A Gentle Introduction to Lasso and Ridge Regression

昆山杜克大学
DUKE KUNSHAN
UNIVERSITY

Leo Tian-Lai Chen
tc289@duke.edu

Jordan Zi-Chao Chen
zc142@duke.edu

Ikea Yi-Jia Xue
yx179@duke.edu

Alan Cheng-Lin Zhang
cz155@duke.edu

***K*eywords** Lasso · Ridge Regression

1 Introduction

This paper introduces Lasso and Ridge regression to machine learning novices with a working knowledge of linear algebra, vector calculus, and optimization. It is divided into four distinct sections. The purpose and history of inventing the Ridge and Lasso method are discussed in the Goal section. The Model section discusses the fundamental concepts and algorithms underlying these two techniques. The Training section introduces two effective approaches for optimizing Lasso and Ridge regression, which are QR decomposition and Proximal Gradient Descent. In the Application section, Lasso and Ridge regression techniques are applied to a real-world data set, and their performance is compared to that of ordinary linear regression.

2 Goals

The common model in linear regression is Ordinary Least Square (OLS):

$$\mathbf{Y} = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_p x_p \tag{1}$$

which can be written in matrix form as:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{w} \tag{2}$$

The matrix $\mathbf{X}$ is a $n \times (p+1)$ data matrix, and $\boldsymbol{w}$ represents unknown weights or coefficients of size $(p+1) \times 1$. Both include the intercept terms. The weights are computed via minimizing the residual sum-of-squares,

$$\text{min} \;\; RSS = (y - \mathbf{X}w)^T (y - \mathbf{X}w) \tag{3}$$

which yields

$$\hat{\boldsymbol{w}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1} \mathbf{X}^T\mathbf{Y} \tag{4}$$

The simple linear regression model, on the other hand, has two significant shortcomings in terms of prediction accuracy and model interpretability.

Prediction Accuracy When n, the number of observations, is not significantly greater than p, the number of features, OLS performs poorly. The issue is determined by the high degree of variability and overfitting. Even worse, if p is larger than n, the model fails because basic least squares has no unique solution.

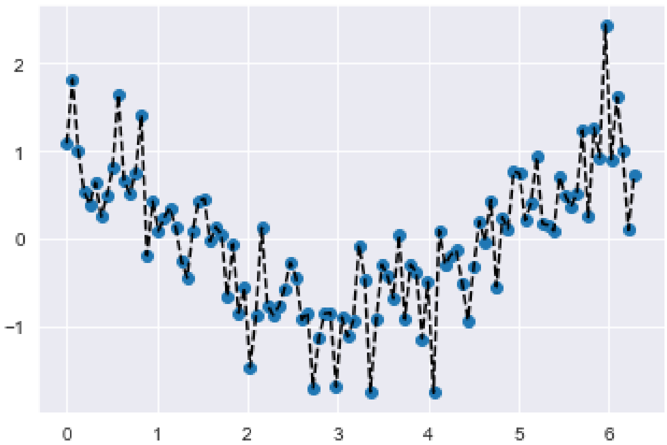


Figure 1: OLS regression result and Relationship between response and one of the predictors

An visualization of the overfitting problem in OLS is shown in Figure 1[1]. The example data set contains 100 variables, each with 100 values (n=100, p=100). In this case, n is not greater than p. It can be seen in the figure that OLS performs poorly and is completely overfit.

Model Interpretability OLS can hardly produce zero-weighted features, and yet certain characteristics are genuinely irrelevant to the response. With irrelevant features, such a result is difficult to interpret.

These difficulties can be addressed by incorporating regularization into least square fitting. The primary objective of Lasso and Ridge regression is to minimize the weights. While fitting all features, this class of approaches continuously reduces the model's variance with acceptable increase of bias [1]. The following sections provide background information on Lasso and Ridge regression.

Because of the drawbacks of the OLS, the model of the Ridge and Lasso regression was invented. Hoerl and Kennard introduced the model of the Ridge regression in 1970, which cost function is

$$\text{min} \;\; RSS + \lambda \sum_{j=1}^{p} w_j^2 \tag{5}$$

And Tibshirani invented Lasso in 1995, which cost function is

$$\text{min} \;\; RSS + \lambda \sum_{j=1}^{p} |w_j| \tag{6}$$

In the second term of the cost functions, λ is called the "tuning variable". Below are some brief introductions of the properties of the cost functions of the Ridge and Lasso, and the properties will be discussed in details in the **Model** section.

Statistically speaking, Ridge and Lasso regression are designed to solve some potential issues of naive solution in equation 4. Specifically, in the naive solution of the OLS ($\hat{\boldsymbol{w}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Y}$), $\mathbf{X}^T\mathbf{X}$ is not always a unit matrix in which all the diagonal elements are equal to 1 and all the other elements are zero. Under this case, the sensitivity of the estimates will increase. For the ridge and lasso regression, penalty term adds a bias-variance trade-off into the model and thus it can work well in situations where the OLS have high variance. As when λ increases, the decreasing of the flexibility of the ridge and lasso regression leads to the reduction in the variance of the predictions.

In practice, Ridge regression can be used when the number of independent variables is greater than the sample size, or there is significant multi-collinearity among the independent variables. Under these situations, $\mathbf{X}'\mathbf{X}$ is problematic which leads to the large distance between $\boldsymbol{w}$ and $\hat{\boldsymbol{w}}$. It works best under situations where RSS estimation has high variance [2]. Nowadays, Ridge regression has many applications in various fields including economy, environmental science and biomedicine [3][4][5].

