第八章：集成学习

主讲：连德富 特任教授 | 博士生导师

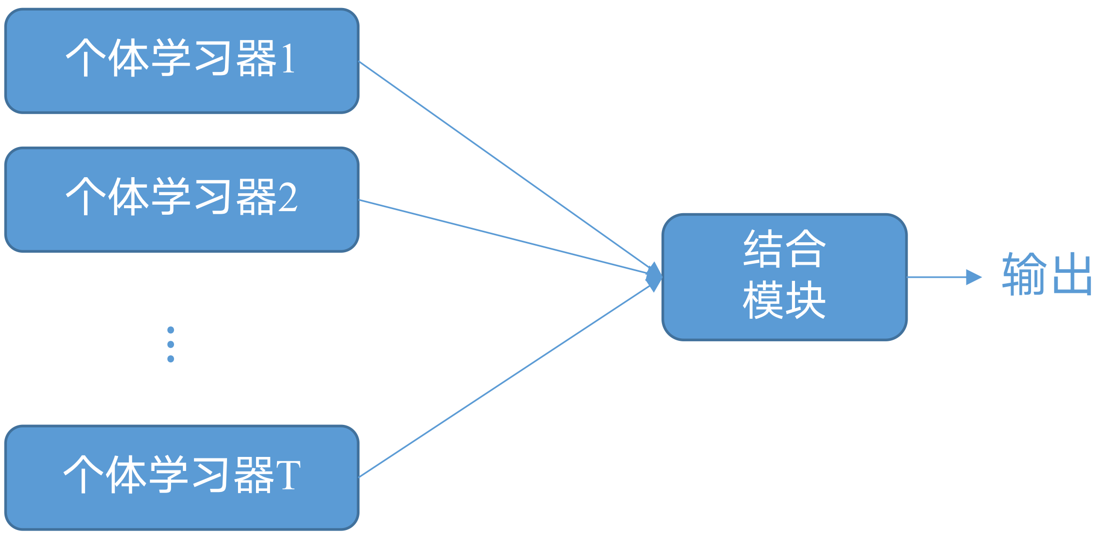
邮箱：liandefu@ustc.edu.cn

手机：13739227137

主页：http://staff.ustc.edu.cn/~liandefu

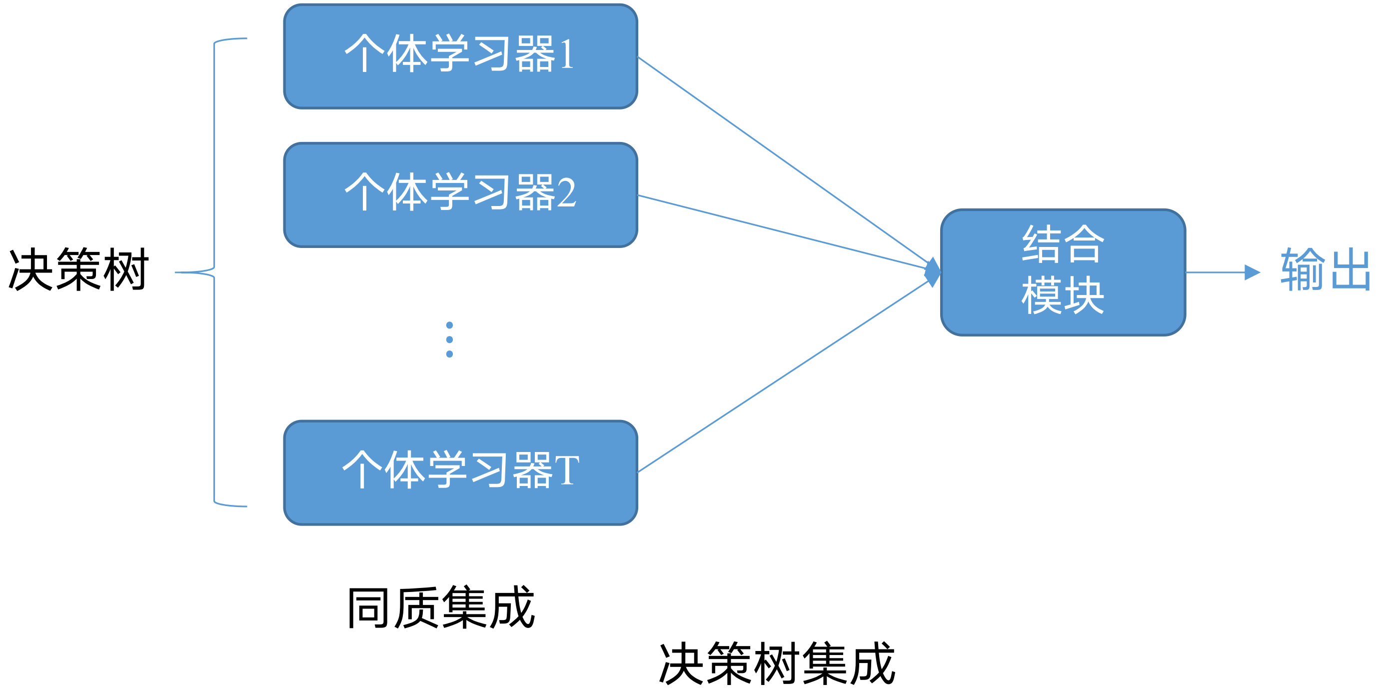
个体与集成

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



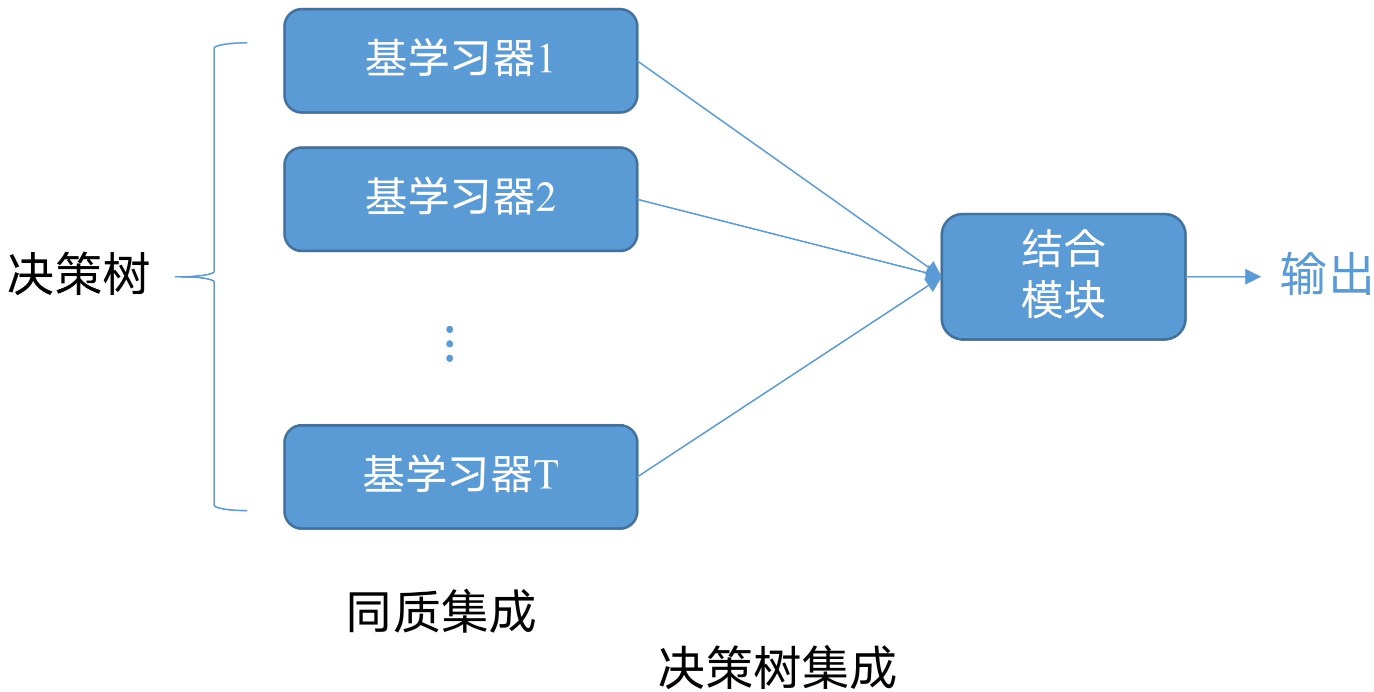
个体与集成

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



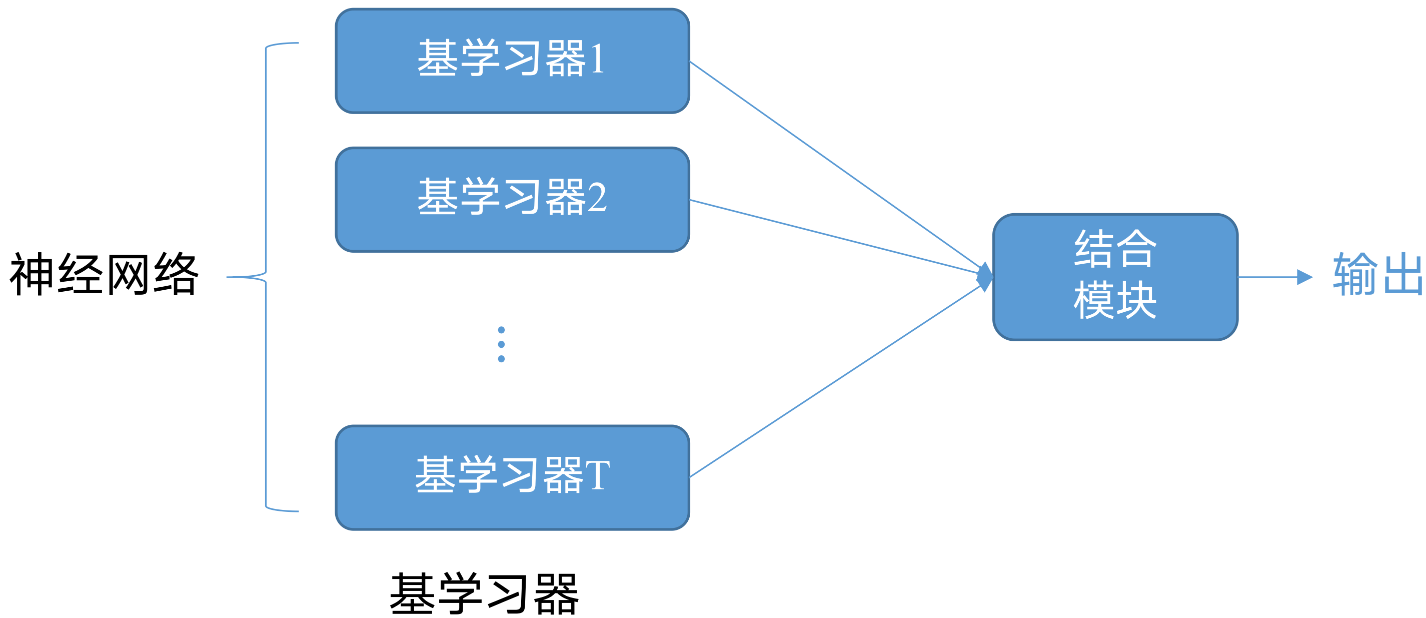
个体与集成

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



个体与集成

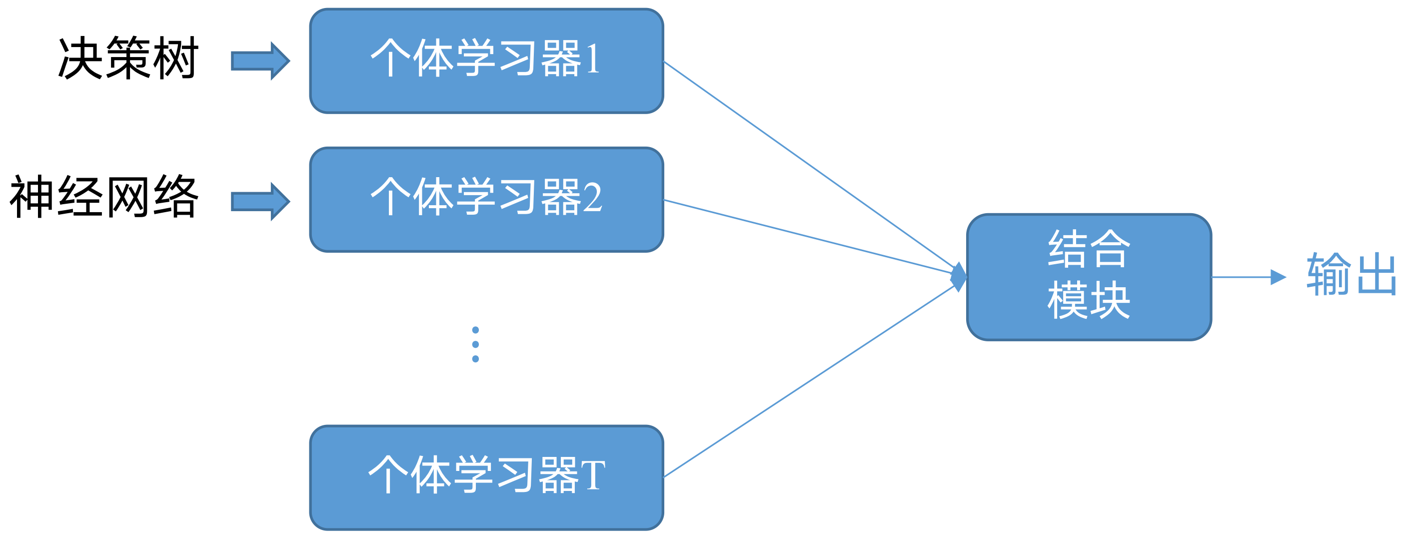
- 集成学习(ensemble learning)通过构建并结合多个学习器来提升性能



神经网络集成

个体与集成

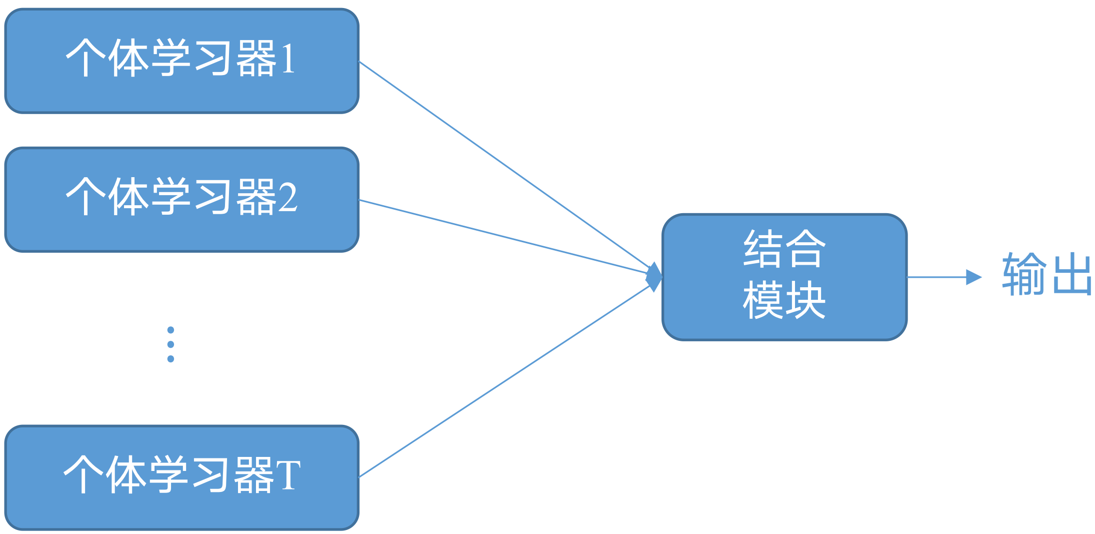
- 集成学习(ensemble learning)通过构建并结合多个学习器来提升性能



异质集成

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



集成学习获得比单一学习器显著优越的泛化性能，对弱学习器尤为明显。

集成学习很多理论研究都是针对弱学习器进行的

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，X 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	√	√	√

集成提升性能

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

集成不起作用

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

集成起负作用

- 集成个体应：好而不同

个体与集成 – 简单分析

- 考虑二分类问题，假设基分类器的错误率为：

$$P(\mathrm{h}(\boldsymbol{x}) \neq f(\boldsymbol{x})) = \epsilon$$

- 假设集成通过简单投票法结合T个分类器

