Advanced Regression

Linear Regression Review: Lets understand basic terminologies used in Linear regression.

1. Simple linear regression equation: Simple linear regression assumes that there is a linear relationship between the input values and the output values. Mathematically, we can express this relationship as:

$$Yi = \beta_0 + \beta_1 x_i + \epsilon_i$$

2. Predicted values

$$\hat{Y}_i = bo + b_1 x_i$$

3. Residuals

$$E_i = y_i - \hat{y}_i$$

4. Residual sum of squares (RSS) or Cost

$$\Sigma_{i=1}^{n} \varepsilon_i^2 = \Sigma_{i=1}^{n} (y_i - \hat{y}_i)^2 = \Sigma_{i=1}^{n} (y - b_i - b_i x_i)$$

Common Interview Questions

1. In Ridge regression, as the regularization parameter increases, do the regression coefficients decrease?

2. Is it true that the L1 term in Lasso has the following purposes: performing feature selection, compensating for overfitting, and smoothing?

3. What are the assumptions of linear regression?

4. What type of penalty is used on regression weights in Ridge regression?

5. Which regularization is used to reduce the over fit problem?

CNN Key Concepts

- Estimating coefficient in SLR
- Matrix Representation for SLR
- Estimating coefficient in MLR
- Assumptions of linear regression (Bias-variance tradeoff etc.)
- Polynomial regression
- Nonlinearity in data
- Ridge Regularisation
- Lasso Regularisation

Simple Linear Regression - SLR

Estimating Coefficients in SLR: There are two methods to obtain our model coefficients by minimising RSS.

Gradient Descent: Gradient descent is an iterative optimisation algorithm to find the minimum of a cost function; this means we apply a certain update rule over and over again, and following that, our model coefficients or betas would gradually improve according to our objective function.

To perform gradient descent, we initialise the weights to some value (e.g., all zeros) and repeatedly adjust them in the direction that decreases the cost function. We repeat this procedure until the betas converge, or stop changing much. Ultimately, the final betas would be close to the optimum.

Using normal equations to solve for model coefficients:

In normal equations, we calculate the model coefficients b0 and b1 at which our cost function, i.e., RSS, is minimum by using derivatives. In order to do this:

- Take the derivative of the cost function w.r.t. b0 and b1,
- Set each equation to 0 and
- Solve the two equations to get the best values for the parameters b0 and b1.
- The results computed using these normal equations and the gradient descent approach are generally the same; just the methods are different.

Matrix Representation in SLR:

Here X matrix, i.e., the matrix with the predictors, is also known as the design matrix. Here, the first column in the design matrix is a column of 1’s for the intercept term β0 and the second column contains the predictor values. There are n rows in the matrix for ‘n’ observations. The error vector consists of residuals for each observation, which is added to the product of the design matrix and the parameter vector to obtain the response vector, Y.

$$y_1 = \beta_0 + \beta_1 x_1 + \epsilon_1$$

$$y_2 = \beta_0 + \beta_1 x_2 + \epsilon_2$$

$$\dots$$

$$y_n = \beta_0 + \beta_1 x_n + \epsilon_n$$

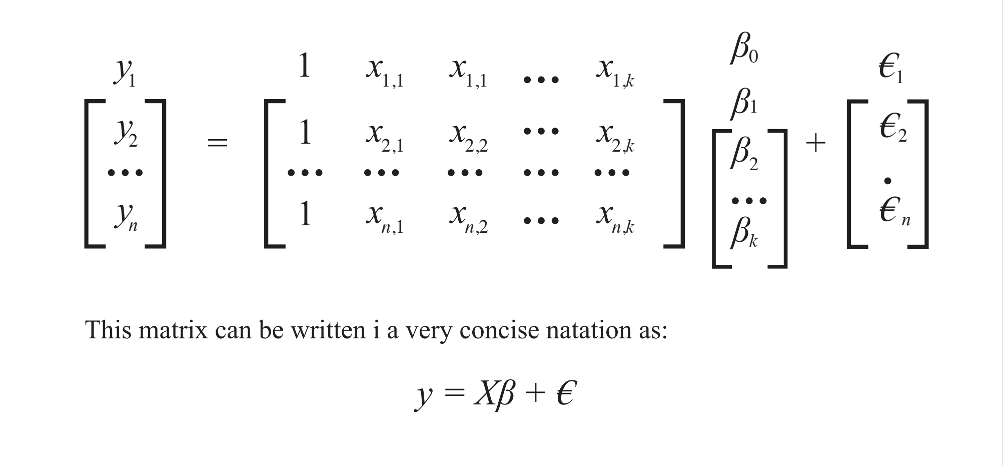
The equations above have been converted to their matrix/vector equivalent as shown below:

