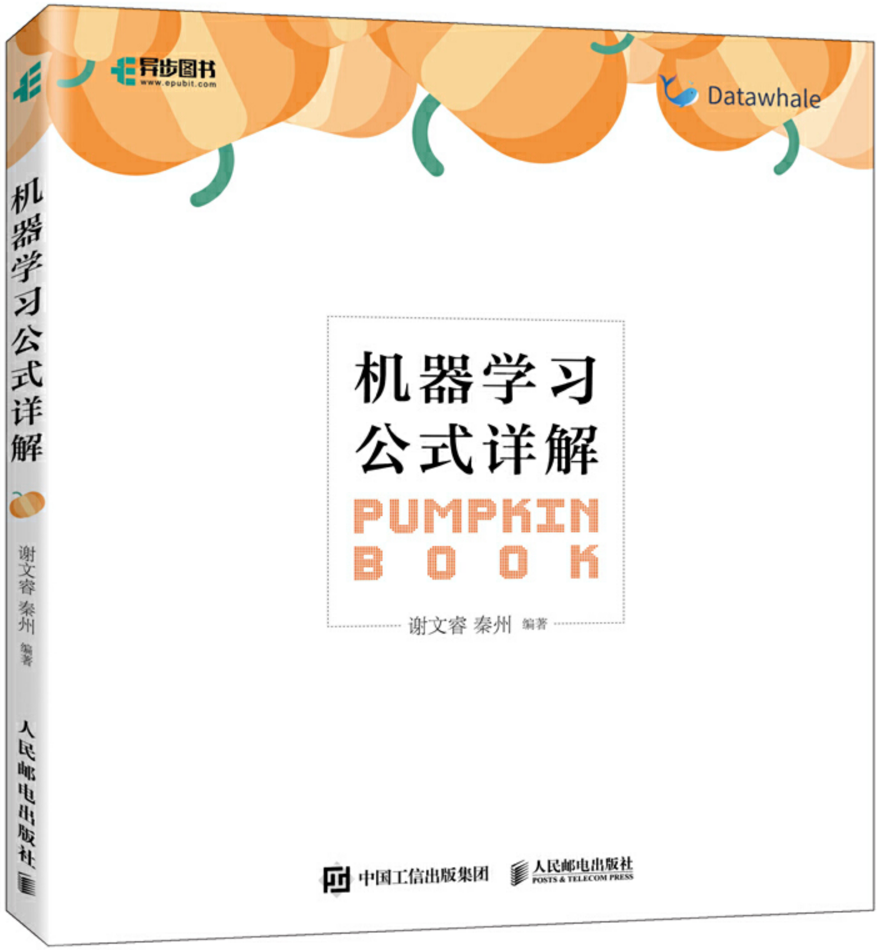
Datawhale 异步社区




# 《机器学习公式详解》（南瓜书）

## 第8章 集成学习

本节主讲：秦州

# 本节大纲

西瓜书对应章节：8.1、8.2

1. 个体与集成
2. Adaboost算法

# 个体与集成

“三个臭皮匠，顶个诸葛亮”，集成学习通过集合 **多个个体学习器** 的结果来提升预测结果的准确性和泛化能力。

“君子和而不同” 个体学习器需要比随机猜想要强一些，个体学习器的预测结果也要具有一定的多样性。

| | **样本a** | **样本b** | **样本c** | | **样本a** | **样本b** | **样本c** | | **样本a** | **样本b** | **样本c** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 学习器1 | 1 | 1 | 0 | | 1 | 0 | 0 | | 1 | 1 | 0 |
| 学习器2 | 1 | 0 | 1 | | 0 | 1 | 0 | | 1 | 1 | 0 |
| 学习器3 | 0 | 1 | 1 | | 0 | 0 | 1 | | 1 | 1 | 0 |
| 集成结果 | 1 | 1 | 1 | | 0 | 0 | 0 | | 1 | 1 | 0 |

# 个体与集成



集成个体学习器的收敛性保证：

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

两个基本结论：

- 收敛速率随着个体学习器数量T呈指数下降
- $\epsilon = 0.5$的个体集成器对收敛没有作用

# Adaboost算法1



学习$T$个个体学习器$h_t$和相应的权重$\alpha_t$，使得他们的加权和

$$H(\boldsymbol{x})=\sum_{t=1}^{T}\alpha_t h_t(\boldsymbol{x})$$

能够最小化损失函数

$$\ell_{\exp}(H\mid\mathcal{D})=\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[e^{-f(\boldsymbol{x})H(\boldsymbol{x})}\right]$$

