Performance Analysis & Implementation of Regularization Techniques In Linear Regression

[C K Lekhana | GitHub | LinkedIn]

Abstract:

Ordinary Least Squares is an unbiased coefficient estimation method minimising RSS. It often struggles with overfitting and multicollinearity due to larger variances in the estimation results. To overcome this problem of overfitting, regularization techniques such as lasso and ridge regression were introduced. The study mathematically demonstrates how the penalty terms in lasso and ridge regression mitigate overfitting and multicollinearity. The empirical analysis highlights that the regularization techniques have better prediction performance, stabilizes coefficients and leverages lasso for feature selection. This work aims at deeper understanding of loss functions, regularization and gradient descent optimization, providing a strong foundation for future machine learning projects.

1. Mathematical Foundations of Ordinary Least Squares, Ridge Regression and Lasso Regression

This section provides mathematical derivations of the listed Linear Regression models: OLS, Ridge and Lasso. Here, only the relevant equations required to understand the models at a high level are considered. For a deeper understanding, you can refer to the book "The Elements of Statistical Learning Data Mining, Inference, and Prediction" by Trevor Hastie, Robert Tibshirani & Jerome Friedman.

1.1 Ordinary Least Squares (OLS)

Linear Regression maps a series of input variables (x_i) to a continuous output variable (y), in the following manner.

$$y = f(X) \tag{1}$$

$$X: input\ variables$$

$$y: output\ variable$$

$$f: mapping\ funtion$$

The mapping function is a linear equation in the form below:

$$f(x) = \beta_0 + \sum_{j=1}^{p} x_j \beta_j \tag{2}$$

$$\beta_i: coefficients\ of\ the\ input\ variables\ x_i$$

$$p: number\ of\ input\ variables$$

Our objective here is to find the "best fit" line, implying estimating β_i such that the error between the actual value y and the predicted value $\hat{y}$ is minimum or 0.

$$loss = y - \hat{y} \tag{3}$$

This difference is called as loss. But, as a standard loss function considered is Residual Sum of squares, given by

$$RSS = \sum_{i=1}^{N} (y_i - f(X_i))^2 = \sum_{i=1}^{N} (y - \hat{y})^2 \tag{4}$$

$$N: number\ of\ instances\ /\ data\ points\ in\ the\ dataset$$

Ordinary Least Squares is one such estimation method that minimizes RSS while estimating the coefficients β_i. OLS follows the below steps to estimate β_i.

Step1: Represent the equation 2 in linear algebra notations.

$$Y = X\mathrm{\beta} \tag{5}$$

$$Y: N \times 1\ matrix\ of\ actual\ values$$

$$X: N \times (p+1)\ matrix\ of\ predictor\ variables$$

$$\beta: (p+1) \times 1\ coefficients\ matrix$$

Step 2: Based on step 1, we can now define RSS as,

$$RSS = (Y - X\beta)^T (Y - X\beta) \tag{6}$$

Step 3: Define first order and second order gradients of RSS with respect to β.

$$\frac{\partial \mathrm{RSS}}{\partial \beta} = -2X^T(Y - X\beta) \tag{7}$$

$$\frac{\partial RSS}{\partial \beta\, \partial \beta^T} = -2X^T X \tag{8}$$

Step 4: Assume X has full column rank (linearly independent matrix) and $X^T X$ is positive definite (results in non-zero vector). Then, we set the first order derivative in equation 7 to 0.

$$\frac{\partial RSS}{\partial \beta} = 0$$

$$-2X^T(Y - X\beta) = 0$$

$$X^T X\beta = X^T y$$

$$\hat{\beta} = (X^T X)^{-1} X^T Y \tag{9}$$

$$\hat{y} = X\hat{\beta} \qquad (10)$$

Equation 9 fetches the estimated coefficients β_i and Equation 10 is used to obtain the predicted values for X using the estimated coefficients $\hat{\beta}$.

Following are the limitations of this method:

i. We are assuming that the input variables are linearly independent, but in actuality there actually exists subsets of correlated features. Then the X becomes singular matrix, thus we cannot obtain coefficients through this method.

ii. It is an unbiased method, meaning, the loss function RSS actually penalizes all those coefficients with large errors irrespective of their direction. This directly results in having larger variances in error and leads to overfitting. Overfitting fits training data perfectly but underperforms in test set leading to very high prediction error rates in test set.

These are the two precise problems we need to overcome these limitations. The solutions to these problems can be broadly discussed as subset-based solutions and the other as regularization-based solutions. This study is focused on regularization-based methods only. Section 1.2 discusses the two important regularization methods.

1.2 Regularization Methods

Regularization methods introduce a penalty term L to the loss function RSS. The penalty term added will shrinks the coefficients near to 0 (in Ridge) or to 0 (in Lasso), therefore reducing the variance in errors, while tackling the problem of multi-collinearity. Intention of adding penalty term is used introduce minimal bias in the model to obtain smaller variance in error, this is called bias-variance trade off. The use cases of both Lasso and Ridge become clearer by the end of both mathematical presentations and empirical analysis.

