机器学习

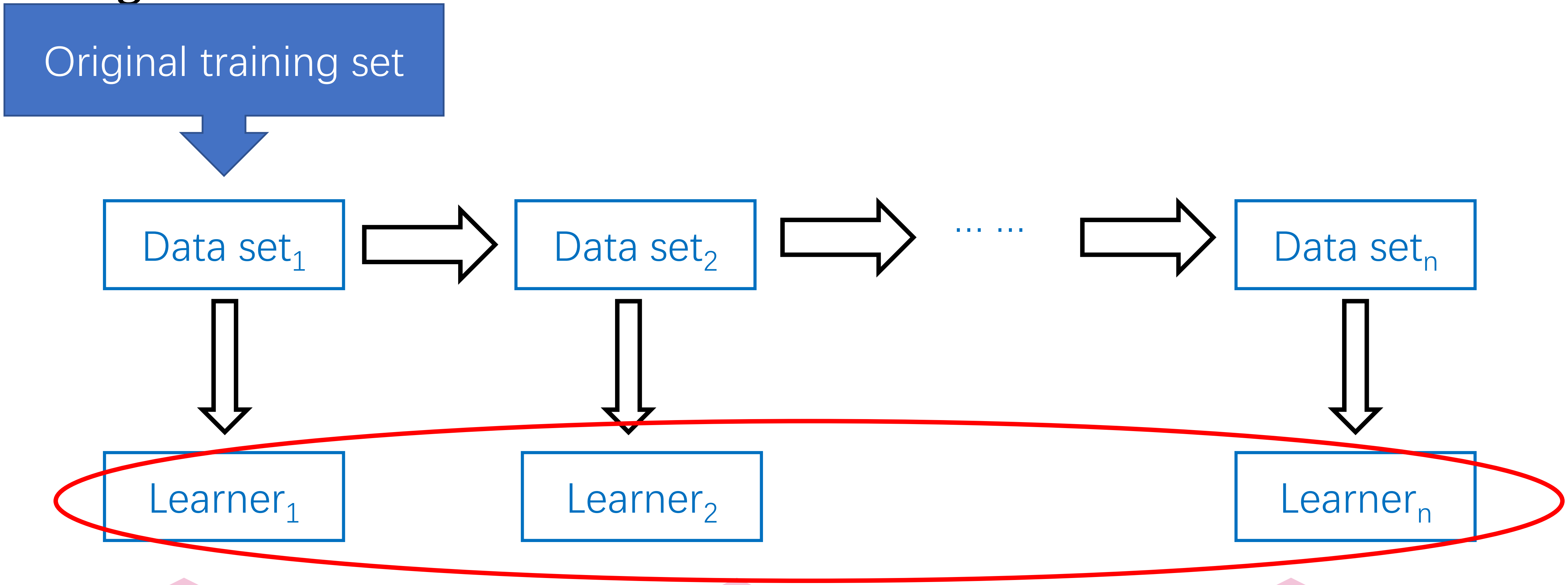
苏州大学计算机科学与技术学院
自然语言处理实验室
主讲：周夏冰
邮箱：zhouxiabing@suda.edu.cn

集成学习

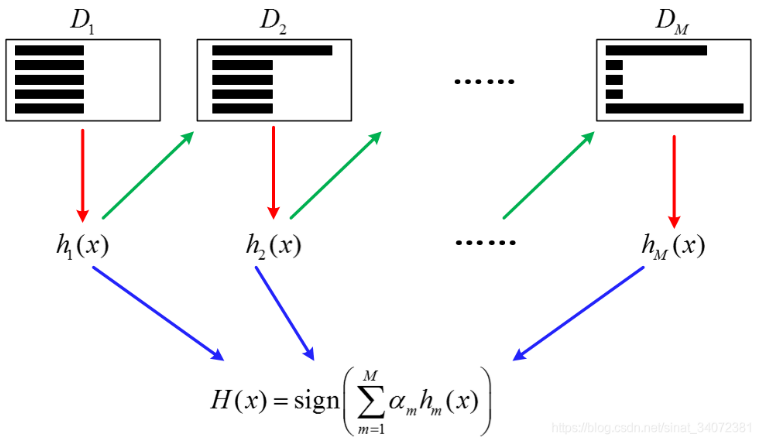
- 集成学习经典的两种
  - **Boosting:** 学习器之间存在着强依赖关系，串行
  - **Bagging**：个体学习器之间不存在强依赖，并行
    - 随机森林

