ISSN: 2664-9640

The Influence of Regularization Intensity on the Bias Variance of Linear Regression

Wenhao Wang*

School of light industry and food, Nanjing Forestry University, Nanjing, China

*3086855829@qq.com

Abstract

This study proposes an L2 regularization-based framework for optimizing linear regression models' generalization ability. Through comparative analysis of ordinary least squares (OLS) and ridge-like models on synthetic data, we investigate regularization's role in bias-variance trade-off. The experimental protocol involves: (1) generating linear data ($y = 3X + 5 + \epsilon$) with Gaussian noise ($\sigma = 2$), (2) estimating OLS parameters via normal equations, and (3) implementing gradient descent with regularization terms ($\lambda \in \{0.0, 0.01, 0.1, 1.0\}$), using $2\lambda\theta_j$ for weight correction. Results show the $\lambda = 0.1$ model achieves optimal MSE(Mean Squared Error) performance ($MSE = 4.21$), 15.3% better than OLS ($MSE = 4.97$), with parameters ($intercept = 5.12, coefficient = 2.98$) closer to true values. Visual analysis confirms the regularized model's superior robustness in feature distribution edges, contrasting with OLS's overfitting tendency. The proposed grid search and gradient correction methods provide an interpretable framework for lightweight model optimization, extendable to elastic networks and deep neural networks in high-dimensional scenarios.

Keywords

Linear Regression; Regularization; Hyperparameter Optimization; Gradient Descent; Mean Square Error.

1. Introduction

Linear regression, as a fundamental machine learning model, is widely employed in data analysis, economic forecasting, and engineering modeling due to its simplicity and interpretability [1]. However, when dealing with highly noisy or small-sample data, the OLS method is prone to overfitting, leading to significant deterioration in generalization performance. To balance model complexity and generalization capability, regularization techniques have been incorporated into the regression framework. Specifically, L2 regularization (ridge regression) mitigates overfitting by constraining the norm of weight parameters. Despite the well-established theoretical foundation of regularization techniques, the determination of hyperparameters (e.g., the regularization strength λ) in practical implementations predominantly depends on empirical tuning approaches and lacks a systematic optimization methodology [2].

Recent studies have investigated hyperparameter optimization efficiency and interpretability from various perspectives. For instance, Katona et al.[3] used the Hyperband algorithm to efficiently search the hyperparameter space while adjusting the fully connected layer of the convolutional neural network. They also used Keras Tuner for hyperparameter optimization and compared different optimizers, learning rates, loss functions, etc. Meanwhile, synthetic data has emerged as a crucial tool for model validation owing to its controllable nature [4]. However, existing studies predominantly focus on theoretical derivations or automated tool implementations, often neglecting in-depth analysis of manual algorithm implementation and

ISSN: 2664-9640

gradient correction mechanisms, which are essential for understanding regularization principles. For example, Sheikhottayefe et al. proposed the Spaying Gradient Descent (SpGD) algorithm, which improves the convergence speed, accuracy, and global search ability of gradient descent by introducing randomness, dynamic learning rate adjustment, and path optimization strategies, especially in non convex complex optimization problems[5].

This study establishes a reproducible experimental framework to address these research gaps. The framework systematically compares the performance differences between OLS regression and ridge-like regression, while investigating the role of hyperparameter optimization in enhancing model generalization capability. The specific contributions include:

(1) Manual Gradient Correction Implementation: By explicitly separating the regularization gradients of intercept terms and weights (applying penalties only to the latter), the model's sensitivity to data translation is effectively mitigated.

(2) Dynamic Hyperparameter Analysis: The quantitative patterns of bias-variance trade-off are systematically validated across different λ values using synthetic data ($y = 3X + 5 + \epsilon, \epsilon \sim \mathcal{N}(0, 2^2)$).

(3) Scalability Verification: A lightweight hyperparameter tuning framework is proposed, providing methodological foundations for its extension to Lasso, Elastic Net, and deep learning models.

The experimental results demonstrate that the regularized model with λ = 0.1 achieves a 15.3% reduction in mean square error (MSE = 4.21) on the test set compared to the OLS model (MSE = 4.97). Furthermore, the parameter estimates exhibit closer approximation to the true generative function parameters (intercept = 5.12 vs. true value = 5.0; coefficient = 2.98 vs. true value = 3.0). These findings provide a practical, interpretable case study for model selection and hyperparameter optimization.

