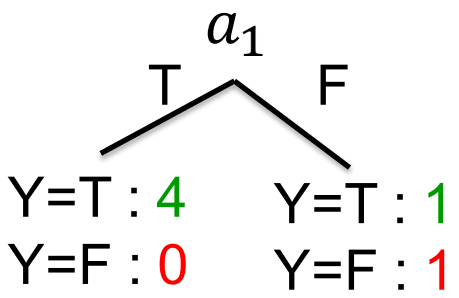
决策树算法的核心

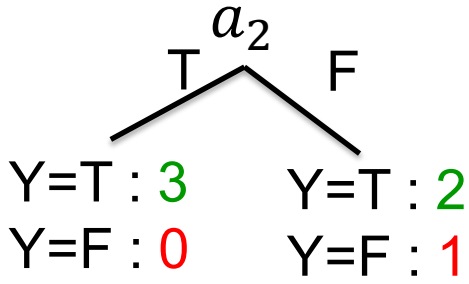
决策树：最优划分属性选择

- 寻找最优划分方式：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

- 假设具有2个属性的6个训练样本的训练计划，选择 a_1 or 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

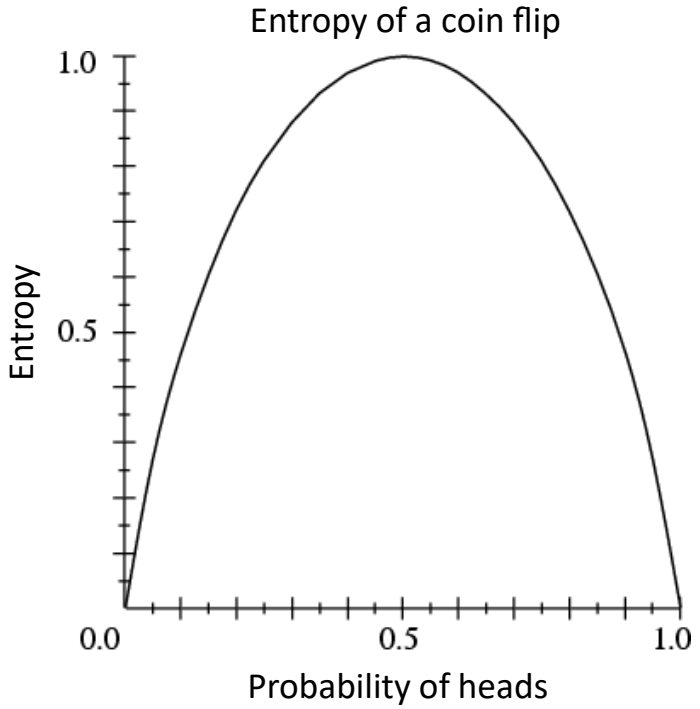
决策树：信息熵

- 随机变量Y的信息熵 $H(Y)$定义如下：

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

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

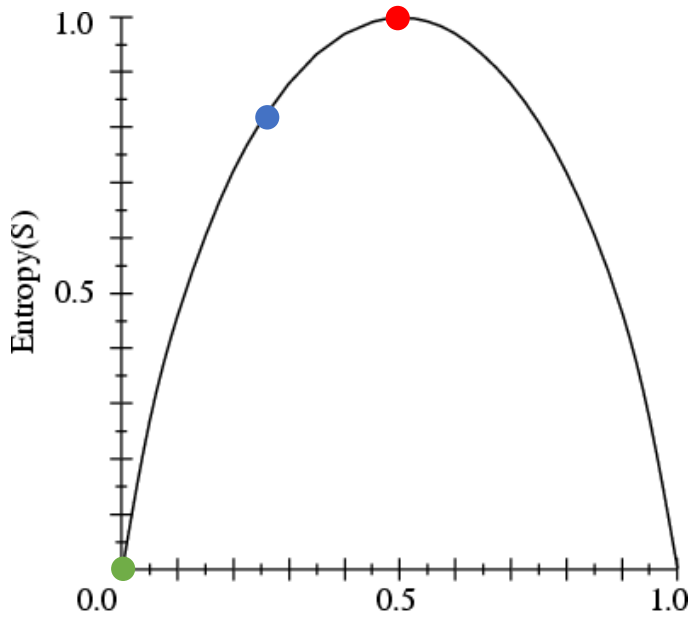
$H(Y)$=0

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

$H(Y)$=0.81

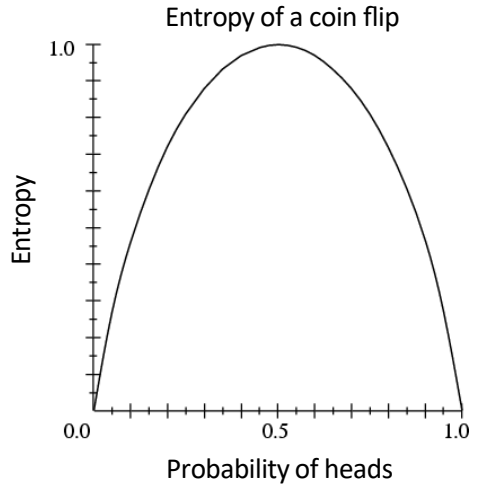
P(Y=T) = 1/2	P(Y=F) = 1/2

$H(Y)$=1



决策树：信息熵

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



P(Y=T) = 5/6

P(Y=F) = 1/6

$$H(Y) = -\frac{5}{6}\log_2\frac{5}{6} - \frac{1}{6}\log_2\frac{1}{6} = 0.65$$

	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

决策树：条件熵

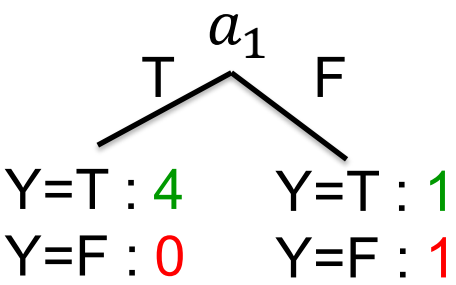
- 随机变量Y以随机变量a 为条件的条件熵$H(Y|a)$定义为

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

例如：