# Adaboost算法2



**前向分布求解算法：** 每一轮只学习一个学习器$h_t$和相应的权重$\alpha_t$，第$t$轮的优化目标

$$\left(\alpha_t, h_t\right)=\underset{\alpha, h}{\arg \min } \ell_{\exp }\left(H_{t-1}+\alpha h \mid \mathcal{D}\right)$$

根据指数损失函数的定义式(8.5)，有

$$\begin{aligned}
\ell_{\exp }\left(H_{t-1}+\alpha h \mid \mathcal{D}\right) & =\mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}}\left[e^{-f(\boldsymbol{x})\left(H_{t-1}(\boldsymbol{x})+\alpha h(\boldsymbol{x})\right)}\right] \\
& =\sum_{i=1}^{|D|} \mathcal{D}\left(\boldsymbol{x}_i\right) e^{-f\left(\boldsymbol{x}_i\right)\left(H_{t-1}\left(\boldsymbol{x}_i\right)+\alpha h\left(\boldsymbol{x}_i\right)\right)} \\
& =\sum_{i=1}^{|D|} \mathcal{D}\left(\boldsymbol{x}_i\right) e^{-f\left(\boldsymbol{x}_i\right) H_{t-1}\left(\boldsymbol{x}_i\right)} e^{-f\left(\boldsymbol{x}_i\right) \alpha h\left(\boldsymbol{x}_i\right)}
\end{aligned}$$

# Adaboost算法3



因为$f(x_i)$和$h(x_i)$仅可取值$\{-1,1\}$，可以推得

$$\begin{aligned}\ell_{\exp}\left(H_{t-1}+\alpha h \mid \mathcal{D}\right) &= \sum_{i=1}^{|D|} \mathcal{D}\left(\boldsymbol{x}_i\right) e^{-f\left(\boldsymbol{x}_i\right) H_{t-1}\left(\boldsymbol{x}_i\right)}\left(e^{-\alpha}+\left(e^{\alpha}-e^{-\alpha}\right) \mathbb{I}\left(f\left(\boldsymbol{x}_i\right) \neq h\left(\boldsymbol{x}_i\right)\right)\right) \\ &= \sum_{i=1}^{|D|} \mathcal{D}\left(\boldsymbol{x}_i\right) e^{-f\left(\boldsymbol{x}_i\right) H_{t-1}\left(\boldsymbol{x}_i\right)} e^{-\alpha}+\sum_{i=1}^{|D|} \mathcal{D}\left(\boldsymbol{x}_i\right) e^{-f\left(\boldsymbol{x}_i\right) H_{t-1}\left(\boldsymbol{x}_i\right)}\left(e^{\alpha}-e^{-\alpha}\right) \mathbb{I}\left(f\left(\boldsymbol{x}_i\right) \neq h\left(\boldsymbol{x}_i\right)\right)\end{aligned}$$

做一个简单的符号替换，令$\mathcal{D}_t'\left(\boldsymbol{x}_i\right)=\mathcal{D}\left(\boldsymbol{x}_i\right) e^{-f\left(\boldsymbol{x}_i\right) H_{t-1}\left(\boldsymbol{x}_i\right)}$，并且注意到$e^{-\alpha}$和$e^{\alpha}-e^{-\alpha}$与求和变量$i$无关，可以提取出来，有

$$\ell_{\exp}\left(H_{t-1}+\alpha h \mid \mathcal{D}\right)=e^{-\alpha} \sum_{i=1}^{|D|} \mathcal{D}_t'\left(\boldsymbol{x}_i\right)+\left(e^{\alpha}-e^{-\alpha}\right) \sum_{i=1}^{|D|} \mathcal{D}_t'\left(\boldsymbol{x}_i\right) \mathbb{I}\left(f\left(\boldsymbol{x}_i\right) \neq h\left(\boldsymbol{x}_i\right)\right)$$

# Adaboost算法4



$$\ell_{\exp}\left(H_{t-1}+\alpha h \mid \mathcal{D}\right)=e^{-\alpha} \sum_{i=1}^{|D|} \mathcal{D}_t^{\prime}\left(\boldsymbol{x}_i\right)+\left(e^\alpha-e^{-\alpha}\right) \sum_{i=1}^{|D|} \mathcal{D}_t^{\prime}\left(\boldsymbol{x}_i\right) \mathbb{I}\left(f\left(\boldsymbol{x}_i\right) \neq h\left(\boldsymbol{x}_i\right)\right)$$

我们的目的是求解$h_t$使得$\ell_{\exp}$最小化，因此可以忽略掉与$h$无关的项，即求解目标是

$$h_t=\underset{h}{\arg \min }\left(e^\alpha-e^{-\alpha}\right) \sum_{i=1}^{|D|} \mathcal{D}_t^{\prime}\left(\boldsymbol{x}_i\right) \mathbb{I}\left(f\left(\boldsymbol{x}_i\right) \neq h\left(\boldsymbol{x}_i\right)\right)$$

