Linear Regression

Vishwa Raghava Reddy Mamindla

ES21BTECH11019

Linear Regression using Gradient Descent is an algorithm that is used to measure the linear relationship between the independent variable(s) of data and the target variable, i.e., output.

Conditions for using the Linear Regression:

- Linear Regression assumes a **linear relationship** between the target variable and the outcome variable. If not, we can extend it to handle nonlinear relationships by using transformations, such as polynomials, logarithms, or exponentials, on the independent variables. This can improve the fit and flexibility of the model, but also increase the risk of overfitting or multicollinearity.
- It assumes little **to no Multicollinearity and Autocorrelation in error terms**. To deal with correlation in error terms, which reduces the accuracy of time series models, some solutions are error correction models and ARIMA models. These methods can provide unbiased and efficient estimates.
- **No Heteroscedasticity**, it is a problem that occurs when the variance of the error terms in a time series model is not constant. This can distort the estimates of the model, as outliers or inflation can have too much weight. To detect and deal with heteroscedasticity, some methods are graphical analysis, statistical tests, and ARCH models.
- **Normal distribution of error terms** is a common assumption in statistical models, as it makes the estimation and inference easier. This can be violated by outliers or non-linearities, which can make the confidence intervals unstable and the estimates inaccurate. The standard error can be used to measure how much the estimates vary from the true values.

We can see two types of Linear Regression,

- Simple Linear Regression
- Multiple Linear Regression

Simple Linear Regression

Simple Linear Regression is a method to find how two continuous variables are related, i.e., how the independent variable affects the dependent variable. The main idea of it is to find a line that fits the data well, by minimizing the loss function.

Let's consider the line equation to be,

$$y_{pred} = mx + c$$

Here, the slope,

- m > 0 implies the positive(direct) relationship between X and Y, i.e., independent variable and dependent variable respectively.
- And m < 0 implies the negative(inverse) relationship.

And the intercept, c, is the value of Y when X = 0.

Loss Function

The loss function is a method to evaluate how your algorithm models the data. It, basically, computes the distance between the current output of the algorithm and the expected output.

Types of Loss functions are,

- **Classification:** Log loss, Focal loss, KL Divergence, Exponential loss, Hinge loss.
- **Regression:** Mean Square error, Mean Absolute error, Huber loss, Log Cosh loss, Quantile loss.

The loss function, Mean Squared error can be expressed as,

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - y_{pred}\right)^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - mx - c)^2$$

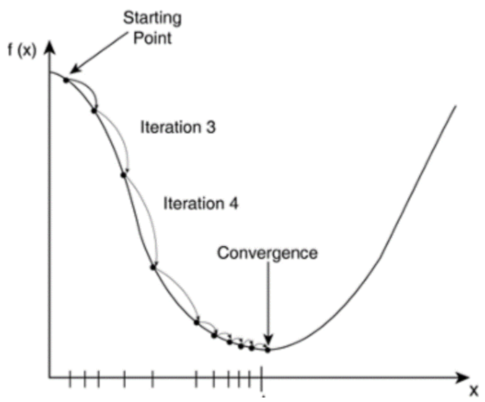
Here, n is the number of samples in the dataset.

Gradient Decent

Gradient Decent, here, we use it to minimize the loss function by optimizing it iteratively. Learning rate(α) is the parameter that is used to vary the values m and c, by custom amount, in each iteration.

Learning rate should be chosen in a way, such that, it should neither be too low which never lets the error to reach the minima nor too high which makes huge jumps skipping the minima in every step.

The working of gradient decent in a simple linear regression with an optimal learning rate looks as follows:



First, the initial values should be selected, we then find the partial derivates of the loss function, with respect to both m and c, by applying the chain rule.

$$D_m = \left(\frac{-2}{n}\right)\left(\sum_{i=1}^{n} x_i(y_i - y_{pred}\right)$$

$$D_c = \left(\frac{-2}{n}\right)\left(\sum_{i=1}^{n}(y_i - y_{pred}\right)$$

Next, the role of learning rate is to increase or decrease by certain amount of D_m and D_c.

$$m = m \pm \alpha(D_m)$$

$$c = c \pm \alpha(D_c)$$

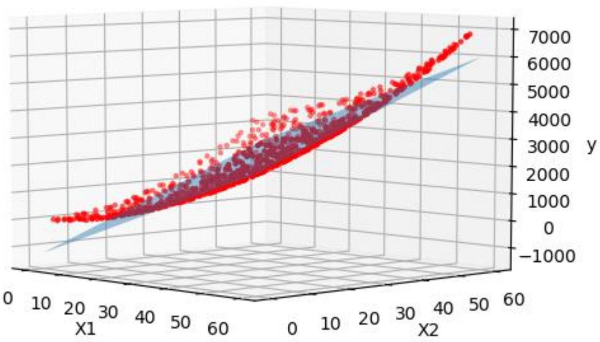
This repeats iteratively until the loss reaches 0(ideally) and the line corresponding to last updated values of m and c is the best fit regression line.

Multiple Linear Regression

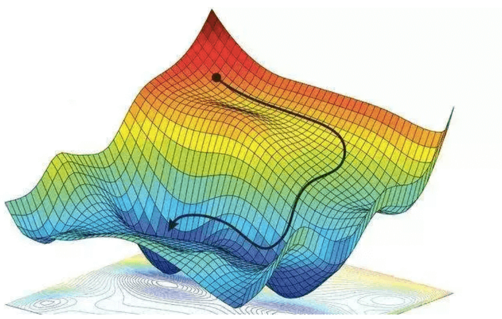
Simple Linear Regression can be extended to Multiple Linear Regression, in which, there will be more than one independent variables.

$$y_{pred} = m_1x_1 + m_2x_2 + \ldots + c$$

The visualization for a two-independent-variables data is as follows,



The same Gradient descent applies here as well, but the graph is in the 3d format as follows,



We iterate penalizing the values m_i and c until the minima is reached.

Note: One potential problem in here is, the linear regression can face challenges when the number of independent variables is larger than the number of observations, which can lead to overfitting or ill-defined solutions.