Comparative Analysis of Linear Regression: Gradient Descent vs. Singular Value Decomposition in Predictive Modeling

Syed Ali Vijdaan Naqvi (2022560)
Bachelors of Artificial Intelligence,
Ghulam Ishaq Khan Institute of
Engineering Sciences and Technology
alivijdaan@gmail.com

Maaz Hamid (2022655)
Bachelors of Artificial Intelligence,
Ghulam Ishaq Khan Institute of
Engineering Sciences and Technology
maazhamid12@gmail.com

Hamza Amin (2022378)
Bachelors of Artificial Intelligence,
Ghulam Ishaq Khan Institute of
Engineering Sciences and Technology
mh4070685@gmail.com

Ali Raza (2022664)
Bachelors of Artificial Intelligence,
Ghulam Ishaq Khan Institute of
Engineering Sciences and Technology
ali.dot.raza@gmail.com

***Abstract*—Linear Regression is one of the fundamental techniques used in Machine Learning. This study compares Gradient Descent and Singular Value Decomposition (SVD) for determining Linear Regression. The aim is to assess their performance, accuracy and computational efficiency. Through extensive analysis, we evaluate their strengths and limitations. This comparative analysis enhances our understanding of Gradient Descent and SVD in linear regression, offering valuable insights into their practical implications and effectiveness for practitioners in selecting appropriate techniques for predictive modeling tasks.**

Keywords*— Machine Learning, Linear Algebra, Predictive Analysis, Linear Regression, Gradient Descent, Singular Value Decomposition (SVD), Optimization Algorithms, Matrix Decomposition

I. Introduction

This project delves into the realm of predictive modeling, focusing on the pivotal technique of linear regression. The study centers on a comparative exploration between Gradient Descent and Singular Value Decomposition (SVD) methods for linear regression analysis. Our primary objective is to scrutinize their effectiveness, accuracy, and computational efficiency across varied datasets. By conducting extensive experimentation and analysis, this research aims to unravel the distinct strengths, limitations, and nuanced disparities between these two widely used approaches. This investigation sheds light on their practical implications in predictive modeling tasks, providing crucial insights for practitioners in selecting optimal techniques.

II. Methodology

A. Data Loading and Preprocessing

The movie dataset was loaded using *pyreadr* from an .RData file and converted to CSV for analysis. A scatter plot visually showcases the relationship between critics' and audience scores, offering a quick insight into their correlation in movie evaluations.

B. Gradient Descent Linear Regression

The implementation of the gradient descent algorithm was applied to determine the optimal line fitting for the association between critics' scores (independent variable) and audience scores (dependent variable). The scatter plot was enhanced by overlaying the regression line, providing a visual depiction of the relationship. The model's efficacy was assessed using Mean Squared Error (MSE). Additionally, the execution time for the Gradient Descent algorithm was measured and reported. To gauge resource utilization, memory usage for crucial variables in the Gradient Descent process was also evaluated.

C. Singular Value Decomposition Linear Regression

Singular Value Decomposition (SVD) was utilized to compute coefficients for the linear regression model. These coefficients were employed to predict audience scores accurately. The resultant regression line was visually compared with the actual data, enhancing comprehension of the model's performance. The execution time for the SVD algorithm was quantified and reported, while the memory usage of pivotal variables within the SVD process was also evaluated for resource assessment.

D. Performance Evaluation and Memory Analysis

The comparative analysis between Gradient Descent and SVD linear regression models encompassed an evaluation of their respective performance metrics. Additionally, a detailed comparison of memory usage, focusing on critical variables, was conducted for both algorithms.

E. Results Visualization

Mean Squared Error (MSE) was graphically presented against intercept and gradient values for Gradient Descent, offering insights into their relationships. Furthermore, the visualization of Linear Regression lines alongside actual data was depicted for both Gradient Descent and Singular Value Decomposition (SVD), providing a comparative view of their predictive capabilities.

