AdaBoost 算法

张腾

2022 年 6 月 16 日

指数损失

设特征空间 $\mathcal{X} \subseteq \mathbb{R}^d$，类别标记集合 $\mathcal{Y} = \{\pm 1\}$，对二分类模型 h，其泛化错误率为

$$\begin{aligned} R(h) &= \mathbb{E}_{(\boldsymbol{x},y)}[\mathbb{I}(y \neq \operatorname{sign}(h)] = \int \sum_{y \in \mathcal{Y}} \mathbb{I}(y \neq \operatorname{sign}(h))\mathbb{P}(\boldsymbol{x}, y)\mathrm{d}\boldsymbol{x} = \mathbb{E}_{\boldsymbol{x}}\left[\sum_{y \in \mathcal{Y}} \mathbb{I}(y \neq \operatorname{sign}(h))\mathbb{P}(y|\boldsymbol{x})\right] \\ &= \mathbb{E}_{\boldsymbol{x}}[1 - \mathbb{P}(\operatorname{sign}(h)|\boldsymbol{x})] \end{aligned}$$

故对 $\forall \boldsymbol{x}$，贝叶斯最优分类器

$$\operatorname{sign}(h) = \operatorname*{arg\,max}_{y \in \mathcal{Y}} \mathbb{P}(y|\boldsymbol{x}) = \begin{cases} +1, & \text{若 } \mathbb{P}(y = +1|\boldsymbol{x}) > \mathbb{P}(y = -1|\boldsymbol{x}) \\ -1, & \text{其它} \end{cases}$$

现将 0-1 错误率替换为指数损失 $\exp(-yh(\boldsymbol{x}))$，即最小化

$$\sum_{y \in \mathcal{Y}} \exp(-yh(\boldsymbol{x}))\mathbb{P}(y|\boldsymbol{x}) = \exp(-h(\boldsymbol{x}))\mathbb{P}(y = +1|\boldsymbol{x}) + \exp(h(\boldsymbol{x}))\mathbb{P}(y = -1|\boldsymbol{x})$$

令关于 $h(\boldsymbol{x})$ 的梯度 $-\exp(-h(\boldsymbol{x}))\mathbb{P}(y = +1|\boldsymbol{x}) + \exp(h(\boldsymbol{x}))\mathbb{P}(y = -1|\boldsymbol{x}) = 0$ 可得

$$\begin{aligned} \exp(h(\boldsymbol{x})) = \sqrt{\frac{\mathbb{P}(y = +1|\boldsymbol{x})}{\mathbb{P}(y = -1|\boldsymbol{x})}} &\Longrightarrow h(\boldsymbol{x}) = \frac{1}{2} \ln \frac{\mathbb{P}(y = +1|\boldsymbol{x})}{\mathbb{P}(y = -1|\boldsymbol{x})} \\ &\Longrightarrow \operatorname{sign}(h) = \begin{cases} +1, & \text{若 } \mathbb{P}(y = +1|\boldsymbol{x}) > \mathbb{P}(y = -1|\boldsymbol{x}) \\ -1, & \text{其它} \end{cases} \end{aligned}$$

这表明若 h 最小化指数损失，则分类器 $\operatorname{sign}(h)$ 可达到贝叶斯最优错误率，取指数损失作为 0-1 错误率的替代损失是合理的。

AdaBoost 算法

Boosting 是一族可将弱学习器提升为强学习器的算法，弱学习器是指泛化性能略优于随机猜测的学习器，例如在二分类问题上精度略高于 50% 的分类器。

该族算法的工作机制是先在初始训练集上训练出一个基学习器 (base learner)，再根据基学习器的表现对训练样本的权重分布进行调整，使得先前基学习器预测错的样本在后续受到更多关注，然后基于调整后的权重分布训练下一个基学习器；如此重复进行，直至基学习器数目达到事先指定的值 T，最终将这 T 个基学习器进行加权线性组合。AdaBoost [1] 就是该族算法的著名代表。

