南昌大学
NANCHANG UNIVERSITY

统计机器学习

主讲人：彭振华

数学与计算机学院

2026年

目录

CONTENTS

- **集成学习(ensemble learning)**通过构建并结合多个学习器来完成学习任务

- 考虑一个简单的例子，在二分类问题中，假定三个分类器在三个测试样本上的表现如下图所示，其中√表示分类正确，×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	√	×	√
集成	√	√	√

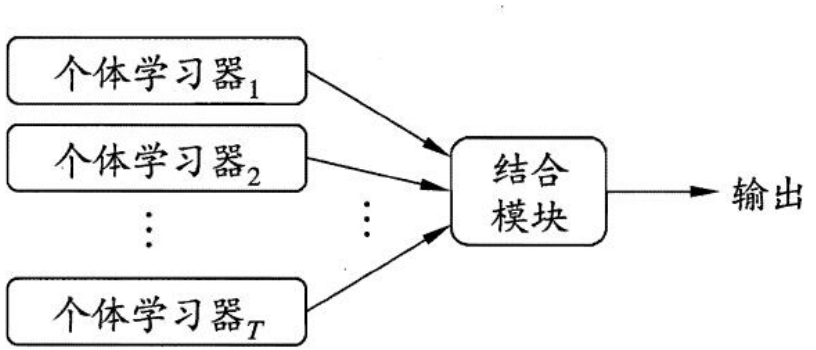
(a) 集成提升性能

	测试例1	测试例2	测试例3
h_1	√	√	×
h_2	√	√	×
h_3	√	√	×
集成	√	√	×

(b) 集成不起作用

	测试例1	测试例2	测试例3
h_1	√	×	×
h_2	×	√	×
h_3	×	×	√
集成	×	×	×

(c) 集成起负作用



集成个体应：好而不同

- 考虑二分类问题，假设基分类器的错误率为：

$$P\left(h_i\left(\boldsymbol{x}\right) \neq f\left(\boldsymbol{x}\right)\right) = \epsilon$$

- 假设集成通过简单投票法结合T个分类器，若有超过半数的基分类器正确则分类就正确

$$F(\boldsymbol{x}) = \operatorname{sign}\left(\sum_{i=1}^{T} h_i(\boldsymbol{x})\right)$$

- 假设基分类器的错误率相互独立，则由Hoeffding不等式可得集成的错误率为

$$\begin{aligned} P\left(\boldsymbol{F}(\boldsymbol{x}) \neq f(\boldsymbol{x})\right) &= \sum_{k=0}^{\lfloor T/2 \rfloor} \binom{T}{k} (1-\epsilon)^k \epsilon^{T-k} \\ &\leq \exp\left(-\frac{1}{2} T \left(1-2\epsilon\right)^2\right) \end{aligned}$$

- 上式显示，在一定条件下，随着集成分类器数目的增加，集成的错误率将指数级下降，最终趋向于0（上面的分析有一个关键假设：基学习器的误差相互独立）

集　成　学　习

- 在概率近似正确学习的框架中，一个概念（类），如果存在一个多项式的学习算法能够学习它，并且正确率很高，称这个概念是**强可学习的**；
- 一个概念（类），如果存在一个多项式的学习算法能够学习它，学习的正确率仅比随机猜测略好，则称这个概念是**弱可学习的**。
- 1989, Schapire，证明：在概率近似正确学习的框架下，一个概念是**强可学习**的充分必要条件是这个概念是**弱可学习**。
- 这意味着，只要找到一个比随机猜测略好的弱学习算法就可以直接将其提升为强学习算法，而不必直接去找很难获得的强学习算法。

集　成　学　习

- 怎样实现**弱学习转为强学习**？
- 例：学习算法A在a情况下失效，学习算法B在b情况下失效，但在a情况下可以用B算法，在b情况下可以用A算法解决。这说明通过某种合适的方式把各种算法组合起来，可以提高准确率。
- 为实现弱学习互补，**面临两个问题**：

（1）怎样获得不同的弱分类器？

（2）怎样组合弱分类器？

集　成　学　习

- 问题1：怎样**获得**不同的弱分类器？

- 使用不同的弱学习算法得到不同基本学习器
  - 参数估计、非参数估计...
- 使用相同的弱学习算法，但用不同的参数
  - K-Mean不同的K，神经网络不同的隐含层...
- 使用不同的训练集

