Adaboost

耿远昊

Datawhale

2021 年 8 月 8 日

章节内容

概述

Adaboost 的全称是 Adaptive Boosting，其含义为自适应 Boosting 算法。其中，自适应是指 Adaboost 会根据本轮样本的误差结果来分配下一轮模型训练时样本在模型中的相对权重，即对错误的或偏差大的样本适度“重视”，对正确的或偏差小的样本适度“放松”，这里的“重视”和“放松”具体体现在了 Adaboost 的损失函数设计以及样本权重的更新策略。本课我们将介绍 Adaboost 处理分类和回归任务的算法原理，包括 SAMME 算法、SAMME.R 算法和 Adaboost.R2 算法。

分类损失

对于 K 分类问题而言，当样本标签 $\mathbf{y} = [y_1, ..., y_K]^T$ 的类别 c 为第 k 类时，记

$$y_k = \begin{cases} 1, & \text{if } c = k \\ -\dfrac{1}{k-1}, & \text{if } c \neq k \end{cases}$$

设模型的输出结果为 $\mathbf{f} = [f_1, ..., f_K]^T$，则记损失函数为

$$L(\mathbf{y}, \mathbf{f}) = \exp(-\frac{\mathbf{y}^T \mathbf{f}}{K})$$

损失函数

由于对任意的向量 $a\mathbb{1}$ 有

$$L(\mathbf{y}, \mathbf{f} + a\mathbb{1}) = \exp(-\frac{\mathbf{y}^T\mathbf{f}}{K} - \frac{a\mathbf{y}^T\mathbb{1}}{K}) = \exp(-\frac{\mathbf{y}^T\mathbf{f}}{K}) = L(\mathbf{y}, \mathbf{f})$$

因此为了保证 $\mathbf{f}$ 的可估性，我们需要作出约束假设，此处选择对称约束条件

$$f_1 + f_2 + ... + f_K = 0$$

损失函数

从概率角度而言，一个设计良好的分类问题损失函数应当保证模型在期望损失达到最小时的输出结果是使得后验概率 $P(c|\mathbf{x})$ 达到最大的类别，这个条件被称为贝叶斯最优决策条件。在本问题下，满足对称约束条件的损失函数期望损失 $\mathbb{E}_{\mathbf{Y}|\mathbf{x}}L(\mathbf{Y}, f)$ 达到最小时，由拉格朗日乘子法可解得模型输出为

$$
\begin{aligned}
k^* &= \arg\max_k f_k^*(\mathbf{x}) \\
&= \arg\max_k (K-1)[\log P(c=k|\mathbf{x}) - \frac{1}{K}\sum_{i=1}^{K}\log P(c=i|\mathbf{x})] \\
&= \arg\max_k P(c=k|\mathbf{x})
\end{aligned}
$$

因此，选择指数损失能够满足贝叶斯最优决策条件。

SAMME

SAMME 算法的全称是 **S**tagewise **A**dditive **M**odeling using a **M**ulti-class **E**xponential loss function，它假定模型的总输出 $\mathbf{f}$ 具有 $\mathbf{f}^{(M)}(\mathbf{x}) = \sum_{m=1}^{M} \beta^{(m)} \mathbf{b}^{(m)}(\mathbf{x})$ 的形式。其中，M 是模型的总迭代轮数，$\beta^{(m)} \in \mathbb{R}^+$ 是每轮模型的加权系数，$\mathbf{b}^{(m)}(\mathbf{x}) \in \mathbb{R}^K$ 是基模型 G 输出类别的标签向量。设样本的标签类别为 k，当基模型预测的样本类别结果为 k' 时，记

$$b_{k'}^{(m)} = \begin{cases} 1, & \text{if } k' = k \\ -\dfrac{1}{k-1}, & \text{if } k' \neq k \end{cases}$$