集成学习(Ensemble learning)

- **Boosting**



集成学习



- **Boosting**
  - **AdaBoost**
  - 基于“基学习器的线性组合”
  - $\boldsymbol{H(x) = \sum_{i=1}^{T} \alpha_t h_t(x)}$
  - 基于一个分布对数据集进行训练
    - 分布——权重
  - 每次根据训练器，重新调整分布
  - 初始分布对每个样本**1/*m***

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:

$\mathcal{D}_1(\boldsymbol{x}) = 1/m$.
for $t = 1, 2, \ldots, T$ **do**
　$h_t = \mathfrak{L}(D, \mathcal{D}_t)$;
　$\epsilon_t = P_{\boldsymbol{x} \sim \mathcal{D}_t}(h_t(\boldsymbol{x}) \neq f(\boldsymbol{x}))$;
　if $\epsilon_t > 0.5$ **then break**
　$\alpha_t = \frac{1}{2} \ln \left(\frac{1-\epsilon_t}{\epsilon_t} \right)$;

$$\mathcal{D}_{t+1}(\boldsymbol{x}) = \frac{\mathcal{D}_t(\boldsymbol{x})}{Z_t} \times \begin{cases} \exp(-\alpha_t), & \text{if } h_t(\boldsymbol{x}) = f(\boldsymbol{x}) \\ \exp(\alpha_t), & \text{if } h_t(\boldsymbol{x}) \neq f(\boldsymbol{x}) \end{cases} = \frac{\mathcal{D}_t(\boldsymbol{x}) \exp(-\alpha_t f(\boldsymbol{x}) h_t(\boldsymbol{x}))}{Z_t}$$

end for

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

Adaboost

- 前向分布算法
  - 加法模型：$f(x) = \sum_{m=1}^{M} \overset{\text{系数}}{\boxed{\beta_m}} \overset{\text{基函数}}{\boxed{b(x;\gamma_m)}}$
  - 损失函数：$L(y, f(x))$
  - 学习：$\min\limits_{\beta_m,\gamma_m} \sum_{i=1}^{N} L(y_i, \sum_{m=1}^{M} \beta_m b(x;\gamma_m))$
  - 思想：从前向后，每一步只学一个基函数和系数，逐渐逼近目标函数
    - $\min\limits_{\beta_m,\gamma_m} \sum_{i=1}^{N} L(y_i, f_{m-1}(x_i) + \beta_m b(x_i, \gamma_m))$

Adaboost

- 目标：最小化指数损失函数
  - 指数损失函数：$L\big(y, f(x)\big) = exp(-yf(x))$
  - $l(y, f(x)) = exp(-y(f_{t-1}(x) + \alpha_t h_t(x)))$
  - 目标函数：$\underset{\alpha_t, h_t}{\operatorname{argmin}} \sum_{i=1}^{N} exp(-y(f_{t-1}(x) + \alpha_t h_t(x)))$
  - $\underset{\alpha_t, h_t}{\operatorname{argmin}} \sum_{i=1}^{N} w_i^t exp(-y_i \alpha_t h_t(x_i))$
  - $w_i^t = exp(-yf_{t-1}(x))$ —— 权重

