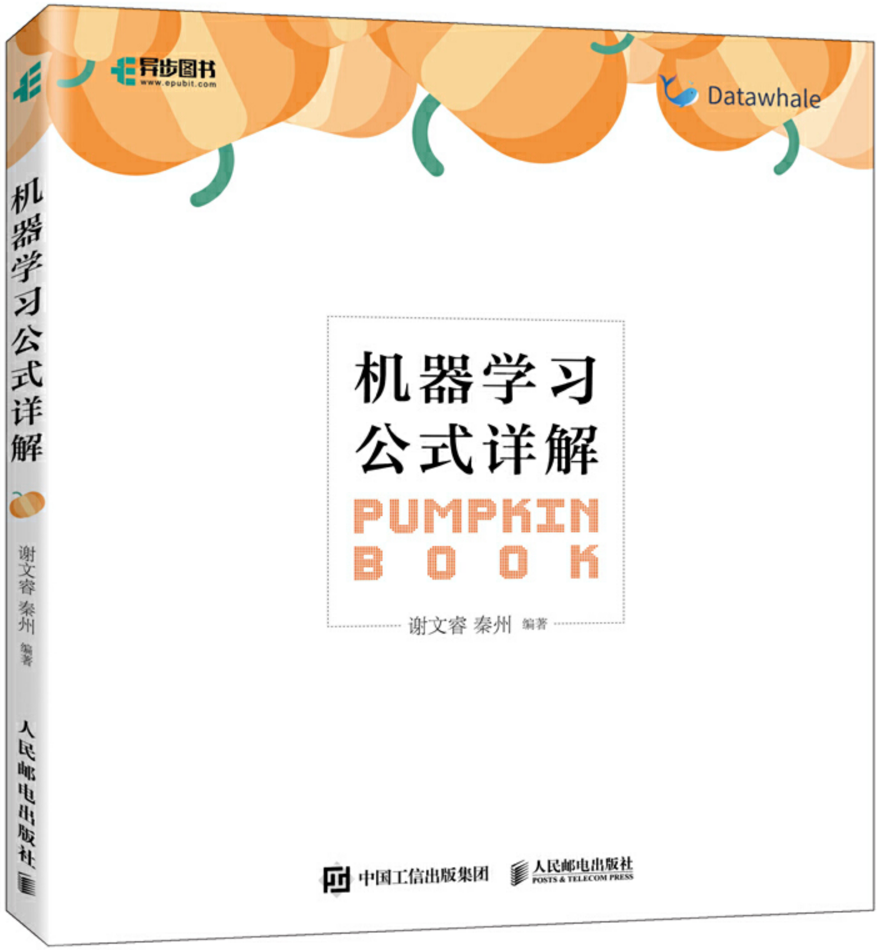
Datawhale 异步社区



《机器学习公式详解》（南瓜书）

第8章 集成学习（下）

本节主讲：秦州

本节大纲

Datawhale 异步社区

南瓜书对应章节：8.3、8.4

0. 增补知识点：GB(Gradient Boosting)/GBDT/XGBoost
1. Bagging
2. 随机森林（Random Forest）
3. 多样性增强方法

Gradient Boosting

将AdaBoost问题一般化，即不限定损失函数为指数函数，也不限定局限于二分类问题，那么更一般的Booting形式为：

$$
\begin{aligned}
\ell\left(H_t \mid \mathcal{D}\right) &= \mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}}\left[\operatorname{err}\left(H_t(\boldsymbol{x}), f(\boldsymbol{x})\right)\right] \\
&= \mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}}\left[\operatorname{err}\left(H_{t-1}(\boldsymbol{x})+\alpha_t h_t(\boldsymbol{x}), f(\boldsymbol{x})\right)\right]
\end{aligned}
$$