设训练数据集 $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}_{i\in[m]}$，基学习器 $h_t : \mathcal{X} \mapsto \mathcal{Y}$，其加权线性组合为

$$H_T(\boldsymbol{x}) = \sum_{t\in[T]} \alpha_t h_t(\boldsymbol{x})$$

在算法的第 t 轮，当前的分类器组合为 $H_t(\boldsymbol{x}) = H_{t-1}(\boldsymbol{x}) + \alpha_t h_t(\boldsymbol{x})$，AdaBoost 最小化指数损失

$$\begin{aligned}\sum_{i\in[m]} \exp(-y_i H_t(\boldsymbol{x}_i)) &= \sum_{i\in[m]} \exp(-y_i H_{t-1}(\boldsymbol{x}_i) - y_i\alpha_t h_t(\boldsymbol{x}_i)) \\ &= \sum_{i\in[m]} \exp(-y_i H_{t-1}(\boldsymbol{x}_i))\exp(-y_i\alpha_t h_t(\boldsymbol{x}_i)) \\ &\propto \sum_{i\in[m]} \mathcal{D}_t(i)\exp(-y_i\alpha_t h_t(\boldsymbol{x}_i))\end{aligned}$$

其中 $\mathcal{D}_t(i) \propto \exp(-y_i H_{t-1}(\boldsymbol{x}_i))$ 是上一轮的分类器组合 H_{t-1} 在样本 $(\boldsymbol{x}_i, y_i)$ 上的归一化指数损失 (在训练数据集上构成一个分布)，亦是本轮样本 $(\boldsymbol{x}_i, y_i)$ 的权重。注意 $y_i, h_t(\boldsymbol{x}_i) \in \{\pm 1\}$，进一步化简有

$$\begin{aligned}\sum_{i\in[m]} \exp(-y_i H_t(\boldsymbol{x}_i)) \propto \sum_{i\in[m]} \mathcal{D}_t(i)\exp(-y_i\alpha_t h_t(\boldsymbol{x}_i)) \\ &= \exp(-\alpha_t)\sum_{i\in[m]} \mathcal{D}_t(i)\mathbb{I}(y_i = h_t(\boldsymbol{x}_i)) + \exp(\alpha_t)\sum_{i\in[m]} \mathcal{D}_t(i)\mathbb{I}(y_i \neq h_t(\boldsymbol{x}_i)) \\ &= \exp(-\alpha_t)\sum_{i\in[m]} \mathcal{D}_t(i) + (\exp(\alpha_t) - \exp(-\alpha_t))\sum_{i\in[m]} \mathcal{D}_t(i)\mathbb{I}(y_i \neq h_t(\boldsymbol{x}_i)) \\ &= \exp(-\alpha_t) + (\exp(\alpha_t) - \exp(-\alpha_t))\sum_{i\in[m]} \mathcal{D}_t(i)\mathbb{I}(y_i \neq h_t(\boldsymbol{x}_i))\end{aligned}$$

其中最后一个等号是因为 $\mathcal{D}_t$ 是一个分布。注意第一项与 h_t 无关，故

$$h_t = \operatorname*{arg\,min}_h \sum_{i\in[m]} \mathcal{D}_t(i)\mathbb{I}(y_i \neq h(\boldsymbol{x}_i))$$

即基分类器 h_t 的选取应最小化加权错误率，记 $\epsilon_t = \sum_{i\in[m]} \mathcal{D}_t(i)\mathbb{I}(y_i \neq h_t(\boldsymbol{x}_i))$，令关于 α_t 的梯度

$$-\exp(-\alpha_t) + (\exp(\alpha_t) + \exp(-\alpha_t))\epsilon_t = 0$$

可得

$$\exp(2\alpha_t) = \frac{1}{\epsilon_t} - 1 = \frac{1-\epsilon_t}{\epsilon_t} \Longrightarrow \alpha_t = \frac{1}{2}\ln\frac{1-\epsilon_t}{\epsilon_t}$$