- 问题2：怎样**组合**弱分类器？

多专家组合

- 一种并行结构，所有的弱分类器都给出各自的预测结果，通过“组合器”把这些预测结果转换为最终结果。 eg.投票（voting）及其变种、混合专家模型

多级组合

- 一种串行结构，其中下一个分类器只在前一个分类器预测不够准（不够自信）的实例上进行训练或检测。 eg. 级联算法（cascading）

$$h_1(x) \in \{-1,+1\}$$
$$h_2(x) \in \{-1,+1\}$$
$$\vdots$$
$$h_T(x) \in \{-1,+1\}$$

$$H_T(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

Weak classifiers strong classifier

slightly better than random

- 两个问题如何解决：
  - 每一轮如何改变训练数据的权值或概率分布？
  - AdaBoost：提高那些被前一轮弱分类器错误分类样本的权值，降低那些被正确分类样本的权值
  - 如何将弱分类器组合成一个强分类器？
  - AdaBoost：加权多数表决，加大分类误差率小的弱分类器的权值，使其在表决中起较大的作用，减小分类误差率大的弱分类器的权值，使其在表决中起较小的作用。

- 输入：二分类的训练数据集 $T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}$
- $x_i \in \mathcal{X} \subseteq \mathbf{R}^n \quad y_i \in \mathcal{Y}=\{-1,+1\}$
- 输出：最终分类器G(*x*)
- 1 初始化训练数据的起始权值分布

$$D_1=(w_{11},\cdots,w_{1i},\cdots,w_{1N}) \quad w_{1i}=\frac{1}{N}, \quad i=1,2,\cdots,N$$

- 2 对m个弱分类器 m=1,2，。。。M

a、在权值D_m下训练数据集，得到弱分类器

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

b、计算G_m的训练误差

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

$$\sum_{i=1}^{N}w_{mi}=1$$

c、计算G_m的系数 $\alpha_m=\frac{1}{2}\log\frac{1-e_m}{e_m}$

当$e_m\leqslant\frac{1}{2}$时，$\alpha_m\geqslant 0$

d、更新训练数据集的权值分布

$$D_{m+1}=(w_{m+1,1},\cdots,w_{m+1,i},\cdots,w_{m+1,N})\quad w_{m+1,i}=\frac{w_{mi}}{Z_m}\exp(-\alpha_m y_i G_m(x_i))$$

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

Z是规范化因子

$$Z_m=\sum_{i=1}^{N}w_{mi}\exp(-\alpha_m y_i G_m(x_i))$$

- 3、构建弱分类器的线性组合，

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

- 得到最终分类器：

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

序号	1	2	3	4	5	6	7	8	9	10
x	0	1	2	3	4	5	6	7	8	9
y	1	1	1	-1	-1	-1	1	1	1	-1

- 初始化

$$D_1 = (w_{11}, w_{12}, \cdots, w_{110})$$

$$w_{1i} = 0.1, \quad i = 1, 2, \cdots, 10$$

- 对m=1
- a、在权值分布为D_1的数据集上，阈值取2.5，分类误差率最小，基本弱分类器：

$$G_1(x) = \begin{cases} 1, & x < 2.5 \\ -1, & x > 2.5 \end{cases}$$

- b、$G_1(x)$的误差率：

$$e_1 = P(G_1(x_i) \neq y_i) = 0.3$$

- c、$G_1(x)$的系数：

$$\alpha_1 = \frac{1}{2}\log\frac{1-e_1}{e_1} = 0.4236$$

- d、更新训练数据的权值分布

$$D_2 = (w_{21}, \cdots, w_{2i}, \cdots, w_{210})$$

$$w_{2i} = \frac{w_{1i}}{Z_1} \exp(-\alpha_1 y_i G_1(x_i)), \quad i = 1, 2, \cdots, 10$$