F. Time Frame

1. Data Loading and Preprocessing: 1 day
2. Gradient Descent and Linear Regression: 2 days
3. SVD Linear Regression: 2 days

4. Performance Evaluation and Memory Analysis: 2 days
5. Results Visualization: 1 day

G. Literature Review

The literature review for this project draws upon Kaggle as a pivotal source for real-world datasets and fundamental principles gleaned from Linear Algebra textbooks. Kaggle, a leading platform for collaborative data-driven projects, provided the movie dataset, forming the basis for exploring the relationship between critics' scores and audience ratings. The project employs well-established techniques in machine learning, utilizing Gradient Descent for optimization and Singular Value Decomposition (SVD) for parameterization in linear regression. These methodologies align with widely acknowledged practices in the field, while the incorporation of mathematical concepts from Linear Algebra textbooks underlines the project's grounding in theoretical foundations. In essence, this literature review highlights the symbiotic relationship between practical data sourcing and theoretical rigor, contributing to a comprehensive exploration of linear regression models applied to movie ratings.

III. Linear Regression

Linear regression is a foundational statistical technique used for modeling the relationship between dependent and independent variables. It operates on the principle of fitting a linear equation to observed data points, aiming to minimize the difference between predicted and actual values.

$$\hat{y} = mx + c \tag{1}$$

Equation (1) shows the equation for Linear Regression modeled after the well-known equation for a straight line, $y = mx + c$

$\hat{y}$	Predicted values
m	Gradient
x	Data values
c	Y - intercept

Table. 1

The process of finding the linear regression revolves around the residuals or errors in our dataset. For simplicity sake, we will call them errors. These errors are in short the difference between the actual data points and the predicted values.

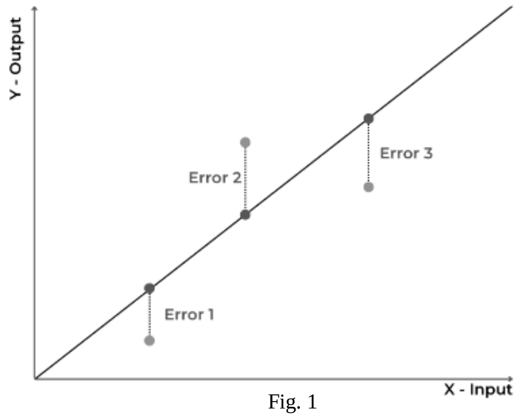

Fig. 1

Linear regression uses a special equation commonly known as the Cost Function or the Mean Squared Error Function (MSE). The steps needed to find the MSE are as follows:

1. Finding the difference between y and $\hat{y}$: $y - \hat{y}$
2. Squaring the difference: $(y - \hat{y})^2$
3. Finding the mean of all squares: $\frac{1}{n}\sum_{i=0}^{n}(y - \hat{y})^2$

These steps give our Cost Function. We can substitute $\hat{y}$ with equation (1) to give the finalized equation

$$E = \frac{1}{n}\sum_{i=0}^{n}\left(y_i - (mx_i + c)\right)^2 \tag{2}$$

Our research uses a *Movie Dataset* for the application of linear regression. The main features we have used as weights and biases for linear regression are the critic scores and the audience scores on Rotten Tomatoes of several movies.

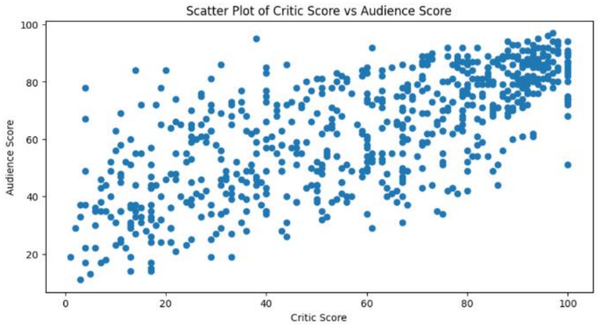

Fig. 2

IV. GRADIENT DESCENT

After passing an array of data through the MSE Functions we can plot a graph between the MSE acquired and the weight or gradient as well as a plot showing the correlation between MSE and bias or y-intercept.

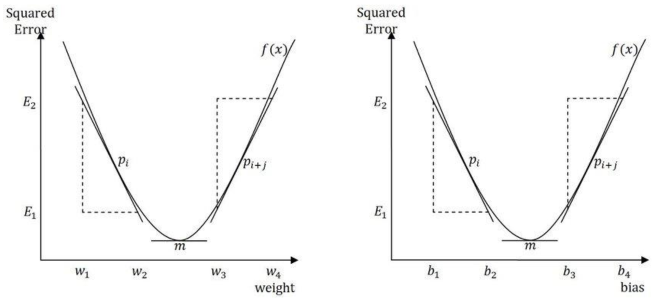

Fig. 3a & b

Our goal is to find the global minimum of the function shown above. By finding the lowest MSE we are able to figure out the optimal y-intercept and gradient needed for the best fit line of our data or in other words the regression line. This is where gradient descent can come into play.