P(a_1 =T) = 4/6

P(a_1 =F) = 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

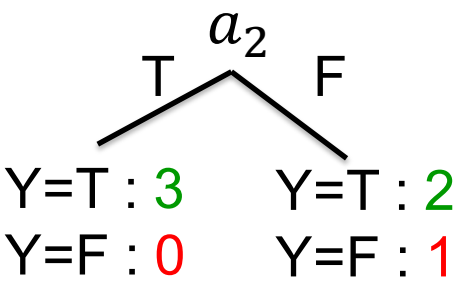
- 随机变量Y以随机变量a 为条件的条件熵$H(Y|a)$定义为

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

例如：

P(a_2 =T) = 3/6

P(a_2 =F) = 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：

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

划分前的信息熵　　划分后的信息熵

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

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)：

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

- 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：

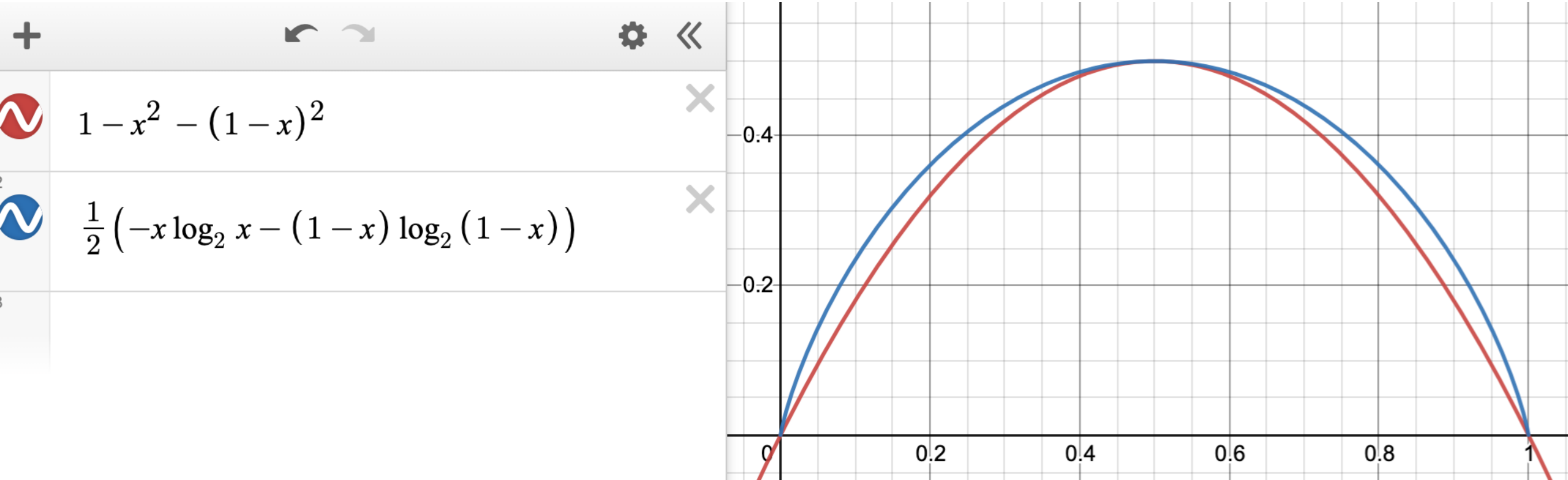
$$Gini\_index(D,a) = -\sum_{v=1}^{V} P(a = a^v) Gini(a = a^v)$$

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

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
- 此叶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

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

提纲

Boosting (adaptive boosting, 自适应提升)

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.)