更近一步，由于$\alpha>\frac{1}{2}$，易证得$e^\alpha-e^{-\alpha}>0$恒成立，因此求解目标为：

$$h_t=\underset{h}{\arg \min } \sum_{i=1}^{|D|} \mathcal{D}_t^{\prime}\left(\boldsymbol{x}_i\right) \mathbb{I}\left(f\left(\boldsymbol{x}_i\right) \neq h\left(\boldsymbol{x}_i\right)\right)$$

# Adaboost算法5



$$h_t=\arg\min_h \sum_{i=1}^{|D|} \mathcal{D}_t'\left(\boldsymbol{x}_i\right) \mathbb{I}\left(f\left(\boldsymbol{x}_i\right) \neq h\left(\boldsymbol{x}_i\right)\right)$$

其中 $\mathcal{D}_t'\left(\boldsymbol{x}_i\right)=\mathcal{D}\left(\boldsymbol{x}_i\right) e^{-f\left(\boldsymbol{x}_i\right) H_{t-1}\left(\boldsymbol{x}_i\right)}$

观察$\mathcal{D}_t'\left(\boldsymbol{x}_i\right)$的形式可以发现它仅与$t-1$轮及以前的学习器有关，因此在求解$h_t$时，对于每个样本$i$，他其实已经固定了，如果把$\mathcal{D}_t'\left(\boldsymbol{x}_i\right)$看做样本$i$在$t$轮学习时的权重分布，我们要依据这个权重求解学习器$h_t$以满足上面的最优化式子。

同时，为了确保$\mathcal{D}_t'\left(\boldsymbol{x}_i\right)$是一个分布，通常我们对其进行规范化后作为下一个学习器的输入样本权重，即$\mathcal{D}_t\left(\boldsymbol{x}_i\right)=\frac{\mathcal{D}_t'\left(\boldsymbol{x}_i\right)}{\sum_{i=1}^{|D|} \mathcal{D}_t'\left(\boldsymbol{x}_i\right)}$，其中分母是常数，因此这个变换不会影响上述最小化的求解。

# Adaboost算法6

有意思的一点是，$t$轮的样本权重可以通过$t-1$轮样本权重计算，而无需从头算起，以$t+1$轮为例，根据迭代公式，有：

$$
\begin{aligned}
\mathcal{D}_{t+1}\left(\boldsymbol{x}_i\right) &= \mathcal{D}\left(\boldsymbol{x}_i\right) e^{-f\left(\boldsymbol{x}_i\right) H_t\left(\boldsymbol{x}_i\right)} \\
&= \mathcal{D}\left(\boldsymbol{x}_i\right) e^{-f\left(\boldsymbol{x}_i\right)\left(H_{t-1}\left(\boldsymbol{x}_i\right)+\alpha_t h_t\left(\boldsymbol{x}_i\right)\right)} \\
&= \mathcal{D}\left(\boldsymbol{x}_i\right) e^{-f\left(\boldsymbol{x}_i\right) H_{t-1}\left(\boldsymbol{x}_i\right)} e^{-f\left(\boldsymbol{x}_i\right) \alpha_t h_t\left(\boldsymbol{x}_i\right)} \\
&= \mathcal{D}_t\left(\boldsymbol{x}_i\right) e^{-f\left(\boldsymbol{x}_i\right) \alpha_t h_t\left(\boldsymbol{x}_i\right)}
\end{aligned}
$$

这便是《机器学习》式8.19

# Adaboost算法7

下面求解学习器$h_t$的权重$\alpha_t$。损失函数$\ell_{\exp}\left(H_{t-1}+\alpha h \mid \mathcal{D}\right)$对$\alpha$求导有:

$$
\begin{aligned}
\frac{\partial \ell_{\exp}\left(H_{t-1}+\alpha h_t \mid \mathcal{D}\right)}{\partial \alpha} &= \frac{\partial\left(e^{-\alpha} \sum_{i=1}^{|D|} \mathcal{D}_t^{\prime}\left(\boldsymbol{x}_i\right)+\left(e^{\alpha}-e^{-\alpha}\right) \sum_{i=1}^{|D|} \mathcal{D}_t^{\prime}\left(\boldsymbol{x}_i\right) \mathbb{I}\left(f\left(\boldsymbol{x}_i\right) \neq h\left(\boldsymbol{x}_i\right)\right)\right)}{\partial \alpha} \\
&= -e^{-\alpha} \sum_{i=1}^{|D|} \mathcal{D}_t^{\prime}\left(\boldsymbol{x}_i\right)+\left(e^{\alpha}+e^{-\alpha}\right) \sum_{i=1}^{|D|} \mathcal{D}_t^{\prime}\left(\boldsymbol{x}_i\right) \mathbb{I}\left(f\left(\boldsymbol{x}_i\right) \neq h\left(\boldsymbol{x}_i\right)\right)
\end{aligned}
$$