$$H(\boldsymbol{x}) = \operatorname{sign}\left(\sum_{i}^{T} \mathrm{h}_i(\boldsymbol{x})\right)$$

- 假设基分类器的错误率相互独立，令$n(T)$表示T个基分类器中对$\boldsymbol{x}$预测正确的个数

若有超过半数的基分类器正确则分类就正确

$$P(H(\boldsymbol{x}) \neq f(\boldsymbol{x})) = P(n(T) \leq \lfloor T/2 \rfloor) = \sum_{k=0}^{\lfloor T/2 \rfloor} \binom{T}{k} (1-\epsilon)^k \epsilon^{T-k}$$

个体与集成 – 简单分析

- Hoeffding's inequality: $P\left(\frac{n(T)}{T} \le \mathbb{E}\left[\frac{n(T)}{T}\right] - \delta\right) \le \exp\left(-2\delta^2 T\right)$

$$P(n(T) \le \mathbb{E}[n(T)] - T\delta) \le \exp\left(-2\delta^2 T\right)$$

$$\mathbb{E}[n(T)] = T(1-\epsilon)$$

$$\xrightarrow{k = \mathbb{E}[n(T)] - T\delta} \quad P(n(T) \le k) \le \exp\left(-2\frac{(\mathbb{E}[n(T)] - k)^2}{T}\right)$$

$$\begin{aligned} P(H(\boldsymbol{x}) \ne f(\boldsymbol{x})) &= P(n(T) \le \lfloor T/2 \rfloor) \\ &\le P(n(T) \le T/2) \\ &\le \exp\left(-2\frac{(\mathbb{E}[n(T)] - T/2)^2}{T}\right) \\ &\le \exp\left(-2\frac{(T - T\epsilon - T/2)^2}{T}\right) \le \exp\left(-\frac{1}{2}T(1-2\epsilon)^2\right) \end{aligned}$$

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

个体与集成 – 简单分析

- 上面的分析有一个关键假设： 基学习器的误差相互独立
- 现实任务中，个体学习器是为解决同一个问题训练出来的，显然不可能互相独立！
- 事实上，个体学习器的“准确性”和“多样性”本身就存在冲突