SAMME

对于第 m 轮迭代而言，上一轮的模型输出为 $\mathbf{f}^{(m-1)}(\mathbf{x})$，本轮需要优化得到的 $\beta^{*(m)}$ 和 $\mathbf{b}^{*(m)}$ 满足

$$(\beta^{*(m)}, \mathbf{b}^{*(m)}) = \arg\min_{\beta^{(m)}, \mathbf{b}^{(m)}} \sum_{i=1}^{n} L(\mathbf{y}_i, \mathbf{f}^{(m-1)}(\mathbf{x}_i) + \beta^{(m)} \mathbf{b}^{(m)}(\mathbf{x}_i))$$

由于 $\mathbf{f}^{(m-1)}(\mathbf{x}_i)$ 在第 m 轮为常数，记

$$w_i = \exp(-\frac{1}{K} \mathbf{y}_i^T \mathbf{f}^{(m-1)}(\mathbf{x}_i))$$

此时有

$$(\beta^{*(m)}, \mathbf{b}^{*(m)}) = \arg\min_{\beta^{(m)}, \mathbf{b}^{(m)}} \sum_{i=1}^{n} w_i \exp(-\frac{1}{K} \beta^{(m)} \mathbf{y}_i^T \mathbf{b}^{(m)}(\mathbf{x}_i))$$

SAMME

设当轮预测正确的样本索引集合为 T，则损失可表示为

$$\begin{aligned}\tilde{L}(\beta^{(m)}, \mathbf{b}^{(m)}) &= \sum_{i=1}^{n} w_i \exp(-\frac{1}{K}\beta^{(m)}\mathbf{y}_i^T\mathbf{b}^{(m)}(\mathbf{x}_i)) \\ &= \sum_{i\in T} w_i \exp[-\frac{\beta^m}{K-1}] + \sum_{i\notin T} w_i \exp[\frac{\beta^{(m)}}{(K-1)^2}] \\ &= \sum_{i\in T} w_i \exp[-\frac{\beta^m}{K-1}] + \sum_{i\notin T} w_i \exp[-\frac{\beta^m}{K-1}] - \\ &\quad \sum_{i\notin T} w_i \exp[-\frac{\beta^m}{K-1}] + \sum_{i\notin T} w_i \exp[\frac{\beta^{(m)}}{(K-1)^2}]\end{aligned}$$

SAMME

$$\tilde{L}(\beta^{(m)}, \mathbf{b}^{(m)}) = \exp[-\frac{\beta^{(m)}}{K-1}] \sum_{i=1}^{n} w_i + \{\exp[\frac{\beta^{(m)}}{(K-1)^2}] - \exp[-\frac{\beta^{(m)}}{K-1}]\} \sum_{i=1}^{n} w_i \mathbb{I}_{\{i \notin T\}}$$

注意到 $\mathbf{b}^{(m)}$ 仅与 $\sum_{i=1} w_i \mathbb{I}_{\{i \notin T\}}$ 有关（因为基学习器的好坏控制了样本是否能够正确预测），且此项前的系数非负（因为 $\beta^{(m)}$ 非负），因此得到

$$\mathbf{b}^{*(m)} = \arg\min_{\mathbf{b}^{(m)}} \sum_{i=1}^{n} w_i \mathbb{I}_{\{i \notin T\}}$$

SAMME

在得到 $\mathbf{b}^{*(m)}$ 后，通过求 $\tilde{L}$ 关于 $\beta^{(m)}$ 的导数并令之为 0 可解得

$$\beta^{*(m)} = \frac{(K-1)^2}{K}[\log\frac{1-err^{(m)}}{err^{(m)}} + \log(K-1)]$$

其中，样本的加权错误率为

$$err^{(m)} = \sum_{i=1}^{n} \frac{w_i}{\sum_{i=1}^{n} w_i} \mathbb{I}_{\{i \notin T\}}$$

