# Report

Ankit Kumar Bhakar

22M1077

## Linear Regression from Scratch 1 Introduction to Linear Regression

Linear regression is a fundamental statistical technique used for modeling the relationship between a dependent variable and one or more independent variables. This report delves into the concepts of linear regression, model evaluation, and various techniques used to analyze and optimize linear regression models. The implementation of linear regression from scratch using Python is also explored.

## 2 Model Implementation

To understand linear regression, we implemented it from scratch using Python. We generated input data matrix X and target variable t. X was generated using a normal distribution with zero mean and variance 5. The target variable was calculated using a linear combination of X and random noise. We also created functions to calculate mean squared error (MSE), weight estimation using the pseudo-inverse method, gradient descent optimization, and other essential components of linear regression.

## 3 Model Evaluation

Model evaluation is crucial to determine the performance and generalization capability of the model. We focused on two key metrics: the root mean squared error (RMSE) and the normalized RMSE (NRMSE). RMSE quantifies the average error between predicted and actual values, while NRMSE normalizes RMSE with respect to the variance of the target variable, making it more interpretable and comparable across different datasets.

## 4 Analysis and Observations

Several experiments were conducted to analyze the behavior of linear regression under various conditions:

Effect of Training Samples: Increasing the number of training samples led to a decrease in NRMSE, showing better generalization with more data points.

Impact of Dimensions: As the dimension of the feature space increased, NRMSE decreased sharply. This suggests that more dimensions lead to a better fit for the data.

Influence of Noise Variance: Higher noise variance resulted in increased NRMSE, indicating that noise negatively impacts model accuracy.

Role of Regularization (Lambda): The choice of lambda influenced the model's performance. Higher lambda led to lower NRMSE, but very high values caused instability.

Optimal Learning Rate (Eta): Proper choice of the learning rate is crucial. Small values led to slow convergence, while large values resulted in divergence. An optimal range around 0.01 proved effective.

Regularization (Lambda) Impact on Nearly Zero Weights: For nearly zero initial weights, regularization effectively controlled NRMSE.

Optimal Lambda with Noise Variance: Optimal lambda reduced NRMSE, and its influence was more pronounced with higher noise variance.

## Comparison with Pseudo-Inverse and Gradient Descent

We compared the two main methods of estimating weights: pseudo-inverse and gradient descent. Pseudo-inverse was faster for small datasets, while gradient descent showed better adaptability and scalability. Gradient descent's performance improved with increasing training samples.

## 6 Conclusion

Linear regression is a foundational technique in machine learning, providing insights into relationships between variables. Through our analysis, we've learned that var- ious factors like training samples, dimensions, noise variance, regularization, and learning rate influence the model's behavior. Understanding these nuances is crucial for effectively applying linear regression in real-world scenarios. By implementing the model from scratch and conducting experiments, we've gained practical insights into its inner workings and optimization techniques.

## Resources


- https://towardsdatascience.com/simple-linear-regression-in-python-numpy-only- 130a988c0212
- https://towardsdatascience.com/gradient-descent-from-scratch-e8b75fa986cc
- https://youtu.be/xrPZbHrxrWo
- https://towardsdatascience.com/linear-regression-using-gradient-descent-97a6c8700931
- Boxplot code taken from : https://matplotlib.org/stable/gallery/statistics/boxplot$_d$emo.html