# AdaBoost by Hand

## $G_1(x)$

$X = \{0, 1, 2\}$

Best split?

temp

{hot, mild} | {cool}
---|---
No | Yes
$-1$ | $+1$
2 right | 1 right
2 wrong | 1 wrong

$err = 3/6 = 1/2$

$G_1(x)$

temp

{hot} | {mild, cold}
---|---
$-1$ | $+1$
2 right | 2 right
1 wrong | 1 wrong

$err = 2/6 = 1/3$

$G_1(x)$

$\alpha_1 = \log\left(\frac{2/3}{1/3}\right) = \log(2)$

## 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 $T_m(x) = \{-1, 1\}$ & the true labels $y \in \{-1, 1\}$.

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

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

It's useful for us to define the <u>margin</u> as:

$$m(x) = y \cdot f(x)$$

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

Pf: If $m(x) = y f(x) > 0$ then $y$ & $f(x)$ have the same sign. Then

If $m(x) > 0 \Rightarrow \hat{y} = \text{sign}(f(x)) \Rightarrow \hat{y} = y$

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

$\Rightarrow \hat{y} \neq y$

**Notes:**

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 bin. class. can be written in terms of the margin.

Claim: Log loss for bin. class can be written as:

$$\boxed{L(y, f(x)) = \log\left[1 + e^{-yf(x)}\right]}$$

if $y \in \{-1, +1\}$ & the classification rule

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

Pf: Consider the $y^* = \{0, 1\}$ log. reg.

$f(x)$ is the logit in this case:

$$f(x) = ax + b.$$

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

$$\therefore L(y^*, \hat{y}^*) = -\left[y^* \log \hat{y}^* + (1 - y^*) \log(1 - \hat{y}^*)\right]$$

Idea: $y^* = 1 \Rightarrow y = 1$

$y^* = 0 \Rightarrow y = -1$

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

If $y = 1$,

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

$$= \log[1+e^{-f(x)}] = \log[1+e^{-yf(x)}] \ *$$

If $y = -1$,

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

$$= -\log\left(1 - \frac{1}{1+e^{-f(x)}}\right)$$

$$= -\log\left(1 - \frac{e^{f(x)}}{e^{f(x)}+1}\right)$$

$$= -\log\left(\frac{e^{f(x)}+1-e^{f(x)}}{1+e^{f(x)}}\right)$$

$$= -\log([1+e^{f(x)}]^{-1}) = \log[1+e^{f(x)}]$$

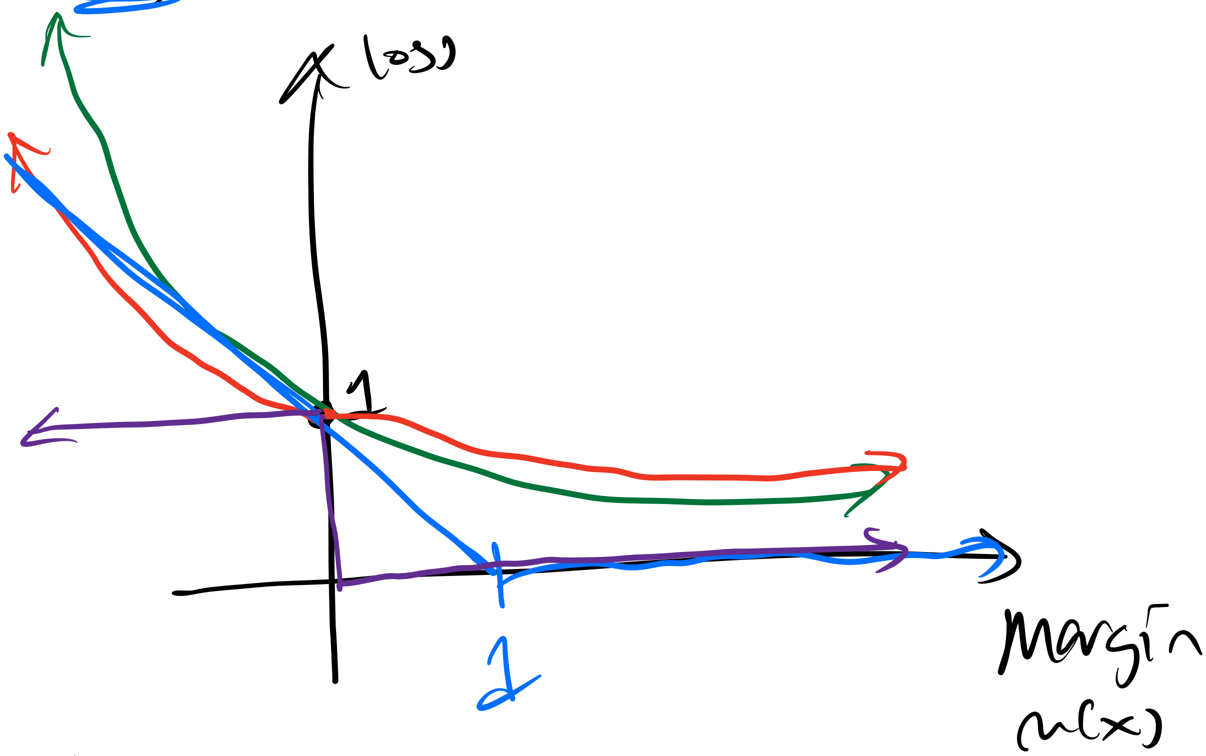
$$= \log[1+e^{-yf(x)}] \ *$$

EX: of Losses for Bin. Class.

