Lecture 22: Boosting [classification]

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

one whose accuracy is only slightly better than guessing

Idea:

(1) sequentially train a series of weak classifiers

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

predict ± 1 (one-split tree = stump)

to repeatedly modified training data.

(2) combine them via a weighted majority vote

$$\hat{f}(x) = \text{Sign}\left(\sum_{m=1}^{M} \alpha_m \hat{f}_m(x)\right)$$

weights reflect the accuracy of each individual classifier

What do I mean by "modified" training data?

Going to re-weight training observations to focus training on pts I get wrong.

→ higher weight if $\hat{f}_{m-1}(x_n)$ is incorrect

→ lower weight if $\hat{f}_{m-1}(x_n)$ is correct.

First / simplest implementation : AdaBoost

1. $w_n = 1/N$ ← weight for n^{th} training pt
2. For $m = 1, \dots, M$
   a. fit $\hat{f}_m$ by minimizing weighted loss L using weights w_n
   b. compute weighted mis-class err rate
   $$err_m = \frac{\sum_n w_n \mathbb{1}(y_n \neq \hat{f}_m(x_n))}{\sum_n w_n}$$
   c. $$\alpha_m = \log\left((1 - err_m)/err_m\right) > 0$$
   d. update weights:
   $$w_n \leftarrow w_n \cdot \exp\left(\alpha_m \mathbb{1}(y_n \neq f_m(x_n))\right)$$
   $= w_n$ if f_m correct

   $= w_n e^{\alpha_m}$ if f_m incorrect

③ $\hat{f}(x) = \text{Sign}\Big(\underbrace{\sum_{m=1}^{M} \alpha_m \hat{f}_m(x)}^{h(x)}\Big)$

What boosting is doing?

Additive model:

$$h(x) = \sum_{m=1}^{M} \beta_m b(x; \gamma_m)$$

coef.

param for basis fn

basis functions

How do we fit? For some loss L

$$\{\hat{\beta}_m, \hat{\gamma}_m\} = \underset{\{\beta_m, \gamma_m\}}{\text{argmin}} \sum_n L\Big(y_n, \sum_m \beta_m b(x; \gamma_m)\Big)$$

Problem: can be difficult

→ lots of parameters

→ depending on L can be difficult

Soln: greedy approach

Forward Stagewise Additive Modeling

Do one at a time.

$\hat{G}_0(x) = 0$

For $m = 1, \dots, M$

(a) $\beta_m, \gamma_m = \underset{\beta, \gamma}{\operatorname{argmin}} \sum_n L(y_n, \hat{G}_{m-1}(x) + \beta b(x;\gamma))$

(b) $\hat{G}_m(x) = \hat{G}_{m-1}(x) + \beta_m b(x;\gamma_m)$

At end return $\hat{G}_m(x) = \sum_{m=1}^{M} \beta_m b(x;\gamma_m)$

Ex. regression

$$L(y, f(x)) = (y - f(x))^2$$

then $L(y_n, \hat{G}_{m-1}(x) + \beta b(x;\gamma))$

$$= (\underbrace{y_n - \hat{G}_{m-1}(x)}_{r_{nm}} - \beta b(x;\gamma))^2$$

$$= (r_{nm} - \beta b(x;\gamma))^2$$

So SAM basically fits sequentiall to residuals of prev. fit.

What does SAM have to do w/ boosting?

AdaBoost $\approx$ SAM w/ $L(y,h) = e^{-yh}$

exponential loss

That is at each AdaBoost is effectively solving

$$\beta_m, \hat{f}_m = \underset{\beta,\, f_m}{\operatorname{argmin}} \sum_n L\left(y_n, \hat{G}_{m-1}(x) + \beta f_m(x)\right)$$

exp. loss $:= e^{-yh}$

$$\underset{\beta,\, f_m}{\operatorname{argmin}} \sum_n \exp\left(-y_n\left[\hat{G}_{m-1}(x) + \beta f_m(x)\right]\right)$$

$$\hat{G}_{m-1}(x) = \sum_{i=1}^{m-1} \beta_m \hat{f}_m(x)$$

$$= \underset{\beta,\, f_m}{\operatorname{argmin}} \sum_n \underbrace{\exp\left(-y_n \hat{G}_{m-1}(x)\right)}_{\text{no } \beta,\, f_m} \exp\left(-\beta y_n f_m(x)\right)$$

regard as same weight w_{nm}

$$= \underset{\beta,\, f_m}{\operatorname{argmin}} \sum_n w_{nm} \exp\left(-\beta y_n f_m(x)\right)$$

weighted exp. loss

$w_{nm} L(y_n, \beta f_m(x))$