$$\begin{array}{cccccc} y_1 & & 1 \quad\quad x_1 & & & \epsilon_1 \\ \begin{bmatrix} y_1 \\ \dots \\ y_n \end{bmatrix} & = & \begin{bmatrix} 1 & x_1 \\ \dots & \dots \\ 1 & x_n \end{bmatrix} & \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} & + & \begin{bmatrix} \epsilon_2 \\ \dots \\ \epsilon_n \end{bmatrix} \end{array}$$

Multiple Linear Regression - MLR

Matrix Representation in MLR:

Here, each row in the X matrix belongs to each of the ‘n’ observations and each column corresponds to each predictor along with the first column of 1’s. Since there are k predictors, we have k+1 elements in the matrix. Finally, we add the product of the design matrix X and the coefficient vector to the error vector to get the Y vector.



$$\begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{1,1} & x_{1,1} & \cdots & x_{1,k} \\ 1 & x_{2,1} & x_{2,2} & \cdots & x_{2,k} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & x_{n,1} & x_{n,2} & \cdots & x_{n,k} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \cdots \\ \beta_k \end{bmatrix} + \begin{bmatrix} \text{€}_1 \\ \text{€}_2 \\ \cdot \\ \text{€}_n \end{bmatrix}$$

This matrix can be written i a very concise natation as:

$$y = X\beta + \text{€}$$

Assumptions of Linear Regression:

1 There is a linear relationship between X and Y: X and Y should display a linear relationship of some form; otherwise, there is no use of fitting a linear model between them.

2. Error terms are distributed normally with mean equal to 0 (not X, Y):

- There is no problem if the error terms are not distributed normally if you wish to just fit a line and not make any further interpretations.
- However, if you wish to draw some inferences on the model that you have built, then you need to have a notion of the distribution of the error terms. One particular repercussion if the error terms are not distributed normally is that the p-values obtained during the hypothesis test to determine the significance of the coefficients become unreliable.
- The assumption of normality is made, as it has been observed that the error terms generally follow a normal distribution with mean equal to 0 in most cases.

3 Error terms are independent of each other: The error terms should not be dependent upon one another.

4. Error terms have constant variance (homoscedasticity): Variance should not increase (or decrease) with a change in error values. Also, variance should not follow any pattern with a change in error terms.

Handling Non-Linear Data

Identify non-linearity in data: The linear regression model assumes that there is a linear relationship between the predictors and the response variable. However, if the true relationship is nonlinear, then virtually all of the conclusions that we draw from the fit do not hold much credibility. In addition, the prediction accuracy of the model can be reduced significantly.

Handling Non-Linear Data:

1. Polynomial regression
2. Data transformation
3. Nonlinear regression

Polynomial regression:

This matrix can be written i a very concise natation as:

$$y= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 \dots\dots ++\beta_k x^k \text{€}$$

If $x_i = x^i$ and $j = 1, 2 ,\dots,k$ then the model is a multiple linear regression model with k predictor variables, $x_1, x_2, x_3,\dots,x_k$ So, the linear regression model $y = X\beta + \varepsilon$ includes polynomial predictors. Thus, polynomial regression can be considered an extension of multiple linear regression and, hence, we can use the same technique used in multiple linear regression to estimate the model coefficients for polynomial regression.