样本 $\mathbf{x}_i$ 在第 m 轮的预测类别为 $k_i^* = \arg\max_k \mathbf{f}^{(m)}(\mathbf{x}_i)$，其中

$$\mathbf{f}^{(m)}(\mathbf{x}_i) = \mathbf{f}^{(m-1)}(\mathbf{x}_i) + \beta^{*(m)}\mathbf{b}^{*(m)}(\mathbf{x}_i)$$

SAMME

Algorithm 1: Adaboost 方法的 SAMME 实现

Data: 训练样本 $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_n)$ 和 $\mathbf{y} = (\mathbf{y}_1, ..., \mathbf{y}_n)$、基分类器 G、迭代轮数 M、测试样本 $\mathbf{x}$

Result: 测试样本的预测类别 $c(\mathbf{x})$

for $i \leftarrow 1$ **to** n **do**
  $w_i \leftarrow \frac{1}{n}$
end
for $m \leftarrow 1$ **to** M **do**
  $G^* \leftarrow \arg\min_G \sum_{i=1}^n w_i \mathbb{I}_{\{i \notin T\}}$
  $err^{(m)} \leftarrow \sum_{i=1}^n \frac{w_i}{\sum_{i=1}^n w_i} \mathbb{I}_{\{i \notin T\}}$
  $\beta^{*(m)} \leftarrow \frac{(K-1)^2}{K}[\log \frac{1-err^{(m)}}{err^{(m)}} + \log(K-1)]$
  for $i \leftarrow 1$ **to** n **do**
    $\mathbf{b}^{*(m)}(\mathbf{x}_i) \leftarrow G^*(\mathbf{x}_i)$
    $w_i \leftarrow w_i \cdot \exp(-\frac{1}{K}\beta^{*(m)} \mathbf{y}_i^T \mathbf{b}^{*(m)}(\mathbf{x}_i))$
  end
  $\mathbf{f}^{(m)} \leftarrow \mathbf{f}^{(m-1)} + \beta^{*(m)} \mathbf{b}^{*(m)}$
end
$c(\mathbf{x}) \leftarrow \arg\max_k \mathbf{f}^{(M)}(\mathbf{x})$

SAMME

事实上，我们还能通过一些多分类的性质来改写算法的局部实现，使得一些变量前的系数得到简化。记

$$\alpha^{*(m)} = \log \frac{1 - err^{(m)}}{err^{(m)}} + \log(K - 1)$$

此时，w_i 每轮会被更新为

$$\tilde{w}_i = w_i \cdot \exp[\frac{1 - K}{K} \alpha^{*(m)}] \exp(\alpha^{*(m)} \mathbb{1}_{\{i \notin T\}})$$

SAMME

对 $\mathbf{w}$ 进行归一化操作后，不会对下一轮算法1中 G^* 和 $err^{(m)}$ 的结果产生任何影响。同时，如果把算法1第 12 行的 $\beta^{*(m)}$ 替换为 $\alpha^{*(m)}$，由于它们的输出结果只相差常数倍 $\frac{(K-1)^2}{K}$，因此最后的预测结果 $c(\mathbf{x})$ 也不会产生任何变化。