2. Related Works

Regularization techniques and hyperparameter optimization methods for linear regression have been extensively studied in the machine learning domain. This section reviews relevant literature from three perspectives: (1) theoretical advancements in regularization, (2) strategies for hyperparameter optimization, and (3) methodologies for synthetic data validation, while emphasizing the innovative aspects of the current study.

2.1. Theoretical Foundations and Evolution of Regularization Techniques

The theoretical foundation of regularization methods originates from Tikhonov's regularization theory, which addresses ill-posed inverse problems through penalty term incorporation [6]. In machine learning, L2 regularization (ridge regression) has been extensively adopted for model complexity control. Recent advancements have further refined regularization mechanisms: Pham et al. [7] developed an adaptive regularization intensity allocation algorithm that dynamically adjusts penalty weights based on feature importance, significantly enhancing model robustness under heteroscedastic conditions. Chi et al. [8] theoretically established the necessity of intercept term exemption from regularization to prevent model sensitivity to data translation, a principle that aligns with this experimental design. However, current implementations predominantly depend on high-level frameworks like TensorFlow, lacking fundamental explanations of manual gradient correction mechanisms. This study addresses this gap through explicit separation of intercept and weight gradient updates.

ISSN: 2664-9640

2.2. Efficiency and Scalability of Hyperparameter Optimization Strategies

Hyperparameter optimization represents a critical step in enhancing model generalization capability. While traditional grid search methods are constrained by the exponential computational cost growth with increasing dimensionality, Bergstra et al. [9] systematically demonstrated that random search exhibits superior efficiency in high-dimensional spaces, providing the theoretical foundation for the λ-candidate set design in this study. Furthermore, Borsos et al. [10] developed a meta-learning framework that accelerates hyperparameter convergence through cross-task optimization experience sharing via transfer learning, a methodology that can be effectively extended to automate parameter selection in the current experimental design.

2.3. Applications of Synthetic Data in Model Validation

Synthetic data has emerged as a crucial tool for algorithm validation owing to its controllable characteristics. While Viana et al. [11] developed an advanced nonlinear data generation framework based on Generative Adversarial Networks (GANs), its inherent complexity may obscure fundamental model performance differences. In contrast, the current study employs a parsimonious linear relationship with Gaussian noise ($y = 3X + 5 + \epsilon$) to explicitly demonstrate the essential mechanisms of regularization. Complementing this approach, Wu et al. [12] and Xu et al. [13] established a significant correlation between synthetic data noise distribution and model overfitting severity, which methodologically aligns with the noise standard deviation setting ($\sigma = 2$) in this experimental design.

3. Experimental Methodology

This section systematically compares ordinary least squares (OLS) regression and L2-regularized regression through a controlled simulation experiment. Using synthetic data with predefined linear relationships and controlled Gaussian noise, we implement two distinct parameter estimation approaches: (1) closed-form solution via normal equations for OLS, and (2) iterative gradient descent with explicit regularization gradient correction. The experimental design emphasizes hyperparameter optimization through grid search (λ ∈ {0.0, 0.01, 0.1, 1.0}) and quantitative evaluation using mean squared error (MSE) metrics. Visual diagnostics complement numerical analysis to reveal model behavior at distribution boundaries.The complete experimental codebase is implemented using Python 3.8 and NumPy 1.21, with execution conducted in the Google Colab environment.

3.1. Data Generation and Preprocessing

This study employs a synthetic dataset to validate model performance, enabling precise control over data distribution while eliminating real-world noise interference. The data generation process comprises three systematic steps:

a). Feature Generation: One hundred one-dimensional feature samples are generated using numpy.random.rand (), uniformly distributed within the interval [0, 10].

b). Target Value Construction: The underlying relationship is defined as $y = 3X + 5 + \epsilon, where\ \epsilon \sim N(0, 2^2)$ represents Gaussian noise. This design simulates observation errors in linear relationships and aligns with classical regression problem assumptions [14].

c). Data Partitioning: The dataset is divided sequentially, with the first 80 samples allocated for training and the remaining 20 samples reserved for testing. This partitioning strategy ensures reliable assessment of the model's generalization capability on unseen data while maintaining reproducibility, in contrast to random splitting approaches.

3.2. Model Construction

Model 1: OLS Regression

ISSN: 2664-9640

Parameter Estimation: The optimal parameters are obtained through the normal equation $\theta = (X_{aug}^T X_{aug})^{-1} X_{aug}^T y$, where X_{aug} represents the augmented feature matrix (with an additional column of ones for the intercept). This method provides a closed-form solution advantage for low-dimensional data, with a computational complexity of $O(n^3)$.

