Machine Learning Course

Linear Regression Assignment Report

Professor:
Dr. Mahdi Eftekhari

Author:
Amirhossein Abolhassani

Fall 2024

Contents

1 Introduction [1]

Linear regression is one of the simplest and most widely used methods in supervised learning for modeling the relationship between a dependent variable and one or more independent variables. Assuming a linear relationship between input features and the target variable, linear regression aims to find the best-fitting line that minimizes prediction error.

When only one independent variable is present, the method is called simple linear regression. However, real-world problems often involve multiple factors affecting the target variable. In such cases, multiple linear regression is used, where the model considers several independent variables to predict the dependent variable. Mathematically, this relationship is modeled as:

$$y = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + \epsilon$$

Here, y is the predicted value, w_0 is the intercept, $w_1, w_2, \ldots, w_n$ are the coefficients of the independent variables $x_1, x_2, \ldots, x_n$, and ϵ is the model's error.

In practice, linear regression can be solved using various techniques, such as the Closed Form Solution, which minimizes the Mean Squared Error (MSE), or optimization algorithms like gradient descent. Additionally, regularization techniques like L_2 regularization (Ridge Regression) can prevent overfitting by penalizing large coefficient values. These methods ensure model stability, especially when dealing with multicollinearity or noisy data.

This report explores simple and multiple linear regression, focusing on the closed-form solution with and without L_2 regularization, as well as optimization using Stochastic Gradient Descent and Batch Gradient Descent. The performance of these methods is compared using training and test data.

2 Simple Linear Regression

2.1 What is the Cost Function for Linear Regression?

The cost function for linear regression, known as Mean Squared Error (MSE), is defined as:

$$J(\theta) = \frac{1}{n} \sum_{i=1}^{n} (y^i - \hat{y}^i)^2$$

2.2 Training Linear Regression

2.2.1 Closed-Form Solution

In the closed-form solution, the following equation is solved to find θ:

$$\theta = (X^T X)^{-1} X^T \vec{y}$$

However, $X^T X$ may not be invertible (as observed with the training data in this assignment). In such cases, the Moore-Penrose pseudo-inverse is used:

$$\theta = (X^T X)^{+} X^T \vec{y}$$

The resulting model:

$$y = \theta_0 + \theta_1 x$$

$$\theta_0 = -3.8957, \theta_1 = 1.1930$$

[1] The introduction is written with GPT-3.

2.2.2 Stochastic Gradient Descent

This method uses Stochastic Gradient Descent (SGD) to optimize the cost function.
The resulting model:

$$y = \theta_0 + \theta_1 x$$

$$\theta_0 = -3.8481, \theta_1 = 1.0570$$

2.2.3 Batch Gradient Descent

This method uses Batch Gradient Descent (BGD) to optimize the cost function.
The resulting model:

$$y = \theta_0 + \theta_1 x$$

$$\theta_0 = -3.5858, \theta_1 = 1.1619$$

2.3 Visualizing the Model with Data

The models obtained from all three methods are plotted alongside the data in Figure 1. Batch Gradient Descent performs better than Stochastic Gradient Descent.