由于 $\exp[\frac{1-K}{K}\alpha^{*(m)}]$ 是样本公共项，故我们可以每次都利用

$$\tilde{w}_i = w_i \cdot \exp(\alpha^{*(m)} \mathbb{1}_{\{i \notin T\}})$$

来更新，而不影响归一化结果。

SAMME

此时，算法1的迭代循环可进行如下重写：

Algorithm 2: SAMME 算法迭代循环的优化实现

for $m \leftarrow 1$ **to** M **do**
  $G^* \leftarrow \arg\min_G \sum_{i=1}^n w_i \mathbb{I}_{\{i \notin T\}}$
  $err^{(m)} \leftarrow \sum_{i=1}^n w_i \mathbb{I}_{\{i \notin T\}}$
  $\alpha^{*(m)} \leftarrow \log \frac{1-err^{(m)}}{err^{(m)}} + \log(K-1)$
  for $i \leftarrow 1$ **to** n **do**
    $\mathbf{b}^{*(m)}(\mathbf{x}_i) \leftarrow G^*(\mathbf{x}_i)$
    $\tilde{w}_i \leftarrow w_i \cdot \exp(\alpha^{*(m)} \mathbb{1}_{\{i \notin T\}})$
  end
  for $i \leftarrow 1$ **to** n **do**
    $w_i \leftarrow \frac{w_i}{\sum_{i=1}^n w_i}$
  end
  $\mathbf{f}^{(m)} \leftarrow \mathbf{f}^{(m-1)} + \alpha^{*(m)} \mathbf{b}^{*(m)}$
end

SAMME.R

许多分类器都能够输出预测样本所属某一类别的概率，但是 SAMME 算法只能利用分类的标签信息，而不能利用这样的概率信息。SAMME.R 算法通过损失近似的思想，将加权分类模型的概率输出信息与 boosting 方法相结合。SAMME.R 中的字母“R”代表“Real”，意味着模型每轮迭代的输出为实数。

SAMME.R

不同于 SAMME 在第 m 轮需要同时考虑得到最优的 $\beta^{(m)}$ 和 $\mathbf{b}^{(m)}$，SAMME.R 将其统一为 $\mathbf{h}^{(m)} \in \mathbb{R}^K$，它需要满足对称约束条件 $\sum_{i=1}^{K} h_k = 0$ 以保证可估性。此时，损失函数为

$$L(\mathbf{h}^{(m)}) = \exp[-\frac{1}{K}\mathbf{y}^T(\mathbf{f}^{(m-1)}(\mathbf{x}) + \mathbf{h}^{(m)}(\mathbf{x}))]$$

为了与概率联系，我们需对损失 L 的后验概率进行最小化，即

$$\begin{aligned}\mathbf{h}^{*(m)} &= \arg\min_{\mathbf{h}^{(m)}} \mathbb{E}[L|\mathbf{x}] \\ &= \arg\min_{\mathbf{h}^{(m)}} \mathbb{E}_{\mathbf{y}}[\exp[-\frac{1}{K}\mathbf{y}^T(\mathbf{f}^{(m-1)}(\mathbf{x}) + \mathbf{h}^{(m)}(\mathbf{x}))]|\mathbf{x}]\end{aligned}$$

SAMME.R

设样本 $\mathbf{y}$ 对应的标签为 $S(\mathbf{y})$，则

$$\begin{aligned}\mathbb{E}[L|\mathbf{x}] &= \mathbb{E}_{\mathbf{y}}[\exp[-\frac{1}{K}\mathbf{y}^T\mathbf{f}^{(m-1)}(\mathbf{x})]\exp[-\frac{1}{K}\mathbf{y}^T\mathbf{h}^{(m)}(\mathbf{x})]]|\mathbf{x}] \\ &= \sum_{k=1}^{K}[\exp[-\frac{1}{K}\mathbf{y}^T\mathbf{f}^{(m-1)}(\mathbf{x})]\exp[-\frac{1}{K}\mathbf{y}^T\mathbf{h}^{(m)}(\mathbf{x})]]\Big|_{S(\mathbf{y})=k} P(S(\mathbf{y})=k|\mathbf{x}) \\ &= \sum_{k=1}^{K}[\exp[-\frac{1}{K}\mathbf{y}^T\mathbf{f}^{(m-1)}(\mathbf{x})]\Big|_{S(\mathbf{y})=k} P(S(\mathbf{y})=k|\mathbf{x})]\exp(-\frac{h_k^{(m)}(\mathbf{x})}{K-1})\end{aligned}$$

