## AdaBoost

Recall that, in general, boosting is an additive stagewise model, i.e.,

$$f(x) \;=\; \sum_{m=1}^{M} \alpha_m \, T_m(x).$$

Where $T_m(x) = \{-1, +1\}$ and the true labels $y \in \{-1, +1\}$.

Our prediction $\hat{y}$ is defined as:

$$\hat{y} \;=\; \text{sign}\big(f(x)\big).$$

It's useful for us to define the *margin* as:

$$m(x) \;=\; y \,\cdot\, f(x).$$

**Claim:** An obs $(x, y)$ is classified correctly iff the margin is positive, $m(x) > 0$.

*Proof:* If $m(x) = y \cdot f(x) > 0$ then $y$ and $f(x)$ have the same sign. Then if $m(x) > 0 \implies \hat{y} = \text{sign}\big(f(x)\big) \implies \hat{y} = y$.

If $y\, f(x) < 0 \implies \hat{y}$ is the opposite sign of $y$. $\implies \hat{y} \neq y$.

**Things to notice:**

1. $f(x) = 0$ is the decision boundary.
2. The margin can be thought of as like a residual for binary classification.
3. Loss functions for binary classification problem can be written in terms of the margin.

**Claim:** Log loss for binary classification (starting from the $y^* = \{0, 1\}$ logistic regression setup) can be written as:

$$L\big(y, f(x)\big) \;=\; \log\big[\,1 + e^{-\,y\,f(x)}\big],$$

if we switch to the convention of taking $y \in \{-1, +1\}$ and the classification rule is $\hat{y} = \text{sign}\big(f(x)\big)$.

*Proof:* Consider the $y^* = \{0, 1\}$ logistic regression scenario. Recall $f(x)$ is just the linear predictor in this case: $f(x) = a\,x + b$, and thus the soft prediction is,

$$\hat{y}^* \;=\; \text{sigmoid}\big(f(x)\big) \;=\; \frac{1}{1 + e^{-\,f(x)}}.$$

The binary log-loss in general takes the form:

$$L\big(y^*, \hat{y}^*\big) \;=\; -\Big[y^* \,\log\big(\hat{y}^*\big) \;+\; (1-y^*)\,\log\big(1-\hat{y}^*\big)\Big].$$

Here's the main idea we use to simplify this expression:

$$y^* = 1 \implies y = 1, \text{ and } \quad y^* = 0 \implies y = -1.$$

So in each case,

$$L\big(y^*, \hat{y}^*\big) \;=\; \begin{cases} -\log\big(\hat{y}^*\big), & \text{if } y^* = 1 \iff y = 1, \\ -\log\big(1-\hat{y}^*\big), & \text{if } y^* = 0 \iff y = -1. \end{cases}$$

If $y = 1$,

$$-\log\big(\hat{y}^*\big) \;=\; -\log\Big[\big(1+e^{-\,f(x)}\big)^{-1}\Big] \;=\; \log\big[\,1+e^{-\,f(x)}\big].$$

And if $y = -1$,

$$-\log\big(1-\hat{y}^*\big) \;=\; -\log\Big(1 - \frac{1}{1+e^{-f(x)}}\Big) \;=\; -\log\Big(1 - \frac{e^{f(x)}}{e^{f(x)}+1}\Big) =$$

$$-\log\Big(\frac{e^{f(x)}+1-e^{f(x)}}{e^{f(x)}+1}\Big) \;=\; -\log\Big(\frac{1}{1+e^{f(x)}}\Big) \;=\; \log\big[\,1{+}e^{f(x)}\big] \;=\; \log\big[\,1{+}e^{-\,y\,f(x)}\big].$$