Adaboost

- 目标：最小化指数损失函数

  - $\underset{\alpha,h_t}{argmin}\sum_{i=1}^{N} w_i^t exp(-y_i\alpha_t h_t(x_i))$
  - $\sum_{i=1}^{N} w_i^t exp(-y_i\alpha_t h_t(x_i)) = \sum_{y_i\neq h_t(x_i)} w_i^t \mathrm{exp}(\alpha_t) + \sum_{y_i=h_t(x_i)} w_i^t exp(-\alpha_t)$
  - $= \left(\frac{\sum_{y_i\neq h_t(x_i)} w_i^t}{\sum_{i=1}^{N} w_i^t} e^{\alpha_t} + \frac{\sum_{y_i=h_t(x_i)} w_i^t}{\sum_{i=1}^{N} w_i^t} e^{-\alpha_t}\right) \sum_{i=1}^{N} w_i^t$
  - $= \left(e^{-\alpha_t}(1-\epsilon_t) + e^{\alpha_t}\epsilon_t\right) \sum_{i=1}^{N} w_i^t$
  - $\frac{\partial l}{\partial \alpha_t} = -e^{-\alpha_t}(1-\epsilon_t) + e^{\alpha_t}\epsilon_t = 0$
  - $\alpha_t = \frac{1}{2} ln(\frac{1-\epsilon_t}{\epsilon_t})$

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: $\boxed{\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)}$;
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

- $w_i^t = exp(-y_i f_{t-1}(x_i))$
- $w_i^{t+1} = exp(-y_i f_t(x_i))$
- $= exp\left(-y_i\left(f_{t-1}(x_i) + \alpha_t h_t(x_i)\right)\right)$
- $= exp\left(-y_i f_{t-1}(x_i)\right) exp\left(-y_i \alpha_t h_t(x_i)\right)$
- $= w_i^t exp\left(-y_i \alpha_t h_t(x_i)\right)$

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:

$\mathcal{D}_1(\boldsymbol{x}) = 1/m$.

for $t = 1, 2, \ldots, T$ **do**

$\quad h_t = \mathfrak{L}(D, \mathcal{D}_t)$;

$\quad \epsilon_t = P_{\boldsymbol{x} \sim \mathcal{D}_t}(h_t(\boldsymbol{x}) \neq f(\boldsymbol{x}))$;

$\quad$ **if** $\epsilon_t > 0.5$ **then break**

$\quad \alpha_t = \frac{1}{2} \ln \left(\frac{1-\epsilon_t}{\epsilon_t} \right)$;

$$\boxed{\begin{aligned}\mathcal{D}_{t+1}(\boldsymbol{x}) &= \frac{\mathcal{D}_t(\boldsymbol{x})}{Z_t} \times \begin{cases} \exp(-\alpha_t), & \text{if } h_t(\boldsymbol{x}) = f(\boldsymbol{x}) \\ \exp(\alpha_t), & \text{if } h_t(\boldsymbol{x}) \neq f(\boldsymbol{x}) \end{cases} \\ &= \frac{\mathcal{D}_t(\boldsymbol{x})\exp(-\alpha_t f(\boldsymbol{x}) h_t(\boldsymbol{x}))}{Z_t}\end{aligned}}$$