$$D_2 = (0.0715, 0.0715, 0.0715, 0.0715, 0.0715, 0.0715, 0.1666, 0.1666, 0.1666, 0.0715)$$

$$f_1(x) = 0.4236 G_1(x)$$

- 弱基本分类器$G_1(x)=sign[f_1(x)]$在更新的数据集上有3个误分类点

- 对m=2
- a、在权值分布D_2上，阈值v=8.5时，分类误差率最低

$$G_2(x)=\begin{cases}1, & x<8.5\\ -1, & x>8.5\end{cases}$$

- b、误差率 $e_2=0.2143$.
- c、计算 $\alpha_2=0.6496$
- d、更新权值分布 $D_3=(0.0455, 0.0455, 0.0455, 0.1667, 0.1667, 0.1667, 0.1060, 0.1060, 0.1060, 0.0455)$

$$f_2(x)=0.4236G_1(x)+0.6496G_2(x)$$

- 分类器$G_2(x)=sign[f_2(x)]$有三个误分类点

- 对m=3
- a、在权值分布D_3上，阈值v=5.5时，分类误差率最低

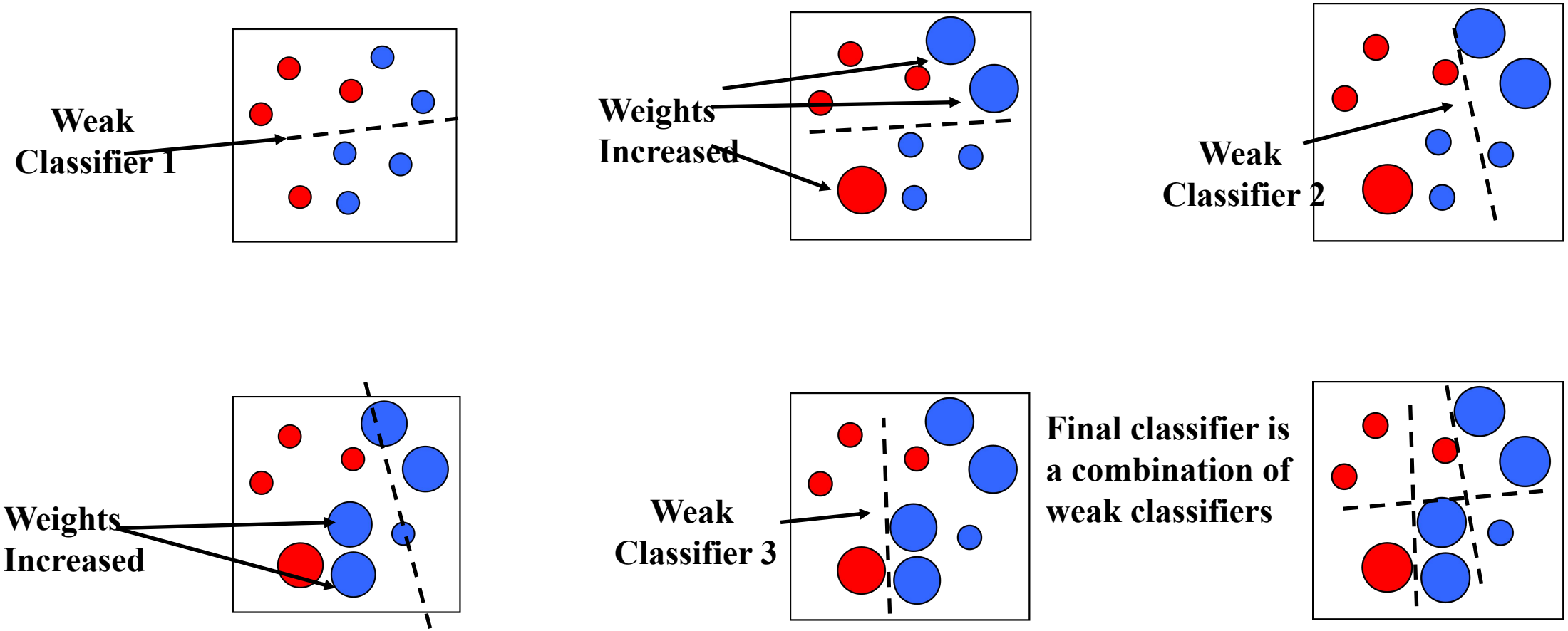
$$G_3(x)=\begin{cases}1, & x>5.5\\ -1, & x<5.5\end{cases}$$