如何产生“好而不同”的个体学习器是集成学习研究的核心

集成学习

集成学习大致可分为两大类

个体学习器之间存在强依赖关系，必须串行生成的序列化方法

- 代表性方法是Boosting

个体学习器不存在强依赖关系，可同时生成的并行化方法

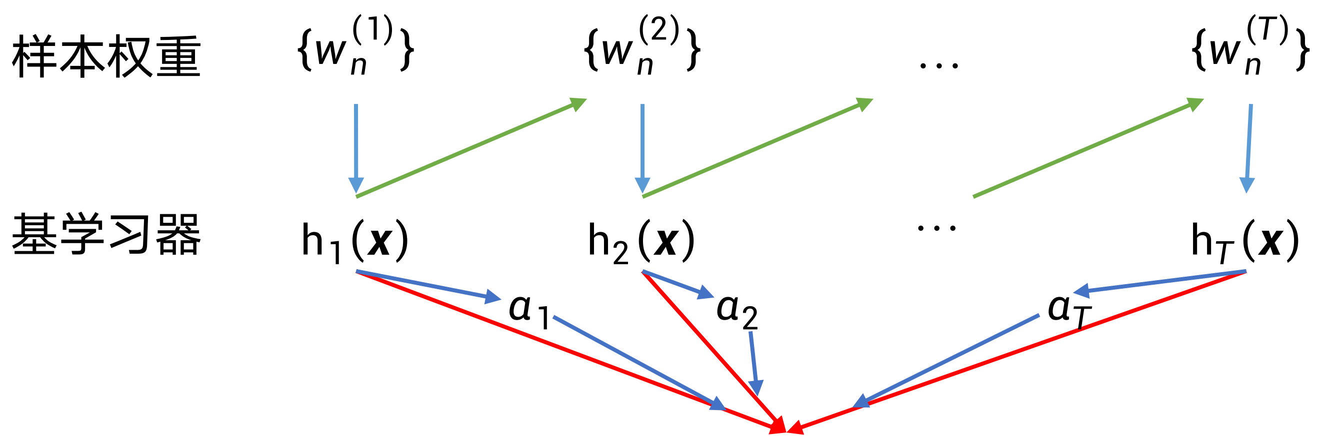
- 代表性方法是Bagging和随机森林

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

如何更新权重 w
如何学习
如何做集成 α

$$D=\{(\boldsymbol{x}_1,y_1,w_1),(\boldsymbol{x}_2,y_2,w_2)\cdots,(\boldsymbol{x}_m,y_m,w_m)\}$$



$$H(x)=\sum_{t=1}^{T}a_t\mathrm{h}_t(\boldsymbol{x})$$

AdaBoost

- 基学习器的线性组合

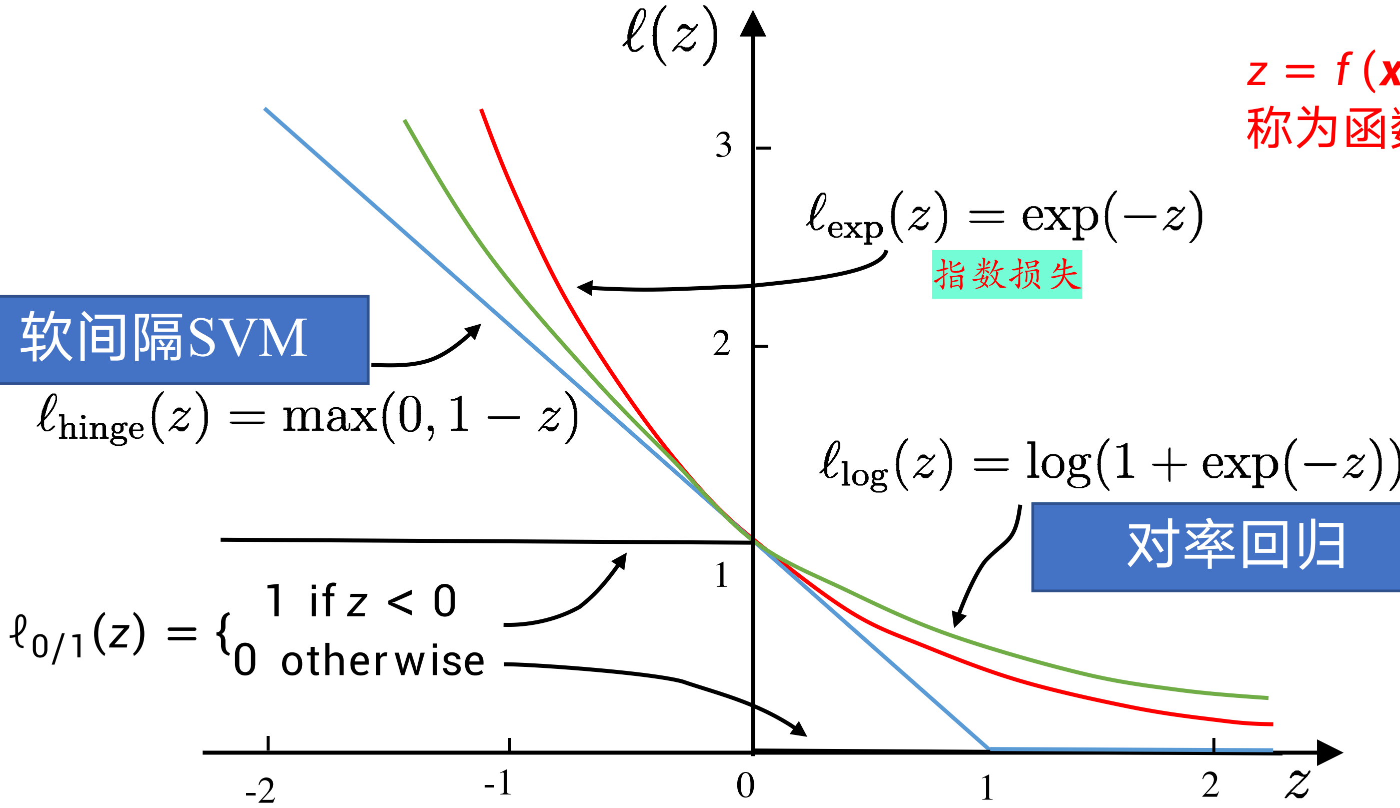
$$H(\boldsymbol{x}) = \sum_{t}^{T} a_t h_t(\boldsymbol{x})$$

- 最小化指数损失函数

$$\ell(H|D) = \mathbb{E}_{\boldsymbol{x}\sim\mathfrak{D}}[\exp\left(-f(\boldsymbol{x})H(\boldsymbol{x})\right)]$$

$z = f(\boldsymbol{x})H(\boldsymbol{x})$

称为函数间隔



指数损失函数的一致性

带有噪声

- 假设$f(\boldsymbol{x})$具不确定性，即$y=f(\boldsymbol{x})$随机变量，则损失写成

$$\begin{aligned}\ell(H|D) &= \mathbb{E}_{\boldsymbol{x},y}[\exp(-yH(\boldsymbol{x}))] \\ &= \mathbb{E}_{\boldsymbol{x},y}[\mathbb{I}(y=1)\exp(-H(\boldsymbol{x}))+\mathbb{I}(y=-1)\exp(H(\boldsymbol{x}))] \\ &= \mathbb{E}_{\boldsymbol{x}}[\mathbb{E}_{y|\boldsymbol{x}}\mathbb{I}(y=1)\exp(-H(\boldsymbol{x}))+\mathbb{E}_{y}\mathbb{I}(y=-1)\exp(H(\boldsymbol{x}))] \\ &= \mathbb{E}_{\boldsymbol{x}}[P(y=1|\boldsymbol{x})\exp(-H(\boldsymbol{x}))+P(y=-1|\boldsymbol{x})\exp(H(\boldsymbol{x}))]\end{aligned}$$

H(x)是基学习器的线性组合

- 若$H(\boldsymbol{x})$能令指数损失函数最小化，则上式对$H(\boldsymbol{x})$的偏导值为0

$$\frac{\partial\ell(H|D)}{\partial H(\boldsymbol{x})} = P(\boldsymbol{x})(-P(y=1|\boldsymbol{x})\exp(-H(\boldsymbol{x}))+P(y=-1|\boldsymbol{x})\exp(H(\boldsymbol{x}))) = 0$$

$$\Rightarrow\quad P(y=1|\boldsymbol{x})\exp(-H(\boldsymbol{x})) = P(y=-1|\boldsymbol{x})\exp(H(\boldsymbol{x}))$$

$$\Rightarrow\quad H(\boldsymbol{x}) = \frac{1}{2}\ln\frac{P(y=1|\boldsymbol{x})}{P(y=-1|\boldsymbol{x})}$$

指数损失函数的一致性

$$H(\boldsymbol{x}) = \frac{1}{2}\ln\frac{P(y=1|\boldsymbol{x})}{P(y=-1|\boldsymbol{x})}$$

$\Rightarrow$

$$P(y=1|\boldsymbol{x}) = \frac{1}{1+\exp{-2\cdot H(\boldsymbol{x})}}$$

$$\operatorname{sign}(H(\boldsymbol{x})) = \begin{cases} +1, \text{if } P(y=1|\boldsymbol{x}) > P(y=-1|\boldsymbol{x}) \\ -1, \text{if } P(y=1|\boldsymbol{x}) < P(y=-1|\boldsymbol{x}) \end{cases}$$

$\Rightarrow$

$$\operatorname{sign}(H(\boldsymbol{x})) = \arg\max_{y\in\{\pm 1\}} P(y|\boldsymbol{x})$$