比如当我们研究是是回归问题时，$f(x) \in \mathbb{R}$且损失函数为平方损失函数

$\operatorname{err}\left(H_t(\boldsymbol{x}), f(\boldsymbol{x})\right)=\left(H_t(\boldsymbol{x})-f(\boldsymbol{x})\right)^2$

Gradient Boosting 1

类似于 AdaBoost，第t轮得到α_t, $h_t(\boldsymbol{x})$,可先对损失函数在$H_{t-1}(\boldsymbol{x})$处进行泰勒展开：

$$
\begin{aligned}
\ell\left(H_t \mid \mathcal{D}\right) & \approx \mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}}\left[\operatorname{err}\left(H_{t-1}(\boldsymbol{x}), f(\boldsymbol{x})\right)+\left.\frac{\partial \operatorname{err}\left(H_t(\boldsymbol{x}), f(\boldsymbol{x})\right)}{\partial H_t(\boldsymbol{x})}\right|_{H_t(\boldsymbol{x})=H_{t-1}(\boldsymbol{x})}\left(H_t(\boldsymbol{x})-H_{t-1}(\boldsymbol{x})\right)\right] \\
& =\mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}}\left[\operatorname{err}\left(H_{t-1}(\boldsymbol{x}), f(\boldsymbol{x})\right)+\left.\frac{\partial \operatorname{err}\left(H_t(\boldsymbol{x}), f(\boldsymbol{x})\right)}{\partial H_t(\boldsymbol{x})}\right|_{H_t(\boldsymbol{x})=H_{t-1}(\boldsymbol{x})} \alpha_t h_t(\boldsymbol{x})\right] \\
& =\mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}}\left[\operatorname{err}\left(H_{t-1}(\boldsymbol{x}), f(\boldsymbol{x})\right)\right]+\mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}}\left[\left.\frac{\partial \operatorname{err}\left(H_t(\boldsymbol{x}), f(\boldsymbol{x})\right)}{\partial H_t(\boldsymbol{x})}\right|_{H_t(\boldsymbol{x})=H_{t-1}(\boldsymbol{x})} \alpha_t h_t(\boldsymbol{x})\right]
\end{aligned}
$$

Gradient Boosting 2

上式中括号内第1项为常量 $\ell\left(H_{t-1} \mid \mathcal{D}\right)$，因此最小化 $\ell\left(H_{t} \mid \mathcal{D}\right)$ 只需要最小化第二项即可。先不考虑α_t，求解如下优化问题即可得到$h_t(\boldsymbol{x})$:

$$h_t(\boldsymbol{x}) = \underset{h}{\arg\min} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\left.\frac{\partial \operatorname{err}(H_t(\boldsymbol{x}), f(\boldsymbol{x}))}{\partial H_t(\boldsymbol{x})}\right|_{H_t(\boldsymbol{x})=H_{t-1}(\boldsymbol{x})} h(\boldsymbol{x})\right] \quad \text{s.t. constraints for } h(\boldsymbol{x})$$

解得$h_t(\boldsymbol{x})$之后，再求解如下优化问题可得权重项α_t:

$$\alpha_t = \underset{\alpha}{\arg\min} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\operatorname{err}\left(H_{t-1}(\boldsymbol{x}) + \alpha h_t(\boldsymbol{x}), f(\boldsymbol{x})\right)\right]$$

以上就是梯度提升(Gradient Boosting)的理论框架，即每轮通过梯度(Gradient)下降的方式将个体弱学习器提升(Boosting)为强学习器。可以看出 AdaBoost 是其特殊形式。

