Linear regression

Case Study – Performance Parameters

What is Linear Regression?

The modeling between the dependent and the independent variable is called the linear regression model method.

Consider a simple linear regression model as $y = B_0 + B_1X + E$

Where, y is the **dependent variable**, X is the **independent variable**, and the terms (B_0 and B_1) are the parameters of the model. Based on the equation of a straight line, we infer that B_0 is the intercept and B_1 is the slope. These parameters are also called **regression coefficients**.

The one parameter which might not be observable in the graph is E, which denotes error in the data which did not lie on the straight line and represents the distance between observed y and predicted y.

The independent variables are **inferred** and **tweaked** by the analyst to obtain better results.

Assessing the Linear Regression equation

We have a regression equation, i.e., $y = B_0 + B_1X+E$

But how good is the equation at predicting values of y, for given independent variables (X), for that we have the following **performance parameters.**

R^2 :

This is the measure of association; It represents the percentage of **variance** in values of y that can be explained by knowing the values of X. r^2 values from a low of 0.0 to the highest as 1.0.

s.e.b. :

s.e.b is the standard error of the predicted value of B_1. A t-test for **statistical significance** of the coefficient is conducted by dividing the value of B_1 by its standard error.

F

F is a test for statistical significance of the regression equation. It is obtained by dividing the explained variance by the unexplained variance. Ideally, an F-value of greater than 4.0 is usually statistically significant but you must consult an F-table to be sure. If F is significant, then the regression equation helps us to understand the relationship between X and Y.

Assumptions of Linear Regression

In theory, there are several important assumptions that must be satisfied if linear regression is to be used. These are:

1. Both the independent (X) and the dependent (Y) variables are measured at the interval or ratio level.

2. The relationship between the independent (X) and the dependent (Y) variables is linear.

3. Errors in prediction of the value of Y are distributed in a way that approaches the normal curve.

4. Errors in prediction of the value of Y are all independent of one another.

5. The distribution of the errors in prediction of the value of Y is constant regardless of the value of X.

Known Methods of Linear Regression :

Least Square Method

Let's take a sample size n, full of observations x_i to y_i where (i=1,2,3…n)

These will satisfy our previous linear regression model :

$y_i = B_0 + B_1x_i + E_i$ (i=1,2,3…n)

The main goal behind least squares estimates our B_0 and B_1 by minimizing the sum of squares of the distance between the observations and the line in our scatter plot. When the vertical difference between the observations and the line in our scatter plot is taken, it's sum of squares is minimized to obtain our estimated B_0 and B_1, this method is also known as **direct regression.**

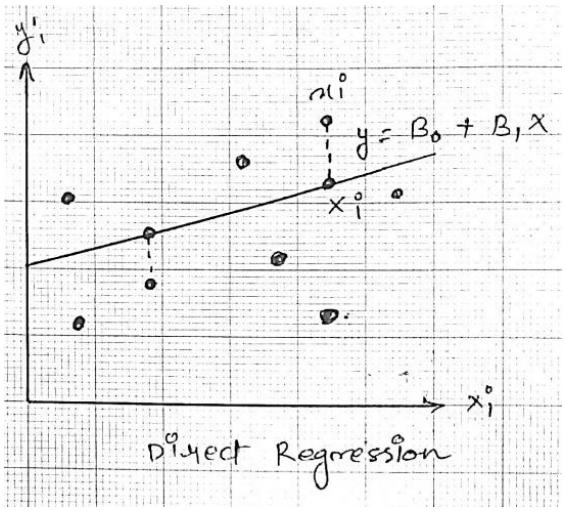
Alternatively, the sum of squares of the differences between the observations and the line in the horizontal direction in our scatter plot can also be minimized to obtain our estimated regression coefficients, this is known as inverse regression.

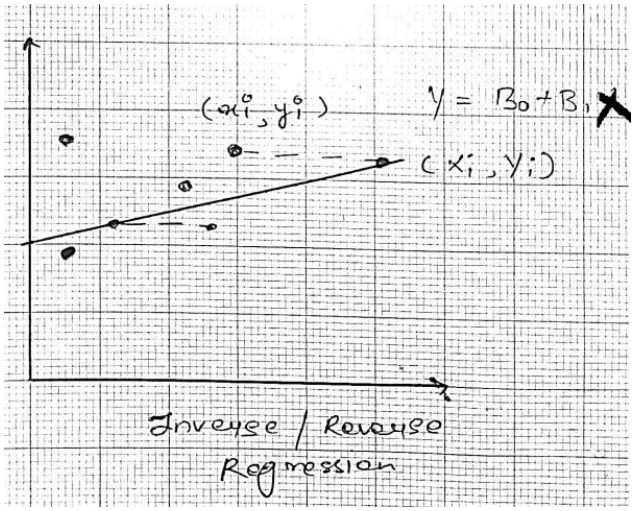
Advantages of LSM –

- Easy to use and understand.
- Applicability : Able to integrate LSM in all fields available
- Theoretical Underprinting : Maximum-Likelihood solution

