FYS-STK4155, Nov 10, 2022

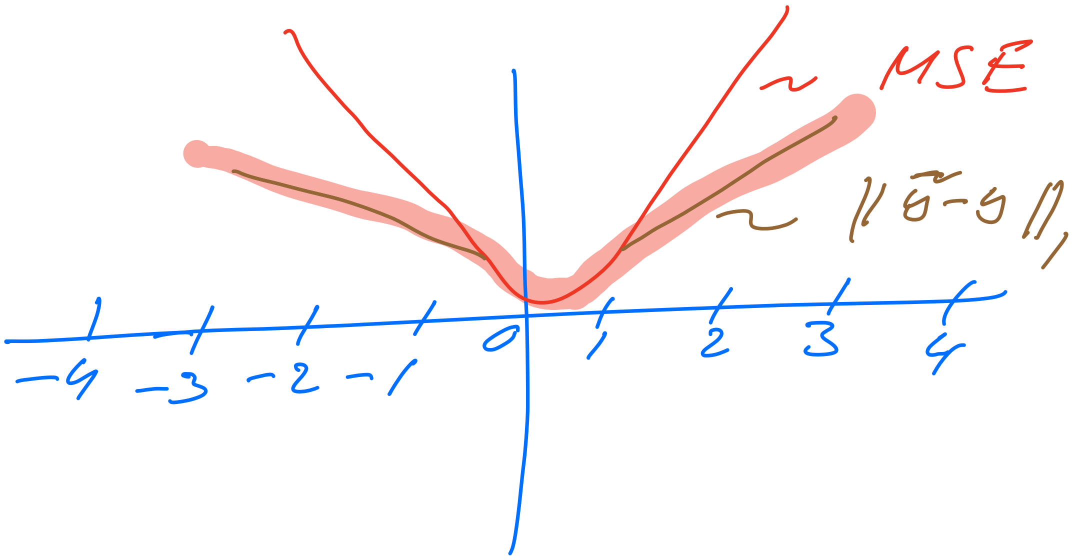
Regression cases have mainly utilized MSE

$$C(\beta) = \frac{1}{n}\sum_{i=0}^{n-1}(y_i - \tilde{y}_i)^2$$

with many outliers you can easily run into a poor MSE.

Huber function combines $\|(y-\tilde{y})\|_2^2$ and $\|y-\tilde{y}\|_1$

$$C(y,\tilde{y}) = \begin{cases} \|(y-\tilde{y}\|_2^2 & \text{if } \|y-\tilde{y}\|_1 \leq \delta \\ \delta\|y-\tilde{y}\|_1 - \delta^2/2 & \text{otherwise} \end{cases}$$



Boosting for classification

AdaBoost

our dataset

$$D = \{ (x_0\, y_0), (x_1\, y_1) \dots (x_{n-1}\, y_{n-1}) \}$$

Each x_i has an associated class $y_i = \{-1, 1\}$

Define a set of weak classifiers (learner)

$$\{ b_1, b_2, \dots, b_M \}$$

each of these outputs

$$b_j(x_i) \in \{-1, +1\}$$

After $m-1$ iterations we have

$$f_{m-1}(x_i) = \alpha_1 b_1(x_i) + \alpha_2 b_2(x_i) + \dots \alpha_{m-1} b_{m-1}(x_i)$$

which we use in next iteration

$$f_m(x_i) = f_{m-1}(x_i) + \alpha_m b_m(x_i)$$

How do we find α_i?

We define an error

$$E_{rr}^{(m)} = \sum_{i=0}^{n-1} e^{-y_i f_m(x_i)}$$

$$y_i \in \{-1, 1\}$$

$$f_m(x_i) \in \{-1, 1\}$$

if correct prediction

$$y_i f_m(x_i) = +1$$

if wrong $y_i f_m(x_i) = -1$

$$E_n^{(m)} = \sum_{n=0}^{n-1} e^{-y_i f_{m-1}(x_i)} e^{-y_i \alpha_m b_m(x_i)}$$

$$w_n^{(1)} = 1 \qquad w_n^{(m)} = e^{-y_i f_{m-1}(x_i)}$$

For $m > 1$

$$E_n^{(m)} = \sum_{n=0}^{n-1} w_n^{(m)} e^{-y_i \alpha_m b_m(x_i)}$$

$$y_i \alpha_m b_m(x_i) = \pm 1$$

$$= \sum_{\substack{i \\ y_i = b_m(x_i)}} w_n^{(m)} e^{-\alpha_m}$$

$\alpha_m > 0$

$$+ \sum_{\substack{n \\ y_i \neq b_m(x_i)}} w_n^{(m)} e^{+\alpha_m}$$

$$= \sum_{n=0}^{n-1} w_n^{(m)} e^{-\alpha_m} + \sum_{\substack{n \\ y_i \neq b_m(x_i)}} w_n^{(m)}$$

$$\times \left(e^{\alpha_m} - e^{-\alpha_m} \right)$$

$$\frac{d\,\mathcal{E}_n^{(m)}}{d\alpha_m} = 0 =$$

$$- \sum_{\substack{i \\ y_i = h_m(x_i)}} w_i^{(m)} e^{-\alpha_m}$$

$$+ \sum_{y_i \neq h_m(x_i)} w_i^{(m)} e^{\alpha_m} = 0$$

α_m does not depend -i-

$$e^{-\alpha_m} \sum_{y_i = h_m(x_i)} w_i^{(m)} =$$

$$e^{\alpha_m} \sum_{y_i \neq h_m(x_i)} w_i^{(m)}$$

$$-\alpha_m \log \left[\sum_{y_i = h_m(x_i)} w_i^{(m)} \right]$$

$$= \alpha_m + \log\left[\sum_{y_i \neq b_m(x_i)} w_i^{(m)}\right]$$

$$\alpha_m = \frac{1}{2}\log\left[\frac{\sum_{y_i = b_m(x_i)} w_i^{(m)}}{\sum_{y_i \neq b_m(x_i)} w_i^{(m)}}\right]$$

Define Error rate

$$\varepsilon_m = \frac{\sum_{y_i \neq b_m(x_i)} w_i^{(m)}}{\sum_{i=0}^{n-1} w_i^{(m)}}$$

$$\alpha_m = \frac{1}{2}\log\left[\frac{1 - \varepsilon_m}{\varepsilon_m}\right]$$

Algorithm

Define $D = \{(x_0, y_0), (x_1, y_1), \ldots (x_{n-1}, y_{n-1})\}$

initialize weights

$$w_0^{(0)} = \frac{1}{n} = w_1^{(0)} = \ldots = w_{n-1}^{(0)}$$

Define error function

$e^{-y f_m}$

Define a weak learner

$b(x) \in \{-1, 1\}$

(Decision tree)

for $m = 1 : M$

$\quad Err = \sum_{j=0}^{n-1} w_j^{(m)}$

$\quad\quad y(x_i) \neq f_m(x_i)$

(optimize) Find $f_m(x)$

compute

$\alpha_m = \frac{1}{2} \log \left(\frac{1 - E_m}{E_m} \right)$

$\alpha_m > 0$

update weights

$w_i^{(m+1)} = w_i^{(m)} e^{-y_i \alpha_m f_m(x_i)}$

normalize weights

$\sum_{i=0}^{n-1} w_i^{(m+1)}$

end

Return $f_M(x) = f_{M-1}(x) + \alpha_M h_M(x)$