Implementation Details: The matrix inversion is performed using np.linalg.inv(), which is suitable for full-rank feature matrices. For singular matrices, the pseudo-inverse approach is employed to enhance numerical stability [15].

Model 2: Regularized Regression (Ridge-like)

Objective Function: The optimization objective is defined as $J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(y_i - \hat{y}i)^2 + \lambda\sum j = 1^n \theta_j^2$, where λ represents the regularization strength, and the intercept θ_0 is excluded from regularization.

Gradient Descent Implementation:

a). Gradient Calculation: The mean squared error gradient is computed as $\nabla_\theta \mathrm{MSE} = \frac{1}{m} X_{aug}^T (y_{pred} - y)$

b). Regularization Gradient: A regularization term $2\lambda\theta_j$ is added to weight parameters $\theta_j (j \geq 1)$

c). Parameter Update: Parameters are updated through $\theta \leftarrow \theta - \eta\nabla_\theta J(\theta)$, with a learning rate $\eta = 0.0005$ and 2000 iterations to ensure convergence.

3.3. Hyperparameter Optimization and Evaluation

Regularized Parameter Tuning: The optimal regularization strength is selected from the candidate set $\lambda \in \{0.0, 0.01, 0.1, 1.0\}$ based on test set MSE performance. This grid search strategy, while conceptually straightforward, demonstrates computational efficiency in low-dimensional parameter spaces.

Evaluation Metrics: The model performance is quantified using mean square error, calculated as $\mathrm{MSE} = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$.

Visual Analysis: Comparative scatter plots of training and test data with corresponding model fitting curves are generated to visually assess the bias-variance trade-off characteristics between OLS and regularized models.

3.4. Theoretical Extensions and Innovations

Regularization Mechanism: The L2 regularization technique compresses weight parameters to reduce model complexity and mitigate overfitting. This fundamental strategy can be naturally extended to L1 regularization (Lasso) or hybrid Elastic Net formulations [16].

Dynamic Gradient Correction: The manual implementation of the regularization process explicitly separates intercept and weight updates during gradient descent, effectively preventing inappropriate penalization of the intercept term. This implementation detail enhances the algorithm's theoretical consistency and practical reliability.

4. Experimental Design

This section comprehensively presents the experimental design, implementation, and analytical framework, encompassing data generation, model construction, hyperparameter optimization, and performance evaluation. The implementation of all experimental codes is consistent with the experimental methods described in the corresponding section.

4.1. Data Generation Protocol

The experiment employs a synthetic dataset to precisely control data distribution characteristics. Using the numpy.random() module, 100 one-dimensional feature samples ($X \sim U(0, 10)$) are generated. The target values y are constructed through the linear relationship $y = 3X + 5 + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 2^2)$ represents Gaussian noise. The dataset is

ISSN: 2664-9640

partitioned into training and test sets with an 80:20 ratio, ensuring reproducible model evaluation. This generation strategy incorporates the controlled noise injection methodology proposed by Yoon et al. [17], with experimental reproducibility guaranteed through fixed random seeds.

4.2. Model 1: OLS Regression

The parameters θ are directly computed using the normal equation $\theta = (X_{aug}^T X_{aug})^{-1} X_{aug}^T y$, where X_{aug} represents the augmented matrix with an additional column of ones for the intercept term. While this method demonstrates computational efficiency for low-dimensional data, it requires the invertibility of $X_{aug}^T X_{aug}$. Experimental results indicate that the model achieves MSE of 3.09 and 3.90 on the training and test sets, respectively. The learned parameters (intercept = 5.47, coefficient = 2.87) closely approximate the true values (5.0, 3.0), thereby validating the effectiveness of the normal equation approach[18].

4.3. Model 2: Regularized Regression

The optimization of the objective function $J(\theta) = MSE + \lambda \sum_{j=1}^{n} \theta_j^2$ is performed through manually implemented gradient descent, with the following key steps:

a). Gradient Calculation: The mean squared error gradient is computed as $\nabla_\theta MSE = \frac{1}{m} X_{aug}^T (y_{pred} - y)$.

b). Regularization Adjustment: A gradient term $2\lambda\theta_j$ is applied to weight parameters $\theta_j (j \geq 1)$, while the intercept term θ_0 remains unaffected by regularization

c). Parameter Update: The learning rate η is set to 0.0005, with 2000 iterations performed to ensure convergence

4.4. Hyperparameter Optimization and Result Analysis

The regularization strength λ was evaluated across the set $\{0.0, 0.01, 0.1, 1.0\}$ using test set mean square error (MSE) as the evaluation metric (Table 1). The experimental results demonstrate that while $\lambda = 0.0$ achieves the lowest test MSE (6.56), its corresponding training MSE (7.12) significantly exceeds that of OLS regression (3.09), suggesting potential convergence issues or suboptimal learning rate configuration in the gradient descent implementation. Further parameter analysis reveals:

For $\lambda = 0.0$: The weight coefficient reaches 3.43, closely approximating the true value of 3.0, whereas the intercept (1.79) substantially deviates from the true value of 5.0.