Adaboost 再推导

$$
\begin{aligned}
h_t(\boldsymbol{x}) &= \underset{h}{\arg\min}\,\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\left.\frac{\partial\,\mathrm{err}\left(H_t(\boldsymbol{x}), f(\boldsymbol{x})\right)}{\partial H_t(\boldsymbol{x})}\right|_{H_t(\boldsymbol{x})=H_{t-1}(\boldsymbol{x})} h(\boldsymbol{x})\right] \\
&= \underset{h}{\arg\min}\,\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\left.\frac{\partial e^{-f(\boldsymbol{x})H_t(\boldsymbol{x})}}{\partial H_t(\boldsymbol{x})}\right|_{H_t(\boldsymbol{x})=H_{t-1}(\boldsymbol{x})} h(\boldsymbol{x})\right] \\
&= \underset{h}{\arg\min}\,\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[-f(\boldsymbol{x})e^{-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x})} h(\boldsymbol{x})\right] = \underset{h}{\arg\min}\,\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[-f(\boldsymbol{x})h(\boldsymbol{x})\right]
\end{aligned}
$$

由$f(\boldsymbol{x}), h(\boldsymbol{x}) \in \{-1, 1\}$，有

$$
f(\boldsymbol{x})h(\boldsymbol{x}) = 1 - 2\mathbb{I}(f(\boldsymbol{x}) \neq h(\boldsymbol{x}))
$$

因此，得到《机器学习》式8.18

$$
h_t(\boldsymbol{x}) = \underset{h}{\arg\min}\,\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_t}[\mathbb{I}(f(\boldsymbol{x}) \neq h(\boldsymbol{x}))]
$$

GBDT 和 XGBoost

GBDT 以Gradient Boosting为基本框架，并使用CART作为个体学习器。

1. 针对回归问题，GBDT 采用平方损失作为损失函数。$\operatorname{err}\left(H_t(\boldsymbol{x}), f(\boldsymbol{x})\right)=\left(H_t(\boldsymbol{x})-f(\boldsymbol{x})\right)^2$
2. 针对二分类问题，GBDT采用对数似然损失函数 $\operatorname{err}\left(H_t(\boldsymbol{x}), f(\boldsymbol{x})\right)=\log\left(1+\exp\left(-H_t(\boldsymbol{x}) f \boldsymbol{x}\right)\right)$

XGBoost 即eXtreme Gradient Boosting的缩写，XGBoost 与GBDT的关系可以类比为LIBSVM和SVM的关系，即XGBoost是GBDT的一种高效实现和改进。

Bagging

Bagging是并行式集成学习的代表。我们可采样出T个含m训练样本的采样集，基于每个采样集训练一个基学习器然后将他们结合起来进行预测。

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

图 8.5 Bagging 算法

自助采样法（booststrap sampling）：

假设从n个样本有放回地抽出n个样本，n次抽样后，有的样本会重复被抽到，有的样本没有被抽到，取没有被抽到的样本作为验证集，它们占比约为:

$$lim_{n\to\infty}\left(1-\frac{1}{n}\right)^n=\frac{1}{e}\approx 36.6\%$$

随机森林

随机森林（Random Forest）是Bagging的一个扩展变体，在以决策树为基学习器构建Bagging集成的基础上，进一步在决策树的训练过程中引入了属性的随机选择。

假设样本包含d个属性对基决策树的每个节点，先从该节点的属性结合中随机选择包含k（$k \leq d$）个属性的子集用来进行最优划分。

随机森林训练效率通常优于Bagging，因为每个节点的划分只需要部分属性参与，而随机森林的泛化误差通常低于bagging，因为属性的扰动为每个基决策树提供了更高的鲁棒性（不易过拟合到训练集上）。

多样性增强

1. 数据样本扰动
    - 对输入扰动敏感的基学习器：决策树、神经网络等
    - 对输入扰动不敏感的基学习器：线性学习器、支持向量机、朴素贝叶斯、k近邻等
2. 输入属性扰动
    - 对包含有大量冗余属性的数据能够大幅加速训练效率
3. 输出属性扰动
    - 随机改变一些训练样本的标记
    - Dropout
4. 算法参数扰动
    - L1、L2正则化等

预告

下一节：聚类、性能和距离度量、原型聚类和密度聚类
西瓜书对应章节：第9章

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