SMAI Assignment 7 Report

Name: Akshat Maheshwari
Roll No. : 20161024

L1 (Lasso) Regularization:

$$\sum_{i=1}^{n}(y_i - \sum_{j} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

This is the function for the cost function using the Lasso Regularization(L1 Regularization).

Lasso Regularization adds penalty to the absolute value of the magnitude of the weights.

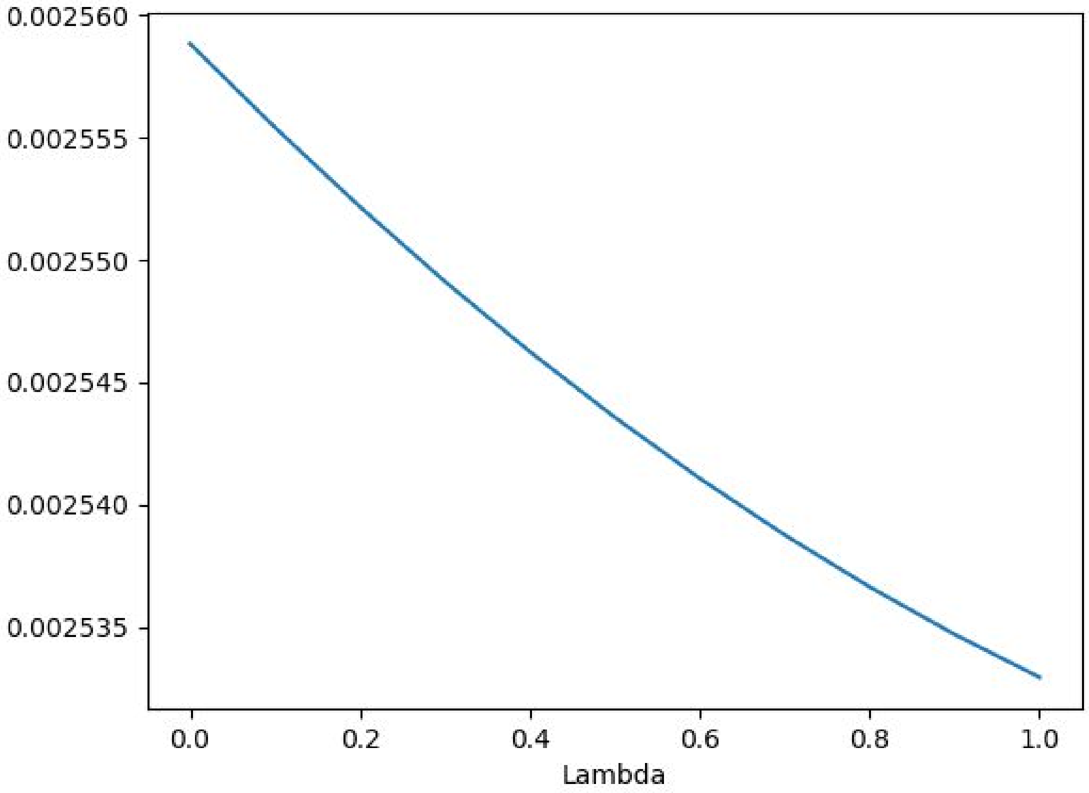
In gradient descent using Lasso Regularization, we have to add the product of lambda and the signum of the magnitudes of the weights.

Overall, the L1 Regularization is as follows:

$$E_{L_1} = E + \lambda \sum_{k=1}^{N} |\beta_k|$$

$$\frac{\partial E_{L_1}}{\partial \beta_l} = \frac{\partial E}{\partial \beta_l} + \lambda \sum_{k=1}^{N} \mathrm{sgn}(\beta_k)\delta_{kl} = \frac{\partial E}{\partial \beta_l} + \lambda \mathrm{sgn}(\beta_l)$$

The graph on adding Lasso Regularization is as follows:



L2 (Ridge) Regularization:

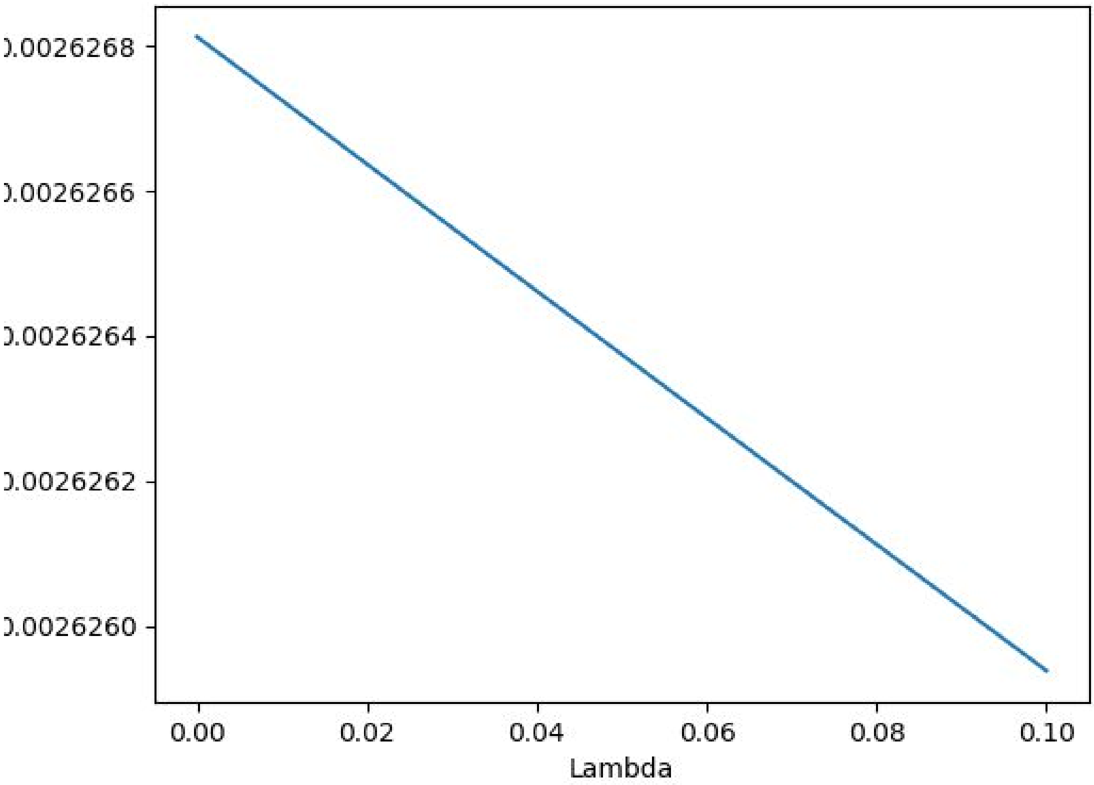
$$L_{ridge}(\hat{\beta}) = \sum_{i=1}^{n}(y_i - x_i'\hat{\beta})^2 + \lambda\sum_{j=1}^{m}\hat{\beta}_j^2 = ||y - X\hat{\beta}||^2 + \lambda||\hat{\beta}||^2.$$