Ada Boosting：思路描述

- Ada Boosting 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。

Ada Boosting：思路描述

- Ada Boosting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。

Ada Boosting：思路描述

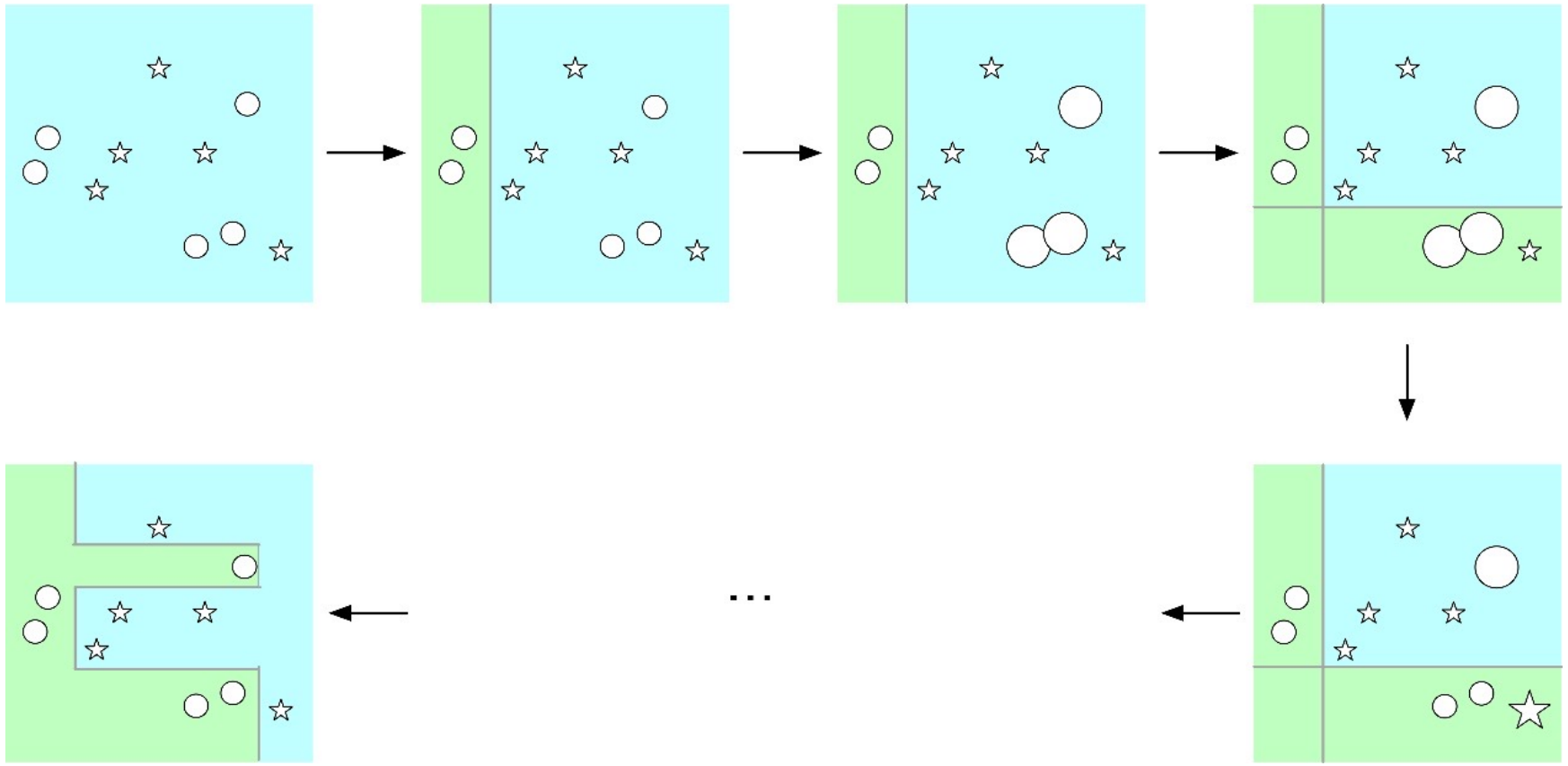


图4.9 Ada Boosting算法学习过程示意图

Ada Boosting：算法描述

- 给定包含N个标注数据的训练集合D，$D = \{(x_1, y_1), \ldots, (x_N, y_N)\}$。

  $x_i (1 \leq i \leq N) \in X \subseteq R^n, y_i \in Y = \{-1, 1\}$

- Ada Boosting算法将从这些标注数据出发，训练得到一系列弱分类器

  $G_m(x): X \to \{-1, 1\}$，并将这些弱分类器线性组合得到一个强分类器。

$$G(\boldsymbol{x}) = \sum_{m=1}^{M} \alpha_m G_m(\boldsymbol{x})$$

来最小化指数损失函数

$$Loss_{exp}(G|D) = E_{\boldsymbol{x} \sim D}[e^{-yG(\boldsymbol{x})}]$$

Ada Boosting：算法描述---数据样本权重初始化

