Lecture 24: Boosting

Loss Functions

Regression:

1. squared error loss

$$L(y, f(x)) = (\underbrace{y - f(x)}_{r})^2$$

2. absolute loss

$$L(y, f(x)) = |\underbrace{y - f(x)}_{r}|$$

Classification: (binary)

Two parameterizations for Y

1. $Y \in \{0, 1\}$ $\frac{(Y+1)}{2}$ 2. $Y \in \{-1, 1\}$

$2Y - 1$

losses: 0-1 loss

$$L(y, f(x)) = \mathbb{1}(y \neq f(x)) = \begin{cases} 0 & y = f(x) \\ 1 & y \neq f(x) \end{cases}$$

If I use $-1/1$ encoding then

$$y \in \{-1, 1\}, \quad f(x) \in \{-1, 1\}$$

correct classification = signs of $y, f(x)$ matching

incorrect " = " don't match

Also note for any classifier f there is some fn h so that

$$f(x) = \text{sign}(h(x))$$

like a disc. fn &

idea: $h(x) \gg 0$ if class 1
$h(x) \ll 0$ if class -1

example: linear classifier

$$f(x) = \text{sign}(w^T x)$$

margin: $yh(x)$

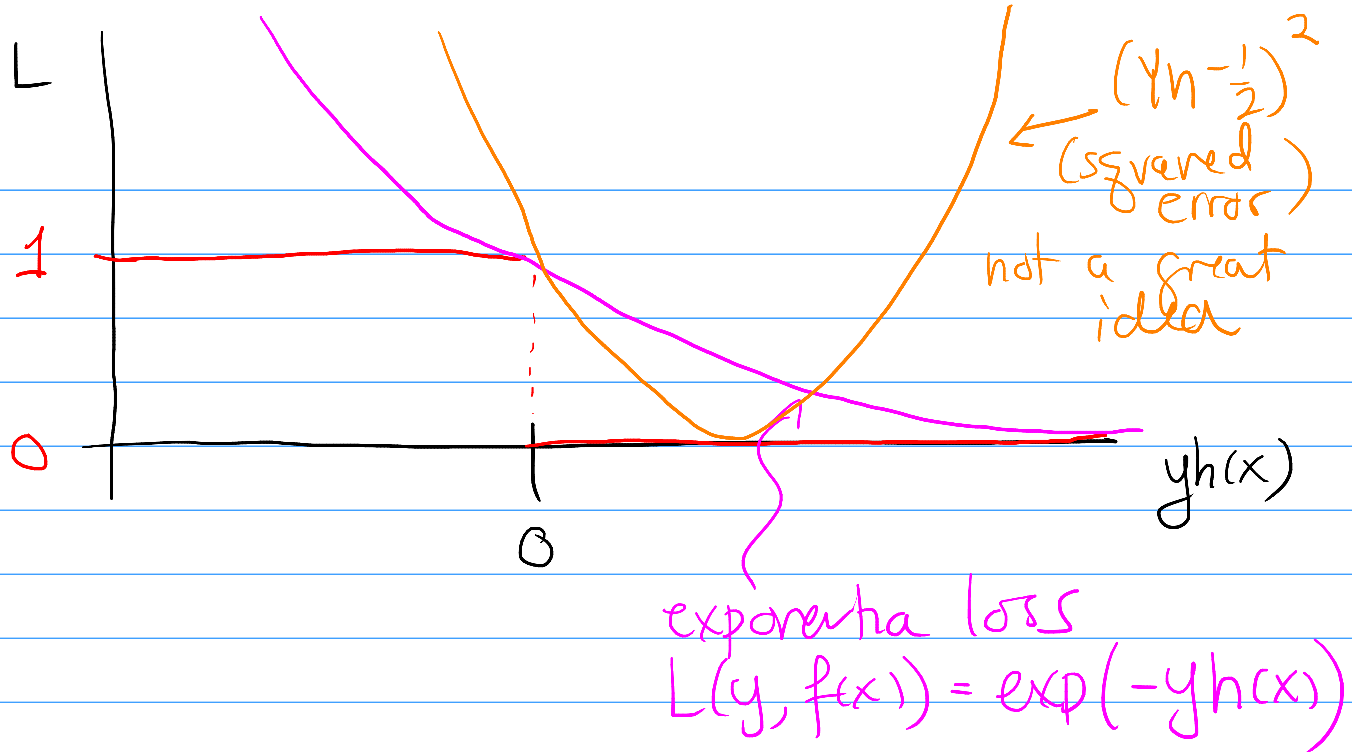
$yh(x) > 0$ correct classification

$yh(x) < 0$ incorrect "