- b、误差率 $e_3=0.1820.$
- c、计算 $\alpha_3=0.7514$
- d、更新权值分布 $D_4=(0.125,0.125,0.125,0.102,0.102,0.102,0.065,0.065,0.065,0.125)$

$$f_3(x)=0.4236G_1(x)+0.6496G_2(x)+0.7514G_3(x)$$

- 分类器$G_3(x)=sign[f_3(x)]$误分类点为0

$$G(x)=\text{sign}[f_3(x)]=\text{sign}[0.4236G_1(x)+0.6496G_2(x)+0.7514G_3(x)]$$


Weak
Classifier 1
Weights
Increased
Weak
Classifier 2
Weights
Increased
Weak
Classifier 3
Final classifier is
a combination of
weak classifiers

由：

$$Z_m = \sum_{i=1}^{N} w_{mi} \exp(-\alpha_m y_i G_m(x_i))$$

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

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

定理：AdaBoost算法最终分类器的训练误差界为：

$$\frac{1}{N} \sum_{i=1}^{N} I(G(x_i) \neq y_i) \leqslant \frac{1}{N} \sum_{i} \exp(-y_i f(x_i)) = \prod_{m} Z_m$$

$$\frac{1}{N}\sum_{i=1}^{N} I(G(x_i) \neq y_i) \leqslant \frac{1}{N}\sum_{i} \exp(-y_i f(x_i)) = \prod_{m} Z_m$$

证明： 前面部分很明显，

证后面，由

$$w_{mi}\exp(-\alpha_m y_i G_m(x_i)) = Z_m w_{m+1,i}$$

$$\begin{aligned}
&\frac{1}{N}\sum_{i}\exp(-y_i f(x_i)) \\
&= \frac{1}{N}\sum_{i}\exp\left(-\sum_{m=1}^{M}\alpha_m y_i G_m(x_i)\right) \\
&= \sum_{i} w_{1i}\prod_{m=1}^{M}\exp(-\alpha_m y_i G_m(x_i)) \\
&= Z_1\sum_{i} w_{2i}\prod_{m=2}^{M}\exp(-\alpha_m y_i G_m(x_i)) \\
&= Z_1 Z_2\sum_{i} w_{3i}\prod_{m=3}^{M}\exp(-\alpha_m y_i G_m(x_i)) \\
&= \cdots \\
&= Z_1 Z_2 \cdots Z_{M-1}\sum_{i} w_{Mi}\exp(-\alpha_M y_i G_M(x_i))
\end{aligned}$$

$$\prod_{m=1}^{M} Z_m$$

定理：二分类问题AdaBoost的训练误差界为

$$\prod_{m=1}^{M} Z_m = \prod_{m=1}^{M} [2\sqrt{e_m(1-e_m)}] = \prod_{m=1}^{M} \sqrt{(1-4\gamma_m^2)} \leqslant \exp\left(-2\sum_{m=1}^{M}\gamma_m^2\right)$$

$$\gamma_m = \frac{1}{2} - e_m$$

$$\begin{aligned} Z_m &= \sum_{i=1}^{N} w_{mi} \exp(-\alpha_m y_i G_m(x_i)) \\ &= \sum_{y_i = G_m(x_i)} w_{mi} e^{-\alpha_m} + \sum_{y_i \neq G_m(x_i)} w_{mi} e^{\alpha_m} \\ &= (1-e_m)e^{-\alpha_m} + e_m e^{\alpha_m} \\ &= 2\sqrt{e_m(1-e_m)} = \sqrt{1-4\gamma_m^2} \end{aligned}$$

后面，

由e^x和 $\sqrt{1-x}$ 在x=0的泰劳展开得：

$$\sqrt{(1-4\gamma_m^2)} \leqslant \exp(-2\gamma_m^2)$$

进而得证。

- **Bagging = Bootstrap AGGregatING**

输入： 训练集 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$;
基学习算法 $\mathfrak{L}$;
训练轮数 T.