$\operatorname{sign}(H(\boldsymbol{x}))$达到了贝叶斯最优错误率

说明指数损失函数是分类任务原来0/1损失函数的一致的替代函数

AdaBoost

- 假设$f(\boldsymbol{x})$是确定性的，在第t步，优化以下损失函数

$$\ell(H_{t-1} + a_t \mathrm{h}_t | D) = \mathbb{E}_{\boldsymbol{x}}\left[\exp\left(-f(\boldsymbol{x})\left(H_{t-1}(\boldsymbol{x}) + a_t \mathrm{h}_t(\boldsymbol{x})\right)\right)\right]$$

$$= \mathbb{E}_{\boldsymbol{x}}\left[\underbrace{\exp\left(-f(\boldsymbol{x}) H_{t-1}(\boldsymbol{x})\right)}_{\bar{w}_t(\boldsymbol{x})} \exp\left(-f(\boldsymbol{x}) a_t \mathrm{h}_t(\boldsymbol{x})\right)\right]$$

$$= \mathbb{E}_{\boldsymbol{x}}\left[\bar{w}_t(\boldsymbol{x}) \exp\left(-f(\boldsymbol{x}) a_t \mathrm{h}_t(\boldsymbol{x})\right)\right]$$

$$D = \{(\boldsymbol{x}_1, y_1, w_1), \cdots, (\boldsymbol{x}_m, y_m, w_m)\}$$

$$= \mathbb{E}_{\boldsymbol{x}}\left[\bar{w}_t(\boldsymbol{x}) \exp(-a_t)\, \mathbb{I}(f(\boldsymbol{x}) = \mathrm{h}_t(\boldsymbol{x})) + \bar{w}_t(\boldsymbol{x}) \exp(a_t)\, \mathbb{I}(f(\boldsymbol{x}) \neq \mathrm{h}_t(\boldsymbol{x}))\right]$$

$$= \exp(-a_t)\, \mathbb{E}_{\boldsymbol{x}}\left[\bar{w}_t(\boldsymbol{x}) \mathbb{I}(f(\boldsymbol{x}) = \mathrm{h}_t(\boldsymbol{x}))\right] + \exp(a_t)\, \mathbb{E}_{\boldsymbol{x}}\left[\bar{w}_t(\boldsymbol{x}) \mathbb{I}(f(\boldsymbol{x}) \neq \mathrm{h}_t(\boldsymbol{x}))\right]$$

样本权重 $\{w_n^{(t)}\}$

基学习器 $\mathrm{h}_t(\boldsymbol{x})$

a_t

$$H_t(x) = a_t \mathrm{h}_t(\boldsymbol{x}) + H_{t-1}(x)$$

AdaBoost

- 假设$f(\boldsymbol{x})$是确定性的，在第t步，优化以下损失函数（求解h_t）

$$\ell(H_{t-1}+\alpha_t \mathrm{h}_t|D)=\exp(-\alpha_t)\,\mathbb{E}_{\boldsymbol{x}}[\bar{w}_t(\boldsymbol{x})\boxed{\mathbb{I}(f(\boldsymbol{x})=\mathrm{h}_t(\boldsymbol{x}))]}+\exp(\alpha_t)\,\mathbb{E}_{\boldsymbol{x}}[\bar{w}_t(\boldsymbol{x})\mathbb{I}(f(\boldsymbol{x})\neq \mathrm{h}_t(\boldsymbol{x}))]$$

$$\min_{\mathrm{h}_t}\ell(H_{t-1}+\alpha_t \mathrm{h}_t|D)=\exp(-\alpha_t)\,\mathbb{E}_{\boldsymbol{x}}[\bar{w}_t(\boldsymbol{x})\boxed{(1-\mathbb{I}(f(\boldsymbol{x})\neq \mathrm{h}_t(\boldsymbol{x})))]}$$
$$+\exp(\alpha_t)\,\mathbb{E}_{\boldsymbol{x}}[\bar{w}_t(\boldsymbol{x})\mathbb{I}(f(\boldsymbol{x})\neq \mathrm{h}_t(\boldsymbol{x}))]$$

$$D=\{(\boldsymbol{x}_1,y_1,w_1),\cdots,(\boldsymbol{x}_m,y_m,w_m)\}$$

$$=\boxed{(\exp(\alpha_t)-\exp(-\alpha_t))}\mathbb{E}_{\boldsymbol{x}}[\bar{w}_t(\boldsymbol{x})\mathbb{I}(f(\boldsymbol{x})\neq \mathrm{h}_t(\boldsymbol{x}))]$$
$$+\exp(-\alpha_t)\,\mathbb{E}_{\boldsymbol{x}}[\bar{w}_t(\boldsymbol{x})]$$

样本权重 $\{w_n^{(t)}\}$

当$\alpha_t>0$，$\exp(\alpha_t)-\exp(-\alpha_t)>0$

基学习器 $\mathrm{h}_t(\boldsymbol{x})$

α_t

$$\min_{\mathrm{h}_t}\ell(H_{t-1}+\alpha_t \mathrm{h}_t|D)\text{ 等价于 }\min_{\mathrm{h}_t}\mathbb{E}_{\boldsymbol{x}}[\bar{w}_t(\boldsymbol{x})\mathbb{I}(f(\boldsymbol{x})\neq \mathrm{h}_t(\boldsymbol{x}))]$$

$$\bar{w}_t(\boldsymbol{x})=\exp(-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x}))$$

$$H_t(x)=\alpha_t \mathrm{h}_t(\boldsymbol{x})+H_{t-1}(x)$$

AdaBoost

$$\mathrm{h}_t^\star(\boldsymbol{x}) = \underset{\mathrm{h}_t}{\operatorname{argmin}}\, \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\bar{w}_t(\boldsymbol{x})\mathbb{I}(f(\boldsymbol{x}) \neq \mathrm{h}_t(\boldsymbol{x}))]$$

$$= \underset{\mathrm{h}_t}{\operatorname{argmin}}\, \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\frac{\bar{w}_t(\boldsymbol{x})}{\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\bar{w}_t(\boldsymbol{x})]}\mathbb{I}(f(\boldsymbol{x}) \neq \mathrm{h}_t(\boldsymbol{x}))\right] = \underset{\mathrm{h}_t}{\operatorname{argmin}}\, \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_t}[\mathbb{I}(f(\boldsymbol{x}) \neq \mathrm{h}_t(\boldsymbol{x}))]$$

$$\mathcal{D}_t(\boldsymbol{x}) = \mathcal{D}(\boldsymbol{x})\frac{\bar{w}_t(\boldsymbol{x})}{\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\bar{w}_t(\boldsymbol{x})]} = \mathcal{D}(\boldsymbol{x})\frac{\exp(-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x}))}{\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\exp(-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x}))]}$$

$$= \mathcal{D}(\boldsymbol{x})\frac{\exp(-f(\boldsymbol{x})H_{t-2}(\boldsymbol{x}))\exp(-\alpha_{t-1}f(\boldsymbol{x})\mathrm{h}_{t-1}(\boldsymbol{x}))}{\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\exp(-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x}))]}$$

$$= \mathcal{D}_{t-1}(\boldsymbol{x})\exp(-\alpha_{t-1}f(\boldsymbol{x})\mathrm{h}_{t-1}(\boldsymbol{x}))\frac{\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\exp(-f(\boldsymbol{x})H_{t-2}(\boldsymbol{x}))]}{\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\exp(-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x}))]}$$

$$= \frac{\mathcal{D}_{t-1}(\boldsymbol{x})\exp(-\alpha_{t-1}f(\boldsymbol{x})\mathrm{h}_{t-1}(\boldsymbol{x}))}{Z_{t-1}}$$

AdaBoost

$$h_t^\star(\boldsymbol{x}) = \underset{h_t}{\operatorname{argmin}}\, \mathbb{E}_{\boldsymbol{x}}[\bar{w}_t(\boldsymbol{x})\mathbb{I}(f(\boldsymbol{x}) \neq h_t(\boldsymbol{x}))]$$

- 假设$f(\boldsymbol{x})$是确定性的，在第t步，优化以下损失函数（求解a_t）

$$\begin{aligned}\ell(H_{t-1} + a_t h_t^\star | D) &= (\exp(a_t) - \exp(-a_t))\mathbb{E}_{\boldsymbol{x}}[\bar{w}_t(\boldsymbol{x})\mathbb{I}(f(\boldsymbol{x}) \neq h_t^\star(\boldsymbol{x}))] \\ &\quad + \exp(-a_t)\,\mathbb{E}_{\boldsymbol{x}}[\bar{w}_t(\boldsymbol{x})] \\ &= \mathbb{E}_{\boldsymbol{x}}[\bar{w}_t(\boldsymbol{x})]((\exp(a_t) - \exp(-a_t))\epsilon_t + \exp(-a_t))\end{aligned}$$

$$\mathbb{E}_{\boldsymbol{x}}\left[\frac{\bar{w}_t(\boldsymbol{x})}{\mathbb{E}_{\boldsymbol{x}}[\bar{w}_t(\boldsymbol{x})]}\mathbb{I}(f(\boldsymbol{x}) \neq h_t^\star(\boldsymbol{x}))\right] = \epsilon_t$$

$$D = \{(\boldsymbol{x}_1, y_1, w_1), \cdots, (\boldsymbol{x}_m, y_m, w_m)\}$$

样本权重 $\{w_n^{(t)}\}$

$\downarrow$

基学习器 $h_t(\boldsymbol{x})$

$\downarrow$

a_t

$$H_t(x) = a_t h_t(\boldsymbol{x}) + H_{t-1}(x)$$

$$\frac{\partial \ell(H_{t-1} + a_t h_t^\star | D)}{\partial a_t} = 0$$

$$\Rightarrow (\exp(a_t) + \exp(-a_t))\epsilon_t - \exp(-a_t) = 0$$

$$\Rightarrow (\exp(2a_t) + 1)\epsilon_t - 1 = 0$$

$$\Rightarrow \exp(2a_t) = \frac{1-\epsilon_t}{\epsilon_t} \Rightarrow a_t = \frac{1}{2}\ln\frac{1-\epsilon_t}{\epsilon_t}$$

