Exercise 10.2

$$f^*(x) = \underset{f(x)}{\arg\min}\, E_{Y|x}\left[e^{-Yf(x)}\right]$$

$$\frac{\partial}{\partial f(x)} E_{Y|x}\left[e^{-Yf(x)}\right] = E_{Y|x}\left[-Y e^{-Yf(x)}\right]$$

$$E_{Y|x}\left[-Y e^{-Yf(x)}\right] = 0 \qquad Y = \begin{cases} -1 & \Pr[Y=-1|x] \\ 1 & \Pr[Y=1|x] \end{cases}$$

$$+(+1)\, e^{+(+1)f(x)} \Pr[Y=-1|x] - (1)\, e^{-1\cdot f(x)} \Pr[Y=1|x] = 0$$

$$e^{f(x)} e^{f(x)} \Pr[Y=-1|x] = \cancel{e^{f(x)}}\cancel{e^{-f(x)}} \Pr[Y=1|x]$$ multiply both sides for $e^{f(x)}$

$$e^{2f(x)} \Pr[Y=-1|x] = \Pr[Y=1|x]$$

$$e^{2f(x)} = \frac{\Pr[Y=1|x]}{\Pr[Y=-1|x]}$$

$$f(x) = \frac{1}{2}\log\left(\frac{\Pr[Y=1|x]}{\Pr[Y=-1|x]}\right)$$

Exercise 10.4

$X = (x_1, \dots, x_{10}) \qquad x_j \sim N(0;1) \qquad x_j$ iid $\qquad N = 2000$

$$Y_i = \begin{cases} 1 & \text{if } \sum_{j=1}^{10} x_j^2 > \chi^2_{10}(0.5) \\ -1 & \text{if } \sum_{j=1}^{10} x_{ij}^2 \leq \chi_{10}(0.5) \end{cases}$$

$$\begin{array}{c|ccccc} & x_1 & x_2 & \dots & x_{10} \\ 1 & x_{11} & x_{12} & \dots & x_{1,10} \\ 2 & & & & \\ \vdots & & & & \\ N & & & & \end{array}$$

obs

apply (x^2, 1, sum)

Statistical boosting $\begin{cases} \text{gradient boosting} \\ \text{likelihood-based boosting} \end{cases}$

From the previous lecture:

AdaBoost : classifier

- weak estimator
- loss function
- iteratively apply a weak estimator to modifications of the data in order to minimize a loss function

↳ AdaBoost : weight more missclassified observations

AdaBoost : - stump - tree - ...

for classification → AdaBoost $e^{-Yf(x)}$

from classification to regression

- loss function : RSS
- weak estimator : $\underbrace{\nu \overbrace{(X^TX)^{-1}X^Ty}^{\text{LS estimator}}}_{\text{weak estimator}}$

$\nu \to \alpha$

ν

penalty parameter $0 < \nu < 1$

↳ makes our LSE "weak" default $= 0.1$

- modification of the data : $y \to u$ residuals

focusing on the not explain part of the variation outcome

L_2 Boost algorithm for linear regression

1. Initialization: initialize the regression coefficient estimate $\hat{\beta}^{[0]} = (0, \dots, 0)$

   (first modification of the data: $u = y - X\beta = y - 0 = y$)

2. for m from 1 to m_stop

   a) fit the weak estimator to the modification of the data
   $$\hat{b}^{[m]} = \nu (X^T X)^{-1} X^T u \qquad \nu = 0.1$$

   b) update the estimate $\hat{\beta}^{[m]} = \hat{\beta}^{[m-1]} + \hat{b}^{[m]}$

   c) modify the data: $u = y - X^T \hat{\beta}^{[m]}$

3. final estimate
$$\hat{\beta}_{L_2Boost} = \sum_{m=1}^{m\_stop} \hat{b}^{[m]} = \hat{\beta}^{[m\_stop]}$$

Note:

$m \to \infty$, $\hat{\beta}_{L_2Boost} \longrightarrow \hat{\beta}_{OLS}$

need of an early stop (find the "right" m_stop) in order to not overfit (to find the best balance between bias and variance for the prediction error)

- m_stop is the crucial tuning parameter

  if it is too small: too much bias (our model does not explain the outcome variation)

  if it is too big: too much variance (we overfit the data)

complexity parameter

boosting has a second tuning parameter, ν (is not so important, because smaller values ⇒ more steps (iterations)
larger values ⇒ less steps

X must be centred $E[x_j] = 0$
(it is ok to standardize)

L_2 Boost algorithm in general

- the goal is to minimize the loss function. At each step we want to identify the direction of the greatest decrease of the loss function

$\rightsquigarrow$ negative gradient : $-\dfrac{\partial L(y, f(x))}{\partial f(x)}$

e.g. $\dfrac{\partial \sum \frac{1}{2}(y - x^T\beta)^2}{\partial \underbrace{x^T\beta}_{f(x)}} = \dfrac{\cancel{2}}{\cancel{2}} \underbrace{\sum (y - x^T\beta)}_{\text{residuals}}$

$\frac{1}{K} \exp\{-\frac{1}{2}(y - x^T\beta)\}$

In general:

1. Initialization : $\hat{f}(x) = 0$ or $\hat{f}(x) = \bar{y}$
2. for m from 1 to m_{stop}
   - (a) derive $U = -\dfrac{\partial L(Y, f(x))}{\partial f(x)}$ RSS lin. reg.
   - (b) fit our weak estimator : $\hat{b} = g(\nu, x, U)$ lin. reg. residuals; $\nu (X^TX)^{-1}X^Ty$ lin reg
   - (c) update the estimate $\hat{f}^{[m]} = \hat{f}^{[m-1]} + \hat{b}$
3. finalization $\hat{f}_{boost} = \hat{f}^{[m_{stop}]}$

GAM : $y = \sum_{j=1}^{p} f_j(x_j)$

L_2 Boost for High Dimensional Data

- one of the advantages of boosting is that we can handle HDD

$\rightarrow$ componentwise version of boosting

Componentwise boosting

- Linear ^Gaussian^ regression model

① Initialization $\hat{\beta}_j^{[0]} \equiv 0 \quad j = 1, \dots, p$

$$\left(u = y - X\hat{\beta}_j^{[0]} = y \right)$$

② for m from 1 to m_stop

a) compute possible updates for each dimension of the regression coefficient vector seperately (fit a weak estimator on each dimension of x)

$$b_j = \frac{\sum_{i=1}^{n} x_i^{(j)} u_i}{\sum_{i=1}^{n} x_i^{(j)2}} \qquad j = 1, \dots, p$$

b) select the best update among the p possibilities

$$j^* : \arg\min_j \sum_{i=1}^{n} (u_i - x^{(j)} b_j)^2$$

c) update the j^*-th regression coefficient

$$\hat{\beta}^{[m]} = \hat{\beta}^{[m-1]} + (0, \dots, 0, b_{j^*}, 0, \dots, 0)$$

d) modify the data $u = y - X\hat{\beta}^{[m]}$

---

fit a GAM in R with boosting

$$y = f_1(x_1) + f_2(x_2) + \dots + f_p(x_p)$$

splines
tree

→ function mboost of the package mboost

$\mu(y) = X^T\beta$ → glmboost

Gaussian regression
logistic regression
⋮

Likelihood-based boosting

- fully statistical approach ⇝ likelihood-based
- weak estimator:

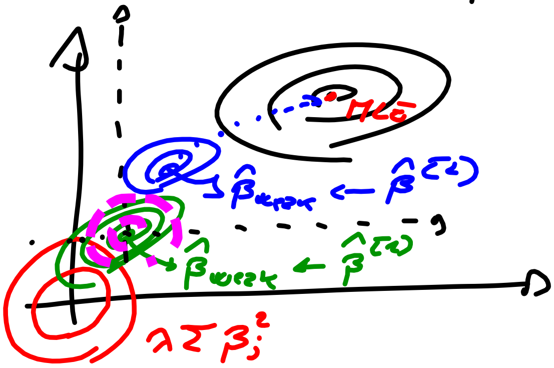
standard estimator $\hat\beta_{MLE} = \arg\max_{\beta} \ell(\beta)$

weak estimator $\hat\beta_{weak} = \arg\max_{\beta} \ell_{pen}(\beta)$

↓ ridge penalty

$$\hat\beta_{weak} = (X^TX + \lambda I)^{-1} X^T y$$

↳ same meaning of the parameter λ



Boosting Ridge (Gaussian regression)

① initialize $\hat\beta^{[0]} = (0, \dots, 0)$ → $U = y - X\hat\beta^{[0]}$

② fit the weak estimator $\hat b = (X^TX + I\lambda)X^Ty$

$$\frac{\partial \ell_{pen}(\beta)}{\partial \beta} = 0 \qquad \ell_{pen} = \frac{1}{2}(y - X\beta)(y - X\beta)^T + \frac{\lambda}{2}\beta\beta^T \qquad \lambda \quad \frac{\lambda}{2}$$

$$\frac{\partial \ell_{pen}}{\partial \beta} = -X^T(y - X\beta) + \lambda\beta = 0$$

Kernel of a Gaussian log-likelihood

$\sigma^2 = 1$ $\exp\left\{-\frac{1}{2}(y - X\beta)(y - X\beta)^T\right\}$

$$-X^Ty + X^TX\beta + \lambda\beta = 0$$

$$\beta(X^TX + \lambda I) = X^Ty$$

$$\hat b = \hat\beta = (X^TX + \lambda I)^{-1}X^Ty$$

(b) $\hat\beta^{[m]} = \hat\beta^{[m-1]} + \hat b$

(c) modification of the data
(add an offset in the log-likelihood)

$$\frac{1}{2}(\underbrace{y - X\hat\beta^{[m]}}_{u} - X\beta)(\underbrace{y - X\hat\beta^{[m]}}_{u} - X\beta)^T$$

3 $\hat\beta_{LBoost} = \sum_{m=1}^{m_{stop}} \hat b^{[m]}$

weak estimator is the ridge estimator

Gaussian regression → likelihood-based boosting → same model when we choose ν and λ in the right way
gradient boosting

Exercise

$$\hat{y} = X\hat{\beta}_{boost} = \sum_{m=1}^{m.stop} S(I-S)^m y = I - (I-S)^{m.stop+1} y$$

where $S = Xb$

using $b = (X^T X + \lambda I)^{-1} X^T y$ Boosting with ridge estimator

show through SVD that Boosting ridge and ridge regression provide different shrinkage effect

ridge regression : $\dfrac{d_j^2}{d_j^2 + \gamma}$

Boosting Ridge : $\left(1 - \left(1 - \dfrac{d_j^2}{d_j^2 + \lambda}\right)^{m.stop+1}\right)$