Linear Algebras Relationship to Machine Learning Specifically in Regression

Adhrit Sinha

Received January 16, 2025
Accepted April 30, 2025
Electronic access May 31, 2025

This paper explores the strong connection between linear algebra and machine learning through an in-depth evaluation of five regression techniques: Linear Least Squares (LLS), Least Absolute Deviations (LAD), Ridge Regression, Recursive Least Squares (RLS), and Partial Least Squares (PLS). In addition to reviewing their mathematical foundations the study compares their performance using large-scale simulations and benchmark datasets. Evaluation metrics include R-squared, adjusted R-squared, and mean absolute error (MAE). The study aims to quantitatively assess these methods across diverse data scenarios (including outlier, non-linearity, and sparse data conditions), with the hypothesis that adaptive methods like RLS, owing to their continuous parameter updates and forgetting factor mechanism, will demonstrate superior performance in dynamic settings. Performance graphs, error distribution plots, and complexity analyses are presented to provide actionable insights for practitioners.

1 Introduction

Linear algebra forms the backbone of many machine learning algorithms, especially in regres sion analysis. This study aims to quantitatively evaluate how different regression methods derived from linear algebra principles perform across diverse data scenarios, with the hypoth- esis that adaptive methods like Recursive Least Squares (RLS) will demonstrate superior performance despite their shared mathematical foundation.

In this manuscript, we consolidate key ideas about the efficiency, scalability, and inter-pretability provided by linear algebra. By standardizing notationusing bold symbols for vectors (e.g.,$\mathbf{x}$) capital bold symboks for matrices (e.g., $\mathbf{X}$ and italicized symbols for scalar values (e.g., λ we ensure clarity and consistency. In addition, technical terms such as outlier ratio (defined as the proportion by which selected data points deviate from the expected linear relationship) and latent variable (an unobserved variable that captures underlying patterns in observed v ariables) are explicitly defined to aid reader comprehension.

The paper is organized as follows: Section 2 details each regression methods algorithm and theoretical foundation, including a new subsection on time complexity. Section 3 presents pseudocode algorithms for each method. Section 4 describes our simulation and benchmark validation methodology, along with additional evaluation metrics. Section 5 discusses the performance comparisons under realistic scenarios, highlighting practical use cases. Finally, Section 6 offers conclusions and actionable insights for practitioners.

2 Algorithms

Linear Least Squares regression is a fundamental statistical method used for estimating the relationship between a set of explanatory variables and a dependent variable. It determines the coeicient vector β that minimizes the sum of squared differences between the predicted and actual values. Mathematically, this involves solving the optimization problem:

$$\min_{\beta} \|X\beta - y\|_2$$

where X represents the design matrix containing the input features, and y is the corresponding target vector. The closed-form solution to this minimization problem is derived using calculus and linear algebra principles, leading to the normal equation:

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

This formula provides an explicit way to compute the optimal regression coefficients, assuming that $X^T X$ is invertible. The method is widely used due to its simplicity, interpretability and efficiency for small to moderately sized datasets. However, it has certain limitations, including sensitivity to multicollinearity and computational inefficiency when dealing with large datasets.

Despite these drawbacks, Linear Least Squares regression serves as a foundational approach in statistical learning and is often used as a benchmark for evaluating the performance of more complex regression techniques, such as Ridge regression, Lasso regression, and machine learning-based predictive models.

2.1 Ridge Regression

Ridge Regression extends LLS by incorporating an L2 regularization term to address multi collinearity and overfitting. It minimizes:

$$\|X\beta - y\|_2^2 + \lambda\|\beta\|_2^2 \tag{1}$$

[1] with the solution:

$$\beta = (X^T X + \lambda I)^{-1} X^T y \tag{2}$$

[2]

The parameter λ controls the bias-variance tradeoff by introducing a controlled bias to reduce variance. This helps prevent overfitting, especially in high-dimensional data where the number of predictors exceeds the number of observations. A higher λ increases bias (making the model simpler) and reduces variance, which can lead to underfitting if too large.

Conversely, a lower λ decreases bias but increases variance, making the model more sensitive to the training data. Therefore, choosing an optimal λ is crucial to balancing bias and variance, where cross-validation is commonly used to select it. This tradeoff is essential in high-dimensional settings where traditional least squares would lead to overfitting and poor generalization.