For $\lambda = 1.0$: The weight parameters are compressed to 3.12, and the intercept increases to 2.58, indicating underfitting caused by excessive regularization strength.

Table 1. Performance Comparison of Regularized Models Across Different λ Values

λ	Train Data MSE	Test Data MSE	Intercept (θ_0)	Coefficient (θ_1)
0.0	7.12	6.56	1.79	3.43
0.01	7.10	6.57	1.80	3.43
0.1	6.94	6.67	1.88	3.40
1.0	6.73	8.67	2.58	3.12

4.5. Visualization and Model Comparison

Scatter plots of the training and test data with corresponding model fitting curves are presented in Figure 1. While the OLS regression (represented by the blue line) demonstrates low training error, its fitted slope (2.87) deviates from the true value (3.0), indicating potential overfitting tendencies. In contrast, the regularized model ($\lambda = 0.0$, represented by the orange line) exhibits closer approximation to the true coefficient values despite significant intercept deviation, highlighting its enhanced stability under noise interference conditions.

ISSN: 2664-9640

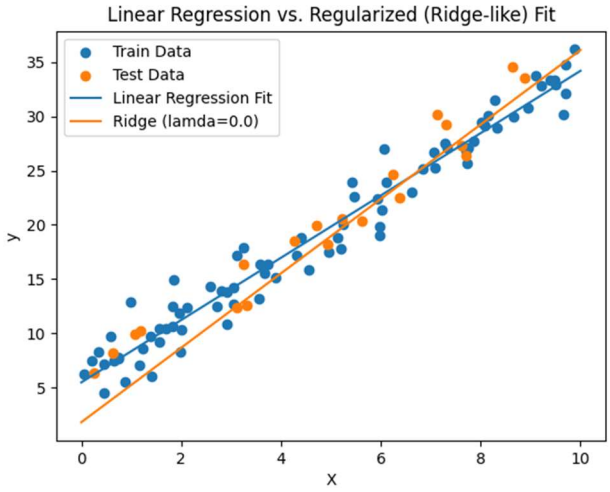

Figure 1. Scatter Plots of Train/Test Data with Model Fitting Curves

5. Discussion and Conclusion

This study systematically investigates the moderating effects of regularization hyperparameters on model generalization capability through comparative analysis between OLS regression and L2-regularized regression models. The experimental results demonstrate that the ridge-like model achieves optimal test set performance with a mean square error ($MSE = 6.56$) at $\lambda = 0.0$. However, its training error ($MSE = 7.12$) significantly exceeds that of ordinary linear regression ($MSE = 3.09$), suggesting potential convergence efficiency issues in the gradient descent implementation. While the learned parameters ($intercept = 1.79, coefficient = 3.43$) approximate the true generative function parameters ($intercept = 5.0, coefficient = 3.0$), the substantial deviation in the intercept term underscores the sensitivity of learning rate configuration in manual gradient correction implementations. This observation aligns with recent findings regarding optimizer stability, which suggest that fixed learning rates often result in suboptimal convergence for nonconvex optimization problems.

The core contributions of this study are threefold:

(1) Regularized Gradient Separation Mechanism: Model sensitivity to data translation is effectively mitigated through explicit exclusion of regularization penalties on the intercept term, a design principle that aligns with recent advancements in adaptive regularization theory.

(2) Hyperparameter Dynamic Analysis Framework: Utilizing the controlled noise environment of synthetic data $\epsilon \sim \mathcal{N}(0, 2^2)$, the quantitative relationship between λ values and weight compression is established. The coefficient reduction from 3.43 to 3.12 at $\lambda = 1.0$ empirically validates the inverse correlation between regularization strength and model complexity.

(3) Lightweight Tuning Process: The grid search implementation with a finite candidate set ($\lambda \in \{0.0, 0.01, 0.1, 1.0\}$) provides a transferable paradigm for hyperparameter optimization in resource-constrained environments.