AdaBoost

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: $\mathcal{D}_1(\boldsymbol{x}) = 1/m$.
2: **for** $t = 1, 2, \ldots, T$ **do**
3: $h_t = \mathfrak{L}(D, \mathcal{D}_t)$;
4: $\epsilon_t = P_{\boldsymbol{x} \sim \mathcal{D}_t}(h_t(\boldsymbol{x}) \neq f(\boldsymbol{x}))$;
5: **if** $\epsilon_t > 0.5$ **then break**
6: $\alpha_t = \frac{1}{2} \ln \left(\frac{1-\epsilon_t}{\epsilon_t} \right)$;
7: $$\mathcal{D}_{t+1}(\boldsymbol{x}) = \frac{\mathcal{D}_t(\boldsymbol{x})}{Z_t} \times \begin{cases} \exp(-\alpha_t), & \text{if } h_t(\boldsymbol{x}) = f(\boldsymbol{x}) \\ \exp(\alpha_t), & \text{if } h_t(\boldsymbol{x}) \neq f(\boldsymbol{x}) \end{cases} = \frac{\mathcal{D}_t(\boldsymbol{x})\exp(-\alpha_t f(\boldsymbol{x}) h_t(\boldsymbol{x}))}{Z_t}$$
8: **end for**

输出: $H(\boldsymbol{x}) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x}) \right)$

AdaBoost

- Boosting算法要求基学习器能对特定的数据分布进行学习

重赋权法：在每一轮训练，根据样本分布为每个样本赋一个权重

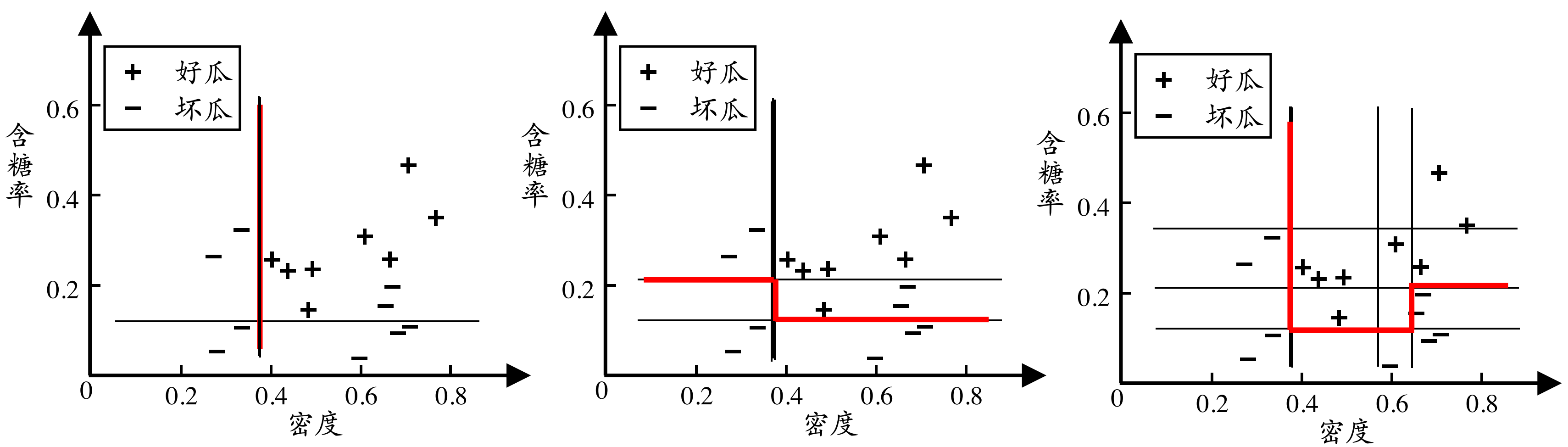
- 检查满足泛化误差大于0.5，条件不满足，就会被放弃，且学习会停止

重采样法：在每一轮学习，根据样本分布对训练集进行重采样，再用重采样而得的样本集对基学习器进行训练

- 获得重启动机会一避免训练过程过早停止：在抛弃不满足条件的当前基学习器之后，可根据当前分布重新对训练样本进行采样，再基于新的采样结果重新训练处基学习器。

AdaBoost

- 从偏差-方差的角度：降低偏差，可对泛化性能相当弱的学习器构造出很强的集成

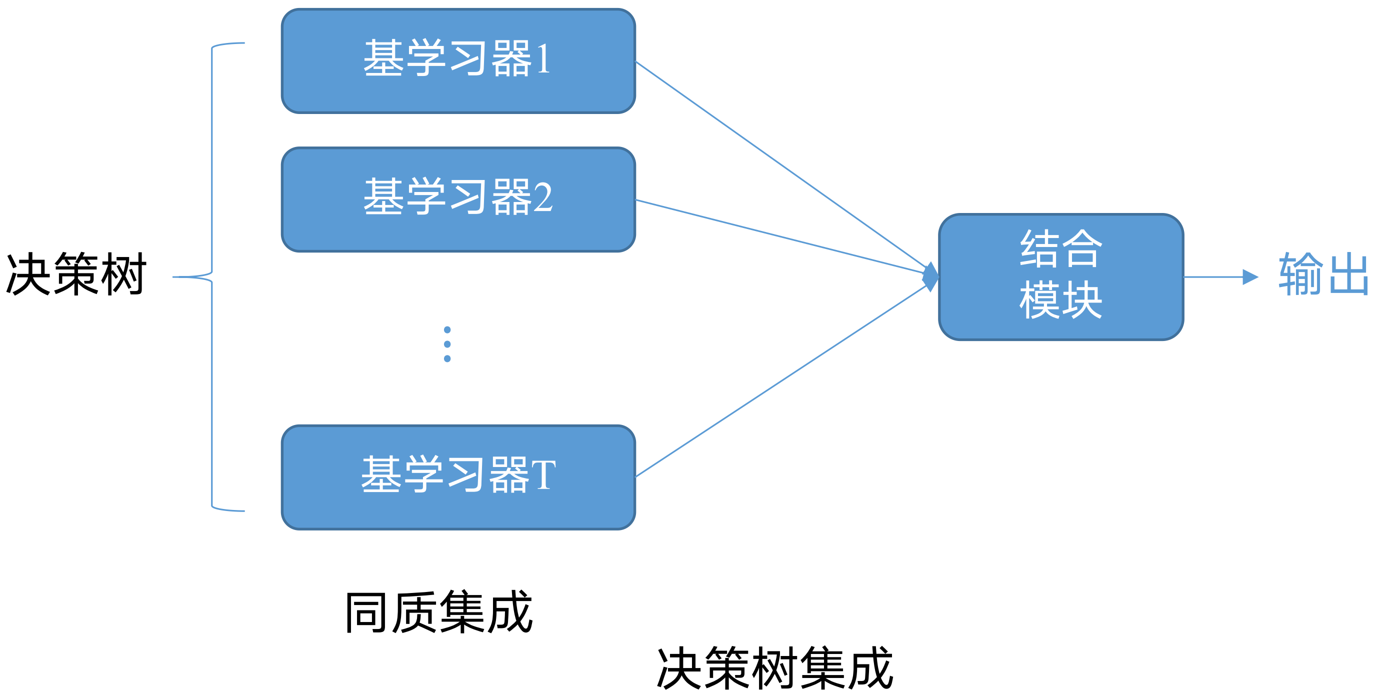


(a) 3个基学习器 (b) 5个基学习器 (c) 11个基学习器

集成模型（红色）和基学习器（黑色）的分类边界

梯度提升树

- 基于决策树的Boosting集成



梯度提升树

Adaboost

$$H(\boldsymbol{x}) = \sum_{t}^{T} a_t \mathrm{h}_t(\boldsymbol{x})$$

先训练学习器再学习权重

$$\min_{H(\boldsymbol{x})} \mathbb{E}_{\boldsymbol{x},y}[\exp(-yH(\boldsymbol{x}))]$$

适用于分类

Gradient Boosting

$$H(\boldsymbol{x}) = \sum_{t}^{T} \mathrm{h}_t(\boldsymbol{x})$$

同时学习权重和学习器

$$\min_{H(\boldsymbol{x})} \mathbb{E}_{\boldsymbol{x},y}[\ell(y, H(\boldsymbol{x}))]$$

适用于回归和分类

梯度提升树

- 与Adaboost一样，通过前向分布法优化 $\min\limits_{H(\boldsymbol{x})} \mathbb{E}_{\boldsymbol{x},y}[\ell(y, H(\boldsymbol{x}))]$

将$H(\boldsymbol{x})$看过一个参数（泛函），$H(\boldsymbol{x})$的学习过程看成是一个梯度下降过程

$$H_t(\boldsymbol{x}) = H_{t-1}(\boldsymbol{x}) + \mathrm{h}_t(\boldsymbol{x})$$

负梯度

$$\mathrm{h}_t(\boldsymbol{x}) \approx -\frac{d\ell(y,\hat{y})}{d\hat{y}}|_{\hat{y}=H_{t-1}(\boldsymbol{x})} = -\ell'(y, H_{t-1}(\boldsymbol{x}))$$

$$\ell(y, H(\boldsymbol{x})) = (y - H(\boldsymbol{x}))^2 \qquad -\ell'(y, H_{t-1}(\boldsymbol{x})) = y - H_{t-1}(\boldsymbol{x})$$