end for

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

Adaboost

- 例：弱分类器由x<v或x>v产生，用adaboost学习一个强分类器

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

$$D_1 = (w_{11}, w_{12}, \cdots, w_{110}) = (0.1, 0.1, \ldots, 0.1)$$

$$h_1(x) = \begin{cases} 1, & x < 2.5 \\ -1, & x > 2.5 \end{cases}$$

$$e_1 = 0.3 \qquad \alpha_1 = \frac{1}{2} log \frac{1 - e_1}{e_1} = 0.4236$$

$$w_{2i} = \frac{w_{1i}}{Z_1} exp(-y_i \alpha_1 h_1(x))$$

$$w_{21} = \frac{0.1}{Z_1} exp(-0.4236) = 0.07143$$

$$Z_1 = \sum w_{1i} exp(-y_i \alpha_1 h_1(x)) = 0.9165$$

Adaboost

- 例：弱分类器由x<v或x>v产生，用adaboost学习一个强分类器

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

$$D_1 = (w_{11}, w_{12}, \cdots, w_{110}) = (0.1, 0.1, \ldots, 0.1)$$

$$h_1(x) = \begin{cases} 1, & x < 2.5 \\ -1, & x > 2.5 \end{cases}$$

$$e_1 = 0.3 \qquad \alpha_1 = \frac{1}{2} log \frac{1 - e_1}{e_1} = 0.4236$$

$$w_{2i} = \frac{w_{1i}}{Z_1} exp(-y_i \alpha_1 h_1(x))$$

$$w_{27} = \frac{0.1}{Z_1} exp(0.4236) = 0.1667$$

$$Z_1 = \sum w_{1i} exp(-y_i \alpha_1 h_1(x)) = 0.9165$$

Adaboost

- 例：弱分类器由x<v或x>v产生，用adaboost学习一个强分类器

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

$$h_1(x) = \begin{cases} 1, & x < 8.5 \\ -1, & x > 8.5 \end{cases} \qquad e_2 = 0.07143 * 3 = 0.2143$$

$$e = 0.1667 * 3 = 0.5001$$

Adaboost

- 例：弱分类器由x<v或x>v产生，用adaboost学习一个强分类器

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

$$h_1(x) = \begin{cases} 1, & x < 8.5 \\ -1, & x > 8.5 \end{cases}$$

$$e_2 = 0.07143 * 3 = 0.2143$$

$$\alpha_2 = \frac{1}{2} log \frac{1 - e_2}{e_2} = 0.9372$$

$$H(x) = sign\big(0.4236 h_1(x) + 0.9372 h_2(x)\big)$$

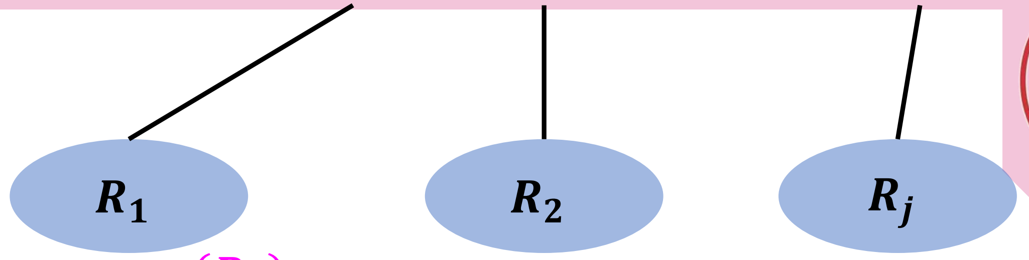
提升树

- 加法模型+前项分布算法
- 基函数：决策树 CART
- $f_M(x) = \sum_{m=1}^{M} T(x;\Theta_m)$

决策树

- $\min_{\theta_m} \sum_{i=1}^{N} L(y_i, f_{m-1}(x_i) + \boxed{T(x_i;\theta_m)})$

提升树



$c_1 = ave(R_1)$

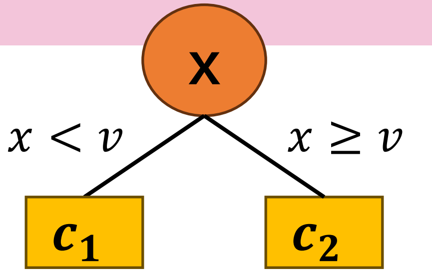
- $\min\limits_{\theta_m} \sum_{i=1}^{N} L(y_i, f_{m-1}(x_i) + T(x_i; \theta_m))$

- $T(x_i; \theta_m) = \sum_{j=1}^{J} c_j I(x_i \in R_j), \Theta = \{(R_1, c_1), (R_2, c_2), \cdots, (R_J, c_J)\}$

- **均方误差为损失函数时：**

  - $L(y, f(x)) = (y - f(x))^2 = [y - f_{m-1}(x) - T(x, \theta_m)]^2 = [\underbrace{r}_{残差} - T(x, \theta_m)]^2$