过程：

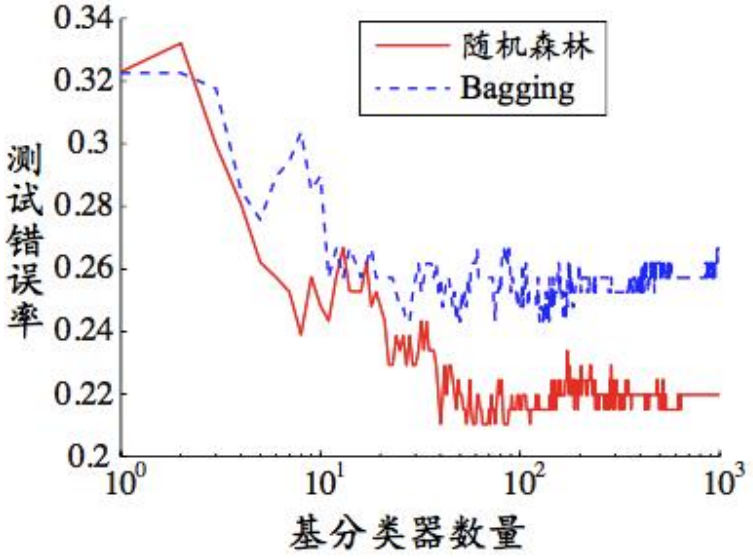
1: **for** $t = 1, 2, \ldots, T$ **do**
2: $h_t = \mathfrak{L}(D, \mathcal{D}_{bs})$
3: **end for**

输出： $H(\boldsymbol{x}) = \arg\max_{y \in \mathcal{Y}} \sum_{t=1}^{T} \mathbb{I}(h_t(\boldsymbol{x}) = y)$

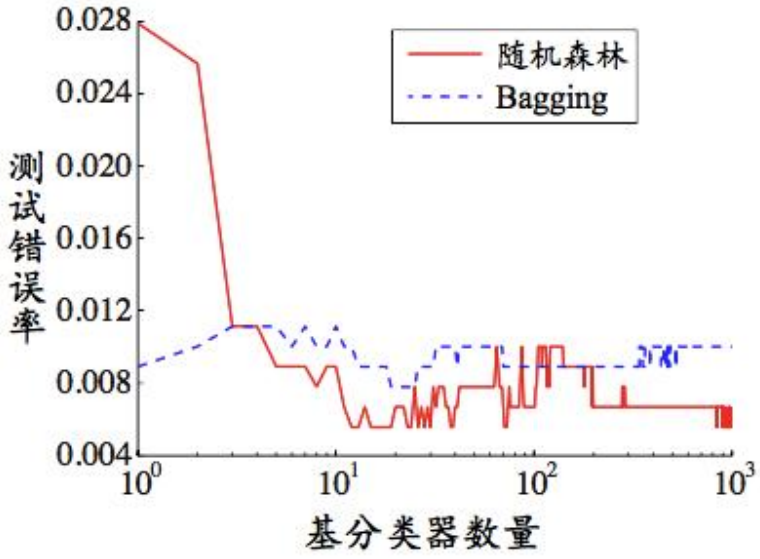
- 时间复杂度低
  - 假定基学习器的计算复杂度为O(m)，采样与投票/平均过程的复杂度为O(s)，则bagging的复杂度大致为T(O(m)+O(s))
  - 由于O(s)很小且T是一个不大的常数
  - 因此训练一个bagging集成与直接使用基学习器的复杂度同阶

- 随机森林(Random Forest，简称RF)是Bagging的一个扩展变体
- 采样的随机性
- 属性选择的随机性

对基决策树的每个结点，先从该结点的属性集合中随机选择一个包含k个属性的子集，然后再从这个子集中选择一个最优属性用于划分



(a) glass 数据集



(b) auto-mpg 数据集

- 多样性度量(diversity measure)用于度量集成中个体分类器的多样性

- 对于二分类问题，分类器h_i与h_j的预测结果列联表(contingency table)为

	$h_i = +1$	$h_i = -1$
$h_j = +1$	a	c
$h_j = -1$	b	d

$$a + b + c + d = m$$

个体学习器准确性越高、多样性越大，则集成效果越好。

常见的多样性度量

- 不合度量(Disagreement Measure)

$$dis_{ij} = \frac{b+c}{m}$$

- 相关系数(Correlation Coefficient)

$$\rho_{ij} = \frac{ad-bc}{\sqrt{(a+b)(a+c)(c+d)(b+d)}}$$

- Q-统计量(Q-Statistic)