Gradient descent is an optimization algorithm that can help us find the optimal MSE. Let us say our initial position (this position is obtained at random) at Fig. 3a is (w_4 , E_2). The goal of this algorithm is for our position to change from the current co-ordinates to where the Squared Error is at its lowest. This is done by taking small *steps* towards the global minimum. These *steps* are the partial derivative of the MSE Function w.r.t. the gradient and y-intercept.

$$\mathrm{D_m} = \frac{-2}{\mathrm{n}} \sum_{\mathrm{i}=0}^{\mathrm{n}} \mathrm{x_i}(\mathrm{y_i} - \hat{\mathrm{y}}_\mathrm{i}) \qquad (3)$$

$$\mathrm{D_c} = \frac{-2}{\mathrm{n}} \sum_{\mathrm{i}=0}^{\mathrm{n}} (\mathrm{y_i} - \hat{\mathrm{y}}_\mathrm{i}) \qquad (4)$$

Through multiple iterations we can change our gradient and y-intercept to give the optimal MSE. Each iteration signifies a step taken. How big of a step needed to be taken is computed through our **learning rate**. The learning rate is denoted by α.

We update the gradient and y-intercept using the following equations.

$$\mathrm{m} = \mathrm{m} - \alpha \times \mathrm{D_m} \qquad (5)$$

$$\mathrm{c} = \mathrm{c} - \alpha \times \mathrm{D_c} \qquad (6)$$

After applying gradient descent on our movie dataset we get the following results.

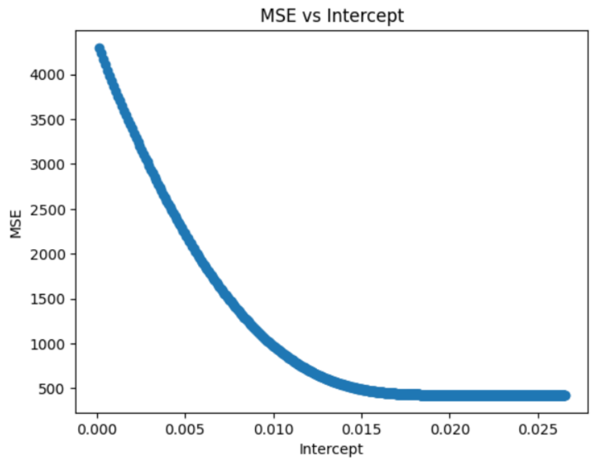

Fig. 4

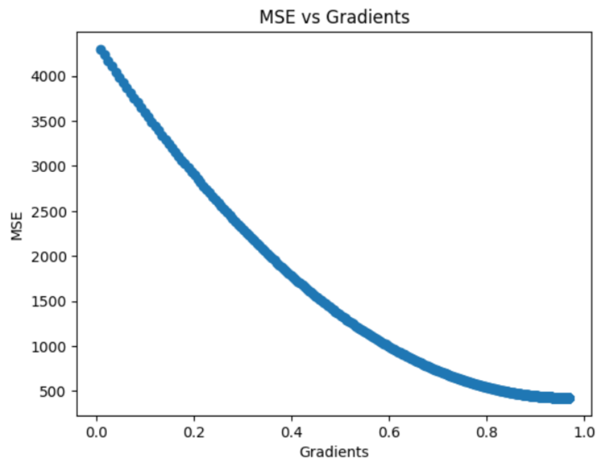

Fig. 5

Fig. 4 and 5 are obtained after applying a learning rate of 0.000001. How this graph change w.r.t the changes in learning rate will be discussed later on. However, we can see that this optimization algorithm slowly has taken steps going from a high MSE to the lowest possible. The curve shown in these graph is one side of the convex function shown before.