## Examples of Losses for Binary Classification

Log-loss $\Longrightarrow$

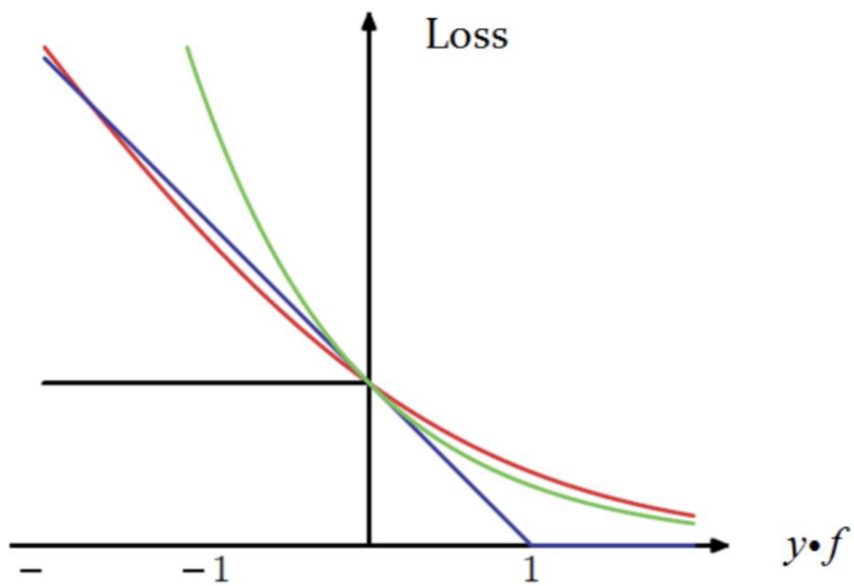
$$\ell\big(y, f(x)\big) \;=\; \log\big(1 + e^{-\,y\,f(x)}\big).$$

Exp. $\Longrightarrow$

$$L\big(y, f(x)\big) \;=\; e^{-\,y\,f(x)}.$$

Hinge $\Longrightarrow$

$$L^{H}\big(y, f(x)\big) \;=\; \max\big(0,\; 1 - y \cdot f(x)\big).$$




**Notice:**

- Loss functions penalize negative margins. How do each of these penalize? Look at the asymptotic behavior for negative margins:

$$\text{Exp.} \;\to\; e^{-m(x)} \quad \text{(exponential penalty)},$$

$$\text{LogLoss} \;\to\; \log\big(1 + e^{-\,m(x)}\big) \approx \text{linear for large negative margins},$$

$$\text{Hinge} \;\to\; \text{linear for negative margins}.$$

## What is AdaBoost doing?

AdaBoost is:

1. optimizing exponential loss, and
2. fitting additive models in steps.

Consider the additive model:

$$f(x) \;=\; \sum_{m=1}^{M} \beta_m \, T_m(x).$$

**Problem:** Find $\beta_1, \ldots, \beta_m$ and $T_1, \ldots, T_m$ such that

$$\frac{1}{n}\sum_{i=1}^{n} L\big(y_i,\; f(x_i)\big) \;=\; \frac{1}{n}\sum_{i=1}^{n} L\Big(y_i,\; \sum_{m=1}^{n} \beta_m \, T_m(x)\Big)$$

is minimized. Finding all trees at once is intractable, so we work in stages.

**Stage 1:** Find $\beta$ and $T$ such that

$$\frac{1}{n}\sum_{i=1}^{n} L\big(y_i,\; \beta \, T(x_i)\big) \;=\; \frac{1}{n}\sum_{i=1}^{n} e^{-\,y_i\,\beta\,T(x_i)} \quad \text{is minimized.}$$

First,

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n} L\big(y_i,\; \beta \, T(x_i)\big) &= \frac{1}{n}\sum_{i=1}^{n} e^{-\,y_i\,\beta\,T(x_i)} \\
&= \frac{1}{n}\Big( \sum_{\text{margin}=1} e^{-\beta} \;+\; \sum_{\text{margin}=-1} e^{\,\beta}\Big).
\end{aligned}$$

Define

$$\text{err} \;=\; \frac{\#\{\text{incorrectly classified points}\}}{n}.$$

and

$$Q(\beta) = \frac{1}{n}\sum_{i=1}^{n} L\big(y_i,\; \beta \, T(x_i)\big) \;=\; \text{err}\cdot e^{\,\beta} \;+\; (1-\text{err})\cdot e^{-\,\beta}.$$

Minimize with respect to $\beta$,

$$\frac{\mathrm{d}Q}{\mathrm{d}\beta} \;=\; \text{err}\, e^{\,\beta} \;-\; (1-\text{err})\, e^{-\,\beta} \;=\; 0.$$

$$\mathrm{err}\, e^{2\beta} \;=\; 1 - \mathrm{err}. \quad \Longrightarrow \quad e^{2\beta} \;=\; \frac{1-\mathrm{err}}{\mathrm{err}}.$$

$$2\,\beta \;=\; \log\!\Big(\tfrac{1-\mathrm{err}}{\mathrm{err}}\Big) \quad \Longrightarrow \quad \beta^* = \beta_1 \;=\; \tfrac{1}{2}\,\log\!\Big(\tfrac{1-\mathrm{err}}{\mathrm{err}}\Big).$$

Here $\beta_1$ is the "amount of say" that the first tree has!

