Lecture 19 -- Random Forests

Tuesday, November 12, 2024 3:29 PM

Random Forest

(1) Fit B (randomized) trees to bootstrap samples

For $b = 1, \dots, B$

(i) draw bootstrap sample S_b from traing data

(ii) Fit a (randomized) tree to S_b to get $\hat{f}_b$.

really overfit these

at each split in tree only consider a random sample of M variables

make $\hat{f}_b$s less collelated

(2) Bag my trees

Out-of-Bag Error (OOB)

Each traing sample x will end up in some of my S_b and not others.

not here

some of my S_b and not others.

not here

$$S_1 \quad S_2 \quad S_3 \quad \cdots \quad S_{10}$$

x in here

Consider bagging only those $\hat{f}_b$ where $x \notin S_b$:

$$\hat{f}_{-x} = \frac{1}{N_{-x}} \sum_{b : x \notin S_b} \hat{f}_b$$

N_{-x} = # bootstraps w/o x

As far as $\hat{f}_{-x}$ is concerned, x is a testing point.

So $\hat{y}_{OOB} = \hat{f}_{-x}(x)$

So If I calculate accuracy metrics based on $\{\hat{y}_{OOB,n}\}_{n=1}^{N}$

then this is basically a test error.

then this is basically a test error.

e.g. $MSE_{OOB} = \frac{1}{N} \sum_{n=1}^{N} (y_n - \hat{y}_{OOB,n})^2$.

Binary Classification

two parameterizations of Y

$$① \; Y = 0 \text{ or } 1 \quad \underset{\frac{Y+1}{2}}{\overset{2Y-1}{\rightleftarrows}} \quad ② \; Y = \pm 1$$

0-1 loss: $L(y, \hat{f}(x)) = \mathbb{1}(y \neq \hat{f}(x))$

If I use ± 1 parameterization then

correct class $\Leftrightarrow$ signs of $y, \hat{f}(x)$ match incorrect $\Leftrightarrow$ signs differ

For any $\hat{f}$ there is some function h so that

$$\hat{f}(x) = \text{Sign}(h(x))$$

<u>idea</u>: $h(x) \gg 0$ for class 1

$h(x) \ll 0$ for class -1

<u>margin</u>: $yh(x) = yh$

$yh > 0 \Leftrightarrow$ correct class

$yh < 0 \Leftrightarrow$ incorrect

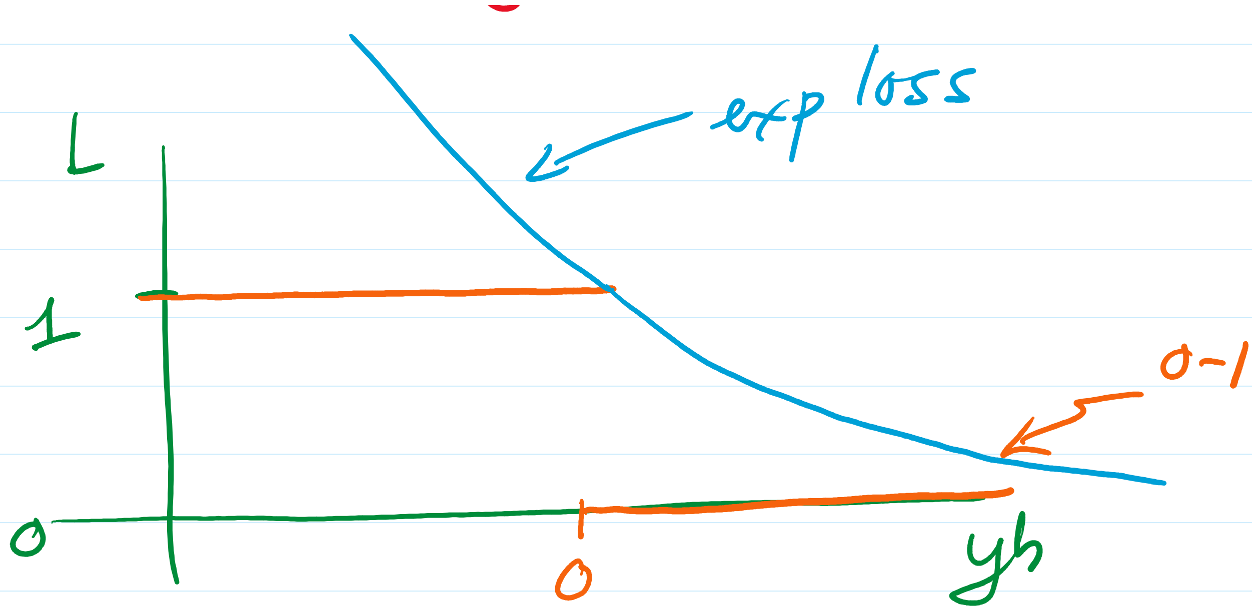
$|yh| \approx$ correctness

$\approx$ residual for regression

<u>0-1 loss</u> : $L(yh) = \mathbb{1}(yh < 0)$

<u>Exponential Loss</u> : $L(yh) = e^{-yh}$

loss



Boosting (Binary classification)

Orig. designed as way of combining a series of weak classifiers to make a stronger one.

one whose perf. isn't much better than guessing

Idea:

1. Sequentially train a series of weak classifiers (one-split tree)

weak classifiers (one-split tree = stump)

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

to repeatedly modified traing data.

→ up weight cases where prev. classifiers wrong

(2) Combine them via a weighted majority vote:

$$\hat{f}(x) = \text{sign}\Big(\underbrace{\sum_{m=1}^{M} \alpha_m \hat{f}_m(x)}_{h(x)}\Big)$$

weights reflect accuracy of each $\hat{f}_m$.

First/Simplest Boosting Algo AdaBoost:

(1) $w_n = 1/N$ ← weight for n^{th} traing point

(2) For $m = 1, \ldots, M$

(a) fit $\hat{f}_m$ by minimizing weighted loss L using weights w_n

loss L using weights w_n

(b) Compute weighted misclass err for $\hat{f}_m$:

$$\text{err}_m = \frac{\sum_n w_n \mathbb{1}(y_n \neq \hat{f}_m(x_n))}{\sum_n w_n}$$

(c) $\alpha_m = \log((1-\text{err}_m)/\text{err}_m)$

(d) Update weights:

$$w_n \leftarrow \underbrace{w_n \cdot \exp(\alpha_m \mathbb{1}(y_n \neq \hat{f}_m(x)))}$$

$= w_n$ if $\hat{f}_m$ is correct

$= e^{\alpha_m} w_n$ if $\hat{f}_m$ is incorrect

3 $\hat{f}(x) = \operatorname{Sign}\left(\sum_m \alpha_m \hat{f}_m(x)\right).$