The regression line or the line of best fit obtained looks like this:

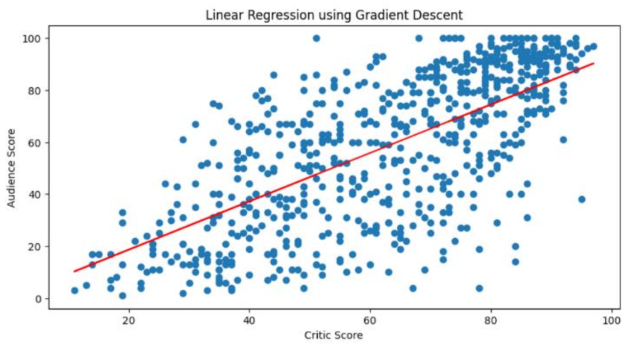

Fig. 6

V. Singular Value Decomposition

Singular Value Decomposition (SVD) is a powerful mathematical technique used to decompose a matrix into three simpler matrices. By revealing the underlying structure and properties of the original matrix, SVD offers numerous advantages in various fields, including:

- *Linear Algebra*: Solving linear systems of equations, especially when dealing with non-square or ill-conditioned matrices.
- *Machine Learning*: Dimensionality reduction, data analysis, feature extraction, and improving algorithm performance.
- *Signal Processing*: Image compression, anomaly detection, and pattern recognition.

For a given matrix A, SVD can be expressed as:

$$A = U\Sigma V^T \tag{7}$$

where:

- U is an $m \times m$ orthogonal matrix containing the left singular vectors.
- Σ is an $m \times n$ diagonal matrix with singular values on its diagonal.
- V^T is an $n \times n$ orthogonal matrix containing the right singular vectors.

The left singular vectors (U) in the matrix U provide an orthogonal basis for the column space of A, revealing relationships within its rows. Singular values (Σ) represent scaling factors, ordered by importance, affecting how singular vectors stretch or compress the data. The right singular vectors (V^T) in the matrix V^Tserve as an orthogonal basis for the row space of A, capturing relationships within its columns. Together, these components constitute the Singular Value Decomposition (SVD) and unveil intrinsic structures within the original matrix A.

A. Pseudoinverse Method

The pseudoinverse is used when A is not a square or full-rank matrix. It is calculated using the SVD of A:

$$A^+ = V\ \Sigma^+ U^T \tag{8}$$

where:

- U , V, Σ are obtained from the SVD of A.
- Σ^+ is the pseudoinverse of Σ, formed by taking the reciprocal of its non-zero singular values and transposing the result.

The pseudoinverse (A^+) is a valuable tool enabling the solution of linear systems, especially when the matrix A is not full rank. Particularly useful in overdetermined or underdetermined systems, the pseudoinverse is obtained by incorporating the components of the singular value decomposition. This calculation offers a means to effectively "invert" non-square or rank-deficient matrices, providing solutions to linear equations in scenarios where traditional inversion methods may be impractical or undefined.

B. SVD in Linear Regression

In the context of linear regression, the pseudoinverse method can be employed to find the coefficients (b) in the linear equation

$$y\ = Xb \tag{9}$$

where:

- y is the target variable
- X is the matrix of features.
- b is the vector of coefficients.

The pseudoinverse method is crucial when the matrix of X is not full rank, providing a solution to linear regression problems that may otherwise be ill-posed.

SVD when applied to movie dataset looks similar to the scatter plot obtained through gradient descent showing that in terms of precision both SVD and gradient descent same.

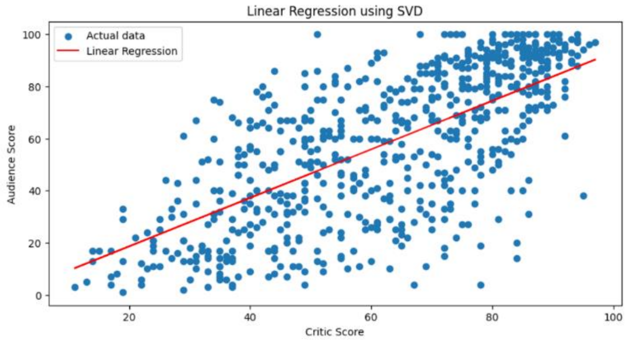

Fig. 7

VI. Comparative analysis

After applying linear regression using both gradient descent and SVD. We were able to find many features that differentiate both hence leading to many insights for practitioners looking to choose a linear regression technique.

	SVD	Gradient Descent
Time Taken (s)	0.0089	1.9367

Table. 2

Although we are limited by the amount of datasets we can use, SVD is a much more stable and preferred method as it is stable and more efficient for small datasets. However, this is always not the case. Gradient descent is much more efficient for large datasets with multiple features. Furthermore, its computational cost is much less than SVD.