提升树



- 例

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

考虑不同的切分点

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

$$L_1(x) = \sum_{i=1}^{10} \left(y_i - T_1(x)\right)^2 = 1.93$$

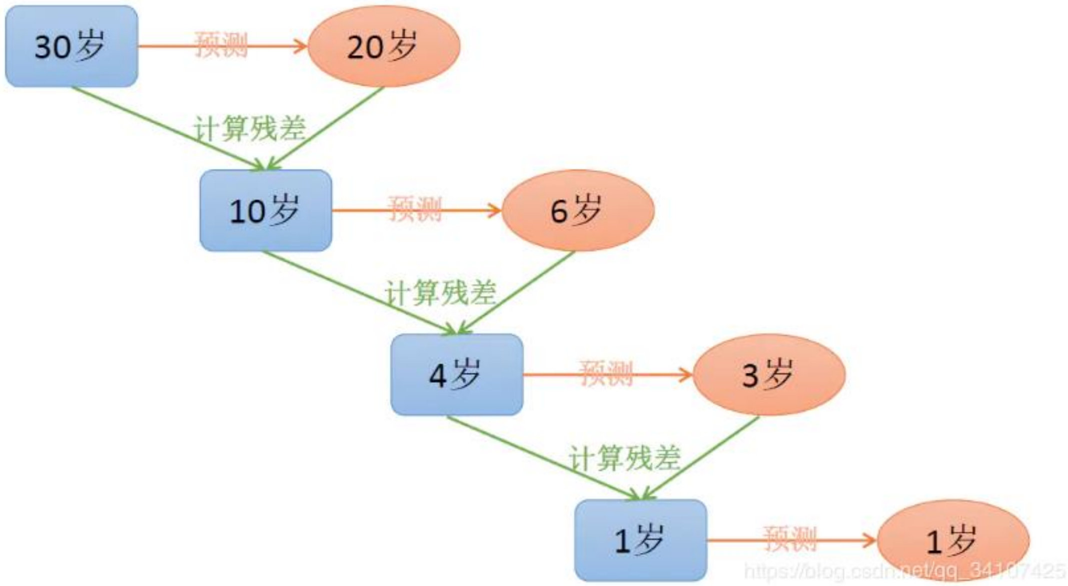
x	1	2	3	4	5	6	7	8	9	10
r	-0.68	-0.51	-0.33	0.16	0.56	0.81	-0.01	-0.21	0.09	0.14

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

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

提升树

- 初始化$f_0(x) = 0$
- $m = 1, 2, \ldots, M$
  - $r_{mi} = y_i - f_{m-1}(x_i), i = 1, 2, \cdots, N$
  - 拟合残差学习回归树，得到$T(x_i; \theta_m)$
  - 更新$f_m(x_i) = f_{m-1}(x_i) + T(x_i; \theta_m)$
- 得到回归问题的提升树：$f_M(x)$



梯度提升树GDBT

- 训练样本$T=\{(x_1,y_1),\cdots,(x_m,y_{,m})\}$，最大迭代次数T，损失函数L
- 1. 初始化弱学习器$f_0(x)=\underset{c}{\operatorname{argmin}}\sum_{i=1}^{m}L(y_i,c)$
- 2. 对迭代轮数t=1,2,...,T，有
  - （1）$r_{ti}=\frac{\partial L(y_i,f(x_i))}{\partial f(x_i)},i=1,2,\cdots,m$

    $L=(y-f)^2$
  - （2）利用$\{(x_1,r_{t1}),\cdots,(x_m,r_{,tm})\}$，构建一颗CART回归树，其叶子节点区域为$R_{tj}$, $\mathrm{j}=1,2,\cdots,J$，J为叶子节点个数
  - （3）对于$j=1,2,\ldots,J$，$c_{tj}=\underset{c}{\operatorname{argmin}}\sum_{x_i\in R_{tj}}L(y_i,f_{t-1}(x_i)+c)$
  - （4）更新$f_t(x_i)=f_{t-1}(x_i)+\sum_{j=1}^{J}c_{tj}I(x_i\in R_{tj})$
- 得到回归问题的提升树：$f_M(x)=f_0+\sum_{t=1}^{T}\sum_{j=1}^{J}c_{tj}I(x_i\in R_{tj})$