Team 07: Assignment 01
Linear Regression with MSE and Huber Loss

Vatsal Jitendra Patel, Latika Liladhar Dekate, and Lakshminarayanaa Rajamannar
Emails: vpate160@asu.edu, ldekate@asu.edu, lrajaman@asu.edu

***Abstract*— The report deals with fitting a linear regression model to a dataset with outliers. The analysis is about implementing and comparing two different loss functions namely, standard Mean Squared Error, and Huber Loss. To find the best model, gradient descent algorithm is being used. The impact of different learning rates (0.01, 0.1, 0.5) and two termination criteria (loss and parameter convergence) on the training process is also studied in this report.**

***Index Terms*— Linear Regression, Gradient Descent, Mean Squared Error (MSE), Huber Loss, Learning Rate, Outlier Robustness.**

I. INTRODUCTION

Linear regression, a fundamental model that is used in machine learning to model the relationship between the variables in the sample dataset. It determines a line that best fits the data points by minimizing a "loss function" which measures the distance between the line predictions from the actual data.

This project explores the impact of the loss function on the model's performance. The comparison between the most common loss function, Mean Squared Error (MSE), with Huber Loss is analyzed in this project. MSE is simple and works on clean data, but sensitive to outliers. Huber loss is less sensitive to outliers, hence robust.

The complete Python code used to run the experiments, generate the plots, and perform the analysis is contained in the Jupyter Notebook, which can be accessed via the following link: GitHubCodelink_Team7

II. INTRODUCTION TO THE PROBLEM

With the given dataset samples with input values in the range [0.037, 0.986] and corresponding outputs in the range [0.307, 1.158]. The true underlying relationship is known to be y = 0.67x + 0.5

There are also outliers in the dataset. The scatter plot below shows the relationship between x and y.

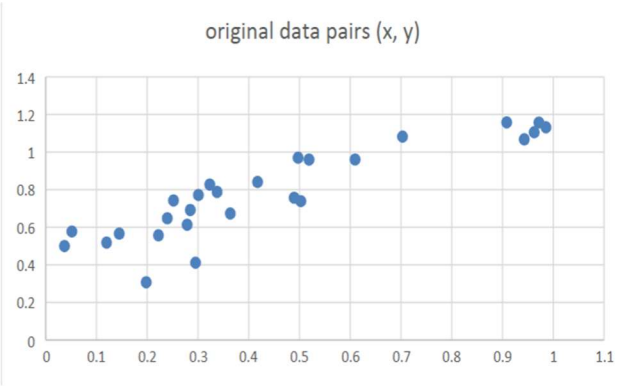

Figure 1: Scatter plot of the dataset

In Figure 1, most points follow a clear line, but a few are scattered away from this trend. These outliers can cause problems for a standard linear regression model that uses MSE, because MSE heavily penalizes large errors. This is the main challenge we are trying to address. Our goal is to find the parameters w (weight) and b (bias) for the line y = wx + b that helps us represent this data, even with the outliers present.

III. APPROACH/METHOD

In this assignment, we have used gradient descent (GD) algorithm, so that we can find w and b. The main approach of this algorithm is to use random variables for these parameters, then iterate and update them, in a way that minimizes the loss function with respect to each parameter.

A. Equations

1) Mean Square Error Loss Function:

i) $L_{\text{MSE}} = \frac{1}{2n}\sum_{i=1}^{n}\left(y_i - (w\,x_i + b)\right)^2$

- Gradients (updating rules):

To update our parameters, we calculate how the loss changes as we change w and b.

ii) $\frac{\partial L_{\text{MSE}}}{\partial w} = -\frac{1}{n}\sum_{i=1}^{n} x_i(y_i - wx_i - b)$

iii) $\frac{\partial L_{\text{MSE}}}{\partial b} = -\frac{1}{n}\sum_{i=1}^{n}(y_i - wx_i - b)$

2) Huber Loss Function:

The Huber Loss function is a hybrid. For small errors, it acts like MSE (quadratic), but for large errors, it acts like Mean Absolute Error (linear). This makes it less sensitive to outliers. We define a threshold to switch between these two behaviors.

iv) $Let\ r\ be\ residual, r = y_i - (wx_i + b)$

v) $L_{Huber}(\mathrm{r}) = \begin{cases} \frac{1}{2}r^2 & if\ |r| \le \delta \\ \delta|r| - \frac{1}{2}\delta^2 & if\ |r| > \delta \end{cases}$

- Gradients (Updating rules):
  Gradients also depend on whether the error is smaller or larger than δ.

vi) $\frac{\partial L_{\mathrm{Huber}}}{\partial w} = -\frac{1}{n}\sum_{i=1}^{n} x_i\, g_i$

vii) $\frac{\partial L_{\mathrm{Huber}}}{\partial b} = -\frac{1}{n}\sum_{i=1}^{n} g_i$

Where:

viii) $g_i = \begin{cases} r_i & if\ |r_i| \le \delta \\ \delta \cdot sign(r_i) & if\ |r_i| > \delta \end{cases}$

For our experiments, we set $\delta = 0.1$

B. Algorithms

The gradient descent (GD) algorithm uses the updating rules derived above to iteratively find the optimal values for the weight `w` and bias `b`. The process is analogous to descending a hill to find its lowest point; at each step, the algorithm determines the direction of steepest descent (the negative gradient) and takes a small step to move closer to the minimum of the loss function.

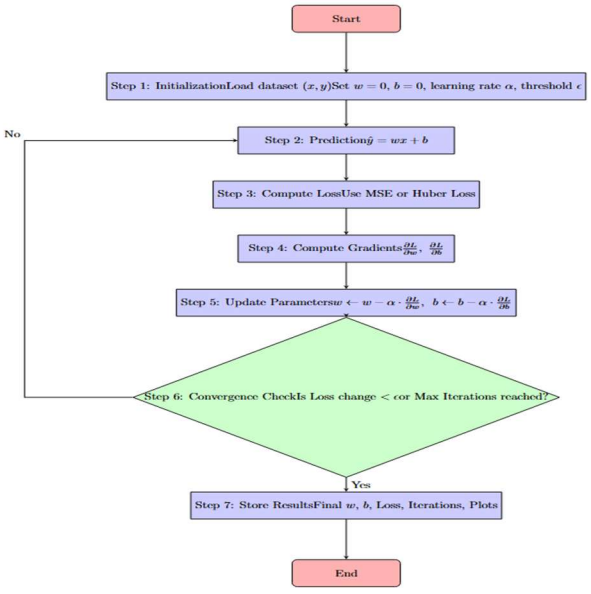

Figure 2: Flowchart representation of the algorithm

IV. Results/Evaluations

A. Experimental stage

For this analysis a total of 12 experiments were executed to compare the performance of the regression model under various conditions. The combination consists of 2 Loss functions (MSE and Huber Loss); 3 Learning rates (0.01 - slow, 0.1 - medium, 0.5 - fast); 2 Termination criteria.

The Loss Convergence stops when loss stops changing much between steps. The Parameter Convergence stops when w and b themselves stop changing much. The convergence threshold is set to 1e-6 for both criteria.

B. Results

The table below summarizes the final parameters, loss, and number of iterations for all 12 experiments. The true parameters for the dataset are known to be w = 0.67 and b = 0.5.

Table 1: Summary of all 12 experimental runs.

