FYS-STK3155/4155, NOV 18

$$P(x=k) = \binom{n}{k} \frac{1}{2}^{k} \left(1-\frac{1}{2}\right)^{n-k}$$

$k=5 \quad n=10$

$$\frac{10!}{5!\,5!} \left(\frac{1}{2}\right)^{10}$$

at least

$$P(x \geq k) = \sum_{k+1}^{n} \binom{n}{k} \frac{1}{2}^{n}$$

Boosting methods
- AdaBoost
- Gradient Boosting (XGB)

Basic idea is to use a weak learner to train continuous improvements in an iterative way

Regression case

$$f_M(x) \simeq y(x) \quad \text{(output)}$$

one possible setup:

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

parameter

weak learner

parameter

$$f_m(x) = f_{m-1}(x) + \beta_m \, b(x; \gamma_m)$$

$f_0(x)$ = initial guess (could be zero)

Example

$$b(x;\gamma) = 1+\gamma x$$

Define a cost/loss function

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

$$= \frac{1}{n}\sum_{i=0}^{n-1}\left(y_i - \underbrace{f_{m-1}(x_i) - \beta(1+\gamma x_i)}_{f_m(x_i)}\right)^2$$

(for every m)

must determine β, γ

guess $f_0(x) = 0$

$m = 1$

$$f_1(x) = f_0(x) + \beta_1 b(x; \gamma_1)$$

we need derivatives of

$$\frac{\partial C}{\partial \beta_m} , \quad \frac{\partial C}{\partial \gamma_m}$$

$$\frac{\partial C}{\partial \beta_1} = 0 = -\frac{2}{n} \sum_{i=0}^{n-1} (1+\gamma_1 x_i)(y_i - f_0(x_i) - \beta_1 b(x_i; \gamma_1))$$

$$\frac{\partial C}{\partial \gamma_1} = -\frac{2}{n}\sum_i \beta_1 x_i \left(y_i - f_0(x) - \beta_1(1+\gamma_1 x_i)\right) = 0$$

Solve by gradient descent methods

Algorithm:

- Define cost/loss function
- $D = \{(x_0, y_0), (x_1, y_1) \ldots (x_{n-1}, y_{n-1})\}$
- Define a weak learner $b(x; \gamma)$

- $f_0(x) = ?$
- Define # iterations M

for $m = 1 : M$

minimize

$$(\hat{\gamma}_m, \hat{\beta}_m) = \underset{\gamma_m, \beta_m}{\operatorname{argmin}} \; C(y, f_m)$$

- gives $\hat{\gamma}_m$ and $\hat{\beta}_m$
- Define new $f_m(x)$

$$f_m(x) = f_{m-1}(x) + \hat{\beta}_m b(x, \hat{\gamma}_m)$$

Classification case (Adaboost)

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

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

emphasize wrong predictions

Define a set of weak classifier

$\{k_1, k_2, \ldots k_M\}$

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

after $m-1$ iterations

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

cost function to train is the total error

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

this is a way to emphasize wrong predictions

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

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

if correct

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

$$e^{-1}$$

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

$$\Rightarrow e^{+1}$$