残差

梯度提升树

- 梯度提升树的算法流程

(1) 初始化 $\mathrm{h}_0(\boldsymbol{x}) = \mathrm{argmin}_{\gamma} \sum_{i=1}^{m} \ell(y_i, \gamma)$

(2) 对 t=1 to T

(a) 计算负梯度:$\tilde{y}_i = -\ell'(y_i, H_{t-1}(\boldsymbol{x}_i))$， $i \in \{1,2,\ldots m\}$

(b) 通过最小化平方误差，用基学习器$\mathrm{h}_t(\boldsymbol{x})$根据$\boldsymbol{x}_i$拟合$\tilde{y}_i$

(c) 使用线搜索确定步长ρ_t，$\rho_t = \mathrm{argmin}_{\rho_t} \sum_{i=1}^{m} \ell(y_i, H_{t-1}(\boldsymbol{x}) + \rho_t \mathrm{h}_t(\boldsymbol{x}))$

(d) 更新$H_t(\boldsymbol{x}) = H_{t-1}(\boldsymbol{x}) + \rho_t \mathrm{h}_t(\boldsymbol{x})$

(3) 输出$H_T(\boldsymbol{x})$

梯度提升树

- 回忆：回归树将空间划分为M个区域$R_1, R_2, \cdots, R_M$，那么回归树的决策函数为

$$\mathrm{h}(\boldsymbol{x}) = \sum_{c=1}^{C} w_c I(\boldsymbol{x} \in R_c)$$

- 如果优化平方损失，在给定区域划分情况下，最优的c_m值是对应区域内的平均值

$$f_i = \ell^{'}(y_i, H_{t-1}(\boldsymbol{x}_i)) \qquad s_i = \ell^{''}(y_i, H_{t-1}(\boldsymbol{x}_i))$$

- 提升树每次优化

$$J^{(t)}(\mathrm{h}_t(\boldsymbol{x}_i)) = \sum_{i=1}^{m} \ell(y_i, H_{t-1}(\boldsymbol{x}_i) + \mathrm{h}_t(\boldsymbol{x}_i)) + \Omega(\mathrm{h}_t)$$

$$\simeq \sum_{i=1}^{m} [\ell(y_i, H_{t-1}(\boldsymbol{x}_i)) + f_i \mathrm{h}_t(\boldsymbol{x}_i) + \frac{1}{2} s_i \mathrm{h}_t^2(\boldsymbol{x}_i)] + \Omega(\mathrm{h}_t)$$

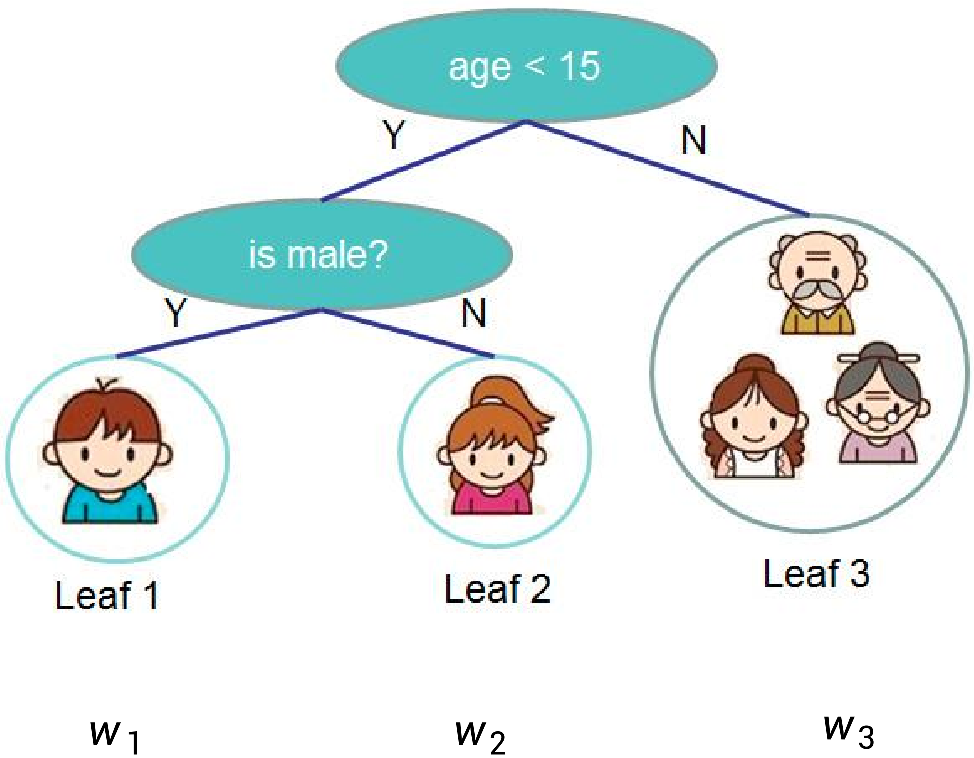
$$= \sum_{i=1}^{m} [f_i \mathrm{h}_t(\boldsymbol{x}_i) + \frac{1}{2} s_i \mathrm{h}_t^2(\boldsymbol{x}_i)] + \Omega(\mathrm{h}_t) + \mathrm{const}$$

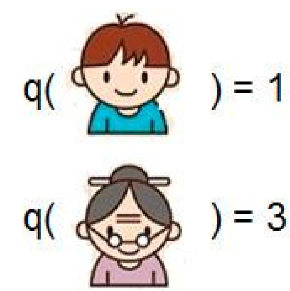
梯度提升树

$$h(\boldsymbol{x}) = \sum_{c=1}^{C} w_c I(\boldsymbol{x} \in R_c)$$

树的复杂度可以定义为

$$\Omega(h) = \gamma C + \frac{1}{2}\lambda \sum_{c=1}^{C} w_c^2$$





梯度提升树

$$\mathrm{h}(\boldsymbol{x}) = \sum_{c=1}^{C} w_c I(\boldsymbol{x} \in R_c)$$

$$J^{(t)}(\mathrm{h}_t(\boldsymbol{x}_i)) \simeq \sum_{i=1}^{m} [f_i \mathrm{h}_t(\boldsymbol{x}_i) + \frac{1}{2} s_i \mathrm{h}_t^2(\boldsymbol{x}_i)] + \Omega(\mathrm{h}_t) + \mathrm{const}$$

$$\sum_{i=1}^{m} f_i \sum_{c=1}^{C} w_c I(\boldsymbol{x}_i \in R_c) \qquad = \sum_{i=1}^{m} [f_i \mathrm{h}_t(\boldsymbol{x}_i) + \frac{1}{2} s_i \mathrm{h}_t^2(\boldsymbol{x}_i)] + \gamma C + \frac{1}{2}\lambda \sum_{c=1}^{C} w_c^2 + \mathrm{const}$$

$$\sum_{c=1}^{C} w_c \sum_{i=1}^{m} f_i I(\boldsymbol{x}_i \in R_c) \qquad = \sum_{c=1}^{C} [(\sum_{i \in R_c} f_i) w_c + \frac{1}{2} (\sum_{i \in R_c} s_i + \lambda) w_c^2] + \gamma C + \mathrm{const}$$

$$= \sum_{c=1}^{C} [F_c w_c + \frac{1}{2} (S_c + \lambda) w_c^2] + \gamma C + \mathrm{const}$$

$$J^{(t)}(\mathrm{h}_t(\boldsymbol{x}_i)) = -\frac{1}{2} \sum_{c=1}^{C} \frac{F_c^2}{S_c + \lambda} + \gamma C$$

给定在给定区域划分情况下

$$w_c^\star = -\frac{F_c}{S_c + \lambda}$$

该树与训练数据的适合度度量，可用于指导分类树的构建

梯度提升树

属性a的阈值 t

叶子c的样本

f_1, s_1 f_4, s_4 f_2, s_2 f_5, s_5 f_3, s_3

$F_L = f_1 + f_4$ $F_R = f_2 + f_3 + f_5$

$\frac{(F_L + F_R)^2}{S_L + S_r + \lambda}$

- 划分点选择准则
- 划分前： $-\frac{1}{2}\sum_{c=1,c\neq\bar{c}}^{C}\frac{F_c^2}{S_c+\lambda} - \frac{1}{2}\boxed{\frac{F_{\bar{c}}^2}{S_{\bar{c}}+\lambda}} + \gamma C$（假设划分叶子为$\bar{c}$）
- 划分后： $-\frac{1}{2}\sum_{c=1,c\neq\bar{c}}^{C}\frac{F_c^2}{S_c+\lambda} - \frac{1}{2}\frac{F_L^2}{S_L+\lambda} - \frac{1}{2}\frac{F_R^2}{S_R+\lambda} + \gamma(C+1)$

划分为左子树L和右子树R

$$\text{gain}(D, a, t) = \frac{1}{2}\left(\frac{F_L^2}{S_L + \lambda} + \frac{F_R^2}{S_R + \lambda} - \frac{(F_L + F_R)^2}{S_L + S_r + \lambda}\right) - \gamma$$

梯度提升树

- GBDT的知名算法库

- XGBoost

dmlc XGBoost eXtreme Gradient Boosting

Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).