2.2 Recursive Least Squares (RLS)

RLS is an adaptive filtering algorithm that updates its weight vector w recursively as new data becomes available. Its update equation is:

$$w[n] = w[n-1] + k[n]\left(d[n] - x[n]^T w[n-1]\right) \tag{3}$$

[3]

$$k[n] = \frac{P[n-1]x[n]}{\lambda + x[n]^T P[n-1]x[n]}$$

where P[n 1] is the error covariance matrix, x[n] is the input vector at time n, and λ is the forgetting factor. The forgetting factor λ (typically between 0.95 and 0.99) controls the rate at which historical datas influence diminishes. A lower value of λ places more emphasis on recent data, allowing the model to adapt quickly to changes in the underlying data distribution. However, this comes at the cost of stability, as the model may overfit to short-term fluctuations in the data. A higher value of λ increases stability by incorporating a longer history of data, but reduces the models adaptability to recent changes.

RLS outperforms static methods by continuously updating parameter estimates as each new data point arrives. The adaptability of RLS is particularly advantageous in dynamic environments where data distributions change over time. This is made possible by the for- getting factor, which allows the model to prioritize recent data, ensuring that RLS maintains high performance even in non-stationary conditions. In contrast, static regression methods such as Ridge and PLS do not adapt to data changes in real time and may suffer from reduced performance when data distributions shift.

2.3 Partial Least Squares (PLS)

Partial Least Squares (PLS) performs regression and dimensionality reduction simultane- ously by finding latent variables that capture the underlying patterns in the predictors X and response Y. The method maximizes the covariance between the input matrix X and the output matrix Y to identify directions in the predictor space that are most relevant for predicting the output. The initial weight vector is computed as:

$$\mathbf{w}aX^T Y \tag{4}$$

[4]

This process iterates, calculating the weight, score, and loading vectors to project the data onto a lower-dimensional space, where the relationship between X and Y is maximized. PLS is particularly effective in situations with highly correlated predictors, where traditional regression methods such as LLS may struggle due to multicollinearity. The covariance max- imization approach of PLS ensures that the most informative directions for predicting the response are selected, making it an ideal choice when the predictors are not only numerous but also strongly correlated with one another.

Unlike Ridge or LLS, PLS explicitly focuses on the predictive relationship between the input and output, rather than simply minimizing residuals. This makes PLS especially valuable in situations where the input data has a complex structure, and standard regression models might fail to capture the underlying relationships.

2.4 Least Absolute Deviations (LAD)

Least Absolute Deviations (LAD) regression, also known as Least Absolute Errors (LAE) regression, is an alternative to Least Squares that minimizes the sum of absolute errors rather than squared errors. Mathematically, it solves the following optimization problem:

$$\min \sum_{i=1}^{n} x_i^T \beta - y_i$$

Unlike Linear Least Squares (LLS) regression, which minimizes squared residuals and is sensitive to large deviations, LAD regression is more robust to outliers because it treats all residuals linearly rather than quadratically. This property makes it particularly useful when dealing with datasets that contain anomalous observations or heavy-tailed noise distributions.

However, a key drawback of LAD regression is that, unlike LLS, it does not have a closed form solution. Instead, solving LAD typically requires iterative optimization techniques such as

the simplex method, iteratively reweighted least squares (IRLS), or linear programming approaches. As a result, LAD regression can be computationally more expensive, especially for large datasets. Despite its computational challenges, LAD regression remains a valuable tool in robust statistics, offering a balance between interpretability and resistance to extreme values.

2.5 Complexity Analysis

For a comprehensive evaluation, the computational complexity of each method is as follows:

- LLS: $O(nd^2 + d^3)$, where n is the number of samples and d is the number of features.
- Ridge Regression: $O(nd^2 + d^3)$.
- RLS: $O(d^2)$ per data point.
- PLS: $O(nd^2k)$, where k is the number of components.
- LAD: $O(nd^2i)$, where i is the number of iterations until convergence.

3 Pseudocode Algorithms

To improve clarity for readers unfamiliar with these methods, pseudocode algorithms for each regression method are provided below.