1. 给初始化每个训练样本的权重
   - $\boldsymbol{w}_1 = (w_{11}, \ldots, w_{1i}, \ldots, w_{1N})$，其中 $w_{1i} = \frac{1}{N}(1 \leq i \leq N)$

Ada Boosting：算法描述---第m个弱分类器训练

2. 对$m = 1,2,\ldots,M$

a) 使用具有分布权重W_m的训练数据来学习得到第m个基分类器（弱分类器）G_m：

$$G_m(x): X \to \{-1,1\}$$

b) 计算$G_m(x)$在训练数据集上的分类误差

$$err_m = \sum_{i=1}^{N} w_{mi} I(G_m(x_i) \neq y_i)$$

这里：$I(\cdot) = 1$，如果$G_m(x_i) \neq y_i$；否则为0

c) 计算弱分类器$G_m(x)$的权重：

$$\alpha_m = \frac{1}{2} \ln \frac{1 - err_m}{err_m}$$

d) 更新训练样本数据的分布权重$\boldsymbol{w}_{m+1} = \{w_{m+1,1}, \ldots, w_{m+1,N}\}$，其中：

$$w_{m+1,i} = \frac{w_{m,i}}{z_m} e^{-\alpha_m y_i G_m(x_i)}$$

其中z_m是归一化因子以使得$\boldsymbol{w}_{m+1}$为概率分布，$z_m = \sum_{i=1}^{N} w_{m,i}\, e^{-\alpha_m y_i G_m(x_i)}$