记 $w=\exp[-\frac{1}{K}\mathbf{y}^T\mathbf{f}^{(m-1)}(\mathbf{x})]$，则

$$\mathbb{E}[L|\mathbf{x}] = \sum_{k=1}^{K} w|_{S(\mathbf{y})=k}\cdot P(S(\mathbf{y})=k)\exp(-\frac{h_k^{(m)}(\mathbf{x})}{K-1})$$

SAMME.R

不难发现对于样本 $\mathbf{y}$ 而言，越大的 w 意味着上一轮的模型结果越糟糕，此时负责预测 $P(S(\mathbf{y}) = k)$ 的基模型就要加大对该样本的重视程度以获得较小的损失。

但是，此时基模型本身是不带权重的，SAMME.R 采用的近似方法是，考虑以 w 为权重的基模型 G，用其输出 $P_w(s(\mathbf{y}) = k|\mathbf{x})$ 的概率值来代替 $w|_{S(\mathbf{y})=k} \cdot P(S(\mathbf{y}) = k|\mathbf{x})$，这种行为合法的原因在于权重对于总体损失的惩罚方向是一致的，G 通过权重 w 将原本作用于 L 的损失近似地“分配”给了基分类器的损失。

SAMME.R

此时，损失函数近似为

$$\mathbb{E}[L|\mathbf{x}] = \sum_{k=1}^{K} P_w(s(\mathbf{y}) = k|\mathbf{x}) \exp(-\frac{h_k^{(m)}(\mathbf{x})}{K-1})$$

由对称约束条件，结合拉格朗日乘子法可得

$$h_{k'}^{*(m)} = (K-1)[\log P_w(S(\mathbf{y}) = k'|\mathbf{x}) - \frac{1}{K}\sum_{k=1}^{K} \log P(S(\mathbf{y}) = k|\mathbf{x})]$$

SAMME.R

Algorithm 3: Adaboost 方法的 SAMME.R 实现（输入和输出同 SAMME）

```
for i ← 1 to n do
    w_i ← 1/n
end
for m ← 1 to M do
    G* ← 以 w 为权重训练的基模型
    for i ← 1 to n do
        for k ← 1 to K do
            P_k^(m)(x_i) ← P_w(S(y_i) = k|x)
        end
    end
    for i ← 1 to n do
        for k' ← 1 to K do
            h_k'^(m)(x_i) ← (K − 1)[log P_k'^(m)(x) − (1/K) Σ_{k=1}^K log P_k^(m)((x))]
            w_i ← w_i · exp(−((K−1)/K) y_i^T [log P_1^(m), ..., log P_K^(m)])
        end
    end
    for i ← 1 to n do
        w_i ← w_i / Σ_{i=1}^n w_i
    end
end
c(x) ← arg max_k Σ_{m=1}^M h_k^(m)(x)
```

Adaboost.R2

利用权重重分配的思想，Adaboost 还可以应用于处理回归问题。其中，Adaboost.R2 算法是一种最常使用的实现。

设训练集特征和目标分别为 $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_n)$ 和 $\mathbf{y} = (y_1, ..., y_n)$，权重 $\mathbf{w}$ 初始化为 $(w_1, ..., w_n)$。在第 m 轮时，根据权重训练基预测器得到 G^*，计算每个样本的相对误差

$$e_i = \frac{|y_i - G^*(\mathbf{x}_i)|}{\max_i |y_i - G^*(\mathbf{x}_i)|}$$

设样本的加权相对误差率为 $E^{(m)} = \sum_{i=1}^{n} w_i e_i$，则相对误差率与正确率的比值为 $\beta^{(m)} = \frac{E^{(m)}}{1-E^{(m)}}$，即预测器权重 $\alpha^{(m)} = \log \frac{1}{\beta^{(m)}}$。