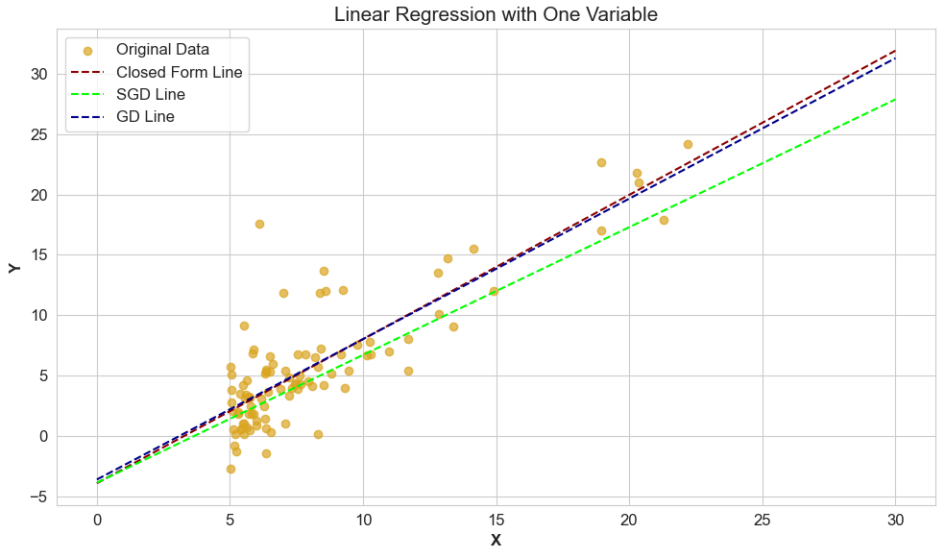

Figure 1: Lines obtained from the three methods

2.4 Comparing Model Predictions

After training, each model is tested with the input data $X = [6.2, 12.8, 22.1, 30]$. The closeness of each model's output to the closed-form solution indicates how well the optimization method minimizes the cost function.

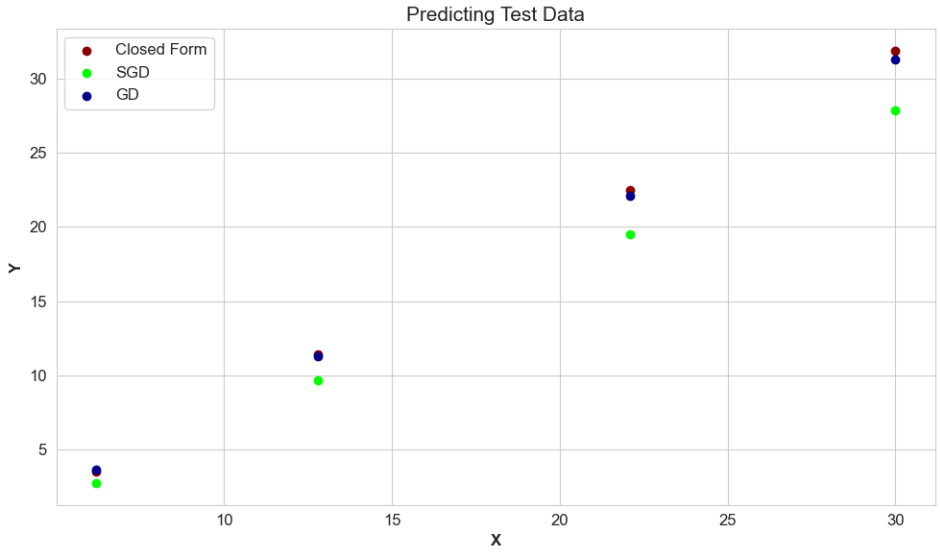

Figure 2: Model outputs for different methods

2.5 Comparing Trained Models

Since the closed-form solution provides the optimal parameters, comparing each model's line to it reveals its quality (Figure 3).

