# Statistical Learning for Engineers (EN.530.641)
# Homework 5

Jin Seob Kim, Ph.D.

Out: 10/07/2022
due: 10/14/2022 by midnight EST



**1** (Exercise Problem 3.5 in ESL)
Show that (3.85) is equivalent to (3.41) in ESL. Also give correspondence between $\beta^c$'s and the original $\beta$'s in (3.41).

**2** Solve Exercise Problem 3.7 in p.95 of ESL. To repeat:
Assume $y_i \sim \mathcal{N}(\beta_0 + \mathbf{x}_i^\top \boldsymbol{\beta}, \sigma^2)$, $i = 1, 2, ..., N$, and the parameters $\beta_j$, $j = 1, ..., p$ are each distributed as $\mathcal{N}(0, \tau^2)$, independently of one another. Assuming $\sigma^2$ and $\tau^2$ are known, and $\beta_0$ is not governed by a prior (or has a flat improper prior), show that the (minus) log-posterior density of $\boldsymbol{\beta}$ is proportional to

$$\sum_{i=1}^{N} \left( y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

where $\lambda = \sigma^2/\tau^2$.

**3** (Shrinkage or regularization effects)
Let us revisit one problem in a previous homework. But this time, we will include regularization terms. Specifically, we will consider Ridge regression and Lasso regression models.
As before, let us generate an artificial training data set as follows. First, we will define the output $Y$ by the following additive error model

$$Y = 0.5X^2 + 0.5X + 1 + \epsilon \tag{1}$$

where $\epsilon \sim \mathcal{N}(0, 1)$. To complete a training data, let us generate the corresponding $X$ as

$$X = 5\eta - 2$$

where $\eta$ denotes uniform random number. Assume $N = 100$. Let $\alpha$ be the complexity parameter (in ESL, it is $\lambda$).

(a) Perform linear regression without regularization (i.e., $\alpha = 0$). Then perform Ridge regression with $\alpha = 10$ and $\alpha = 100$. Compare three results by plotting original data points and the fitted models (three lines in this case: $\alpha = 0, 10, 100$).

(b) Perform Lasso regression with $\alpha = 0.1$ and $\alpha = 1$. Compare three results by plotting original data points and the fitted models (three lines in this case: $\alpha = 0, 0.1, 1$). Compare them and also compare them with the results in part (a).

Now you will apply polynomial regression with $p = 10$. To repeat, let $p$ denote the highest order of polynomial of interest. Then the model in our case is

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \cdots \beta_p X^p.$$

In other words, the regression problem becomes a multi-dimensional input case where

$$\mathbf{X} = [X_1, X_2, \cdots, X_p]^T = [X, X^2, \cdots, X^p]^T.$$

Hence you can generate the corresponding training data from $X$ and $Y$. Then as we discussed in the class, the same approach for linear regression, Ridge, and Lasso can be applied.

(c) Repeat part (a) with polynomial regression as above with $\alpha = 0, 10^{-5}, 1$.

(d) Repeat part (b) with polynomial regression as above with $\alpha = 0, 10^{-7}, 1$.

(e) Plot the graph of $\beta_i$'s vs $\log \alpha$ for part (c) and (d). From those plots what can you say about the difference between Ridge and Lasso regression?

Practically when you use Scikit-learn, you will need `Ridge` in `sklearn.linear_model` for the Ridge regression. When you look into parameters, you will find `alpha` which is the complexity parameter. Also try to use `svd` and `sag` as solver parameters. Note that `sag` denotes Stochastic Average Gradient-Descent, which is a variant of Stochastic Gradient-Descent (SGD). In other words, when you use `sag`, essentially you are solving the problem by SGD.

For the Lasso regression, you need `Lasso` in `sklearn.linear_model`. You can vary `alpha` in the parameters to apply different $\alpha$'s.

## Submission Guideline

- For analytic parts (e.g., problems 1 and 2, and the results of problem 3), submit your homework answers in a single pdf format, including plots, to "HW5_analytical" on the gradescope.
- No more than two (2) homework problems may be on the same page. In other words, for each problem your answers should be on a separate set of pages. Then when submitting, you should assign the pages to each problem on Gradescope.
- *Show your work.*
- Submit all your python codes in a single .zip file that contains codes for each problem (name them by including the problem number). Name your single zip file submission as "YourName_HW5.zip". For example, "JinSeobKim_HW5.zip" for a single zip file. Submission will be done through "HW5_computational" on the gradescope.
- Just in case you have related separate files, please make sure to include *all the necessary files*. If TAs try to run your function and it does not run, then your submission will have a significant points deduction.
- Make as much comments as possible so that the TAs can easily read your codes.