7 Boosting

Boosting is a meta-algorithm for improving the accuracy of a base learning procedure. In general, boosting can be viewed as a functional gradient descent algorithm for generating potentially complex parametric or nonparametric models through additive basis expansion. See Bühlmann and Hothorn (2007) for a comprehensive review of boosting algorithms.

AdaBoost (**Ada**ptive **Boost**ing) is the first practically feasible boosting algorithm proposed by Freund and Schapire (1997) in the context of classification. The AdaBoost algorithm is mainly discussed in this section although there are many different versions of boosting algorithms available in the machine learning literature for classification. Given a base classifier, boosting can be described as a procedure that combines the outputs of many base (or "weak") classifiers to form a powerful committee. Here a weak learner (base learner) means a classifier whose error rate is only slightly better than random guessing. For instance, the perceptron algorithm and stumps (a classification tree with two terminal nodes only) are weak learners. The boosting algorithm iteratively applies the weak learner to reweighted data to produce a sequence of classifiers $f_t(x), t = 1, 2, \ldots, T$ and take a weighted majority vote for the final prediction. For the training data, $\mathcal{D}_n = \{(x_i, y_i) : \ x_i \in \mathcal{X}, y_i \in \{-1, +1\}, i = 1, \ldots, n\}$, and a collection of ± 1 valued weak learners, $\mathcal{F}$, the AdaBoost algorithm is described as follows.

1. Initialize the data weights and the starting classifier:

$$D_1(i) = 1/n, \ i = 1, \ldots, n \ \text{ and } \ F_0(x) = 0.$$

2. For $t = 1, \ldots, T$,

   (a) Train a classifier using the distribution D_t, that is, choose $f_t \in \mathcal{F}$ minimizing the error rate

$$\epsilon_t = \sum_{i=1}^{n} D_t(i) I(f_t(x_i) \neq y_i).$$

   (b) Define $\alpha_t = \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon_t}$. Here $\alpha_t > 0$ if $\epsilon_t < 1/2$, which can be always assumed since if f_t has $\epsilon_t > 1/2$, then $-f_t$ has $\epsilon_t < 1/2$.

   (c) Update

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \exp\{-\alpha_t y_i f_t(x_i)\},$$

   where

$$Z_t = \sum_{i=1}^{n} D_t(i) \exp\{-\alpha_t y_i f_t(x_i)\}.$$

   That is,

$$D_{t+1}(i) \propto \begin{cases} D_t(i) e^{\alpha_t} & \text{if } y_i \neq f_t(x_i) \\ D_t(i) e^{-\alpha_t} & \text{if } y_i = f_t(x_i). \end{cases}$$

   Update $F_t(x) = F_{t-1}(x) + \alpha_t f_t(x)$.

Remark 14.

D_t**:** a distribution of weights at step t. Initial weights are $1/n$. The weights of incorrectly classified examples get increased so that the weak learner is forced to focus on difficult examples in the next step.

α_t**:** the weight assigned to f_t. α_t gets smaller as ϵ_t gets larger.

$F_T = \sum_{t=1}^{T} \alpha_t f_t$: a weighted majority vote of T weak learners.

$Z_t = 2\sqrt{\epsilon_t(1-\epsilon_t)}$ by simple algebra as shown below.

$$\begin{aligned}
Z_t &= \sum_{i=1}^{n} D_t(i) \exp\{-\alpha_t y_i f_t(x_i)\} \\
&= \sum_{i:y_i=f_t(x_i)} D_t(i) e^{-\alpha_t} + \sum_{i:y_i\neq f_t(x_i)} D_t(i) e^{\alpha_t} \\
&= e^{-\alpha_t} \sum_{i=1}^{n} I(f_t(x_i) = y_i) D_t(i) + e^{\alpha_t} \sum_{i=1}^{n} I(f_t(x_i) \neq y_i) D_t(i) \\
&= e^{-\alpha_t}(1-\epsilon_t) + e^{\alpha_t}\epsilon_t \\
&= \sqrt{\frac{\epsilon_t}{1-\epsilon_t}}(1-\epsilon_t) + \sqrt{\frac{1-\epsilon_t}{\epsilon_t}}\epsilon_t \\
&= 2\sqrt{\epsilon_t(1-\epsilon_t)}.
\end{aligned}$$