Ada Boosting：算法描述---弱分类器组合成强分类器

3. 对以线性加权形式来组合弱分类器$f(x)$

$$f(x) = \sum_{i=1}^{M} \alpha_m G_m(x)$$

得到强分类器$G(x)$

$$G(x) = sign(f(x)) = sign(\sum_{i=1}^{M} \alpha_m G_m(x))$$

Ada Boosting：算法解释

- 第m个弱分类器$G_m(x)$在训练数据集上产生的分类误差：
  - 该误差为被错误分类的样本所具有权重的累加

$$err_m = \sum_{i=1}^{N} w_{m,i} I(G_m(x_i) \neq y_i)$$

  - 这里：$I(\cdot) = 1$，如果$G_m(x_i) \neq y_i$；否则为0

Ada Boosting：算法解释

$$\alpha_m = argmin_\alpha Loss_{exp}(G_m|D)$$
$$= argmin_\alpha E_{x\sim D}\left[e^{-y\alpha_m G_m(x)}\right]$$
$$= argmin_\alpha E_{x\sim D}[e^{-\alpha_m} I(G_m(x) = y) + e^{\alpha_m} I(G_m(x) \neq y)]$$
$$= argmin_\alpha e^{-\alpha_m}(1-\epsilon_m) + e^{\alpha_m}\epsilon_m$$

- 计算第m个弱分类器$G_m(x)$的权重α_m：

$$\alpha_m = \frac{1}{2} ln \frac{1-err_m}{err_m}$$

1. 当第m个弱分类器$G_m(x)$ 错误率为1，即

$$err_m = \sum_{i=1}^{N} w_{mi} I(G_m(x_i) \neq y_i) = 1$$

意味每个样本分类出错，则$\alpha_m = \frac{1}{2} ln \frac{1-err_m}{err_m} \to -\infty$，给予第$m$个弱分类器$G_m(x)$很低权重。

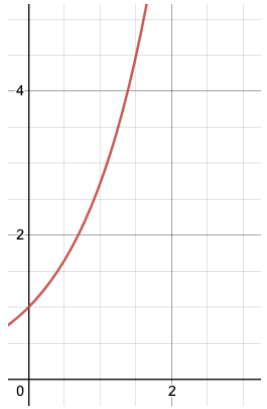
2. 当第m个弱分类器$G_m(x)$错误率为$\frac{1}{2}$，$\alpha_m = \frac{1}{2} ln \frac{1-err_m}{err_m} = 0$。如果错误率$err_m$小于$\frac{1}{2}$，权重$\alpha_m$为正($err_m < \frac{1}{2}$、$\alpha_m > 0$)。可知权重$\alpha_m$随$err_m$减少而增大，即错误率越小的弱分类器会赋予更大权重。

3. 如果一个弱分类器的分类错误率为$\frac{1}{2}$，可视为其性能仅相当于随机分类效果。

Ada Boosting：算法解释

- 在开始训练第$m+1$个弱分类器$G_{m+1}(x)$之前对训练数据集中数据权重进行调整

$$w_{m+1,i}=\begin{cases}\dfrac{w_{m,i}}{Z_m}e^{-\alpha_m}, & G_m(x_i)=y_i\\ \dfrac{w_{m,i}}{Z_m}e^{\alpha_m}, & G_m(x_i)\neq y_i\end{cases}$$



  - 可见，如果某个样本无法被第m个弱分类器$G_m(x)$分类成功，则需要增大该样本权重，否则减少该样本权重。这样，被错误分类样本会在训练第$m+1$个弱分类器$G_{m+1}(x)$时会被“重点关注”。
  - 在每一轮学习过程中，Ada Boosting算法均在划重点（重视当前尚未被正确分类的样本）

