Linear Regression

Part III: Various Algorithms

Tae Geun Kim

Table of Contents

Table of Contents

Table of Contents

LASSO

Find $\hat{\beta}$ for Lasso

Lasso cost function is given as :

$$\begin{aligned}\mathrm{PRSS}^{\mathrm{lasso}}(\beta) &= \frac{1}{2}\mathrm{RSS}(\beta) + \lambda\|\beta\|_1 \\ &= \frac{1}{2}\sum_{i=1}^{N}\left[y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right]^2 + \lambda\sum_{j=1}^{p}|\beta_j|\end{aligned}$$

Find $\hat{\beta}$ for Lasso

Lasso cost function is given as :

$$
\begin{aligned}
\mathrm{PRSS}^{\mathrm{lasso}}(\beta) &= \frac{1}{2}\mathrm{RSS}(\beta) + \lambda\|\beta\|_1 \\
&= \frac{1}{2}\sum_{i=1}^{N}\left[y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right]^2 + \lambda\sum_{j=1}^{p}|\beta_j|
\end{aligned}
$$

We can decompose RSS term as follows:

$$
\begin{aligned}
\frac{\partial}{\partial\beta_j}\mathrm{RSS}(\beta) &= -\sum_{i=1}^{N} x_{ij}\left[y_i - \sum_{k\neq j}^{p} x_{ik}\beta_k - x_{ij}\beta_j\right] \\
&= -\sum_{i=1}^{N} x_{ij}\left[y_i - \sum_{k\neq j} x_{ik}\beta_k\right] + \beta_j\sum_{i=1}^{N} x_{ij}^2 \\
&\equiv -\rho_j + \beta_j z_j
\end{aligned}
$$

Find $\hat{\beta}$ for Lasso

Now, focus on the L_1 term:

$$\lambda \sum_{j=1}^{p} |\beta_j| = \lambda |\beta_j| + \lambda \sum_{k \neq j}^{p} |\beta_k|$$

Find $\hat{\beta}$ for Lasso

Now, focus on the L_1 term:

$$\lambda \sum_{j=1}^{p} |\beta_j| = \lambda |\beta_j| + \lambda \sum_{k \neq j}^{p} |\beta_k|$$

And differentiate it with **subdifferential**:

$$\partial_{\beta_j} \lambda \sum_{j=1}^{p} |\beta_j| = \partial_{\beta_j} \lambda |\beta_j| = \begin{cases} \{-\lambda\} & \text{if } \beta_j < 0 \\ [-\lambda, \lambda] & \text{if } \beta_j = 0 \\ \{\lambda\} & \text{if } \beta_j > 0 \end{cases}$$

Find $\hat{\beta}$ for Lasso

Now, focus on the L_1 term:

$$\lambda \sum_{j=1}^{p} |\beta_j| = \lambda |\beta_j| + \lambda \sum_{k \neq j}^{p} |\beta_k|$$

And differentiate it with **subdifferential**:

$$\partial_{\beta_j} \lambda \sum_{j=1}^{p} |\beta_j| = \partial_{\beta_j} \lambda |\beta_j| = \begin{cases} \{-\lambda\} & \text{if } \beta_j < 0 \\ [-\lambda, \lambda] & \text{if } \beta_j = 0 \\ \{\lambda\} & \text{if } \beta_j > 0 \end{cases}$$

And we need some theorems for subdifferential:

- **Moreau-Rockafellar theorem**: If f, g are both convex with subdifferentials ∂f, ∂g then the subdifferential of $f + g$ is $\partial f + \partial g$
- **Stationary condition**: A point x_0 is the **global minimum** of a convex function f iff the **zero** is contained in the subdifferential.

Find $\hat{\beta}$ for Lasso

Then let's put it together :

$$\partial_{\beta_j} \mathrm{PRSS}^{\mathrm{lasso}}(\beta) = -\rho_j + \beta_j z_j + \partial_{\beta_j} \lambda |\beta_j|$$

$$0 = \begin{cases} -\rho_j + \beta_j z_j - \lambda & \text{if } \beta_j < 0 \\ [-\rho_j - \lambda,\ -\rho_j + \lambda] & \text{if } \beta_j = 0 \\ -\rho_j + \beta_j z_j + \lambda & \text{if } \beta_j > 0 \end{cases}$$

Find $\hat{\beta}$ for Lasso

Then let's put it together :

$$\partial_{\beta_j} \mathrm{PRSS}^{\mathrm{lasso}}(\beta) = -\rho_j + \beta_j z_j + \partial_{\beta_j} \lambda |\beta_j|$$

$$0 = \begin{cases} -\rho_j + \beta_j z_j - \lambda & \text{if } \beta_j < 0 \\ [-\rho_j - \lambda, \ -\rho_j + \lambda] & \text{if } \beta_j = 0 \\ -\rho_j + \beta_j z_j + \lambda & \text{if } \beta_j > 0 \end{cases}$$

We know that $\beta_j = 0$ is a global minimum, thus, there should be the zero in closed interval of the second case.

$$0 \in [-\rho_j - \lambda, \ -\rho_j + \lambda] \ \Rightarrow \ \begin{cases} -\rho_j - \lambda \leq 0 \\ -\rho_j + \lambda \geq 0 \end{cases} \ \Rightarrow \ -\lambda \leq \rho_j \leq \lambda$$