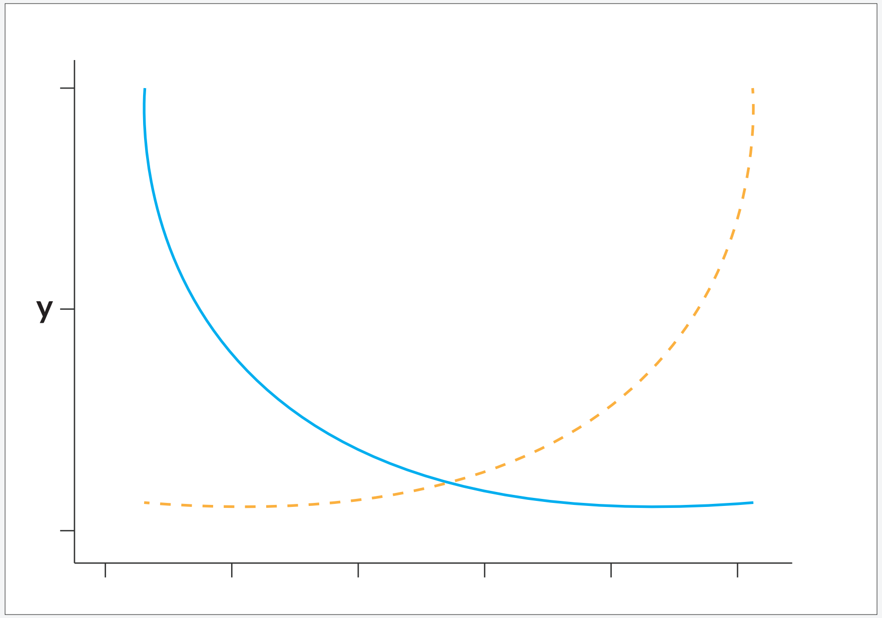
Data regression: If the residual plot indicates the presence of nonlinear relations in the data, then a simple approach is to use nonlinear transformations of the predictors. For instance, for a predictor x, these transformations can be log(x), sqrt(x), exp(x), etc., in the regression model.

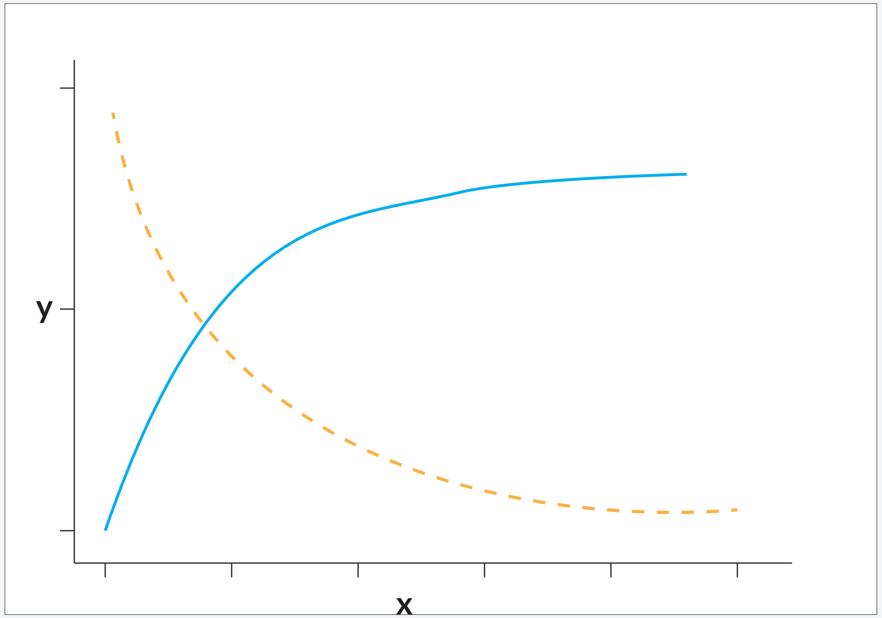
We need to remember that although log is the most commonly used function for transformations, it is not the only one that we can use. There are a few other functions that we can use depending upon the shape of the data.

Essentially, to handle nonlinear data, we may have to try different transformations on the data to determine a model that fits it well. Hence, we may try polynomial models or transformations of the x-variable(s) or the y-variable, or both. These transformations can be square root, logarithmic or reciprocal transformations, although this is not an exhaustive list. Generally, one of these transformations helps.

When we transform the response variable, we will change both its errors and variance. Hence, we should transform the response variable only if the error terms are not normal or if the residuals exhibit non-constant variance, as seen in the residual plots. Transformations that can be applied on the response variable can include natural log, square root or inverse, i.e., ln(y), √y or 1/y, respectively.

However, if nonlinear trends are observed between the predictor and response variables, then we should first try to transform the predictor variable(s). Despite these guidelines, the transformations may not always work.





Regularisation

Regularisation: Regularisation helps with managing model complexity by essentially shrinking the model coefficient estimates towards 0. This discourages the model from becoming too complex, thus avoiding the risk of overfitting.