This is the function for the cost function using the Ridge Regularization(L2 Regularization).

Ridge Regularization adds penalty to the square of the values of the weights.

In gradient descent using Lasso Regularization, we have to add the product of twice of lambda and the magnitudes of the weights.

The graph on adding Ridge Regularization is as follows:



Part 3:

In case of L2 regularization,

- If ƛ = 0, the solution is same as in regular least-squares linear regression
- If ƛ → ∞, the solution w → 0
- Positive ƛ will cause the magnitude of the weights to be smaller than in the usual linear regression.

In case of L1 regularization,
ƛ might help in reducing some of the unnecessary weights to 0, instead of a finite value, which is not the case in L2 regularization. L2 regularization does not lead to weights becoming 0, unless ƛ → ∞.

In case of high bias(overfitting), we need to reduce the contribution of the weights, so we need to increase the value of lambda.
In case of high variance(underfitting), we need to increase the contribution of the weights a little bit, so the value of lambda needs to reduce a little bit.

Part 4:

In the case of ML, both ridge regression and Lasso find their respective advantages. Ridge regression does not completely eliminate (bring to zero) the coefficients in the model whereas lasso does this along with automatic variable selection for the model. This is where it gains the upper hand. While this is preferable, it should be noted that the assumptions considered in linear regression might differ sometimes.
Both these techniques tackle overfitting, which is generally present in a realistic statistical model. It all depends on the computing power and data available to perform these techniques on a statistical software. Ridge regression is faster compared to lasso but then again lasso has the advantage of completely reducing unnecessary parameters in the model.

The Lasso Regularization works with the absolute values of the magnitudes of the weights, whereas the Ridge Regularization works with the squares of the magnitudes of the weights.

Part 5:

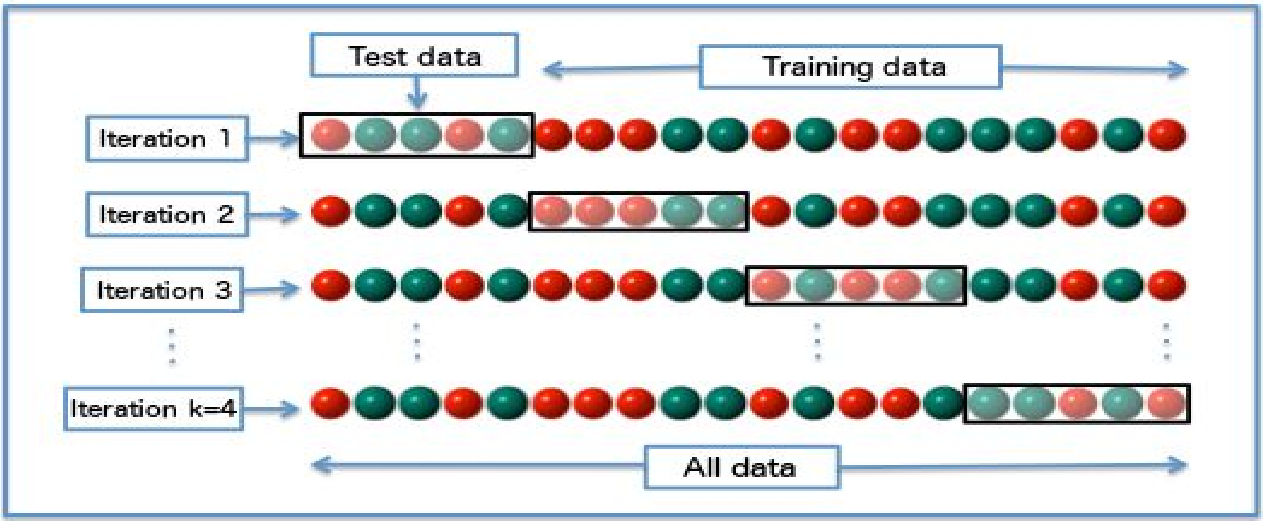
Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample.

The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation.

When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=5 becoming 5-fold cross-validation.

Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model.

It is a popular method because it is simple to understand and because it generally results in a less biased or less optimistic estimate of the model skill than other methods, such as a simple train/test split.



So basically, this method is used for improving the efficiency of the model. This method allows us to choose the best value of theta which gives a lower error on the cross-validation set.

In case of the leave-one-out cross-validation, we just need to split the data into number of parts , which is equal to the length of the training dataset.