得到 α_t 和 h_t 后，下一轮的样本权重

$$\mathcal{D}_{t+1}(i) \propto \exp(-y_i H_t(\boldsymbol{x}_i)) \propto \mathcal{D}_t(i)\exp(-y_i\alpha_t h_t(\boldsymbol{x}_i))$$

伪代码见算法 1。

Algorithm 1: AdaBoost 算法

输入: 训练数据集 $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}_{i \in [m]}$，基学习算法 $\mathcal{L}$，迭代轮数 T

$\mathcal{D}_1(i) \leftarrow {}^1/_m$; // 初始化权重分布
for $t \leftarrow 1$ **to** T **do**
  $h_t \leftarrow \mathcal{L}(\mathcal{S}, \mathcal{D}_t)$; // 在 $\mathcal{S}$ 上以权重 $\mathcal{D}_t$ 训练 h_t
  $\epsilon_t \leftarrow \mathbb{P}_{(\boldsymbol{x} \sim \mathcal{D}_t, y)} \mathbb{I}(h_t(\boldsymbol{x}) \neq y)$; // 计算加权错误率
  if $\epsilon_t > 0.5$ **then** break ; // 若基分类器比随机猜测还差则中止算法
  $\alpha_t \leftarrow \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$; // 计算 h_t 的权重系数
  for $i \leftarrow 1$ **to** m **do**
    if $h_t(\boldsymbol{x}_i) = y_i$ **then**
      $\mathcal{D}_t(i) \leftarrow \mathcal{D}_t(i) \exp(-\alpha_t)$;
    else
      $\mathcal{D}_t(i) \leftarrow \mathcal{D}_t(i) \exp(\alpha_t)$;
    end
  end
  $s \leftarrow \sum_{i \in [m]} \mathcal{D}_t(i)$;
  for $i \leftarrow 1$ **to** m **do** $\mathcal{D}_{t+1}(i) \leftarrow {}^{\mathcal{D}_t(i)}/_s$; // 归一化权重
end

输出: $\operatorname{sign}(\sum_{t \in [T]} \alpha_t h_t(\boldsymbol{x}))$

求解异或问题

设基学习器为决策树桩 (decision stump)，即只有一层的决策树，数据集

$$\mathcal{S} = \begin{Bmatrix} (\boldsymbol{x}_1 = (+1, 0),\ y_1 = +1) & (\boldsymbol{x}_2 = (-1, 0),\ y_2 = +1) \\ (\boldsymbol{x}_3 = (0, +1),\ y_3 = -1) & (\boldsymbol{x}_4 = (0, -1),\ y_4 = -1) \end{Bmatrix}$$

如图 1(a) 所示。由于决策树桩只有一层，即只能挑选两个特征其中之一并以某一阈值进行分裂，因此分界面为平行于坐标轴的直线，最多分对三个样本。

第 1 轮所有样本权重均为 ${}^1/_4$，因此分对任意三个样本即可，不妨设学到的决策树桩为

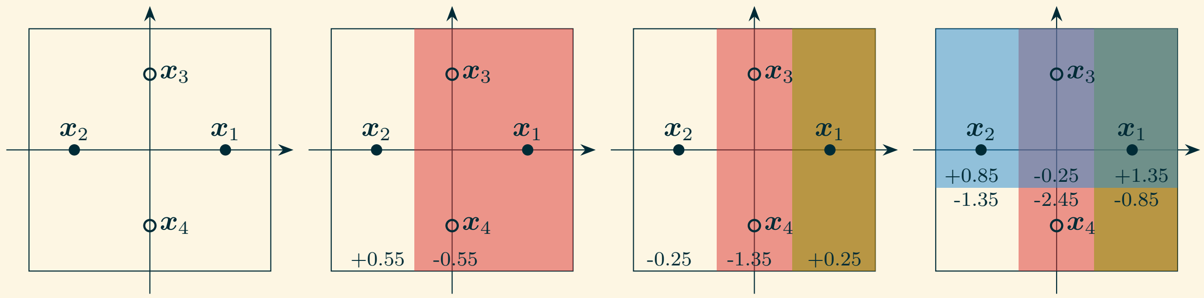
$$h_1(\boldsymbol{x}) = \begin{cases} -1, & \text{若 } x_1 > -0.5 \\ +1, & \text{其它} \end{cases}$$

