Lecture September 24

- Short note about Project

<u>scaling</u>:

- OLS, with intercept or not, no change in MSE

Do it yourself.

$X \rightarrow X - \text{mean}(X)$

$y \rightarrow y - \text{mean}(y)$

$\beta_0 = \text{mean}(y)$

- <u>Ridge</u>: regularization term

$\lambda \|\beta\|_2^2$

$= \lambda \sum_{j=0}^{p-1} \beta_j^2$

β_0 not included.

- same with Lasso.

$[\, 1 \;\; x_i^1 \;\; x_i^2 \;\; \cdots \;\; x_i^{p-1} \,]$

$$X = \begin{bmatrix} 1 & x_0^1 & x_0^2 & \cdots & x_0 \\ 1 & x_1^1 & x_1^2 & \cdots & \\ 1 & \vdots & & & \\ 1 & \vdots & & & \end{bmatrix}$$

if you keep the intercept column, when comparing own code with sklearn fit_intercept = False.

- <u>Coding</u>

  Python

  - numpy
  - Pandas
  - CV $\left\{ \begin{array}{l} k=5 \quad \boxed{\text{Train} \mid - \mid - \mid \ldots \mid \text{Test}} \\ \text{KFold in sklearn.} \end{array} \right.$

Classification & Logistic regression

- Linear Regression

$$y = f(x) + \varepsilon$$

$$\varepsilon \sim N(0, \sigma^2)$$

$$\tilde{y} = X\beta$$

$$\mathbb{E}[y] = X\beta$$

$$\text{var}[y] = \sigma^2$$

$$y \sim N(X\beta, \sigma^2)$$

— Binary classification

$$y_i = 1 \Rightarrow p(y_i = 1 | x_i; \beta)$$

$$y_i = 0 \Rightarrow 1 - p(y_i = 1 | x_i; \beta)$$

$$\underline{y = p(x) + \varepsilon}$$

$\varepsilon \sim$ Binomial distribution

y, ε are iid

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

$$P(D|\beta)$$

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmax}}\, P(D|\beta)$$

$$C(\beta) = \underline{-\log P(D|\beta)}$$

$$P(D|\beta) = \prod_{i=0}^{n-1} P(y_i=1|\beta)^{y_i} \Big(\underbrace{1 - P(y_i=1|\beta)}_{P(y_i=0|\beta)}\Big)^{1-y_i}$$

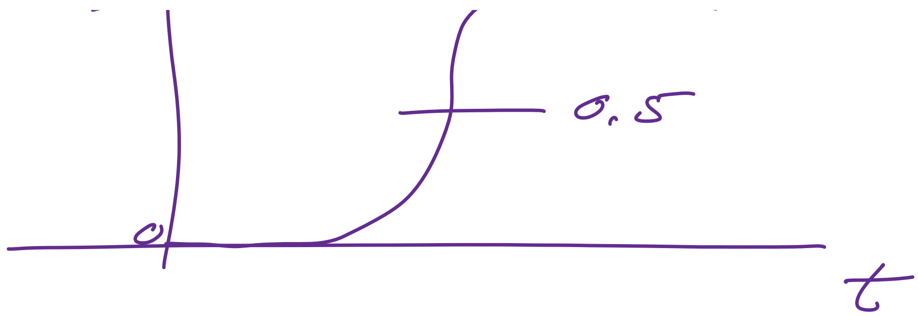
$$P(y_i=1|x_i;\beta) = \frac{e^{t(x_i;\beta)}}{1+e^{t(x_i;\beta)}}$$

$$= \frac{e^t}{1+e^t}$$

$$1 - P(y_i=1|x_i;\beta) = \frac{1}{1+e^t}$$

$$0 \leq P(y_i|x_i;\beta) \leq 1$$

$1 \neq$



$$t = t(x, \beta) = \beta_0 + \beta_1 x$$

$$\left(t = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots \beta_{p-1} x_{i,p-1} \right)$$

$$C(\beta) = - \sum_{i=0}^{n-1} \left\{ y_i \log p_i + (1 - y_i) \log(1 - p_i) \right\}$$

$$\left(p_i = p(y_i = 1 \mid x_i \beta) \right)$$

$$p_i = \frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}}$$

$$C(\beta) = - \sum_{i=0}^{n-1} \left\{ y_i \left(\beta_0 + \beta_1 x_i - \log\left(1 + e^{\beta_0 + \beta_1 x_i}\right)\right) - (1 - y_i) \log\left(1 + e^{\beta_0 + \beta_1 x_i}\right) \right\}$$

$$\frac{\partial C(\beta)}{\partial \beta_0} = 0 = -\sum (y_i - p_i)$$

$$\frac{\partial C(\beta)}{\partial \beta_1} = 0 = -\sum x_i (y_i - p_i)$$

$$\frac{\partial C}{\partial \beta} = 0 = \boxed{-X^T(y - p)}$$

$p(x_i y_i | \beta)$

$$\frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}}$$

non-linear
dependence on β

$y_i = [0, 1] \rightarrow \tilde{y}_i = [-1, 1]$

$$\frac{\partial^2 C}{\partial\beta\,\partial\beta^T} = X W X$$

$0 \le p_i \le 1$

$y_i = 1$

$$\left(\begin{array}{l} W_{ii} = \underline{p_i}\,(1 - p_i) \\ w_{ij} = 0 \quad i\,j \;\; i \neq j \end{array} \right)$$

$$W \geq 0$$

In linear Regression

$$\frac{\partial^2 C}{\partial\beta\,\partial\beta^T} = X^T X = H$$

(Hessian)

$$\boxed{g = -X^T(y - P) = 0}$$

$$f(s) = 0$$

Newton-Raphson

Taylor expand

$$f(s) = f(x) + (s-x)f'(x) + \frac{(s-x)^2}{2} f''$$

$2!$

$$+ \cdots$$

$$= f(x) + (s-x)f'(x) = 0$$

$$s = x - f(x)/f'(x)$$

suggests an iterative procedure:

$$s \Rightarrow x_{n+1} = x_n - f(x_n)/f'(x_n)$$

stop when $|x_{n+1} - x_n| \le \delta$

$$\delta \sim 10^{-10}$$

$$\underline{g(\beta)} = -X^T(y - p(\beta)) = 0$$

Generalization of Newton-Raphson to more than one variable applied to $g(\beta) = 0$

$$\beta_{n+1} = \beta_n - g(\beta_n)/H(\beta)$$

$$\frac{\partial g}{\partial \beta^T} = \frac{\partial g^T}{\partial \beta} = H$$

$$\underline{\beta}_{n+1} = \beta_n + \frac{1}{H} X^T(y - p(\beta_n))$$

$$\frac{1}{H} \longrightarrow \quad \eta = \text{learning rate}$$

$$\boxed{\beta_{n+1} = \beta_n - \eta\, g(\beta_n)}$$

η is a parameter

$$\eta = [10^{-5}, 10^{-9}, \dots]$$

gradient descent.