1.2.1 Ridge/L2 Penalty Regression

In this regularization technique, we add $L2$ penalty to RSS, which penalizes the coefficients of the variables contributing to large variance by shrinking them near to 0. By this the model becomes simpler due to shrinkage of coefficients and thus solves the problem of multicollinearity. This can be better understood while performing empirical analysis. The following are the equations that introduce $L2$ penalty to RSS.

$$\hat{\beta}_{ridge} = argmin\left\{\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_i\beta_i\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2\right\} (11)$$

(Note: $argmin()$ is a notation that indicates that the function inside () is to be minimized to obtain the results)

Equation 11 in linear algebra notations is given by,

$$\hat{\beta} = (Y - X\beta)^T(Y - X\beta) + \lambda\beta^T\beta \qquad (12)$$

λ in Equation 11 & 12 is the regularization parameter that determines the scale/ rate of shrinkage. Equation 12 further can be reduced to,

$$\hat{\beta} = (X^TX + \lambda I)^{-1}X^TY \qquad (13)$$

1.2.2 Lasso/L1 Penalty Regression

In this method, a penalty $L1$ similar to $L2$ is introduced to RSS, and thus this method also shrinks the coefficient but it also reduces the coefficients of correlated variables to 0 as well. This characteristic is leveraged to perform feature selection. The following equation gives the mathematical structure to lasso regression.

$$\hat{\beta}_{lasso} = argmin\left\{\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_i\beta_i\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j|\right\} (14)$$

It is important to note that the resulting $\hat{y}\ is\ non-linear\ and\ open\ form$. Since this penalty results in a lot of 0 coefficients, it simplifies the model, due to sparsity in the resulting coefficients β. Since, a closed form solution is not possible, we look at other iterative solutions that make implementation of Lasso regression possible. Here, we explore the Coordinate Descent implementation of Lasso Regression. [Medium article]

In Coordinate Descent Method, each of the coefficients(features) in equation (2) is updated independently by controlling for the other coefficients using a thresholding technique defined as:

$$\beta_j = \sum\left(\frac{S(\rho_j, threshold)}{X_j^2}\right) \qquad (15)$$

where $S(\rho, threshold)$ is the soft threshold function defined as:

$$S(\rho_j, threshold) = \begin{cases} \rho_j + threshold, & if\ \rho_j < -threshold \\ 0, & if - threshold < \rho_j < threshold \\ \rho_j - threshold, & if\ \rho_j > threshold \end{cases} \qquad (16)$$

where ρ_j is defined as:

$$\rho_j = \sum_{i=1}^{N} X_{ij}(y_i - (\sum_{k\neq j} X_{ik}\theta_k)) \qquad (17)$$

2. Methodology

This section details of the experimental setup considered to validate the hypotheses listed in section 2.

2.1 Dataset Description

This study uses two kinds of datasets: one synthetic dataset and Concrete Compressive Strength (CCS) Dataset [Kaggle] is considered. The synthetic dataset generated is to ensure perfectly controlled multicollinear data, for effective analysis of algorithms. The CCS dataset is used to demonstrate effective understanding of regularization methods in real world.

2.2 Model Implementation

For custom OLS and Ridge Regression implementation, both Linear Algebra based and iterative (Gradient Descent) based models is designed, but for custom Lasso Regression model only iterative (Coordinate Descent) solution is considered. All custom implementations of Linear, Ridge & Lasso Regression Numpy is primarily used. Synthetic Data Generation and preprocessing of CCS Data is performed using scikit-learn. Scikit-learn is also used for dataset splitting and evaluation metrics calculation. Matplotlib is used as a visualization tool.

2.3 Performance Metrics

Mean Squared Error (MSE) and R2 Score is accounted for all models in the experimentation. While MSE provides insight into the error percentage in the final model, R2 score provides information on how better the model is performing in comparison to when averaged coefficients.

$$MSE = \frac{\sum(y-\hat{y})^2}{N}$$

$$R2\ Score = 1 - \frac{RSS}{TSS}$$

$$RSS : Residual\ Sum\ of\ Squares$$

$$TSS : Total\ Sum\ of\ Squares$$

3. Results & Discussion

In this section, the results obtained for all the models built and trained are presented and discussed to build the skill of interpretation of model results. The models are built & trained in broadly three groups: (1) Linear, Ridge & Lasso Regression models from Scikit-Learn, (2) Linear, Ridge & Lasso Regression modelled using SGDRegressor for Gradient Descent (GD) implementation & (3) Custom implementations of 5 models : Linear Regression (OLS & GD method), Ridge Regression (Closed Form & GD method) and Lasso Regression (Coordinate Descent method).

Regarding the dataset, two heatmaps was programmed for both the synthetic and the CCS dataset. Both of which showed high correlations between certain features, which can be clearly observed from the following heatmaps. [Heatmap 1 & Heatmap 2].

In terms of the linear models built using scikit-learn, all had similar performance with no significant differences in their R2 Score but lasso regression model showed a little higher MSE in comparison, which can be associated with the presence of high collinearity in the datasets. In custom implementations of the iterative models, the Ridge regression with gradient descent performed badly in comparison with the other models. It was also observed that the lasso with coordinate descent converged faster than other two gradient descent implementations. The faster convergence could be because the coordinate descent aims at finding the best value for each coefficient while keeping the others constant. The performance of the model was mostly similar in both the synthetic and CCS dataset. However, because the CCS dataset did not show a very high collinearity, the linear model mostly performed better than the regularization models. Since, the datasets did not present any signs of overfitting during the model building and training, the performance of the regularization models overcoming overfitting could not be explored thoroughly. We were able to determine the relevant features in the datasets using lasso regression model, helping with understanding how lasso regression can help in feature selection. The most relevant features were consistent in all the implementation of lasso algorithm considered in the project.

[Refer Notebook for detailed visualization and interpretation]

4. Conclusion

In conclusion, this study was instrumental in solidifying my understanding of regularization techniques and their importance in mitigating common challenges in linear regression. The custom implementations, particularly those built with NumPy, offered invaluable mathematical and computational insights into loss functions and gradient descent optimization. This foundational knowledge will undoubtedly enhance future data analysis efforts and guide more effective model selection. Crucially, this project has provided a practical framework and sharpened my intuition for structured machine learning research.