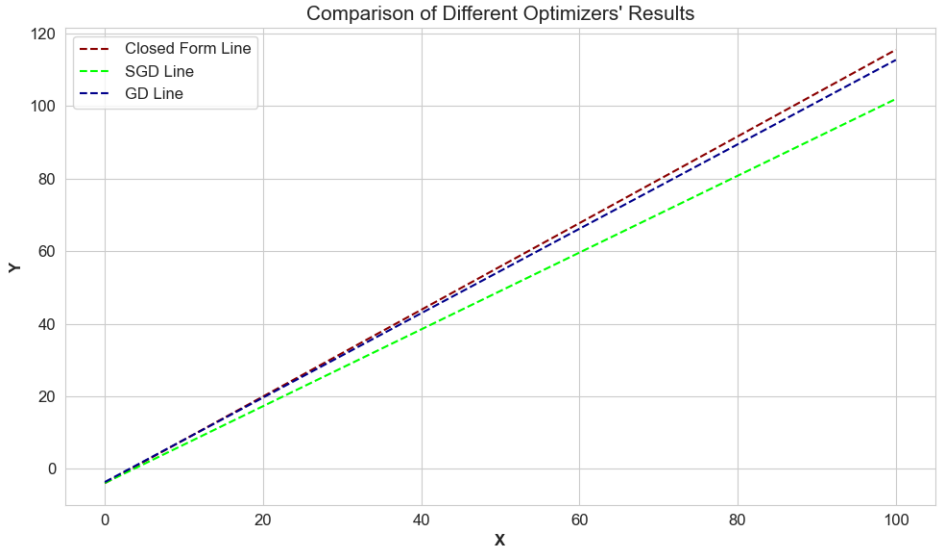

Figure 3: Model outputs for different methods

2.6 Examining Cost Function Behavior

Monitoring the cost function's behavior during training is a key way to assess model learning quality.
As shown in Figure 4, Stochastic Gradient Descent stops improving the cost function after 20 iterations, likely stuck in a local minimum. In contrast, Batch Gradient Descent approaches the global minimum more effectively.

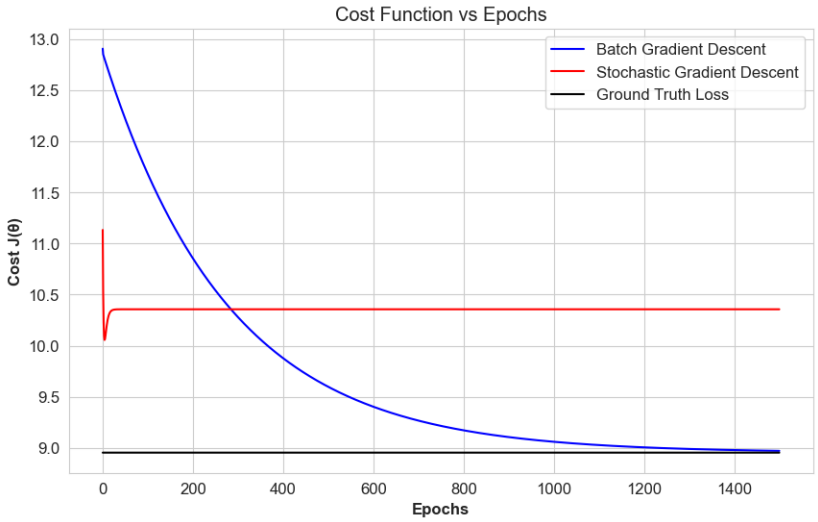

Figure 4: Plot of cost function value versus iteration during training

2.7 Which Optimization Method is Preferred?

Based on Section 2.6 and Figure 4, Batch Gradient Descent outperforms Stochastic Gradient Descent because it considers the entire dataset, allowing it to better navigate toward the cost function's minimum, especially with small datasets that fit in memory.

3 Multiple Linear Regression

3.1 Data Preprocessing

3.1.1 Data Encoding

Some dataset variables are categorical and required encoding. One Hot Encoding (OHE) was used for the *region* feature, while Integer Encoding (IE) was applied to *smoker* and *gender*.

Question: Why were different encoding methods used for these features?

IE is used when the order of feature values matters. For *gender*, which has only two values, the choice of IE versus OHE makes little difference. OHE is used for *region* because its values have no inherent order, ensuring each category is treated equally.

3.1.2 Standardization

During the assignment, it was observed that not standardizing the data caused gradient explosion and parameter instability. Thus, training data was standardized, and test data was standardized using the same parameters.

3.2 Training the Model

3.2.1 Closed-Form Solution

Since $X^T X$ (where X is the training data matrix) is not invertible, the Moore-Penrose pseudo-inverse is used. Figure 5 shows that the closed-form solution's performance improves with more training data, stabilizing after approximately 20 samples.