	SVD	Gradient Descent
Time Complexity	$O(n^3)$	$O(n^2)$

Table. 3

In terms of precision both SVD and Linear Regression produced similar results regarding gradient of regression line:

	SVD	Gradient Descent
Gradient	0.931	0.930

Table. 4

Other than that both SVD and gradient descent have more limitations that we will discuss.

A. *Limitations of SVD*

- *Memory Usage:* For large matrices, SVD might consume a considerable amount of memory.
- *Not Suitable for Sparse Matrices*: SVD doesn't work well with sparse matrices, where most elements are zero.
- *All Features Consideration*: SVD considers all features, which might not be ideal if there are irrelevant or noisy features in the dataset.

B. *Limitations of Gradient Descent*

- *Sensitivity to Learning Rate*: Gradient Descent's performance can be sensitive to the learning rate. An inappropriate learning rate can lead to slow convergence or divergence.
- *Local Optima*: It might converge to a local minimum rather than the global minimum, especially in non-convex problems.
- *Feature Scaling Dependency*: It's sensitive to feature scaling, and poorly scaled features can hinder its performance.

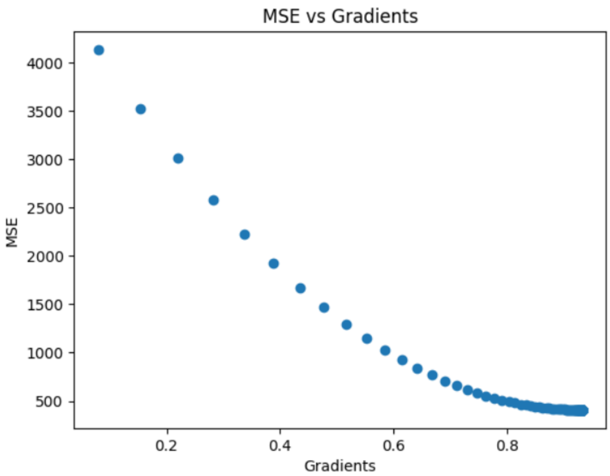

Fig. 8

Fig. 8 is a MSE vs Gradient scatter using the same dataset as Fig. 5 before, the only difference is that the learning rate has been changed from 0.000001 to 0.0001. This proves how sensitive gradient descends performance can be on the learning rate.

VII. Conclusion

In summary, the exploration of Gradient Descent and Singular Value Decomposition (SVD) linear regression models in the context of the movie dataset provided substantial insights into the relationship between critics' scores and audience scores. Both models exhibited successful implementation, demonstrating reasonable performance metrics and offering valuable insights into computational efficiency. These findings underscore the applicability and effectiveness of these regression methodologies in elucidating the association between critics' evaluations and audience perceptions within the movie domain.

Future endeavors could expand upon this study in several directions to enrich the analysis and augment predictive capabilities:

- *Feature Enrichment*: Extending the analysis to include additional features, such as sentiment analysis or genre-specific information, could afford a more nuanced understanding of audience score determinants, enhancing the model's predictive capacity.
- *Model Refinement*: Exploring techniques for hyper parameter optimization in the Gradient Descent model and addressing multicollinearity issues in the SVD model would refine predictive accuracy and robustness.
- *Cross-Validation and Comparative Analysis*: Incorporating cross-validation methodologies and conducting a comparative assessment between linear regression models and alternative algorithms would provide a comprehensive evaluation of predictive performance, ensuring a more thorough understanding and comparison of model efficacy in this domain.

Such advancements hold the potential to further fortify predictive modeling within the movie domain and contribute to broader insights into audience evaluations, facilitating more accurate and insightful predictions in this domain and potentially extending applicability to other related fields.

REFERENCES

FIGURES

[Fig. 1] https://www.mydatamodels.com/blog/regression-metrics/

[Fig. 3a & b] TY - BOOK, AU - Molokwu, Bonaventure, AU - Kobti, Ziad, PY - 2019/07/01, SP - 1, EP - 8, T1 - Spatial Event Prediction via Multivariate Time Series Analysis of Neighboring Social Units using Deep Neural Networks, VL - , DO - 10.1109/IJCNN.2019.8851730, ER -

JOURNALS

[1] https://towardsdatascience.com/why-gradient-descent-and-normal-equation-are-bad-for-linear-regression-928f8b32fa4f

[2] https://www.kdnuggets.com/2020/09/solving-linear-regression.html

DATASET

[1] https://www.kaggle.com/code/nichen301/movie-dataset-a-liner-regression-analysis/input