Report 2

A Comparative Study of Regularization Methods: Ridge, LASSO, and Elastic Net Regression

Name :

Amin Gamal : 202202219

Hamza Abdelmoreed : 202201508

Mohamed Ehab Yousri : 202201236

Date : 3 / 12 / 2024

Table of Contents

Regularization Methods:

- Ridge Regression.

-LASSO Regression.

-Elastic Net Regression

The report should include the following:

1. Theoretical definition on each regularization method.
2. Implementation of the designated method using its original definition.
3. Finding the optimal regularization parameter (λ) in each case using cross validation, providing the appropriate diagrams.

Ridge Regression

[1] Introduction

Ridge regression is a linear regression technique used to prevent overfitting and to deal with singular matrix features and large matrix dimensions. The penalty term, proportional to the square of the magnitude of the model's coefficients.

[2] Mathematical Formulation

$$F(w) = loss(w) + \lambda * ||w||^2$$

- **{ loss(w) }** Mean-Squared Error of the model with weights {w}
- **{ λ }** Regularization parameter control on penalty
- **{ ||w||^2 }** Euclidean norm of weight vector W.

[3] Impact of the Regularization Term

- **Reduced Overfitting**
  Ridge Regression increases the model applicability for unseen data or outliers and helps in avoiding closely fitting noise in the training datasets by adding penalties to high coefficients.

- **Improved Stability**
  Capable of shortening the coefficient of the model as it could allow many high correlations among the features.

[4] Choosing the Regularization Parameter (λ)

- Large λ can lead to more potential bias
- Can be more strong regularization, decreased variance.

Depend on

- Cross-Validation:
  By evaluating the model's performance on a validation set for different values.

Find code implementation here: Math_Report#2.ipynb

[5] Advantages of Ridge Regression

As considering in part 3 impact of Ridge R(w) term point difference between linear regression and ridge regression

- **Improved Model Stability**

  Ridge regression can handle multicollinearity

- **Reduced Overfitting**

  By preventing the model from fitting the noise in the training data.

[6] conclusion

Ridge Regression is a powerful tool to combat overfitting and handle multicollinearity, ensuring your model remains robust even in high-dimensional data settings.

[7] Pseudo code

```
1. Add a bias term (if needed)
   X (training features matrix) / y (target variable) / lambda (regularization parameter)
2. Initialize identity matrix
3. Compute the regularized matrix
4. Compute the weight vector (w)
5. Predicting with Ridge Regression
```

LASSO Regression

-Theoretical definition:

LASSO Regression Which is the Least Absolute Shrinkage and Selection Operator is a linear regression technique that include L1 regularization. Not the same other models , L1 regularization adds a penalty based on the absolute values of the coefficients, which can shrink some coefficients to exactly zero.

This makes LASSO useful for selecting the most important features in a dataset.

Its main goal is to minimize an objective function that balances the model's accuracy and simplicity.

objective function:

$$\min_{\beta} \left(\frac{1}{2n} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^\top \beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right)$$

Where:

-xi: Features of the i-th observation.

-yi: Target value for the i-th observation.

-β: Coefficients of the model.

-λ: Determines the strength of the penalty on large coefficients.

-Impact of the Regularization Term:

The regularization term in LASSO, controlled by λ\\lambdaλ, shapes the model by adding a penalty to the absolute size of the coefficients.

-Small λ: Minimal penalty, model acts like linear regression and may overfit.

-Large λ: Strong penalty, simplifies the model but may underfit by ignoring important features.

-Methodology:

1-From scratch implementation:

-Built LASSO Regression from scratch using gradient descent.

-Included L1 penalty to shrink less important coefficients to zero.

-Used cross-validation to determine the best regularization parameter (λ).

2-Python Built-in Library:

Used the `Lasso` class from the `sklearn` library for comparison.

Performed a grid search (`GridSearchCV`) to find the optimal λ.

3-Comparison:

-Compared test Mean Squared Error (MSE) and prediction results from both implementations.

-Created visualizations to show similarities and differences.

-Results and Analysis:

Housing Dataset

-Custom Implementation: Optimal λ=0.001 = 0.001λ=0.001, Test MSE = 31.83.

-Built-in Implementation: Optimal λ=0.0298 = 0.0298λ=0.0298, Test MSE = 31.83.

Both implementations gave consistent results with similar predictions.

Advertising Dataset

-Custom Implementation: Optimal λ=0.127\lambda = 0.127λ=0.127, Test MSE = 2.91.