$|yh(x)|$ = amount of correct/incorrect

We can write 0-1 loss as a fn of the margin

$$L(y, f(x)) = \mathbb{1}(yh(x) < 0)$$



Boosted Methods

ensemble

Orig. motivated as a method to combine a series of weak classifiers into a stronger one.

Weak classifier: one that is not much better than random chance

typically a "stump" tree w/ 1 split

Boosting:

1. sequentially learn weak classifiers on a series of modified traing data

$\hat{f}_1, \hat{f}_2, \hat{f}_3, \ldots$

use a weighted loss

each classifier will be modified (weighted) to focus on traing data where prev. class. was bad.

(2)

$$\hat{f}(x) = \text{Sign}\left(\overbrace{\sum_{b=1}^{B} \alpha_b \hat{f}_b(x)}^{h(x)}\right)$$

$\hat{f}_b(x)$ is ± 1

weighted a combn of individual f_b

weight α_b is higher for better classifiers and lower for worse.

Ada Boost

(1) $w_n = \frac{1}{N}$ = weight for n^{th} training pt.

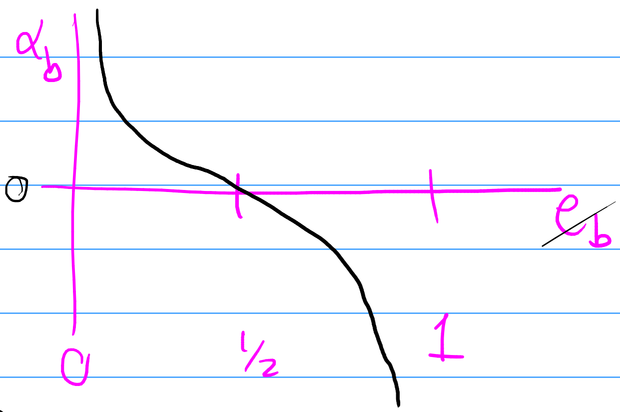
(2) For $b = 1, \ldots, B$

(a) fit $\hat{f}_b$ using weighted loss (w_n)

(b) Compute error for $\hat{f}_b$

e_b = weighted mis. class. err.

(c) $\alpha_b = \frac{1}{2} \log\left(\frac{1-e_b}{e_b}\right)$



(d) update w_n

$$w_n \longleftarrow \exp(\pm \alpha_b) w_n$$

If $\hat{f}_b$ got n^{th} point correct $e^{-\alpha_b} w_n$

" incorrect $e^{\alpha_b} w_n$

(2) $\hat{f}(x) = \text{Sign}\left(\sum_{b=1}^{B} \alpha_b \hat{f}_b(x)\right)$

[as prev.].

What is boosting doing?

General form:

$$\hat{f}(x) = \text{Sign}(h(x))$$
$$h(x) = \sum_{b=1}^{B} \alpha_b \hat{f}_b(x)$$

Generalized Additive method

$$h(x) = \sum_{b=1}^{B} \alpha_b \beta_b(x; \gamma_b)$$

γ_b = params for basis fn.

β_b is some collection of "basis" fns

want to find α_b, γ_b to fit data well

$$\underset{\{\alpha_b, \gamma_b\}_{b=1}^{B}}{\text{argmin}} \text{Loss}(h(x))$$

Lots of params! A problem.

Soln: Use a greedy forward "stagewise" add. modeling approach where do this for one pair α_b, γ_b at a time.

For $b = 1, \ldots, B$

(a) $\alpha_b, \gamma_b = \underset{\alpha, \gamma}{\operatorname{argmin}} \; \text{Loss}\left(\hat{f}_{b-1}(x) + \alpha \beta(x, \gamma)\right)$

(b) $f_b(x) = f_{b-1}(x) + \alpha_b \beta(x, \gamma_b)$

Punchline: Ada-Boost is basically Forward stagewise add. modeling where βs are "stumps" and we use an Exp. loss to measure error.