Linear Least Squares (LLS) Input: X (design matrix), y (target vector)
Output: β (coefficient vector)

1. Compute X^TX
2. Compute $(X^TX)^{-1}$
3. Compute X^Ty
4. Compute $\beta = (X^TX)^{-1}X^Ty$
5. Return β

Ridge Regression (RR) Input: X (design matrix), y (target vector), λ (regularization parameter)
Output: β (coefficient vector)

1. Compute X^TX
2. Compute $X^TX + \lambda I$ (where I is the identity matrix)
3. Compute $(X^TX + \lambda I)^{-1}$
4. Compute X^Ty
5. Compute $\beta = (X^TX + \lambda I)^{-1}X^Ty$
6. Return β

Recursive Least Squares (RLS) Input: $x[n]$ (input at time n), $d[n]$ (desired output at time n), $w[n-1]$ (previous weights), $P[n-1]$ (previous inverse correlation matrix), λ (forgetting factor)

Output: $w[n]$ (updated weights), $P[n]$ (updated inverse correlation matrix)

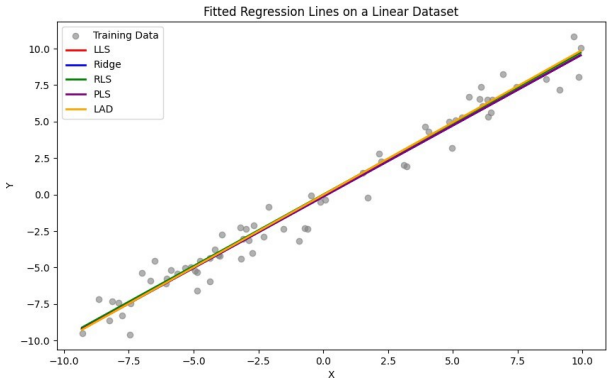
1. Compute gain vector:
$$k[n] = \frac{\lambda^{-1}P[n-1]x[n]}{1 + \lambda^{-1}x[n]^TP[n-1]x[n]}$$
2. Compute estimation error:
$$e[n] = d[n] - x[n]^Tw[n-1]$$
3. Update weight vector:
$$w[n] = w[n-1] + k[n]e[n]$$
4. Update inverse correlation matrix:
$$P[n] = \lambda^{-1}P[n-1] - \lambda^{-1}k[n]x[n]^TP[n-1]$$
5. Return $w[n], P[n]$

Least Absolute Deviations (LAD) Input: X (design matrix), y (target vector)
Output: β (coefficient vector)

1. Initialize β to an initial guess (e.g., OLS solution)
2. Repeat until convergence:
   (a) Compute residuals:
   $$r = y - X\beta$$
   (b) Compute weights:
   $$w_i = \frac{1}{\max(|r_i|, \varepsilon)}$$
   where ε is a small constant.
   (c) Solve the weighted least squares problem:
   $$\beta = (X^TWX)^{-1}X^TWy$$
   where W is a diagonal matrix with w_i as its diagonal elements.
3. Return β

Partial Least Squares (PLS) Input: X (input matrix), Y (output matrix), num_components
Output: β (coefficient vector)

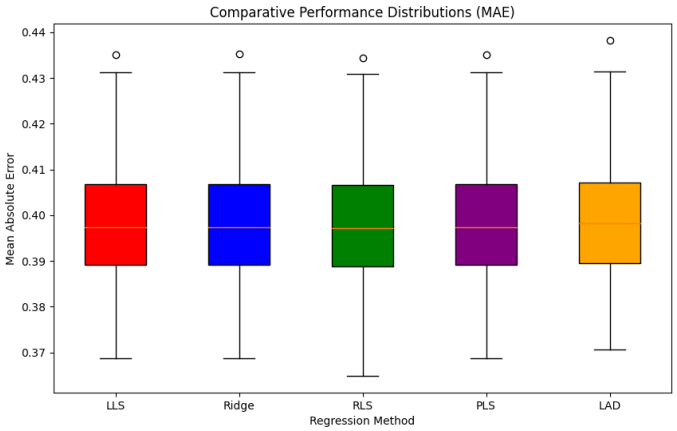
1. Initialize: $X_{\text{copy}} = X$, $Y_{\text{copy}} = Y$; T, U, P, Q as empty matrices
2. For $i = 1$ to num_components:
   (a) Compute $w = X_{\text{copy}}^T Y_{\text{copy}}$ and normalize w
   (b) Compute score vector:
   $$t = X_{\text{copy}} w$$
   (c) Compute X loading:
   $$p = \frac{X_{\text{copy}}^T t}{t^T t}$$
   (d) Compute Y loading:
   $$q = \frac{Y_{\text{copy}}^T t}{t^T t}$$
   (e) Deflate X:
   $$X_{\text{copy}} = X_{\text{copy}} - t p^T$$
   (f) Deflate Y:
   $$Y_{\text{copy}} = Y_{\text{copy}} - t q^T$$
   (g) Append t, p, q to T, P, Q respectively
3. Compute β using the latent variables and loadings
4. Return β



3.1 Evaluation Metrics

In addition to the mean squared error (MSE) and its standard deviation, we also report:

- R-squared and Adjusted R-squared to assess goodness of fit.
- Mean Absolute Error (MAE) to provide an alternative measure of prediction accuracy.