Adaboost.R2

更新权重 w_i 为 $w_i[\alpha^{(m)}]^{1-e_i}$，权重在归一化后进入下一轮训练，由此可如下写出训练算法：

Algorithm 4: Adaboost.R2 算法的训练流程

for $i \leftarrow 1$ **to** n **do**
  $w_i \leftarrow \frac{1}{n}$
end
for $m \leftarrow 1$ **to** M **do**
  for $i \leftarrow 1$ **to** n **do**
    $e_i \leftarrow \frac{|y_i - G^*(\mathbf{x}_i)|}{\max_i |y_i - G^*(\mathbf{x}_i)|}$
  end
  $E^{(m)} \leftarrow \sum_{i=1}^{n} w_i e_i$
  $\beta^{(m)} \leftarrow \frac{E^{(m)}}{1-E^{(m)}}$
  $\alpha^{(m)} \leftarrow \log \frac{1}{\beta^{(m)}}$
  $w_i \leftarrow w_i[\alpha^{(m)}]^{1-e_i}$
  for $i \leftarrow 1$ **to** n **do**
    $w_i \leftarrow \frac{w_i}{\sum_{i=1}^{n} w_i}$
  end
end

Adaboost.R2

在预测阶段，Adaboost.R2 使用的是加权中位数算法。设每个基模型对某一个新测试样本的预测输出为 $y_1, ..., y_M$，基模型对应的预测器权重为 $\alpha^{(1)}, ..., \alpha^{(M)}$，则 Adaboost.R2 的输出值为

$$y = \inf\{y | \sum_{m \in \{m | y_m \leq y\}} \alpha^{(m)} \geq 0.5 \sum_{m=1}^{M} \alpha^{(m)}\}$$

习题

1. 假设有一个 3 分类问题，标签类别为第 2 类，模型输出的类别标签为 [-0.1,-0.3,0.4]，请计算对应的指数损失。
2. 什么是贝叶斯最优决策条件？请针对 Adaboost 的分类问题，给出一个不符合贝叶斯最优决策条件的损失函数。
3. 对于二分类问题，请完成以下任务：
   - 对公式进行化简，写出 $K = 2$ 时的 SAMME 算法流程，并与李航《统计学习方法》一书中所述的 Adaboost 二分类算法对比是否一致。
   - 在 sklearn 源码中找出算法流程中每一行对应的处理代码。

习题

4. 算法2第 12 行中给出了 $\mathbf{f}$ 输出的迭代方案，但在 sklearn 包的实现中使用了 $\mathbb{I}_{\{G^*(\mathbf{x})=S(\mathbf{y})\}}$ 来代替 $\mathbf{b}^{*(m)}(\mathbf{x})$。请根据本文的实现，对 sklearn 包的源码进行修改并构造一个例子来比较它们的输出是否会不同。（提示：修改 AdaboostClassifier 类中的 decision_function 函数和 staged_decision_function 函数）
5. 证明 SAMME.R 算法中 $h^*_{k'}$ 的求解结果。
6. 算法3的第 14 行给出了 w_i 的更新策略，请说明其合理性。
7. 请实现 SAMME、SAMME.R 和 Adaboost.R2 算法。

习题

8. 请结合加权中位数的定义解决以下问题：
   - 当满足什么条件时，Adaboost.R2 的输出结果恰为每个基预测器输出值的中位数？
   - Adaboost.R2 模型对测试样本的预测输出值是否一定会属于 M 个分类器中的一个输出结果？若是请说明理由，若不一定请给出反例。
   - 设 $k \in \{y_1, ..., y_M\}$，记 k 两侧（即大于或小于 k）的样本集合对应的权重集合为 W^+ 和 W^-，请证明使得这两个集合的元素之和差值最小的 k 就是 Adaboost.R2 的输出值 y。
   - 设 $y' = \sup\{y' \big| \sum_{m \in \{m|y_m \leq y'\}} \alpha^{(m)} \leq 0.5 \sum_{m=1}^{M} \alpha^{(m)}\}$，它和定义中给出的 y 值一定相等吗？若不一定请举出反例。
   - 相对于普通中位数，加权中位数的输出结果鲁棒性更强，请结合公式说明理由。