Find $\hat{\beta}$ for Lasso

Then we can get the solution :

$$\hat{\beta}_j = \begin{cases} \frac{\rho_j + \lambda}{z_j} & \text{for } \rho_j < -\lambda \\ 0 & \text{for } -\lambda \leq \rho \leq \lambda \\ \frac{\rho_j - \lambda}{z_j} & \text{for } \rho_j > \lambda \end{cases}$$

Find $\hat{\beta}$ for Lasso

Then we can get the solution :

$$\hat{\beta}_j = \begin{cases} \frac{\rho_j+\lambda}{z_j} & \text{for } \rho_j < -\lambda \\ 0 & \text{for } -\lambda \leq \rho \leq \lambda \\ \frac{\rho_j-\lambda}{z_j} & \text{for } \rho_j > \lambda \end{cases}$$

And it can be denoted with **Soft-thresholding** function.

$$\hat{\beta}_j = \frac{1}{z_j} S(\rho_j, \lambda)$$

$$S(\rho_j, \lambda) = \begin{cases} \rho_j + \lambda & \text{for } \rho_j < -\lambda \\ 0 & \text{for } -\lambda \leq \rho \leq \lambda \\ \rho_j - \lambda & \text{for } \rho_j > \lambda \end{cases}$$

Find $\hat{\beta}$ for Lasso

To find $\hat{\beta}_j$, we need some iterations - called **Coordinate descent**.

Find $\hat{\beta}$ for Lasso

To find $\hat{\beta}_j$, we need some iterations - called **Coordinate descent**.

Coordinate descent update rule :

- For $1 \leq j \leq p$
- Compute $\rho_j = \sum_{i=1}^{N} x_{ij}(y_i - \sum_{k \neq j}^{p} x_{ik}\beta_k)$
- Compute $z_j = \sum_{i=1}^{N} x_{ij}^2$ $\Rightarrow$ If we normalize $\mathbf{X}$ then we can omit this process
- Set $\beta_j = \frac{1}{z_j} S(\rho_j, \lambda)$
- Repeat above processes for the number of iterations or until convergence.

Summary of Lasso

1. **Normalize** input via L_2 norm:

$$z_j = \sum_{i=1}^{N} x_{ij}^2 = 1$$

2. **Center** response:

$$\mathbf{y}^c = \mathbf{y} - \bar{\mathbf{y}}$$

3. Calculate $\hat{\beta}_j$ via **Coordinate descent rule**.

4. Calculate $\hat{\mathbf{y}}^c$:

$$\hat{\mathbf{y}}^c = \frac{\mathbf{X}}{\|\mathbf{X}\|}\hat{\beta}$$

5. Add intercept:

$$\hat{\mathbf{y}} = \bar{\mathbf{y}} + \hat{\mathbf{y}}^c$$

Implementation of Lasso

[Axect's Github]

Principal Components Regression

Principal Components Regression (PCR)

In Ridge regression, we already learned about *principal components*.

$$\mathbf{z}_m = \mathbf{X} v_m \ (1 \leq m \leq p) \quad \text{where} \quad \mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \ \mathbf{V} = (v_m)$$

Principal Components Regression (PCR)

In Ridge regression, we already learned about *principal components*.

$$\mathbf{z}_m = \mathbf{X} v_m \ (1 \leq m \leq p) \quad \text{where} \quad \mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T,\ \mathbf{V} = (v_m)$$

Now, let's take M principal components and regress $\mathbf{y}$ on it.
Since $\mathbf{z}_m$ are orthogonal, the regression is just a sum of univariate regressions:

$$\hat{\mathbf{y}}^{\text{pcr}}_{(M)} = \bar{y}\mathbf{1} + \sum_{m=1}^{M} \hat{\theta}_m \mathbf{z}_m \quad \text{where} \quad \hat{\theta}_m = \frac{\langle \mathbf{z}_m, \mathbf{y} \rangle}{\langle \mathbf{z}_m, \mathbf{z}_m \rangle}$$

Principal Components Regression (PCR)

In Ridge regression, we already learned about *principal components*.

$$\mathbf{z}_m = \mathbf{X} v_m \ (1 \le m \le p) \quad \text{where} \quad \mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T,\ \mathbf{V} = (v_m)$$

Now, let's take M principal components and regress $\mathbf{y}$ on it.
Since $\mathbf{z}_m$ are orthogonal, the regression is just a sum of univariate regressions:

$$\hat{\mathbf{y}}^{\text{pcr}}_{(M)} = \bar{y}\mathbf{1} + \sum_{m=1}^{M} \hat{\theta}_m \mathbf{z}_m \quad \text{where} \quad \hat{\theta}_m = \frac{\langle \mathbf{z}_m, \mathbf{y} \rangle}{\langle \mathbf{z}_m, \mathbf{z}_m \rangle}$$

And corresponding parameter is given as follows:

$$\hat{\beta}^{\text{pcr}}(M) = \sum_{m=1}^{M} \hat{\theta}_m v_m$$

Summary of PCR

1. **Standardize** input $\mathbf{X}$
2. **Center** response $\mathbf{y}$
3. Obtain SVD of input : $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$
4. Set the number of features M
5. Take $\mathbf{z}_m = \mathbf{X}v_m \ (1 \leq m \leq M)$
6. Regress $\mathbf{y}$ on $\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_M$
7. Add intercept term $\bar{y}\mathbf{1}$

Implementation of PCR

Axect's Github

References

- T. Hastie et al., *The Elements of Statistical Learning 2nd ed*, Springer (2009)

Thank you!