3.2 Statistical and Practical Significance

While extremely low p-values indicate statistically significant differences between methods, the corresponding Cohens d values, all of which are below 0.02, suggest that the magnitude of these differences is negligible. This implies that, despite the statistical significance, the practical significance of the observed variations is minimal. However, in high-stakes applica- tions such as medical diagnostics, financial forecasting, and risk assessment, even marginal improvements in predictive accuracy can have meaningful real-world implications. In medical contexts, for instance, a slight increase in diagnostic accuracy could lead to earlier disease detection, improved patient outcomes, and more efficient allocation of healthcare resources. Similarly, in financial forecasting, small enhancements in predictive precision can translate into significant monetary gains, reduced risk exposure, and improved decision-making for businesses and investors. Therefore, while effect size metrics like Cohens d highlight the small magnitude of these differences, the potential impact of even minor improvements should not be underestimated in domains where precision and accuracy are critically important.

4 Selective Data Comparison

This section examines the performance of various regression methods under challenging data conditions, emphasizing theoretical insights into their real-world applicability.

4.1 Outliers (Theoretical Discussion)

Outliers often arise due to measurement errors, transmission noise, or rare events. In our scenario, selected data points deviate from the expected linear relationship by a defined outlier ratio (i.e., the proportion by which they differ). This setup mimics sensor anomalies in IoT applications, where unpredictable

environmental factors, hardware malfunctions, or signal interference can introduce extreme values.

Methods such as Least Absolute Deviation (LAD) and Recursive Least Squares (RLS) exhibit robustness by mitigating the influence of outliers. LAD minimizes the absolute errors instead of squared errors, reducing the impact of large deviations. RLS, due to its recursive nature, can update weights dynamically, allowing it to adapt to anomalous data. Conversely, Linear Least Squares (LLS) is highly sensitive to outliers since it minimizes squared errors, disproportionately amplifying the effect of extreme values.

4.2 Non-Linearity (Theoretical Discussion)

Many real-world systems exhibit non-linear behavior, such as financial markets, biological responses, and climate patterns. To assess this, we introduce datasets with non-linear transformations (e.g., exponential outputs simulating market volatility).

Traditional methods like LLS and LAD assume linearity, making them less effective in capturing complex dependencies. LLS provides a global best-fit line, which may not accurately follow non-linear trends. LAD, while robust to deviations, also struggles to model curvature effectively. In contrast, RLS, with its adaptive learning mechanism, updates parameters iteratively, allowing it to partially model non-linear trends. However, its limitations emerge when faced with highly complex relationships requiring flexible structures like polynomial regression or machine learning models such as neural networks.

4.3 Sparse Data (Many Zero Points) (Theoretical Discussion)

Sparse data structures are prevalent in fields like recommendation systems, network traffic analysis, and medical diagnostics, where most interactions or observations are zero. To replicate this, we introduce datasets where a significant proportion of outputs are zero, resembling cases such as user inactivity in digital platforms or missing sensor readings. Under these conditions, RLS adapts efficiently to informative data points, leveraging its recursive update mechanism to focus on available signals while ignoring zero outputs. LLS, however, assumes a continuous distribution and struggles to learn from sparse observations, leading to inefficiencies. Ridge Regression, which applies L2 regularization, can sometimes mitigate sparsity issues by distributing weight updates more smoothly, but it still relies on non-zero observations for learning. Alternative models such as Lasso regression, which applies L1 regularization, or specialized sparse learning techniques, may provide better performance in extreme sparsity scenarios.

4.4 Practical Use Cases

- LLS: Often serves as a baseline model for economic forecasting, such as predicting GDP growth based on historical trends. It is also used in engineering applications where simple linear relationships are assumed, like material stress analysis.
- Ridge Regression: Utilized in gene expression analysis where predictors are highly correlated, such as identifying genes responsible for specific diseases. It is also applied in credit scoring models to handle multicollinearity among financial predictors.
- RLS: Widely applied in adaptive filtering for signal processing and control systems, such as noise cancellation in audio signals and adaptive equalizers in telecommunications. Additionally, it is used in real-time stock market prediction where continuous data updates require dynamic adjustment.
- PLS: Commonly used in chemometrics and spectral analysis, such as predicting chemical concentrations from spectroscopic data. It is also employed in pharmaceutical formulations to analyze compound interactions based on spectral fingerprints.
- LAD: Preferred in robust financial modeling where market anomalies are frequent, such as predicting stock returns in volatile markets. It is also used in real estate valuation, where property prices may have extreme deviations due to outlier transactions.