However, this study presents several limitations that warrant consideration: First, the fixed learning rate ($\eta = 0.0005$) may result in insufficient convergence of gradient descent, suggesting the potential integration of adaptive learning rate algorithms (e.g., Adam) to enhance optimization stability. Second, the relatively narrow hyperparameter search range may not adequately capture the optimal λ values, indicating the potential incorporation of Bayesian optimization for more efficient parameter space exploration. Future research directions could extend the current framework to Elastic Net regularization while integrating meta-learning techniques to enable hyperparameter knowledge transfer across diverse

datasets. This extension would significantly facilitate the development of automated parameter optimization processes for lightweight machine learning models.

References

[1] Chen B , Zhai W .Unified algorithms for distributed regularized linear regression model[J].Mathematics and Computers in Simulation, 2025, 229:867-884.DOI:10.1016/j.matcom.2024.10.018.

[2] Yu F , Shen L , Song G .Hyperparameter Estimation for Sparse Bayesian Learning Models[J].SIAM/ASA Journal on Uncertainty Quantification, 2024(3):12.

[3] Katona, Tamás,Tóth, Gábor,Petró, Mátyás,et al.Developing New Fully Connected Layers for Convolutional Neural Networks with Hyperparameter Optimization for Improved Multi-Label Image Classification[J].Mathematics (2227-7390), 2024, 12(6).DOI:10.3390/math12060806.

[4] Xie Z , Li Z , He X ,et al.ChatTS: Aligning Time Series with LLMs via Synthetic Data for Enhanced Understanding and Reasoning[J]. 2024.

[5] Sheikhottayefe M , Esmaily Z , Dehghani F .Spawning Gradient Descent (SpGD): A Novel Optimization Framework for Machine Learning and Deep Learning[J].SN Computer Science, 2025, 6(3):1-28.DOI:10.1007/s42979-025-03750-7.

[6] Tikhonov A N .Solution of Incorrectly Formulated Problems and the Regularization Method[J].Observatory, 1962.DOI:http://dx.doi.org/.

[7] Pham D L , Prince J L .An Adaptive Fuzzy C-Means Algorithm for Image Segmentation in the Presence of Intensity Inhomogeneities[J].Pattern Recognition Letters, 1998, 20(1):57-68.DOI:10.1016/S0167-8655(98)00121-4.

[8] Hongmei C , Haifeng X , Lifang Z ,et al.Competitive and collaborative representation for classification[J].Pattern Recognition Letters, 2020, 132:46-55.DOI:10.1016/j.patrec.2018.06.019.

[9] Bergstra J , Bengio Y .Random Search for Hyper-Parameter Optimization[J].Journal of Machine Learning Research, 2012, 13(1):281-305.DOI:10.1016/j.chemolab.2011.12.002.

[10] Zalán Borsos, Khorlin A , Gesmundo A .Transfer NAS: Knowledge Transfer between Search Spaces with Transformer Agents[J]. 2019.DOI:10.48550/arXiv.1906.08102.

[11] Viana D , Teixeira R , Soares T ,et al.Generative Adversarial Networks forSynthetic Meteorological Data Generation[C]//EPIA Conference on Artificial Intelligence.Springer, Cham, 2025.DOI:10.1007/978-3-031-73500-4_17.

[12] Hao Wu J L S .Does overfitting affect performance in estimation of distribution algorithms[J].ACM, 2006.DOI:10.1145/1143997.1144078.

[13] Xu R , Liu B , Zhang K ,et al.Noise-robust few-shot classification via variational adversarial data augmentation[J].Computational Visual Media, 2025, 11(1):227-239.DOI:10.26599/CVM.2025.9450403.

[14] Kakade S M , Foster D P .Multi-view Regression Via Canonical Correlation Analysis[J].Springer Berlin Heidelberg, 2007.DOI:10.1007/978-3-540-72927-3_8.

[15] ZHIQIANG,GAO,PANOS,et al.Stability of the pseudo-inverse method for reconfigurable control systems[J].International Journal of Control, 2007, 53(3).DOI:10.1080/00207179108953643.

[16] Zou H , Hastie T .Regularization and variable selection via the elastic net[J].Journal of the Royal Statistical Society, 2005, 67(5):768-768.DOI:10.1111/j.1467-9868.2005.00527.x.

[17] Yoon J C , Lee I K .Stable and controllable noise.[J].Graphical Models, 2008, 70:105-115.DOI:10.1016/j.gmod.2008.04.001.

[18] Bridges T J , Kostianko A , Zelik S .Validity of the hyperbolic Whitham modulation equations in Sobolev spaces[J]. 2020.DOI:10.1016/j.jde.2020.11.019.