$$Q_{ij} = \frac{ad-bc}{ad+bc} \qquad |Q_{ij}| \geq |\rho_{ij}|$$

- κ-统计量(Kappa-Statistic)

$$\kappa = \frac{p_1 - p_2}{1 - p_2}$$

$$p_1 = \frac{a+d}{m},$$

$$p_2 = \frac{(a+b)(a+c)+(c+d)(b+d)}{m^2}$$

- 提升树是以分类树或回归树为基本分类器的提升方法；提升树被认为是统计学习中性能最好的方法之一。
- 提升树模型 (boosting tree)
  - 提升方法实际采用：加法模型(即基函数的线性组合)与前向分步算法，以决策树为基函数；
  - 对分类问题决策树是二叉分类树，
  - 对回归问题决策树是二叉回归树，
    - 基本分类器x<v或x>v，可以看作是由一个根结点直接连接两个叶结点的简单决策树，即所谓的决策树桩(decision stump)。

$$f_M(x) = \sum_{m=1}^{M} T(x;\Theta_m)$$

$T(x;\Theta_m)$表示决策树； Θ_m为决策树的参数； M为树的个数

- 前向分步算法：
- 首先确定初始提升树： $f_0(x)=0$
- 第m步的模型： $f_m(x)=f_{m-1}(x)+T(x;\Theta_m)$

- 其中，$f_{m-1}(x)$为当前模型，通过经验风险极小化确定下一棵决策树的参数θ_m

$$\hat{\Theta}_m=\arg\min_{\Theta_m}\sum_{i=1}^{N}L(y_i,f_{m-1}(x_i)+T(x_i;\Theta_m))$$

- 由于树的线性组合可以很好地拟合训练数据，即使数据中的输入与输出之间的关系很复杂也是如此，所以提升树是一个高功能的学习算法。

- 回归问题提升树：
- 已知训练数据集：$T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}$，$x_i\in\mathcal{X}\subseteq\mathbf{R}^n$

$$y_i\in\mathcal{Y}\subseteq\mathbf{R}$$

- X为输入空间，y为输出空间，
- 将X划分为J个互不相交的区域R1,R2,..Rj，并且在每个区域上确定输出的常量c_j，那么，树可表示为：

$$T(x;\Theta)=\sum_{j=1}^{J}c_jI(x\in R_j)$$

$$\Theta=\{(R_1,c_1),(R_2,c_2),\cdots,(R_J,c_J)\}$$

- J是回归树的复杂度即叶结点个数

- 前向分步算法：

$$f_0(x)=0$$
$$f_m(x)=f_{m-1}(x)+T(x;\Theta_m),\quad m=1,2,\cdots,M$$
$$f_M(x)=\sum_{m=1}^{M}T(x;\Theta_m)$$

- 在前向分步算法的第m步，给定当前f_{m-1} 需求解

$$\hat{\Theta}_m=\arg\min_{\Theta_m}\sum_{i=1}^{N}L(y_i,f_{m-1}(x_i)+T(x_i;\Theta_m))$$

- 得到第m棵树的参数 $\hat{\Theta}_m$

- 采用平方损失函数时：$L(y,f(x))=(y-f(x))^2$

$$L(y,f_{m-1}(x)+T(x;\Theta_m))$$
$$=[y-f_{m-1}(x)-T(x;\Theta_m)]^2\quad=[r-T(x;\Theta_m)]^2$$

输入：训练数据集 $T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}$

$$x_i \in \mathcal{X} \subseteq \mathbf{R}^n,\ y_i \in \mathcal{Y} \subseteq \mathbf{R}$$

输出：提升树 $f_M(x)$.