Limitations of LSM –

- Sensitivity to outliers
- Test statistics might be unreliable when the data is not normally distributed (but with many datapoints that problem gets mitigated)
- Tendency to overfit data (LASSO or Ridge Regression might be advantageous)





Gradient Descent :

Now, let's suppose we have our data plotted out in the form of a scatter graph, and when we apply a cost function to it, our model will make a prediction. Now this prediction can be very good, or it can be far away from our ideal prediction (meaning its cost will be high). So, to minimize that cost (error), we apply gradient descent to it.

Now, gradient descent will slowly converge our hypothesis towards a global minimum, where the cost would be lowest. In doing so, we must manually set the value of alpha, and the slope of the hypothesis changes with respect to our alpha's value. If the value of alpha is large, then it will take big steps. Otherwise, in the case of small alpha, our hypothesis would converge slowly. (code and analysis available on github)

Advantages of Gradient Descent :

- Computational Efficiency
- Produces a stable error gradient.
- Stable convergence

Limitations of Gradient Descent :

- The stable error might result in a state of convergence that isn't the best the model can achieve.
- Requires the entire training dataset be in memory and available to the algorithm.

Adam Optimizer :

ADAM, which stands for Adaptive Moment Estimation, is an optimization algorithm that is widely used in Deep Learning.

It is an iterative algorithm that works well on noisy data.

It is the combination of RMSProp and Mini-batch Gradient Descent algorithms.

In addition to storing an exponentially decaying average of past squared gradients, Adam also keeps an exponentially decaying average of past gradients, similar to momentum.

We compute the decaying averages of past and past squared gradients respectively as follows:

$m_t = B_1 m_{t-1} + (1-B_1) g_t$

$v_t = B_2 v_{t-1} + (1-B_2) g^2_t$

As mt and vt are initialized as vectors of 0's, the authors of Adam observe that they are biased towards zero, especially during the initial time steps, and especially when the decay rates are small (i.e., B_1B_1 and B_2B_2 are close to 1).

They counteract these biases by computing bias-corrected first and second-moment estimates:

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$$
$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

and then update it with $$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon}\hat{m}_t.$$

Pseudo code for Adam optimizer :

Require: α: Stepsize
Require: $\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates for the moment estimates
Require: $f(\theta)$: Stochastic objective function with parameters θ
Require: θ_0: Initial parameter vector
  $m_0 \leftarrow 0$ (Initialize 1st moment vector)
  $v_0 \leftarrow 0$ (Initialize 2nd moment vector)
  $t \leftarrow 0$ (Initialize timestep)
  while θ_t not converged **do**
    $t \leftarrow t + 1$
    $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Get gradients w.r.t. stochastic objective at timestep t)
    $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)
    $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased second raw moment estimate)
    $\widehat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)
    $\widehat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)
    $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \widehat{m}_t / (\sqrt{\widehat{v}_t} + \epsilon)$ (Update parameters)
  end while
  return θ_t (Resulting parameters)

Advantages of Adam Optimizer :

- Easy to implement.
- Quite computationally efficient.
- Requires little **memory** space.
- Good for non-stationary objectives.
- Works well on problems with noisy or sparse gradients.
- Works well with large data sets and large parameters.

Limitations of Adam Optimizer :

- Adam does not converge to an optimal solution in some.
- Adam can suffer a weight decay problem.
- Recent optimization algorithms have been proven faster and better.

References :

https://towardsdatascience.com/complete-guide-to-adam-optimization-1e5f29532c3d

https://www.tech-quantum.com/adam-optimization-algorithms-in-deep-learning/

https://builtin.com/data-science/gradient-descent

https://www.quora.com/What-are-the-advantages-and-disadvantages-of-least-square-approximation

https://github.com/

https://builtin.com/data-science/gradient-descent

https://www.kdnuggets.com/

http://home.iitk.ac.in/