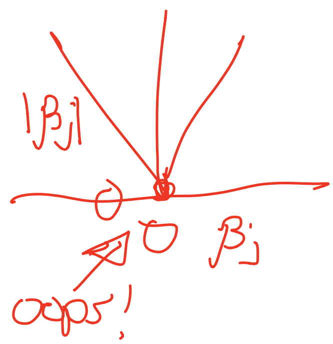
$$C(\beta_1,\beta_2,\dots\beta_p) = \sum_i^N \left(y_i - \sum_j^p x_{ij}\beta_j\right)^2 + \lambda \sum_j^p |\beta_j|$$

$$\hat{\beta}_1,\hat{\beta}_2,\dots\hat{\beta}_p = \underset{\beta_1,\beta_2,\dots\beta_p}{\arg\min}\; C(\beta_1,\beta_2,\dots\beta_p)$$

this problem is trick because of the $|\beta_j|$ term



We will go over the derivation of the coord. descent algorithm for lasso and then, at the end, you will write out pseudocode in groups.

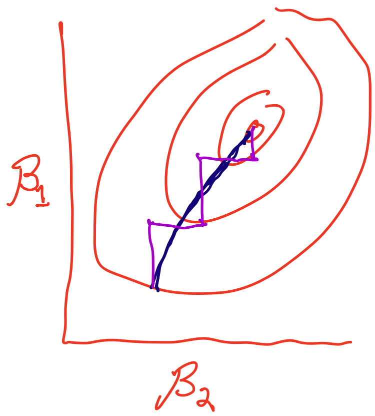
History: (according to Hastie)

- PhD student (Tibshirani's student) Wenjeng Fu had main idea in 1997, called it "shooting alg."
- 2002 Tibshirani and Hastie try something similar but have bug + assume doesn't work
- 2006 The realised their mistake and published stuff with it. Advisor -1, student +1

$$C(\beta_1, \beta_2, \dots \beta_p) = \sum_i^N \left(y_i - \sum_j^p x_{ij}\beta_j\right)^2 + \lambda \sum_j^p |\beta_j|$$

$$\hat\beta_1, \hat\beta_2, \dots \hat\beta_p = \underset{\beta_1, \beta_2, \dots \beta_p}{\operatorname{argmin}} C(\beta_1, \beta_2, \dots \beta_p)$$

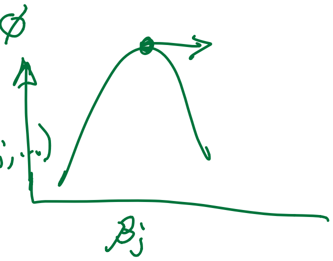
For $P = 2$



ONe approach
"coordinate descent"

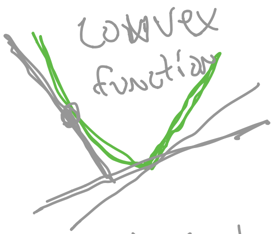
initialize $\hat\beta_1, \hat\beta_2, \dots \hat\beta_p$

while not converged:
    for j in P:
        $\hat\beta_j = \underset{\beta_j}{\operatorname{argmin}} \underbrace{C(\hat\beta_1, \hat\beta_2, \dots \beta_j, \dots \hat\beta_p)}_{\text{ideally find } \frac{\partial}{\partial \beta_j} C() = \emptyset}$

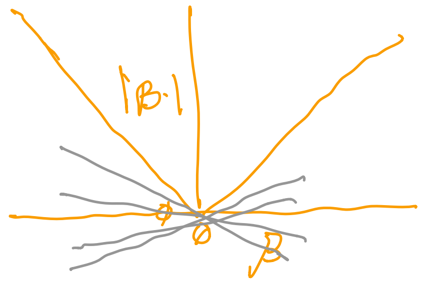


$$C(\hat\beta_1, \hat\beta_2, \dots \beta_k, \dots \hat\beta_p) = \sum_i^N \left(y_i - \sum_{j \neq k}^p x_{ij}\hat\beta_j - x_{ik}\beta_k\right)^2 + \lambda \sum_{j \neq k}^p |\hat\beta_j| + \lambda|\beta_k|$$

$$\frac{\partial C(\cdot)}{\partial \beta_k} = \sum_i^N 2\left(y_i - \sum_{j \neq k}^p x_{ij}\hat\beta_j - x_{ik}\beta_k\right)x_{ik} + \boxed{\lambda \frac{\partial}{\partial \beta_k}|\beta_k|} \leftarrow \text{oops!!}$$



gradients/subgradients provide a lower bound to convex functions



at $\beta_k < 0$, $\frac{\partial|\beta_k|}{\partial\beta_k} = -1$

at $\beta_k > 0$, $\frac{\partial|\beta_k|}{\partial\beta_k} = 1$

at $\beta_k = 0$, subgradient so $-1 \le \frac{\partial|\beta_k|}{\partial\beta_k} \le 1$