-Built-in Implementation: Optimal λ=0.001\lambda = 0.001λ=0.001, Test MSE = 2.91.

Predictions and error metrics were nearly identical, proving the custom method's accuracy.

-Trade-off Between Pros and Cons

Pros:

-Automatically selects features by setting some coefficients to zero.

-Simplifies models, making them easier to understand.

Handles correlated predictors better than standard linear regression.

Cons:

-May exclude important features if λ\lambdaλ is too high, causing underfitting.

-Shrinking coefficients can introduce bias in complex relationships.

Can be unstable when predictors are highly correlated, choosing one over others arbitrarily.

-Pseudocode:

1-Input:

-X: Feature matrix.

-y: Target vector.

-λ: Regularization parameter.

-max_iter: Maximum iterations.

-tol: Convergence tolerance.

2-Initialize:

-Set all coefficients : $\beta=0$.

3-Standardize the features in X (mean = 0, variance = 1).

4- Iterate up to max_iter::

1. Save the current coefficients (β old).
2. For each coefficient jin β:

-Compute the partial residual $rj=y-X\beta+Xj\beta j$

-Update βj using soft-thresholding: $\beta j=\text{sign}(sj)\cdot\max(|sj|-\lambda,0)/(Xj^{\top}Xj)$ where:

- $sj=Xj^{\top}rj$

-Check for convergence: If $// \beta-\beta old // < tol$,break.

Output:

-Final coefficients β.

Applications:

-LASSO is ideal for high-dimensional datasets where feature selection is important.
-It eliminates irrelevant predictors, making it a powerful tool for data science and modeling.

Conclusion:

The LASSO Regression on these two datasets highlights the effectiveness in feature selection and regularization, LASSO Regression simplifies models by reducing insignificant coefficients to zero.
This result showed consistent test MSE values between the from scratch, and the built-in implementation ,This task showed how important regularization is for avoiding overfitting and making models simpler. Overall, LASSO turned out to be a very useful method for regression, especially when working with datasets that have a lot of features.

Elastic Net Regression

[1] Introduction:

Elastic Net Regression is a hybrid approach that incorporates both L1 and L2. By combining these penalty terms, Elastic Net can effectively handle situations where Lasso tends to select too few variables, and where Ridge alone fails to achieve sufficient sparsity.

[2] Mathematical Formulation:

$$\mathbf{F(w) = loss(w) + \lambda\ (\alpha\ ||w|| + ((1 - \alpha) / 2) * ||w||^2)}$$

- **{ loss(w) }** Mean-Squared Error of the model with weights {w}
- **{ λ }** Controls overall strength of regularization.
- **{ $||w||^2$ }** Euclidean norm of weight vector W.
- **{ α }** Balances between the L1 and L2 penalty.

[3] Choosing the Regularization Parameters (λ, α)

For λ :

- Large λ means more regularization and vice versa.

For α :

- α = 1: Pure Lasso, strong sparsity.
- α = 0: Pure Ridge, no sparsity, but stable solutions.
- 0 < α < 1: Mixture between L1 and L2.

Depend on

- Cross-Validation:
  By evaluating the model's performance on a validation set for different values.

Find code implementation here: Math_Report#2.ipynb

[4] Trade off between pros and cons of Elastic Net Regression:

Pros:

- Stability: Elastic Net distributes weights fairly, improving model stability.
- Flexibility : Balance between the benefits of Lasso (sparsity) and Ridge (shrinkage).

Cons:

- Extra parameter to tune: Unlike Ridge or Lasso, Elastic Net introduces another parameter α\alphaα (the mixing ratio) in addition to λ. This increases the complexity of the model selection process.
- Computational cost: The search over both λ and α can increase computational cost.

[5] Summary:

Elastic Net is a flexible method that combines the strengths of Lasso and Ridge. By balancing L1 and L2 penalties it gives us a method that can handle correlated features more gracefully than Lasso alone.

In conclusion, Elastic Net often provides a balanced solution, delivering both stable feature selection and predictive performance at the cost of having an additional parameter to tune.

[6] References:

1- Elements of Statistical Learning: data mining, inference, and prediction. 2nd Edition.

2- Guide on Ridge and Lasso Regression in Python - Analytics Vidhya

3- Scikit-learn ElasticNetCV: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNetCV.html

4- Ridge Regression Implementation in Python | by Amit Yadav | Biased-Algorithms | Medium