令导数等于0，移项可得:

$$
\begin{aligned}
\frac{e^{-\alpha}}{e^{\alpha}+e^{-\alpha}} &= \frac{\sum_{i=1}^{|D|} \mathcal{D}_t^{\prime}\left(\boldsymbol{x}_i\right) \mathbb{I}\left(f\left(\boldsymbol{x}_i\right) \neq h\left(\boldsymbol{x}_i\right)\right)}{\sum_{i=1}^{|D|} \mathcal{D}_t^{\prime}\left(\boldsymbol{x}_i\right)} = \sum_{i=1}^{|D|} \frac{\mathcal{D}_t^{\prime}\left(\boldsymbol{x}_i\right)}{Z_t} \mathbb{I}\left(f\left(\boldsymbol{x}_i\right) \neq h\left(\boldsymbol{x}_i\right)\right) \\
&= \sum_{i=1}^{|D|} \mathcal{D}_t\left(\boldsymbol{x}_i\right) \mathbb{I}\left(f\left(\boldsymbol{x}_i\right) \neq h\left(\boldsymbol{x}_i\right)\right) = \mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}_t}\left[\mathbb{I}\left(f\left(\boldsymbol{x}_i\right) \neq h\left(\boldsymbol{x}_i\right)\right)\right] = \epsilon_t
\end{aligned}
$$

# Adaboost算法8



$$\frac{e^{-\alpha}}{e^{\alpha}+e^{-\alpha}}=\frac{1}{e^{2\alpha}+1}\Rightarrow e^{2\alpha}+1=\frac{1}{\epsilon_t}\Rightarrow e^{2\alpha}=\frac{1-\epsilon_t}{\epsilon_t}\Rightarrow 2\alpha=\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$$
$$\Rightarrow \alpha_t=\frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$$

这便是《机器学习》式8.11

当$\epsilon>\frac{1}{2}$时，上式单调递减，因此误差率越大的学习器分配的权重越少。

# Adaboost算法9



---

**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: $\mathcal{D}_1(\boldsymbol{x}) = 1/m$. — 初始化样本权值分布.

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

3: $h_t = \mathfrak{L}(D, \mathcal{D}_t)$; — 基于分布 $\mathcal{D}_t$ 从数据集 $D$ 中训练出分类器 $h_t$.

4: $\epsilon_t = P_{\boldsymbol{x} \sim \mathcal{D}_t}(h_t(\boldsymbol{x}) \neq f(\boldsymbol{x}))$; — 估计 $h_t$ 的误差.

5: **if** $\epsilon_t > 0.5$ **then break**

6: $\alpha_t = \frac{1}{2} \ln \left( \frac{1-\epsilon_t}{\epsilon_t} \right)$; — 确定分类器 $h_t$ 的权重.

7: 
$$\begin{aligned}\mathcal{D}_{t+1}(\boldsymbol{x}) &= \frac{\mathcal{D}_t(\boldsymbol{x})}{Z_t} \times \begin{cases} \exp(-\alpha_t), & \text{if } h_t(\boldsymbol{x}) = f(\boldsymbol{x}) \\ \exp(\alpha_t), & \text{if } h_t(\boldsymbol{x}) \neq f(\boldsymbol{x}) \end{cases} \\ &= \frac{\mathcal{D}_t(\boldsymbol{x})\exp(-\alpha_t f(\boldsymbol{x}) h_t(\boldsymbol{x}))}{Z_t}\end{aligned}$$
— 更新样本分布, 其中 $Z_t$ 是规范化因子, 以确保 $\mathcal{D}_{t+1}$ 是一个分布.

8: **end for**

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

---

图 8.3 AdaBoost算法

# 预告

下一节：Bagging、随机森林、多样性增强

增补知识点：GB(Gradient Boosting)/GBDT/XGBoost

西瓜书对应章节：8.3 8.5

增补知识点参考资料：略

# 结束语

欢迎加入【南瓜书读者交流群】，我们将在群里进行答疑、勘误、本次直播回放、本次直播PPT发放、下次直播通知等最新资源发放和活动通知。

加入步骤：

1. 关注公众号【Datawhale】，发送【南瓜书】三个字获取机器人“小豚”的微信二维码
2. 添加“小豚”为微信好友，然后对“小豚”发送【南瓜书】三个字即可自动邀请进群