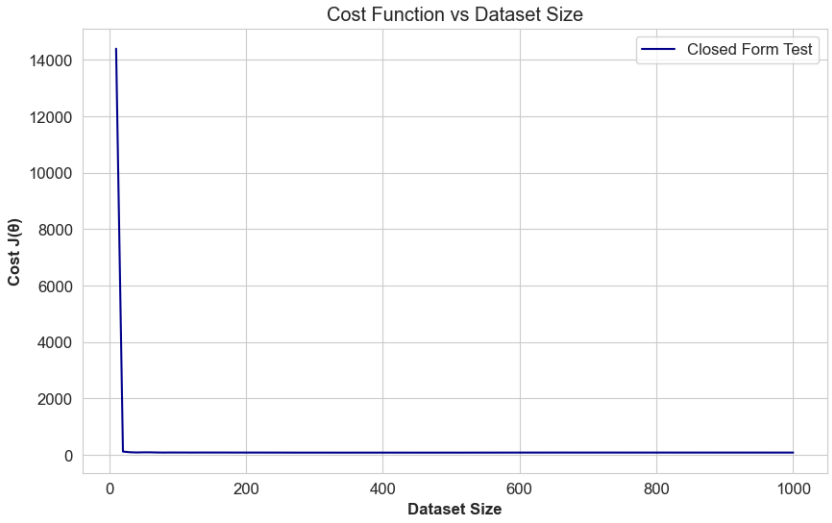

Figure 5: Plot showing mean error versus training dataset size

3.2.2 Stochastic Gradient Descent

The linear regression model is trained using Stochastic Gradient Descent. Figure 6 shows the Test Loss and Train Loss trends for training on the entire Train dataset.

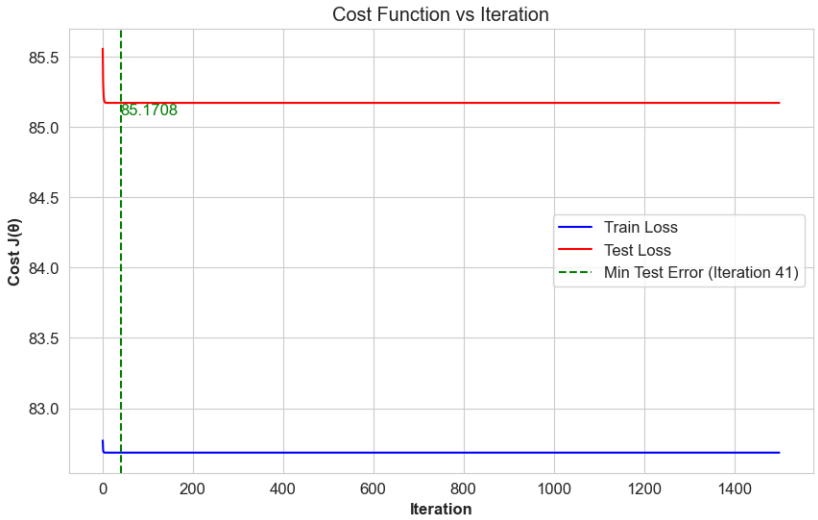

Figure 6: Plot of test and train error behavior for Stochastic Gradient Descent (lowest error at iteration 41 with value 17.85)

The mean error versus dataset size is examined in Figure 7.

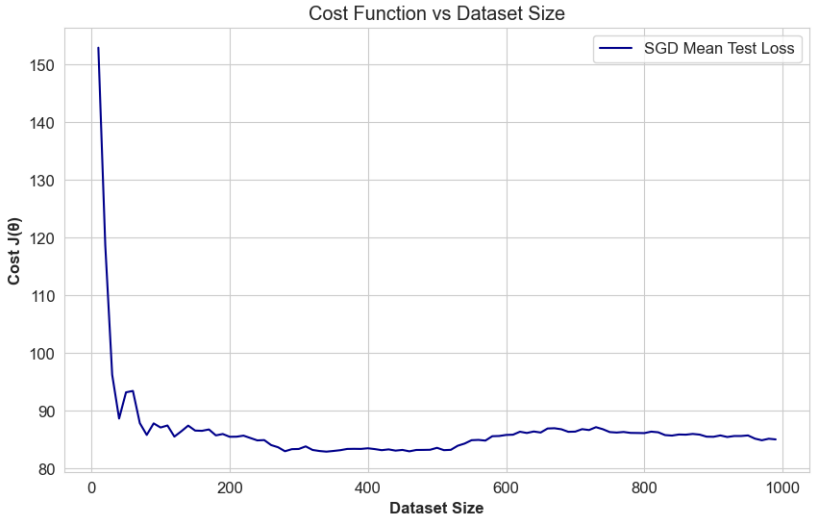

Figure 7: Mean error of the trained model across different dataset sizes

3.2.3 Batch Gradient Descent

The linear regression model is trained using Batch Gradient Descent. Figure 8 shows the Test Loss and Train Loss trends for training on the entire Train dataset.