https://github.com/dmlc/xgboost

- LightGBM

LightGBM

Light Gradient Boosting Machine

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in neural information processing systems* (pp. 3146-3154).

https://github.com/microsoft/LightGBM

Bagging

- 个体学习器不存在强依赖关系、并行化生成

- 基学习器尽可能具有较大的差异

对训练样本进行采样，产生出不同的子集，再从每个子集中训练出一个基学习器

若采样出完全不同的子集，每个基学习器只用到一小部分训练数据，可能不足以进行有效学习，无法确保基学习器的性能

Bagging

- 自助采样法

对数据集 D 有放回采样 m 次得到训练集D'

样本在m次采样中始终不被采样到的概率 $= \lim_{m\to\infty}(1-\frac{1}{m})^m = \frac{1}{e} \approx 0.368$

初始训练集中约有63.2%样本出现在采样集中

Bagging

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: $\quad h_t = \mathfrak{L}(D, \mathcal{D}_{bs})$
3: **end for**

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

对分类任务使用简单投票法　**结合策略**　对回归任务使用简单平均法

Bagging

- 时间复杂度低
  - 假定个体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))
  - O(s)很小且T是一个不大的常数
  - 训练一个bagging集成与直接使用基学习器的复杂度同阶

Bagging

- 由于个体学习器并未使用全部训练数据，因此其余数据可用作验证集对泛化性能进行包外估计（out-of-bag estimate）

- $H^{oob}(\boldsymbol{x})$表示对样本$\boldsymbol{x}$的包外预测，即仅考虑那些未使用样本$\boldsymbol{x}$训练的基学习器在$\boldsymbol{x}$上的预测

$$H^{oob}(\boldsymbol{x}) = \mathrm{argmax}_{y\in\mathcal{Y}} \sum_{i}^{T} \mathbb{I}(\mathrm{h}_t(\boldsymbol{x}) = y) \cdot \mathbb{I}(\boldsymbol{x} \notin D_t)$$

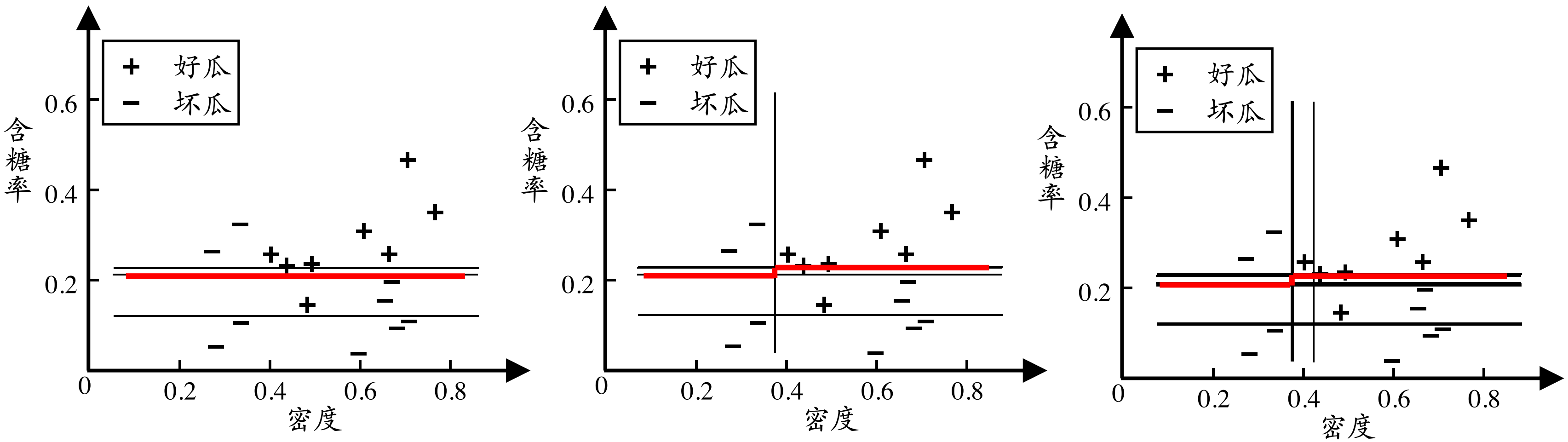
- Bagging泛化误差的包外估计为：

$$\epsilon^{oob} = \frac{1}{|D|} \sum_{(x,y)\in D} \mathbb{I}(H^{oob}(\boldsymbol{x}) = y)$$

- 包外样本还可以用于决策树剪枝、神经网络早停等

Bagging

从偏差-方差的角度：降低方差，在不剪枝的决策树、神经网络等易受样本影响的学习器上效果更好



(a) 3个基学习器　(b) 5个基学习器　(c) 11个基学习器

随机森林

- 随机森林(Random Forest，简称RF)是bagging的一个扩展变种
- 在采样的随机性基础上，进一步引入了属性选择的随机性

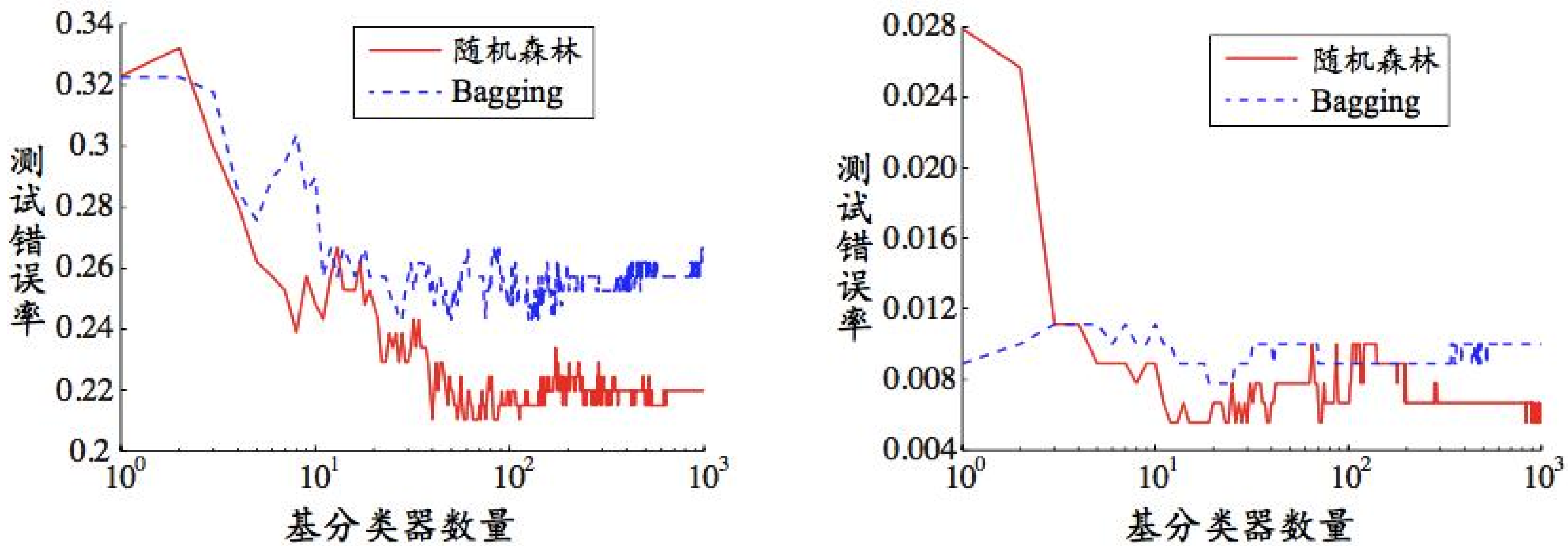
传统决策树在选择划分属性时是在当前节点的属性集合（假定d个属性）中选择一个最优属性

对基决策树的每个节点，先从该节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。 $k = \log_2 d$

属性扰动使得个体学习器相关性进一步减弱，提升了泛化性能

随机森林简单、易实现、计算开销小，很多任务展现强大性能，被誉为“代表集成学习水平的方法”