Ada Boosting：算法解释

- 弱分类器构造强分类器：

$$f(x) = \sum_{m=1}^{M} \alpha_m G_m(x)$$

$$G(x) = sign(f(x)) = sign(\sum_{m=1}^{M} \alpha_m G_m(x))$$

  - $f(x)$是M个弱分类器的加权线性累加。分类能力越强的弱分类器具有更大权重。
  - α_m累加之和并不等于1。
  - $f(x)$符号决定样本x分类为1或-1。如果$\sum_{i=1}^{M} \alpha_m G_m(x)$为正，则强分类器$G(x)$将样本$x$分类为1；否则为-1。

Ada Boosting：优化目标

- Ada Boosting实际在最小化如下指数损失函数(minimization of exponential loss)：

$$\sum_i e^{-y_i f(x_i)} = \sum_i e^{-y_i \sum_{m=1}^{M} \alpha_m G_m(x_i)}$$

- Ada Boosting的分类误差上界如下所示：

$$\frac{1}{N}\sum_{i=1}^{N} I(G(x_i) \neq y_i) \leq \frac{1}{N}\sum_i e^{-y_i f(x_i)} = \prod_m Z_m$$

- 在第m次迭代中，Ada Boosting总是趋向于将具有最小误差的学习模型选做本轮生成的 弱分类器 G_m，使得累积误差快速下降。

Ada Boosting：例子

- 通过一个简单两类分类例子来介绍Ada Boosting算法过程。表4.5.1给出了10个数据点$x_i(\ i \in \{1,2,\ldots,10\})$取值及其所对应的类别标签$y_i \in \{1,-1\}(i \in \{1,2,\ldots,10\})$。

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
x	-9	-7	-5	-3	-1	1	3	5	7	9
y	-1	-1	1	1	-1	-1	-1	-1	1	1

表4.8 两类分类问题数据

- 根据表4.8所给出的数据，要构造若干个弱分类器，然后将这些弱分类器组合为一个强分类器，完成表4.8所示数据的分类任务。

Ada Boosting：例子

- 这里定义每个弱分类器G为一种分段函数，由一个阈值ε构成，形式如下：

$$G(x_i)=\begin{cases}-1 & x_i<\varepsilon\\ 1 & x_i>\varepsilon\end{cases} \quad 或 \quad G(x_i)=\begin{cases}1 & x_i<\varepsilon\\ -1 & x_i>\varepsilon\end{cases}$$

当然，在实际中，可根据需要使用其他的弱分类器。

- Ada Boosting主要步骤：

（1） 数据样本权重初始化

$$\boldsymbol{w}_1=(w_{1,1},\dots,w_{1,i},\dots,w_{1,10})，其中w_{1,i}=\frac{1}{10}(1\le i\le 10)$$

Ada Boosting：例子

$G_1(x)$

	1	2	3	4	5	6	7	8	9	10
x	-9	-7	-5	-3	-1	1	3	5	7	9
y	-1	-1	1	1	-1	-1	-1	-1	1	1

（2） 分别训练M个基分类器（弱分类器）

对$m=1$

- 使用具有分布权重W_1的训练数据来学习得到第$m=1$个基分类器G_1。不难看出，当阈值$\varepsilon=6$时，基分类器G_1具有最小错误率。G_1分类器如下表示：

$$G_1(x_i)=\begin{cases}-1 & x_i<6\\ 1 & x_i>6\end{cases}$$

- 计算$G_1(x)$在训练数据集上的分类误差，样例3、4被错误分类，因此G_1的分类误差为

$$err_1=\sum_{i=1}^{N} w_{1,i} I(G_1(x_i)\neq y_i)=0.1+0.1=0.2$$

- 根据分类误差计算弱分类器$G_1(x)$的权重：

$$\alpha_1=\frac{1}{2}ln\frac{1-err_1}{err_1}=0.6931$$

- 更新下一轮第$m=2$个分类器训练时第i个训练样本的权重：$\boldsymbol{w}_2=\{w_{2,i}\}_1^{10}, w_{2,i}=\frac{w_{1,i}}{Z_1}e^{-\alpha_1 y_i G_1(x_i)}$，可得到数据样本新的权重：

$$\boldsymbol{w}_2=(0.0625,0.0625,0.25,0.25,0.0625,0.0625,0.0625,0.0625,0.0625,0.0625)$$