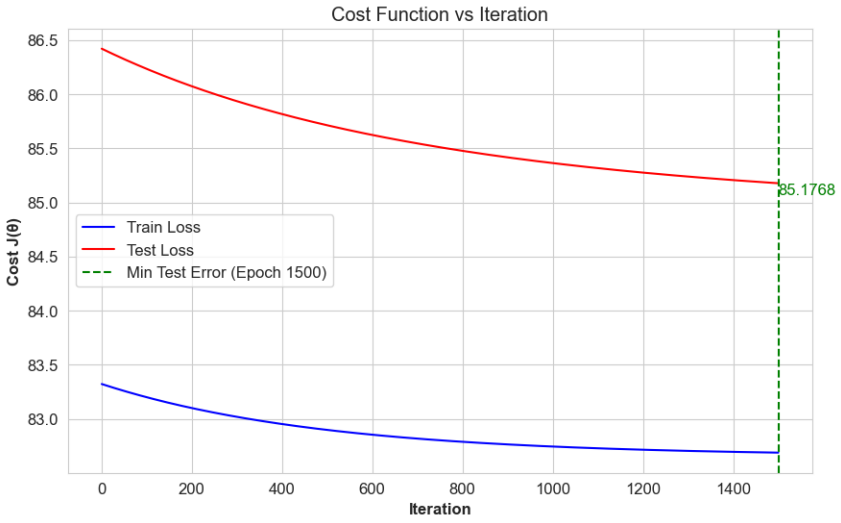

Figure 8: Plot of test and train error behavior for Batch Gradient Descent (lowest error at iteration 15 with value 17.85)

The mean error versus dataset size is examined in Figure 9.

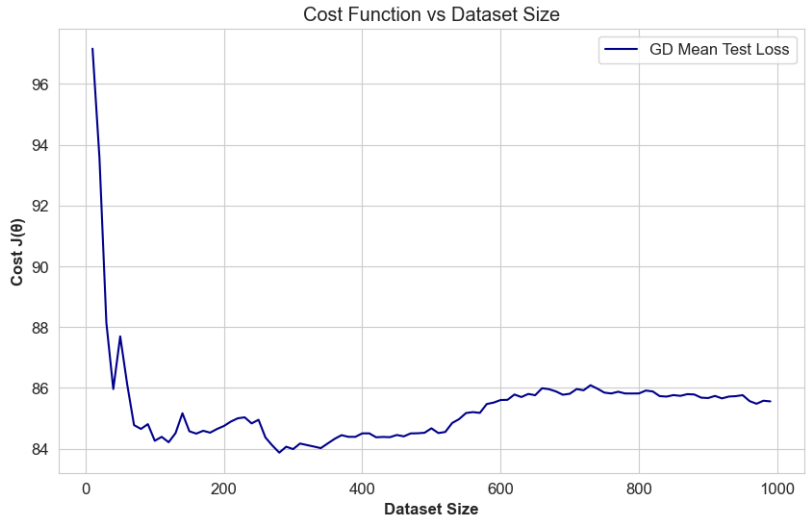

Figure 9: Mean error of the trained model across different dataset sizes

3.2.4 Comparing the Three Methods

Figure 10 shows the trend of test error reduction. The SGD-trained model converges quickly, while BGD converges more slowly but steadily. Both methods achieve errors within 2.0 of the minimum (Figure 10).