（1）初始化 $f_0(x)=0$

（2）对 $m=1,2,\cdots,M$

（a）按式 (8.27) 计算残差

$$r_{mi}=y_i-f_{m-1}(x_i),\quad i=1,2,\cdots,N$$

（b）拟合残差 r_{mi} 学习一个回归树，得到 $T(x;\Theta_m)$

（c）更新 $f_m(x)=f_{m-1}(x)+T(x;\Theta_m)$

（3）得到回归问题提升树 $f_M(x)=\sum_{m=1}^{M}T(x;\Theta_m)$

- X的取值范围[0.5, 10.5],y的取值范围： [5.0,10.10],用树桩做基函数;

x_i	1	2	3	4	5	6	7	8	9	10
y_i	5.56	5.70	5.91	6.40	6.80	7.05	8.90	8.70	9.00	9.05

- 求f1(x)回归树T1(x)，

$$\min_s \left[\min_{c_1} \sum_{x_i \in R_1} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2} (y_i - c_2)^2 \right]$$

- 求切分点s： $R_1 = \{x \mid x \leqslant s\}$， $R_2 = \{x \mid x > s\}$

- 容易求得在R1,R2内部使平方损失误差达到最小值的c_1, c_2:

$$c_1 = \frac{1}{N_1} \sum_{x_i \in R_1} y_i, \quad c_2 = \frac{1}{N_2} \sum_{x_i \in R_2} y_i$$

- 各切分点:

$$1.5,\ 2.5,\ 3.5,\ 4.5,\ 5.5,\ 6.5,\ 7.5,\ 8.5,\ 9.5$$

$$m(s)=\min_{c_1}\sum_{x_i\in R_1}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2}(y_i-c_2)^2$$

当 $s=1.5$ 时，$R_1=\{1\}$，$R_2=\{2,3,\cdots,10\}$，$c_1=5.56$，$c_2=7.50$，

$$m(s)=\min_{c_1}\sum_{x_i\in R_1}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2}(y_i-c_2)^2=0+15.72=15.72$$

s	1.5	2.5	3.5	4.5	5.5	6.5	7.5	8.5	9.5
$m(s)$	15.72	12.07	8.36	5.78	3.91	1.93	8.01	11.73	15.74

- 回归树T1

$$T_1(x)=\begin{cases}6.24, & x<6.5\\ 8.91, & x\geqslant 6.5\end{cases}$$

$$f_1(x)=T_1(x)$$

$$r_{2i} = y_i - f_1(x_i)$$

x_i	1	2	3	4	5	6	7	8	9	10
r_{2i}	−0.68	−0.54	−0.33	0.16	0.56	0.81	−0.01	−0.21	0.09	0.14

- 用f1拟合数据的平方误差：$L(y, f_1(x)) = \sum_{i=1}^{10}(y_i - f_1(x_i))^2 = 1.93$
- 第二步：求T2，

$$T_2(x) = \begin{cases} -0.52, & x < 3.5 \\ 0.22, & x \geqslant 3.5 \end{cases}$$

$$f_2(x) = f_1(x) + T_2(x) = \begin{cases} 5.72, & x < 3.5 \\ 6.46, & 3.5 \leqslant x < 6.5 \\ 9.13, & x \geqslant 6.5 \end{cases}$$

$$L(y, f_2(x)) = \sum_{i=1}^{10}(y_i - f_2(x_i))^2 = 0.79$$

$$T_3(x)=\begin{cases}0.15, & x<6.5\\ -0.22, & x\geqslant 6.5\end{cases} \qquad L(y,f_3(x))=0.47$$

$$T_4(x)=\begin{cases}-0.16, & x<4.5\\ 0.11, & x\geqslant 4.5\end{cases} \qquad L(y,f_4(x))=0.30$$

$$T_5(x)=\begin{cases}0.07, & x<6.5\\ -0.11, & x\geqslant 6.5\end{cases} \qquad L(y,f_5(x))=0.23$$

$$T_6(x)=\begin{cases}-0.15, & x<2.5\\ 0.04, & x\geqslant 2.5\end{cases}$$

$$f_6(x)=f_5(x)+T_6(x)=T_1(x)+\cdots+T_5(x)+T_6(x)$$

$$=\begin{cases}5.63, & x<2.5\\ 5.82, & 2.5\leqslant x<3.5\\ 6.56, & 3.5\leqslant x<4.5\\ 6.83, & 4.5\leqslant x<6.5\\ 8.95, & x\geqslant 6.5\end{cases}$$

$$L(y,f_6(x))=\sum_{i=1}^{10}(y_i-f_6(x_i))^2=0.17$$

感 谢 观 看

统 计 机 器 学 习

主讲人：彭振华

数学与计算机学院

2026年