随机森林

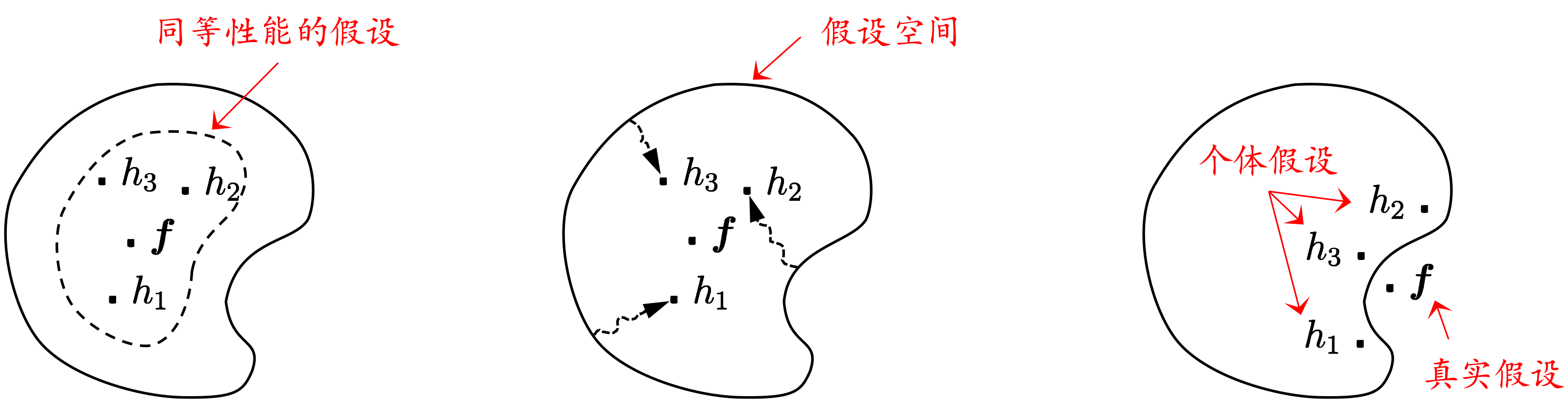


(a) glass 数据集　　(b) auto-mpg 数据集

随着基分类器数目增加，随机森林通常会收敛到更低的泛化误差

结合策略

- 学习器的组合可以从三个方面带来好处



(a) 统计的原因　　(b) 计算的原因　　(c) 表示的原因

学习任务假设空间大，多个假设在训练集上可能达到同等性能，结合多个学习器会减小方差

学习算法往往会陷入局部极小值，有的局部极小点对应的泛化性能可能差，通过多次运行后进行结合，减低陷入差极小点的风险

某些学习任务的真实假设可能不在当前学习算法的假设空间中，结合多个学习器，可以扩大假设空间，学得更好的近似

结合策略—回归

- 简单平均法

$$H(\boldsymbol{x}) = \frac{1}{T}\sum_{i}^{T} \mathrm{h}_i(\boldsymbol{x})$$

- 加权平均法

$$H(\boldsymbol{x}) = \sum_{i}^{T} w_i \mathrm{h}_i(\boldsymbol{x}) \qquad w_i \geq 0 \text{ and } \sum_{i} w_i = 1$$

- 加权平均不一定优于简单平均

结合策略—分类

- 绝对多数投票法（majority voting）假设N个类别

$$H(\boldsymbol{x}) = \begin{cases} c_j & \text{if } \sum_{i=1}^{T} \mathrm{h}_i^j(\boldsymbol{x}) > \frac{1}{2} \sum_{k=1}^{N} \sum_{i=1}^{T} \mathrm{h}_i^k(\boldsymbol{x}) \\ \text{rejection} & \text{otherwise} \end{cases}$$

- 相对多数投票法（plurality voting）

$$H(\boldsymbol{x}) = c_{\operatorname{argmax}_j \sum_{i=1}^{T} \mathrm{h}_i^j(\boldsymbol{x})}$$

- 加权投票法（weighted voting）

$$H(\boldsymbol{x}) = c_{\operatorname{argmax}_j \sum_{i=1}^{T} w_i \mathrm{h}_i^j(\boldsymbol{x})}$$

$$w_i \geq 0 \text{ and } \sum_i w_i = 1$$

结合策略—学习法

- Stacking是学习法的典型代表

先从训练数据集训练出初级学习器，然后生成一个新数据集用于训练次级学习器

Input: Data set $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$;
First-level learning algorithms $\mathfrak{L}_1, \ldots, \mathfrak{L}_T$;
Second-level learning algorithm $\mathfrak{L}$.

Process:
1. **for** $t = 1, \ldots, T$:
2. $\quad h_t = \mathfrak{L}_t(D)$; ← 个体学习器称为初级学习器
3. **end**
4. $D' = \emptyset$;
5. **for** $i = 1, \ldots, m$:
6. $\quad$ **for** $t = 1, \ldots, T$:
7. $\quad\quad z_{it} = h_t(\boldsymbol{x}_i)$;
8. $\quad$ **end**
9. $\quad D' = D' \cup ((z_{i1}, \ldots, z_{iT}), y_i)$; ← 生成一个新数据集
10. **end**
11. $h' = \mathfrak{L}(D')$; ← 用于结合的学习器称为元学习器

Output: $H(\boldsymbol{x}) = h'(h_1(\boldsymbol{x}), \ldots, h_T(\boldsymbol{x}))$

误差-分歧分解

- 对集成学习进行简单理论分析

- 定义学习器h_i的分歧

$$A(\mathrm{h}_i|\boldsymbol{x}) = (\mathrm{h}_i(\boldsymbol{x}) - H(\boldsymbol{x}))^2$$

$$H(\boldsymbol{x}) = \sum_i^T w_i \mathrm{h}_i(\boldsymbol{x})$$

$$w_i \geq 0 \text{ and } \textstyle\sum_i w_i = 1$$

- 集成的分歧

$$\bar{A}(\mathrm{h}|x) = \sum_i^T w_i A(\mathrm{h}_i|x) \qquad = \sum_i^T w_i (\mathrm{h}_i(\boldsymbol{x}) - H(\boldsymbol{x}))^2$$

分歧项代表了个体学习器在样本$\boldsymbol{x}$上的不一致性，即在一定程度上反映了个体学习器的多样性

误差-分歧分解

$$\bar{A}(\mathrm{h}|\boldsymbol{x}) = \sum_{i}^{T} w_i(\mathrm{h}_i(\boldsymbol{x}) - H(\boldsymbol{x}))^2 \qquad = \sum_{i}^{T} w_i(\mathrm{h}_i(\boldsymbol{x}) - f(\boldsymbol{x}) + f(\boldsymbol{x}) - H(\boldsymbol{x}))^2$$

$$= \sum_{i}^{T} w_i(\mathrm{h}_i(\boldsymbol{x}) - f(\boldsymbol{x}))^2 + \sum_{i}^{T} w_i(f(\boldsymbol{x}) - H(\boldsymbol{x}))^2 + 2\sum_{i}^{T} w_i(\mathrm{h}_i(\boldsymbol{x}) - f(\boldsymbol{x}))(f(\boldsymbol{x}) - H(\boldsymbol{x}))$$

$$= \sum_{i}^{T} w_i(\mathrm{h}_i(\boldsymbol{x}) - f(\boldsymbol{x}))^2 + (f(\boldsymbol{x}) - H(\boldsymbol{x}))^2 + 2(H(\boldsymbol{x}) - f(\boldsymbol{x}))(f(\boldsymbol{x}) - H(\boldsymbol{x}))$$

$$= \sum_{i}^{T} w_i(\mathrm{h}_i(\boldsymbol{x}) - f(\boldsymbol{x}))^2 - (f(\boldsymbol{x}) - H(\boldsymbol{x}))^2$$

$$= \sum_{i}^{T} w_i E(\mathrm{h}_i|\boldsymbol{x}) - E(H|\boldsymbol{x}) \quad = \bar{E}(\mathrm{h}|\boldsymbol{x}) - E(H|\boldsymbol{x})$$

$E(\mathrm{h}_i|\boldsymbol{x})$集成$\mathrm{h}_i$的平方误差

$E(H|\boldsymbol{x})$集成H的平方误差

$\bar{E}(\mathrm{h}|\boldsymbol{x})$个体学习器误差的加权均值

误差-分歧分解

$$\bar{A}(h|\boldsymbol{x}) = \bar{E}(h|\boldsymbol{x}) - E(H|\boldsymbol{x})$$

- 上式对所有样本$\boldsymbol{x}$均成立，令$p(\boldsymbol{x})$表示样本的概率密度，则在全样本上有

$$\sum_{i=1}^{T} w_i \int A(h_i|\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} = \sum_{i=1}^{T} w_i \int \bar{E}(h_i|\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} - \int E(H|\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}$$

$$\sum_{i=1}^{T} w_i A_i = \sum_{i=1}^{T} w_i E_i - E$$

$$E = \bar{E} - \bar{A}$$

个体学习器精确性越高($\bar{E}$小)、多样性越大($\bar{A}$大)，则集成效果越好。

$$\bar{A} = \bar{E} - E$$

个体学习器的加权分歧　　个体学习器的加权泛化误差　　集成的泛化误差

多样性增强

- 常见的增强个体学习器的多样性的方法

- 数据样本扰动
- 输入属性扰动
- 输出表示扰动
- 算法参数扰动

多样性增强—数据样本扰动

- 数据样本扰动通常是基于采样法
  - Bagging中的自助采样法
  - Adaboost中的序列采样

数据样本扰动对“不稳定基学习器”很有效

- 对数据样本的扰动敏感的基学习器(不稳定基学习器)
  - 决策树，神经网络等

- 对数据样本的扰动不敏感的基学习器(稳定基学习器)
  - 线性学习器，支持向量机，朴素贝叶斯，k近邻等

多样性增强—属性扰动

- 随机子空间算法(random subspace)
- 从初始属性集中抽取出若干个属性子集，在基于每个属性子集训练一个基学习器

输入： 训练集 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$;
基学习算法 $\mathfrak{L}$;
基学习器数 T;
子空间属性数 d'.

过程：

1: **for** $t = 1, 2, \ldots, T$ **do**
2: $\quad \mathcal{F}_t = \mathrm{RS}(D, d')$
3: $\quad D_t = \mathrm{Map}_{\mathcal{F}_t}(D)$
4: $\quad h_t = \mathfrak{L}(D_t)$
5: **end for**

输出： $H(\boldsymbol{x}) = \underset{y \in \mathcal{Y}}{\arg\max} \sum_{t=1}^{T} \mathbb{I}\left(h_t\left(\mathrm{Map}_{\mathcal{F}_t}(\boldsymbol{x})\right) = y\right)$

多样性增强

- 输出扰动
  - 翻转法(Flipping Output)
  - 输出调剂法(Output Smearing)
  - ECOC法

- 负相关法
  - 显式地通过正则化来强制个体神经网络使用不同的参数

作业

- 习题8.2
- 习题8.6
- 习题8.8