**Stage $m$ for $m > 2$:**
At the $(m-1)^{th}$ stage, we have the current function:

$$f_{m-1}(x) \;=\; \sum_{j=1}^{m-1} \beta_j\, T_j(x).$$

We want to find $T_m$ such that

$$\frac{1}{n}\sum_{i=1}^{n} L(y_i\;\, f_{m-1}(x_i) \;+\; \beta_m\, T_m(x_i)) = \frac{1}{n}\sum_{i=1}^{n} e^{-\,y_i\,[\,f_{m-1}(x_i)\,+\,\beta_m\,T_m(x_i)\,]}$$

is minimized.
We can rewrite:

$$\frac{1}{n}\sum_{i=1}^{n} e^{-\,y_i\,[\,f_{m-1}(x_i)\,+\,\beta_m\,T_m(x_i)\,]} = \frac{1}{n}\sum_{i=1}^{n} e^{-\,y_i\,f_{m-1}(x_i)}\, e^{-\,y_i\,\beta_m\,T_m(x_i)} \;=\; \frac{1}{n}\sum_{i=1}^{n} w_i\, e^{-\,y_i\,\beta_m\,T_m(x_i)}$$

where

$$w_i \;=\; e^{-\,y_i\,f_{m-1}(x_i)}.$$

Expanding this expression:

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n} w_i\, e^{-\,y_i\,\beta_m\,T_m(x_i)} &= \frac{1}{n}\Big[\sum_{i:\,y_i=T_m(x_i)} w_i\, e^{-\,\beta_m} \;+\; \sum_{i:\,y_i\neq T_m(x_i)} w_i\, e^{\,\beta_m}\Big] \\
&= \frac{1}{n}\left[(e^{\beta} - e^{-\beta})\sum_{i=1}^{n} w_i \mathbf{1}[y_i \neq T_m(x_i)] + e^{-\beta}\sum_{i=1}^{n} w_i\right]
\end{aligned}$$

To minimize this, we want

$$\sum_{i=1}^{n} w_i\, \mathbf{1}[\,y_i \neq T_m(x_i)\,]$$

to be as small as possible. That is:

$$T_m^* = \arg\min_{T} \sum_{i=1}^{n} w_i\, \mathbf{1}[\,y_i \neq T(x_i)\,].$$

This is the tree that minimizes the weighted error.

The best $\beta_m$ is, as before,

$$\beta_m \;=\; \tfrac{1}{2}\,\log\Bigl(\tfrac{1-\mathrm{err}_m}{\mathrm{err}_m}\Bigr), \quad \text{where} \quad \mathrm{err}_m \;=\; \frac{\sum_{i=1}^n w_i\,\mathbf{1}\bigl[\,y_i \neq T_m(x_i)\,\bigr]}{\sum_{i=1}^n w_i}.$$

When we update our weights for the next tree, we do:

$$w_i \;\leftarrow\; w_i\,\exp\bigl[-\,\beta_m\,y_i\,T_m(x_i)\bigr] \;=\; \begin{cases} w_i\,e^{-\,\beta_m}, & \text{if } y_i = T_m(x_i),\\ w_i\,e^{\,\beta_m}, & \text{if } y_i \neq T_m(x_i). \end{cases}$$

## Showing equivalence to the $\alpha_m$ formulation:

We can redefine some quantities to show that this is totally equivalent to the $\alpha_m$ formulation. Define:

$$\alpha_m \;=\; 2\,\beta_m,$$

so that

$$\alpha_m = \log\Bigl(\frac{1-\mathrm{err}_m}{\mathrm{err}_m}\Bigr),$$

and redefine

$$w_i \;\leftarrow\; w_i\,\exp\bigl[\alpha_m\,\mathbf{1}(y_i \neq T_m(x_i))\bigr].$$

Then we're done.

**Big Takeaway:** Finally our fitted boosted tree looks like:

$$f(x) \;=\; \sum_{m=1}^M \beta_m\,T_m(x_i) \;=\; \tfrac{1}{2}\,\sum_{m=1}^M \alpha_m\,T_m(x_i) \quad\Longrightarrow\quad \hat{y} \;=\; \mathrm{sign}\bigl(f(x)\bigr).$$

and since at the end of the day we are only using the sign of $f(x)$ for the hard prediction, multiplying by 2 doesn't change the final prediction $\hat{y}$.