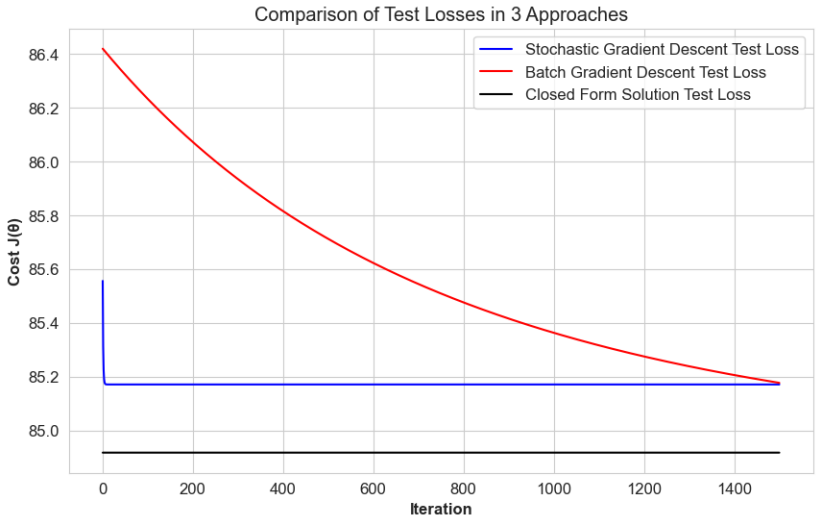

Figure 10: Test error reduction trend for SGD, BGD, and closed-form solution

Another comparison examines the mean test error versus dataset size. As shown in Figure 11, gradient-based methods initially outperform the closed-form solution.

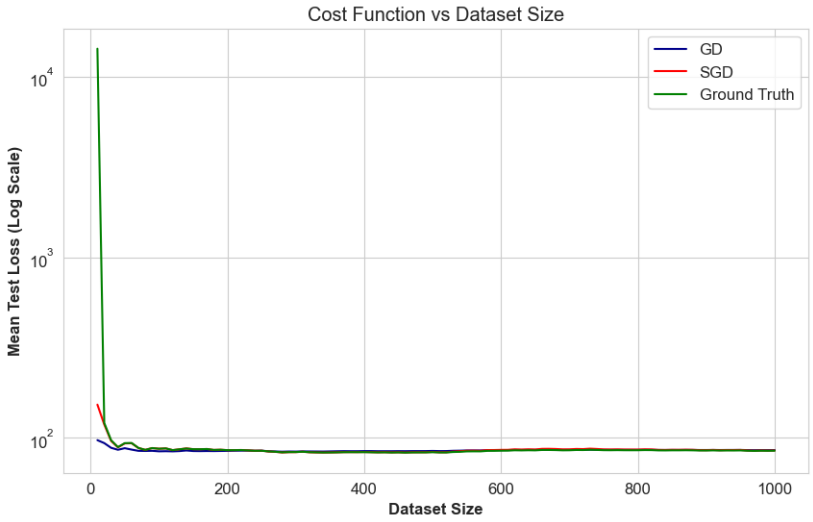

Figure 11: Mean test error versus dataset size

3.3 Adding L_2 Regularization to the Closed-Form Solution

By adding an L_2 regularization term to the cost function, we derive the following equation for θ:

$$\theta = (X^T X + \lambda I)^{-1} X^T \vec{y}$$

To find the optimal λ, the test error for parameters obtained with different λ values is shown in Figure 12.

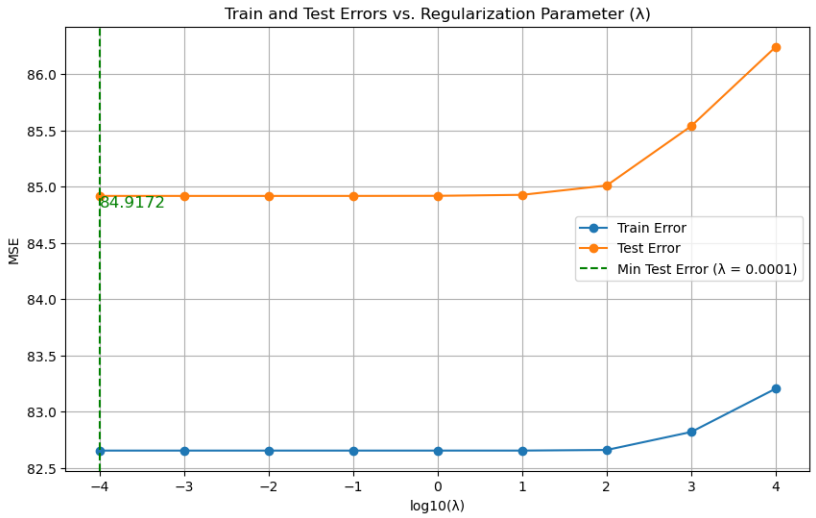

Figure 12: Mean test error for different λ values (lowest test error achieved with $\lambda = 10^{-4}$)