LogLoss $\Rightarrow$ $\ell(y, f(x)) = \log(1 + e^{-yf(x)})$

Exp $\rightarrow$ $L(y, f(x)) = e^{-yf(x)}$

Hinge $\Rightarrow$ $L_H(y, f(x)) = \max(0, 1 - yf(x))$




Notes:

① Loss functions penalize negative margins:

Exp $\rightarrow$ $e^{-M(x)}$ = exponential penalty

logloss $\rightarrow$ $\log(1+e^{-M(x)}) \approx$ linear for large neg. margins

Hinge → linear for neg. margins.

---

What is AdaBoost doing?

AdaBoost is:

① optimizing exp. loss by

② fitting additive models in stages.

Consider the additive model:

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

Problem:

Find $\beta_1, \dots, \beta_M$ & $T_1, \dots, T_M$ such that

$$\frac{1}{N}\sum_{i=1}^{N} L(y_i, f(x_i))$$

$$= \frac{1}{N}\sum_{i=1}^{N} L\left(y_i, \sum_{m=1}^{M} \beta_m T_m(x)\right)$$

is minimized.

Finding all trees @ once is intractable.
we have to work in stages.

<u>Stage #1</u>:

Find $\beta$ & $T$ such that

$$\frac{1}{N}\sum_{i=1}^{N} L(y_i, \beta T(x_i)) \text{ is minimized.}$$

$$= \frac{1}{N}\sum_{i=1}^{N} e^{-y_i \beta T(x_i)}$$

$$= \frac{1}{N}\Bigg( \sum_{(\text{margin}=1)} e^{-\beta} + \sum_{(\text{margin}=-1)} e^{\beta} \Bigg)$$

$(\text{margin}=1) \Updownarrow x_i$ correct

$(\text{margin}=-1) \Updownarrow x_i$ incorrect

Define:

$$err = \frac{\#\ \text{incorrect class pts}}{N}$$

$$\frac{1}{N}\sum_{i=1}^{N} L = err \cdot e^{\beta} + (1-err)e^{-\beta} = Q(\beta)$$

$$\frac{\partial Q}{\partial \beta} = err \cdot e^{\beta} - (1-err)e^{-\beta} \overset{!}{=} 0$$

$$err\, e^{2\beta} - (1-err) = 0$$

$$err\, e^{2\beta} = 1-err$$

$$e^{2\beta} = \frac{1-err}{err}$$

$$\beta_1 = \frac{1}{2}\log\left(\frac{1-err}{err}\right) = \frac{1}{2}\alpha_1$$

$\beta_1$ = amount of say that this tree has!

**Stage # m** $(m \geq 2)$

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:

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

is minimized.

$$= \frac{1}{N}\sum_{i=1}^{N} \underbrace{e^{-y_i f_{m-1}(x_i)}}_{w_i}\, e^{-y_i \beta_m T_m(x_i)}$$

$$\left( w_i = e^{-y_i f_{m-1}(x_i)} \right)$$

$$= \frac{1}{N}\sum_{i=1}^{N} w_i e^{-y_i \beta_m T_m(x_i)}$$

$$= \frac{1}{N}\left( \sum_{(x_i:\, T_m(x_i)=y_i)} w_i e^{-\beta_m} + \sum_{(x_i:\, T_m(x_i)\neq y_i)} w_i e^{\beta} \right)$$

$$= \frac{1}{N}\left[ (e^{\beta}-e^{-\beta}) \left( \sum_{i=1}^{N} w_i \mathbb{1}(y_i \neq T_m(x_i)) \right) + \left( e^{-\beta} \sum_{i=1}^{N} w_i \right) \right]$$

Exercise

To minimize this, I want

to be as small as possible.

$$T_m^* = \arg\min_T \sum_{i=1}^{N} w_i \mathbb{1}(y_i \neq T(x_i))$$

This is the tree that minimizes the weighted error.

The best $\beta_m = \frac{1}{2} \log\left(\frac{1 - err_m}{err_m}\right)$

$$err_m = \frac{\sum_{i=1}^{N} w_i \mathbb{1}(y_i \neq T_m(x_i))}{\sum_{i=1}^{N} w_i}$$

---

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

$$w_i \leftarrow w_i \cdot e^{-\beta_m y_i T_m(x_i)} = \begin{cases} w_i e^{-\beta_m} & \text{if } y_i = T_m(x_i) \\ w_i e^{\beta_m} & \text{if } y_i \neq T_m(x_i) \end{cases}$$

Equivalence to the $\alpha_m$ formulation:

$$\alpha_m = 2\beta_m$$

$$\alpha_m = \log\left(\frac{1-err_m}{err_m}\right) \quad \&$$

$$w_i \leftarrow w_i e^{\alpha \mathbb{1}(y_i \neq T_m(x_i))}$$

Big Takeaway:

$$f(x) = \sum_{m=1}^{M} \beta_m T_m(x_i) = \frac{1}{2}\sum_{m=1}^{M} \alpha_m T_m(x_i)$$

$$\Rightarrow \hat{y} = \text{sign}(f(x))$$

Multiplying by 2 doesn't change my final prediction $\hat{y}$.