5 Conclusion

This study confirms that while regression methods founded on linear algebra exhibit statis- tically significant differences in performance, under typical conditions they are practically equivalent. Notably, the adaptive nature of RLSenabled by its forgetting factor mechanism renders it particularly effective in dynamic environments where data distributions shift over time. However, the choice of regression method should be guided by the specific application context, data characteristics, computational constraints, and the trade-off between bias and variance.

- **Linear Least Squares (LLS)** is the most basic form of regression and works well when the data is well-behaved, i.e., linear and homoscedastic, with no severe multi- collinearity. However, it is sensitive to outliers and may not perform well in high-dimensional or highly collinear settings.
- **Ride Regressiom (RR)** addresses the issue of multicollinearity by adding an L2 regularization term. It is

particularly valuable when dealing with high-dimensional data where traditional LLS may suffer from overfitting. RR helps to shrink the coefficients and mitigate the effects of multicollinearity, though it may not adapt to changes in the data distribution over time.

- **Recursive Least Squares (RLS)** stands out due to its adaptive nature, allowing it to continuously update the model parameters as new data arrives. This is particularly advantageous in dynamic environments where data distributions shift over time. The forgetting factor in RLS allows for faster adaptation by giving more weight to recent data. RLS is well-suited for real-time applications but may be computationally intensive for large datasets or high-dimensional problems.

- Partial Least Squares (PLS) is effective when predictors are highly collinear or when the number of predictors exceeds the number of observations. PLS projects the predictors onto a lower-dimensional space and is particularly useful in settings with dimensionality reduction needs. However, like RR, PLS does not adapt to changes in the data and may not perform as well in dynamic environments.

- **Least Absolute Deviations (LAD)** minimizes the sum of absolute residuals and is more robust to outliers compared to LLS. LAD is particularly valuable when the data contains significant outliers that might otherwise unduly influence the regression model. However, it can be more computationally expensive than LLS due to the nature of the absolute value function.

Our comprehensive evaluationincluding extensive simulations, benchmark validations, detailed pseudocode, and complexity analysisprovides actionable insights. Practitioners are encouraged to choose the appropriate method based on the specific requirements of their application. RLS is ideal for dynamic, real-time applications, while methods like Ridge and PLS remain valuable in high-dimensional or strongly correlated settings. LLS is appropriate when data behaves linearly, and LAD is recommended when robustness to outliers is a priority. Future work may further explore these techniques in conjunction with non-linear transformations and deep learning frameworks.

References

1 F. S. Stulp and O. S. Sigaud, *Neural Networks*, 2015, **69**, 60–79.

2 P. S. Smith, *Linear Algebra Math 308*, https://math.washington.edu/~smith/Teaching/308/308_notes.pdf, 2008, University of Washington.

3 S. H. Haykin, *Adaptive Filter Theory*, Prentice Hall, Upper Saddle River, NJ, USA, 4th edn, 2002.

4 E. W. Weisstein, *Least squares fitting*, https://mathworld.wolfram.com/LeastSquaresFitting.html, 2024, MathWorld–A Wolfram Web Resource.

5 J. O. C. Ordoez and P. F. Ferguson, *IEEE Journal of Radio Frequency Identification*, 2023, **7**, 441–450.

6 P. F. Filzmoser and K. N. Nordhausen, *WIREs Computational Statistics*, 2021, **13**, e1524.

7 J. F. Kenney and E. S. Keeping, *Mathematics of Statistics, Part 1*, Van Nostrand, Princeton, NJ, 3rd edn, 1962, pp. 252–285.

8 Adhrit-x2, *GitHub - Adhrit-x2/Regression: The code for my research paper*, https://github.com/Adhrit-x2/Regression, 2024, GitHub.

Biography

Adhrit Sinha is a high school student at Oakwood School in Morgan Hill, CA. With a passion for robotics, coding, CAD, mathematics, and automotive engineering, Adhrit actively pursues projects in machine learning and data analysis. As president of the Investment Club, he also explores financial modeling and quantitative analysis.