The sum of 'updated' weights for correctly classified examples by f_t is

$$\sum_{i:y_i=f_t(x_i)} D_{t+1}(i) = \frac{1}{Z_t} \sum_{i:y_i=f_t(x_i)} D_t(i) e^{-\alpha_t} = \frac{\sqrt{\epsilon_t(1-\epsilon_t)}}{Z_t} = \frac{1}{2}.$$

Therefore, $\sum_{i:y_i\neq f_t(x_i)} D_{t+1}(i) = 1/2$. Thus, the next classifier f_{t+1} may be almost independent of f_t as the weights are updated to make the new weighted problem maximally difficult for the current weak learner.

Theorem 21. *For the AdaBoost algorithm,* $R_n(F_T) \leq \prod_{t=1}^{T} 2\sqrt{\epsilon_t(1-\epsilon_t)}$.

Proof. Note that for any $\tau \in \mathbb{R}$, $I(\tau \leq 0) \leq e^{-\tau}$. Thus,

$$\begin{aligned}
R_n(F_T) &= \frac{1}{n}\sum_{i=1}^{n} I(y_i F_T(x_i) \leq 0) \leq \frac{1}{n}\sum_{i=1}^{n} \exp\{-y_i F_T(x_i)\} \\
&= \frac{1}{n}\sum_{i=1}^{n} \exp\{-y_i \sum_{t=1}^{T} \alpha_t f_t(x_i)\} = \frac{1}{n}\sum_{i=1}^{n}\prod_{t=1}^{T} \exp\{-\alpha_t y_i f_t(x_i)\} \\
&= \frac{1}{n}\sum_{i=1}^{n}\prod_{t=1}^{T} \frac{D_{t+1}(i)}{D_t(i)} Z_t = \frac{1}{n}\Big(\prod_{t=1}^{T} Z_t\Big)\sum_{i=1}^{n} \frac{D_{T+1}(i)}{D_1(i)} \\
&= \prod_{t=1}^{T} Z_t \sum_{i=1}^{n} D_{T+1}(i) = \prod_{t=1}^{T} Z_t = \prod_{t=1}^{T} 2\sqrt{\epsilon_t(1-\epsilon_t)}.
\end{aligned}$$

□

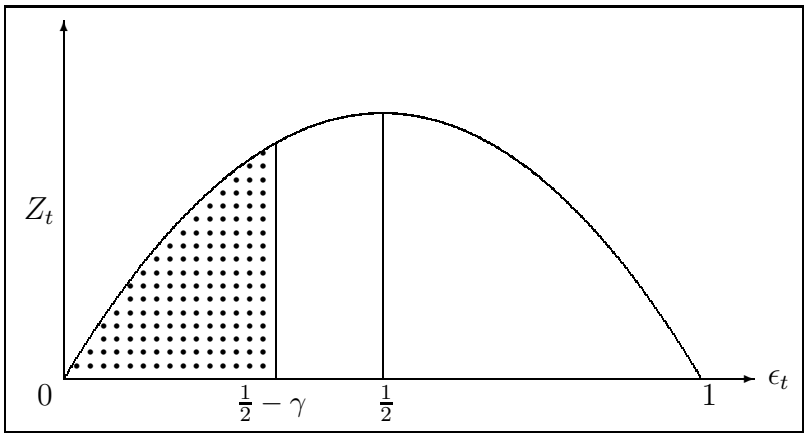

Figure 10: Z_t for small ϵ_t

Remark 15. If for some $\gamma > 0$, $\epsilon_t \leq 1/2 - \gamma$ for all t (see Figure 10), then

$$\begin{aligned}\prod_{t=1}^{T} 2\sqrt{\epsilon_t(1-\epsilon_t)} &\leq \prod_{t=1}^{T} 2\sqrt{\left(\frac{1}{2}-\gamma\right)\left(\frac{1}{2}+\gamma\right)} = \prod_{t=1}^{T} 2\sqrt{\frac{1}{4}-\gamma^2} = \prod_{t=1}^{T}\sqrt{1-4\gamma^2} \\ &\leq \prod_{t=1}^{T}\sqrt{e^{-4\gamma^2}} = e^{-2\gamma^2 T}.\end{aligned}$$

That is, if each weak learner is slightly better than random guess, the training error rate of the boosted procedure drops exponentially fast as the number of rounds, T, increases. So, the boosting algorithm can convert a weak learner into a strong learner.