- 通过加权线性组合得到当前的分类器$f_1(x)=\sum_{i=1}^{M}\alpha_m G_m(x)=0.6931G_1(x)$
- 在上图中，被分类错误数据样本的形状尺寸比其它数据样本形状稍大，以表示被分类错误的样本数据权重增大。

Ada Boosting：例子

			$G_2(x)$						$G_1(x)$	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
x	-9	-7	-5	-3	-1	1	3	5	7	9
y	-1	-1	1	1	-1	-1	-1	-1	1	1

对$m=2$

- 对于具有分布权重为W_2的训练数据，当阈值$\varepsilon=-6$时，基分类器G_2具有最小的错误率。G_2分类器如下表示：

$$G_2(x_i)=\begin{cases}-1 & x_i<-6\\ 1 & x_i>-6\end{cases}$$

- 分类误差

$$err_2=\sum_{i=1}^{N} w_{2,i}I(G_2(x_i)\neq y_i)=0.25$$

- 弱分类器$G_2(x)$的权重

$$\alpha_2=\frac{1}{2}\ln\frac{1-err_2}{err_2}=0.5439$$

- 当进行下一轮分类器训练时，由 $w_{3,i}=\frac{w_{2,i}}{Z_2}e^{-\alpha_2 y_i G_2(x_i)}$，样本权重更新如下：

$$\boldsymbol{w}_3=(0.04166667,0.04166667,0.16666667,0.16666667,0.125,0.125,0.125,0.125,0.04166667,0.04166667)$$

- 通过加权线性组合得到当前的分类器$f_2(x)=0.6931G_1(x)+0.5439G_2(x)$

Ada Boosting：例子

	$G_2(x)$		$G_3(x)$					$G_1(x)$	
	1	2	3	4	5	6	7	8	9	10
x	-9	-7	-5	-3	-1	1	3	5	7	9
y	-1	-1	1	1	-1	-1	-1	-1	1	1

对$m=3$

- 对于具有分布权重为D_3的训练样本数据，当阈值$\varepsilon=-2$时，基分类器G_3具有最小的错误率。G_3分类器表示如下：

$$G_3(x_i)=\begin{cases}-1 & x_i>-2\\ 1 & x_i<-2\end{cases}$$

- 分类误差

$$err_3=\sum_{i=1}^{N} w_{3,i} I(G_3(x_i)\neq y_i)=0.1667$$

- 弱分类器$G_3(x)$的权重

$$\alpha_3=\frac{1}{2}\ln\frac{1-err_3}{err_3}=0.8047$$

- 下一轮弱分类器训练时，由$w_{4,i}=\frac{w_{3,i}}{Z_3}e^{-\alpha_3 y_i G_3(x_i)}$，训练数据样本的权重更新如下：

$$\boldsymbol{w}_4=(0.125,0.125,0.1,0.1,0.075,0.075,0.075,0.075,0.125,0.125)$$

- 通过加权线性组合得到当前的分类器$f_3(x)=0.6931G_1(x)+0.5439G_2(x)+0.8047G_3(x)$

Ada Boosting：例子

			$G_2(x)$		$G_3(x)$				$G_1(x)$	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
x	-9	-7	-5	-3	-1	1	3	5	7	9
y	-1	-1	1	1	-1	-1	-1	-1	1	1

（3） 构造强分类器

- 在$f_3(x)$的基础上，构造强分类器

$$G(x) = sign\big(f_3(x)\big) = sign(0.6931G_1(x) + 0.5439G_2(x) + 0.8047G_3(x))$$

- 这里$sign(\cdot)$是符号函数，其输入值大于0时，符号函数输出为1，反之为-1。由于$G(x)$在训练样本上分类错误率为0，算法终止，得到最终的强分类器。