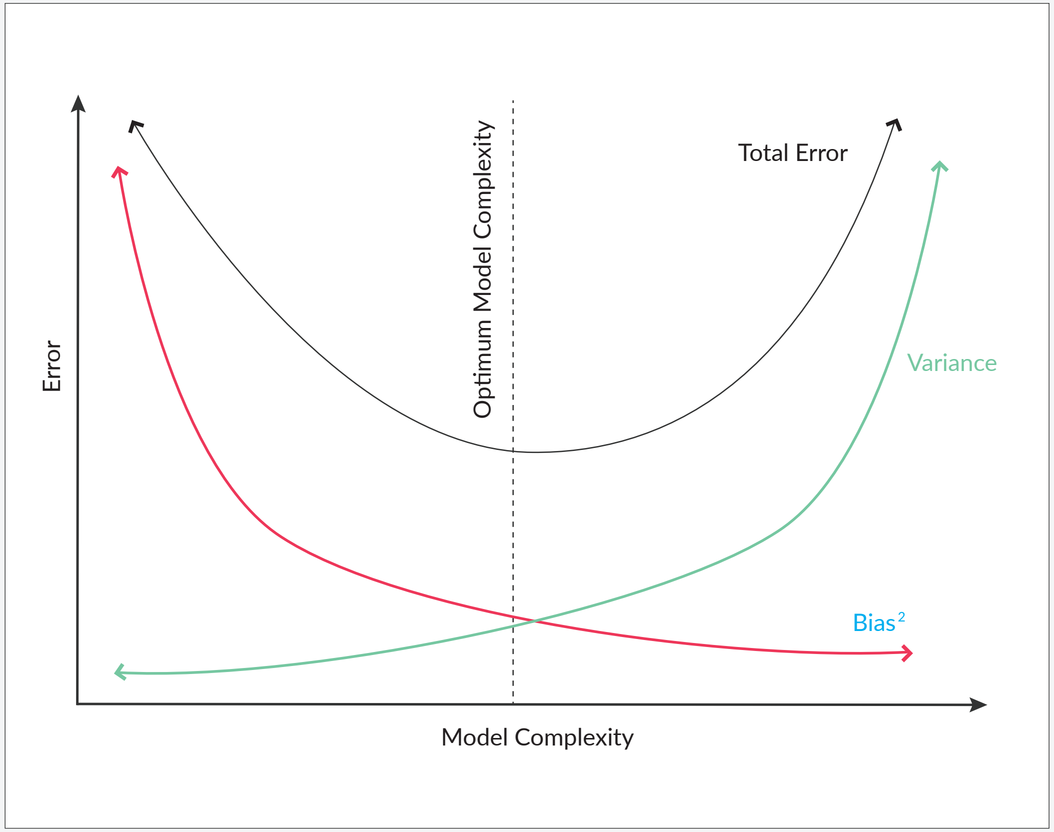
When a model performs really well on the data that is used to train it, but does not perform well with unseen data, we know we have a problem: overfitting. Such a model will perform very well with training data and, hence, will have very low bias; but since it does not perform well with unseen data, it will show high variance.
bias in a model is high when it does not perform well on the training data itself, and variance is high when the model does not perform well on the test data. Please note that a model failing to fit on the test data means that the model results on the test data varies a lot as the training data changes. This may be because the model coefficients do not have high reliability.

What we need is lowest total error, i.e., low bias and low variance, such that the model identifies all the patterns that it should and is also able to perform well with unseen data.

For this, we need to manage model complexity: It should neither be too high, which would lead to overfitting, nor too low, which would lead to a model with high bias (a biased model) that does not even identify necessary patterns in the data.

When we use regularisation, we add a penalty term to the model's cost function.

Here, the cost function would be Cost = RSS + Penalty.