如图 1(b) 所示，此时 $\boldsymbol{x}_1$ 被分错，$\boldsymbol{x}_2$、$\boldsymbol{x}_3$、$\boldsymbol{x}_4$ 被分对，故加权错误率和 h_1 的权重系数分别为

$$\epsilon_1 = \frac{1}{4}, \quad \alpha_1 = \frac{1}{2} \ln 3 = \ln \sqrt{3} \approx 0.55$$

权重分布更新

$$\left[\frac{1}{4}\exp(\ln\sqrt{3}),\ \frac{1}{4}\exp(-\ln\sqrt{3}),\ \frac{1}{4}\exp(-\ln\sqrt{3}),\ \frac{1}{4}\exp(-\ln\sqrt{3})\right] \xrightarrow{\text{归一化}} \left[\frac{1}{2},\ \frac{1}{6},\ \frac{1}{6},\ \frac{1}{6}\right]$$



(a) 异或数据　(b) 第一轮　(c) 第二轮　(d) 第三轮

图 1: 用 AdaBoost 求解异或问题

第 2 轮，$\boldsymbol{x}_1$ 权重最高，决策树桩分对三个样本且必须分对 $\boldsymbol{x}_1$，不妨设学到的决策树桩为

$$h_2(\boldsymbol{x}) = \begin{cases} +1, & \text{若 } x_1 > +0.5 \\ -1, & \text{其它} \end{cases}$$

如图 1(c) 所示，此时 $\boldsymbol{x}_2$ 被分错，$\boldsymbol{x}_1$、$\boldsymbol{x}_3$、$\boldsymbol{x}_4$ 被分对，故加权错误率和 h_2 的权重系数分别为

$$\epsilon_2 = \frac{1}{6}, \quad \alpha_2 = \frac{1}{2}\ln 5 = \ln\sqrt{5} \approx 0.80$$

权重分布更新

$$\left[\frac{1}{2}\exp(-\ln\sqrt{5}),\ \frac{1}{6}\exp(\ln\sqrt{5}),\ \frac{1}{6}\exp(-\ln\sqrt{5}),\ \frac{1}{6}\exp(-\ln\sqrt{5})\right] \xrightarrow{\text{归一化}} \left[\frac{3}{10},\ \frac{1}{2},\ \frac{1}{10},\ \frac{1}{10}\right]$$

第 3 轮，$\boldsymbol{x}_2$ 权重最高，$\boldsymbol{x}_1$ 次之，$\boldsymbol{x}_3$ 和 $\boldsymbol{x}_4$ 最低，因此本轮的决策树桩放弃 $\boldsymbol{x}_3$ 和 $\boldsymbol{x}_4$ 其中一个，分对其余三个即可，不妨设学到的决策树桩为

$$h_3(\boldsymbol{x}) = \begin{cases} +1, & \text{若 } x_2 > -0.5 \\ -1, & \text{其它} \end{cases}$$

如图 1(d) 所示，此时 $\boldsymbol{x}_3$ 被分错，$\boldsymbol{x}_1$、$\boldsymbol{x}_2$、$\boldsymbol{x}_4$ 被分对，故加权错误率和 h_3 的权重系数分别为

$$\epsilon_3 = \frac{1}{10}, \quad \alpha_3 = \frac{1}{2}\ln 9 = \ln 3 \approx 1.10$$

不难发现此时 $\operatorname{sign}(0.55 \cdot h_1 + 0.80 \cdot h_2 + 1.10 \cdot h_3)$ 已可将所有样本都分对。

参考文献

[1] Robert E. Schapire, Yoav Freund, Peter Barlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. In *Proceedings of the 14th International Conference on Machine Learning*, pages 322–330, Nashville, TN, 1997.