$$\frac{\partial C(\cdot)}{\partial \beta_k} = \sum_i^N 2\Big(y_i - \sum_{j\neq k}^p x_{ij}\hat{\beta}_j - x_{ik}\beta_k\Big)(-x_{ik}) + \begin{cases} -\lambda, & \beta_k < 0 \\ [-\lambda, \lambda], & \beta_k = 0 \\ \lambda, & \beta_k > 0 \end{cases}$$

$$\underbrace{-\sum_i^N 2\Big(y_i - \sum_{j\neq k}^p \hat{\beta}_j\Big) x_{ik}}_{g_k} + 2\beta_k \sum_i^N x_{ik}^2 + \begin{cases} -\lambda, & \beta_k < 0 \\ [-\lambda, \lambda], & \beta_k = 0 \\ \lambda, & \beta_k > 0 \end{cases}$$

$\underline{\beta_k < 0}$

$$0 = -g_k + 2\hat{\beta}_k \sum_i^N x_{ik}^2 - \lambda$$

$$2\hat{\beta}_k \sum_i^N x_{ik}^2 = g_k + \lambda$$

$$\hat{\beta}_k = (g_k + \lambda) / 2\sum_i^N x_{ik}^2$$

and this is for when $\beta_k < 0$, so....

$$(g_k+\lambda) / 2\sum_i^N x_{ik}^2 < \emptyset$$

or when $g_k + \lambda < 0$

or $g_k < -\lambda$

$$\hat{\beta}_k = \begin{cases} (g_k+\lambda) / 2\sum_i^N x_{ik}^2, & g_k < -\lambda \\ \emptyset, & -\lambda \le g_k \le \lambda \\ (g_k-\lambda) / 2\sum_i^N x_{ik}, & g_k > \lambda \end{cases}$$

$\underline{\underline{\beta_k > \emptyset}}$ { look above and see it is the same idea so

$$\hat{\beta}_k = (g_k - \lambda) / 2\sum_i^N x_{ik}^2$$

and this occurs at $\beta_k > \emptyset \Rightarrow g_k > \lambda$

$\beta_k = 0$ ...

so $\hat{\beta}_k = 0$

and this is for when

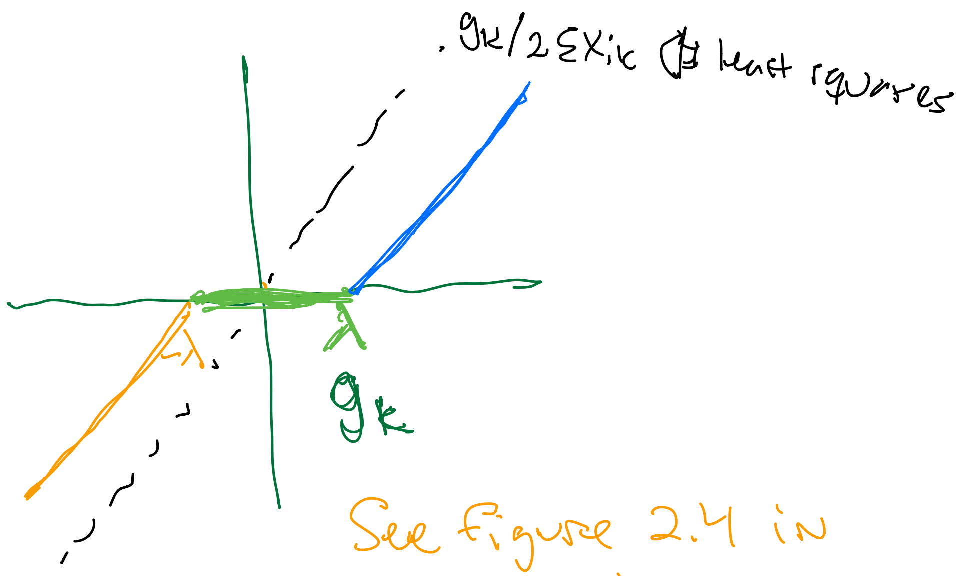
$0 \in \left[(g_k + \lambda) / 2\sum_i^N x_{ik}^2 ,\ (g_k - \lambda) / 2\sum_i^N x_{ik}^2 \right]$

which occurs when

$-\lambda \leq g_k$ and $g_k \leq \lambda$

$$\hat{\beta}_k = \begin{cases} (g_k + \lambda) / 2\sum_i^N x_{ik}^2, & g_k < \lambda \\ 0, & -\lambda \leq g_k \leq \lambda \\ (g_k - \lambda) / 2\sum_i^N x_{ik}, & g_k > \lambda \end{cases}$$

where $g_k = \sum_i^N 2(y_i - \sum_{j \neq k}^p \hat{\beta}_j) x_{ik}$



See figure 2.4 in book.

"Soft thresholding function"