FYS-STK3155/4155, lecture November 25, 2024

FYS-STK3155/4155, Nov 25

ADA BOOST

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

$$y_i = \{-1, 1\} = y(x_i)$$

Set of weak classifiers

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

$$b_j(x_i) = \{-1, 1\}$$

$$C_{m-1}(x_i) = \alpha_1 b_1(x_i) + \ldots$$

$$\alpha_{m-1} b_{m-1}(x_i)$$

$$E_n = \sum_{i=0}^{n-1} e^{-y_i C_m(x_i)}$$

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

$y_i C_m(x_i) = +1$ if predicted correctly

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

$$C_m(x_i) = C_{m-1}(x_i) + \alpha_m b_m(x_i)$$

$$E_{rr} = \sum_{i=0}^{n-1} \left\{ e^{-y_i C_{m-1}(x_i)} \times e^{-y_i \alpha_m b_m(x_i)} \right\}$$

$$w_i^{(0)} = \frac{1}{}$$

$$w_i^{(m)} = e^{-y_i C_{m-1}(x_i)}$$

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

correct classification

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

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

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

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

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

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

$$\frac{d E_m}{d \alpha_m} = - \sum_{y_i = b_m(x_i)} w_i^{(m)} e^{-\alpha_m} + \sum_{y_i \neq b_m(x_i)} w_i^{(m)} e^{\alpha_m} = 0$$

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

Define weighted error

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

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

Algorithm

- Define $D = \{(x_0, y_0), (x_1, y_1) \dots \}$
- initialize weights

$$w_0^{(0)} = w_1^{(1)} = \dots \frac{1}{n}$$

- Define function to optimize

$$E_m = \sum_i e^{-y_i C_m(x_i)}$$

- Define weak learner

$$k_m(x_i) = \{-1, 1\}$$

for $m = 1 : M$

- calculate Err
- compute $\alpha_m = \frac{1}{2} \ln\left[\frac{1 - \varepsilon_m}{\varepsilon_m}\right]$

- update

$$C_m(x) = C_{m-1}(x) + \alpha_m k_m(x)$$

- update weights

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

for $i = 0, 1, \ldots, n-1$

normalize $w_i^{(m+1)}$

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

end

end loop over m

$C_M(x)$

Gradient Boosting

$$f = \{ f(x_0), \ldots \ldots f(x_{n-1}) \}$$

$$f_M(x) = \sum_{m=0}^{M} f_m(x)$$

$$C(f) = \sum_{i=0}^{n-1} \mathcal{L}(y_i, f(x_i))$$

$$\hat{f} = \underset{f}{\operatorname{argmin}}\, C(f)$$

one way to find $f_m(x)$

$$= f_{m-1}(x) - \rho_m \, g_m(x)$$

scalar

gradient of L

$$g_m(x_i) = \left. \frac{\partial L(y_i, f_{m-1}(x_i))}{\partial f(x_i)} \right|_{f(x_i) = f_{m-1}(x_i)}$$

step length ρ_m
(learning rate)

$$\hat{\rho}_m = \arg\min_{\rho} C(\rho)$$

Gradient boosting

$$C(f) = \frac{1}{n} \sum_{i=0}^{n-1} (y_i - f(x_i))^2$$

$(\tilde{y}_i = f(x_i)$, predicted value$)$

$$f_{m+1}(x_i) = f_m(x_i) + r_m(x_i)$$

$$= y_i$$

$$r_m(x_i) = y_i - f_m(x_i)$$

residual

Gradient boosting aims at optimizing $r_m(x_i)$ to be the residual $y_i - f_m(x_i)$

$$\frac{\partial C}{\partial f(x_i)} = -\frac{2}{n}\left(y_i - f(x_i)\right)$$

$$= -\frac{2}{n}\, r_m(x_i)$$

$$f_m(x) = f_{m-1}(x) + r_m(x)$$

$$f_m(x) = f_{m-1}(x) + \arg\min_{r_m} \sum_{i=0}^{n-1} \mathcal{L}\left(y_i, f_{m-1}(x_i) + r_m(x_i)\right)$$

Algorithm for gradient boosting

(i) - initialize a model $f_0(x)$

- set up data set
- Define M

(ii) for $m = 1 : M$

- compute the gradients

$$r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x_i) = f_{m-1}(x_i)}$$

$i = 0, 1, 2, \ldots, n-1$

- fit a decision tree to the residual r_{im}
- For each leaf node

$$\gamma_m = \underset{\gamma}{\operatorname{argmin}}$$

$$\sum_{x_i \in R_{jm}} \mathcal{L}(y_i, f_{m-1}(x_i) + \gamma)$$

specific leaf node

- update model

$$f_m(x) = f_{m-1} + \eta \gamma_m$$

simply our learning rate

$$\eta \sim 10^{-3} - 10^{-1}$$