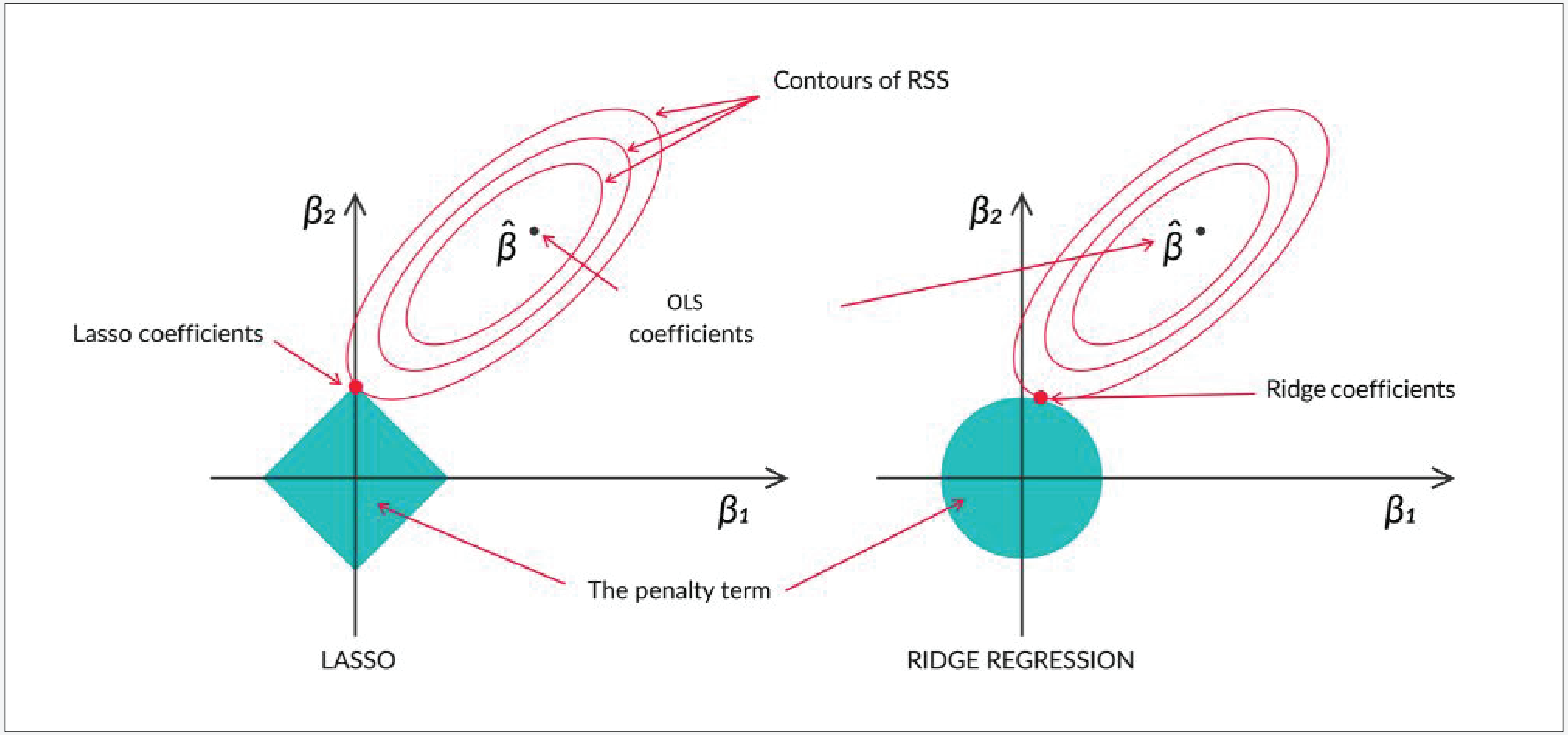
Ridge Vs Lasso Regularisation

Ridge Regularisation:

- Ridge regression has a particular advantage over OLS when the OLS estimates have high variance, i.e., when they overfit. Regularisation can significantly reduce model variance while not increasing bias much.
- The tuning parameter lambda helps us determine how much we wish to regularise the model. The higher the value of lambda, the lower the value of the model coefficients, and more is the regularisation.
- Choosing the right lambda is crucial so as to reduce only the variance in the model, without compromising much on identifying the underlying patterns, i.e., the bias.
- It is important to standardise the data when working with Ridge regression.

Lasso Regularisation:

- The behaviour of Lasso regression is similar to that of Ridge regression.
- With an increase in the value of lambda, variance reduces with a slight compromise in terms of bias.
- Lasso also pushes the model coefficients towards 0 in order to handle high variance, just like Ridge regression. But, in addition to this, Lasso also pushes some coefficients to be exactly 0 and thus performs variable selection.
- This variable selection results in models that are easier to interpret.


Contours of RSS
β2
$\hat{\beta}$
OLS
coefficients
Lasso coefficients
β1
The penalty term
LASSO
β2
$\hat{\beta}$
Ridge coefficients
β1
RIDGE REGRESSION