Experiment	Final w	Final b	Final Loss	Iterations
MSE_LR0.01_loss	0.6020	0.5200	0.006280	1695
MSE_LR0.01_params	0.7321	0.4581	0.005553	8398
MSE_LR0.1_loss	0.6921	0.4771	0.005625	338
MSE_LR0.1_params	0.7332	0.4576	0.005553	1174
MSE_LR0.5_loss	0.7159	0.4658	0.005566	92
MSE_LR0.5_params	0.7334	0.4575	0.00555	280
HUBER_LR0.01_loss	0.5510	0.5577	0.004998	2449
HUBER_LR0.01_params	0.6952	0.4864	0.004177	9816
HUBER_LR0.1_loss	0.6518	0.5076	0.004255	434
HUBER_LR0.1_params	0.6965	0.4858	0.004177	1346
HUBER_LR0.5_loss	0.6771	0.4953	0.004192	113
HUBER_LR0.5_params	0.6966	0.4857	0.004177	319

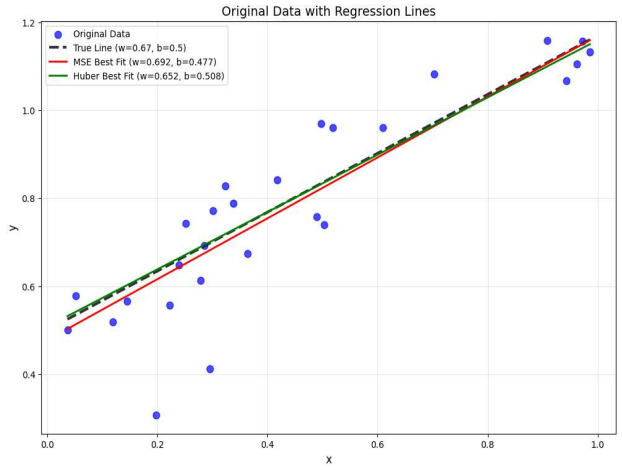

Figure 3. Dataset with Optimal Best Fit Regression Lines by MSE and Huber Loss

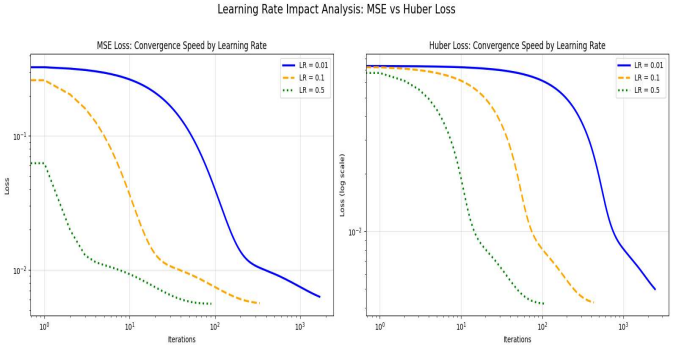

Figure 4: Visualization of Convergence speed affected by Learning Rates

V. Discussions and Conclusions

After understanding and discussing the result, we saw that Mean Squared Error (MSE) is sensitive to outliers present in data, because it squares the errors, creating a unbiased result that pulls the line toward them. In contrast, Huber Loss is more robust because it treats large errors linearly, so it isn't thrown away as much by these outlier points.

Further, after using two termination criteria and three learning rates to train our model, we observed:

For learning rates we used: a low LR (0.01) was very slow, taking thousands of iterations to get to the answer; a high LR (0.5) was super-fast, but for more complex problems, this could be risky and lead to overfitting of the model; and a medium LR (0.1) was among the best rate for our problem, giving us faster convergence without sacrificing accuracy.

For the convergence part, we noticed that using loss convergence was much better than parameter convergence, as a stopping rule. This makes sense because the loss often flattens out while the parameters are still making tiny final adjustments that don't really improve the model in a meaningful way. For practical purposes, stopping when the loss stabilizes is a good approach.

Table 2: Comparison of the Model Performance

Metric	True Parameters	Best MSE Model	Best Huber Model
w (weight)	0.67	0.692	0.677
b (bias)	0.5	0.477	0.495
Total Error	0	0.045	0.012

This helped us realize that just choosing a standard method like MSE isn't always the best approach, and understanding the data is crucial for building a good model. This assignment showed clearly how different loss functions and training settings affect a linear regression model, especially on data with outliers.