The following theorem shows that boosting is, in fact, a gradient descent algorithm for additive modeling in the base learners.

Theorem 22. *The following two statements are equivalent:*

(a) Find f_t by minimizing ϵ_t and set $\alpha_t = \frac{1}{2}\log\frac{1-\epsilon_t}{\epsilon_t}$.

(b) Minimize

$$E_n \exp\{-Y F_t(X)\} = \frac{1}{n}\sum_{i=1}^{n}\exp\{-y_i(\alpha_t f_t(x_i) + F_{t-1}(x_i))\}$$

with respect to $\alpha_t > 0$ and f_t.

Proof. Re-expressing

$$\begin{aligned}\exp\{-y_i\alpha_t f_t(x_i)\} &= e^{-\alpha_t} I(y_i = f_t(x_i)) + e^{\alpha_t} I(y_i \neq f_t(x_i)) \\ &= e^{-\alpha_t} + (e^{\alpha_t} - e^{-\alpha_t}) I(y_i \neq f_t(x_i)),\end{aligned}$$

we have

$$\begin{aligned}
E_n \exp\{-YF_t(X)\} &= \frac{1}{n}\sum_{i=1}^{n} \exp\{-y_i(\alpha_t f_t(x_i) + F_{t-1}(x_i))\} \\
&= \frac{1}{n}[e^{-\alpha_t}\sum_{i=1}^{n} \exp\{-y_i F_{t-1}(x_i)\} \\
&\qquad + (e^{\alpha_t} - e^{-\alpha_t})\sum_{i=1}^{n} \exp\{-y_i F_{t-1}(x_i)\} I(y_i \neq f_t(x_i))] \\
&\leq e^{-\alpha_t}\prod_{u=1}^{t-1} Z_u + (e^{\alpha_t} - e^{-\alpha_t})\prod_{u=1}^{t-1} Z_u \sum_{i=1}^{n} D_t(i) I(y_i \neq f_t(x_i)) \\
&= e^{-\alpha_t}\prod_{u=1}^{t-1} Z_u + (e^{\alpha_t} - e^{-\alpha_t})\prod_{u=1}^{t-1} Z_u \epsilon_t.
\end{aligned}$$

Thus, f_t minimizing $E_n \exp\{-YF_t(X)\}$ is the f_t minimizing ϵ_t. Now with this ϵ_t fixed,

$$\frac{\partial}{\partial \alpha_t} E_n \exp\{-YF_t(X)\} = (-e^{-\alpha_t})\Big(\prod_{u=1}^{t-1} Z_u\Big) + (e^{\alpha_t} + e^{-\alpha_t})\Big(\prod_{u=1}^{t-1} Z_u\Big)\epsilon_t,$$

and

$$\frac{\partial^2}{\partial \alpha_t^2} E_n \exp\{-YF_t(X)\} > 0.$$

Hence,

$$\begin{aligned}
& \frac{\partial}{\partial \alpha_t} E_n \exp\{-YF_t(X)\} = 0 \\
\Rightarrow \qquad & -e^{-\alpha_t} + (e^{\alpha_t} + e^{-\alpha_t})\epsilon_t = 0 \qquad \Rightarrow \qquad e^{2\alpha_t} + 1 = \epsilon_t^{-1} \\
\Rightarrow \qquad & \alpha_t = \frac{1}{2}\log\frac{1-\epsilon_t}{\epsilon_t}.
\end{aligned}$$

□

The theorem suggests that boosting implicitly attempts to minimize the empirical risk with respect to the loss function given by

$$L(f(x), y) = \exp\{-yf(x)\}.$$

It is called the exponential loss function. The loss is another example of a margin-based loss, which is a convex surrogate of the 0-1 loss. It can be shown that the population minimizer of the exponential loss is

$$f_{exp}(x) = \frac{1}{2}\log\frac{\eta(x)}{1-\eta(x)},$$

and thus it is Fisher-consistent. Also, the population minimizer indicates a close link between boosting and logistic regression.