北京航空航天大學 软件学院

提纲

北京航空航天大學 软件学院

决策学习的三种学习方法

- 判别函数方法
  - 不假设概率模型，直接将输入数据$\boldsymbol{x}$映射到特定类别。
  - 比如线性判别分析、感知器、支持向量机等
- 概率判别式学习方法
  - 直接估计$p(y = c_i|\boldsymbol{x})$
  - 比如对数几率回归等
- 概率生成式学习方法
  - 先估计$p(\boldsymbol{x}|y = c_i)$ 和 $p(y)$，再通过贝叶斯公式得到$p(y = c_i|\boldsymbol{x})$
  - 比如朴素贝叶斯、隐马尔科夫模型、受限玻尔兹曼机等

生成式学习模型

- 生成式学习方法从数据中学习联合概率分布$P(X,C)$，然后求出条件概率分布$P(C|X)$作为预测模型，即$P(c_i|\boldsymbol{x}) = \frac{P(\boldsymbol{x},c_i)}{P(\boldsymbol{x})}$。

$$P(\boldsymbol{x},c_i) = \underbrace{P(\boldsymbol{x}|c_i)}_{\text{似然概率}} \times \underbrace{P(c_i)}_{\text{先验概率}} \Rightarrow \underbrace{P(c_i|\boldsymbol{x})}_{\text{后验概率}} = \frac{\overbrace{P(\boldsymbol{x},c_i)}^{\text{联合概率}}}{P(\boldsymbol{x})} = \frac{\underbrace{P(\boldsymbol{x}|c_i)}_{\text{似然概率}} \times \underbrace{P(c_i)}_{\text{先验概率}}}{P(\boldsymbol{x})}$$

- 在表4.10中，一共有12个样本-标签数据，其中
  - (*x=0,c=*阳性)样本出现了6次
  - (*x=0,c=*阴性)样本出现了2次
  - (*x=1,c=*阳性)样本没有出现
  - (*x=1,c=*阴性)样本出现了4次

表4.10 输入数据-类别标签的样本分布

输入 \ 类别	阳性	阴性
$x = 0$	6	2
$x = 1$	0	4

生成学习模型

生成式学习

- 输入样本和类别标签的联合概率分布为：

$$P(x=0,y=\text{阳性})=\frac{6}{12}=\frac{1}{2}、P(x=0,y=\text{阴性})=\frac{2}{12}=\frac{1}{6}、$$

$$P(x=1,y=\text{阳性})=0、P(x=1,y=\text{阴性})=\frac{4}{12}=\frac{1}{3}$$

- 一单给出输入数据，假定输入数据的概率为某个常数，就可以通过计算

$$\frac{P(0,\text{阳性})}{P(0)}、\frac{P(0,\text{阴性})}{P(0)}、\frac{P(1,\text{阳性})}{P(1)}、\frac{P(1,\text{阴性})}{P(1)}$$

- 将输入数据归属到所得结果最大所对应的类别。
- 这里注意，样本-标签数据的联合概率分布累加之和为1。

生成学习模型–朴素贝叶斯

生成式学习

- 朴素贝叶斯基于各特征之间相互条件独立，

$$P(\boldsymbol{x}|y=c_i)=\prod_{d=1}^{D}P(x_d|y=c_i)$$

- 可以计算出后验概率为

$$P_{post}=P(y|\boldsymbol{x})=\frac{P(y)\prod_{d=1}^{D}P(x_d|y)}{P(\boldsymbol{x})}$$

生成学习模型

判别式学习

- 输入样本的类别条件概率分布为：

$$P(y=\text{阳性}|x=0)=\frac{6}{8}=\frac{3}{4}、P(y=\text{阴性}|x=0)=\frac{2}{8}=\frac{1}{4}、$$

$$P(y=\text{阳性}|x=1)=0、P(y=\text{阴性}|x=1)=\frac{4}{4}=1$$

- 这里注意，输入样本为0或为1前提下，类别概率累加之和为1。

北京航空航天大學 软件学院

总结

一、机器学习基本概念

二、回归分析

三、线性判别分析

四、支持向量机

五、决策树

六、Ada Boosting

七、生成学习模型