[1] "https://github.com/suemnjeri/medium-articles/blob/main/regularization/cosine_df_extra_columns.csv"

Lasso was also created to address the same statistical issue as OLS. Although Ridge regression is quite consistent in terms of prediction accuracy, it does not produce weights that are exactly zero. Lasso resolves this issue of interpretability. It shrinks some weights and sets others to 0, preserving important characteristics and generating an easily interpretable model [6].

When dealing with a data collection, researchers frequently want to determine which characteristics of the data set are most critical to the model. The LASSO approach of feature selection may be one method for resolving this challenge. The lasso regression has a variety of real-world applications, including visual field progression prediction, political polling, and genome-wide association study [7][8][9].

3 The Model

3.1 The Ridge Regression

From the above discussion in the **Goal** section, it is known that by adding a tuning variable λ to the original cost function of OLS, the coefficients in the regression models are estimated by minimizing a slightly different cost function, which is:

$$\sum_{i=1}^{n}\left(y_i - w_0 - \sum_{j=1}^{p} w_j x_{ij}\right)^2 + \lambda \sum_{j=1}^{p} w_j^2 = RSS + \lambda \sum_{j=1}^{p} w_j^2 \tag{7}$$

As we can see from the cost function, the Ridge regression also seeks to fit the data well by making RSS as small as possible. However, instead of just minimizing the RSS in OLS, ridge regression adds a term $\lambda \sum_{j=1}^{p} w_j^2$ to the cost function. Here, $\lambda \sum_{j=1}^{p} w_j^2$ is called a "shrinkage penalty". It is important to note here that λ is a hyperparameter of the model, which is to be determined separately.

The second term in the cost function, $\lambda \sum_{j=1}^{p} w_j^2$, will be small (close to zero) and will have small contribution to the overall cost if all the coefficient estimates $w_1, w_2, \cdots, w_n$ is close to zero (except for the intercept w_0, which is not included in the cost function). Thus, the Ridge regression has the effect of shrinking the coefficient estimates w_j to zero. However, the choice of the "tuning parameter" has a great impact on the actual effect of the shrinkage. For example, when $\lambda = 0$, the penalty term $\lambda \sum_{j=1}^{p} w_j^2$ will have no effect in the cost function, and the cost function will be no different with the cost function of OLS. On the other hand, when λ is very large, let's say, it is approaching $+\infty$, then the effect of the shrinkage penalty will also become significantly large that the coefficient estimates $\hat{w}_j$ will approach zero.

Thus, unlike the OLS, which will get only one optimized coefficient vector if given the training data points X_train and y_train, Ridge regression will produce different sets of coefficient estimates $\hat{w}_\lambda$ for each λ that we select. Thus, the selection of λ is very important in the model performance. Strategies like the cross-validation can be employed to find a suitable λ.

3.2 How does the Ridge improve over the Least Squares

The penalty term $\lambda \sum_{j=1}^{p} w_j^2$ adds a bias-variance trade-off into the original OLS model. For example, as λ increases, the model generated by the ridge regression will become simpler as the coefficient estimates $\hat{w}_j$ are all approaching zero, which will lead to a decreased variance (increased ability of generalization of the model) but an increased bias for other datasets D. However, the trade-off by sacrificing some model specialization for the given training dataset for model generalization is extremely useful when we do not have enough observations n in the training dataset, while we have a lot of predictors p (p is close to n or even greater than n, which means the model is very likely to overfit the training dataset).

The book ISLR also gives an example in which a simulated dataset of $n = 50$ and $p = 45$ is given. Figure 2 shows the performance of the Ridge Regression (measured by the test MSE). As we can see, when $\lambda = 0$, there is no additional bias w.r.t. the original OLS model with some variance. As λ increases, the shrinkage of the coefficient estimates leads to a gradual reduction in the variance, at the expense of a slightly larger bias. However, the overall test MSE decreases at an intermediate value of λ. As λ continues to increase to larger values, the model then becomes too simple to perform well with a significantly larger bias.

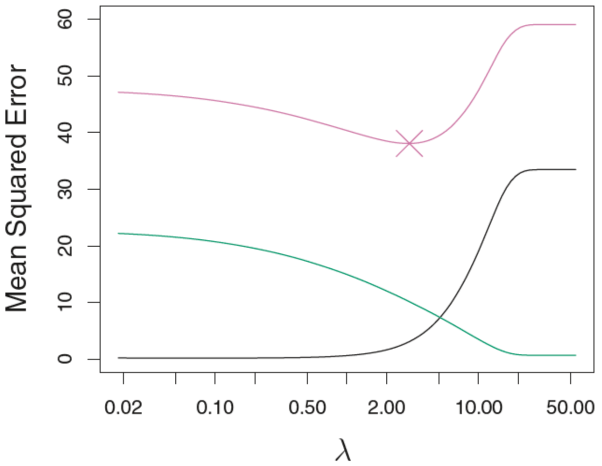


Figure 2: Squared bias (black), variance (green), and test mean squared error (purple) for the ridge regression predictions on a simulated data set [1]

3.3 The Lasso Regression

The cost function of the Lasso Regression is

$$\sum_{i=1}^{n}\left(y_i - w_0 - \sum_{j=1}^{p} w_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|w_j| = RSS + \lambda\sum_{j=1}^{p}|w_j| \tag{8}$$

Compared with the Ridge Regression, the Lasso can also perform the bias-variance trade-off to improve over the OLS, and the selection of λ is also important for the performance of the Lasso. There is only one difference in the cost function of the Lasso Regression: the ℓ_2 penalty (w_j^2) in the Ridge is replaced by the ℓ_1 penalty ($|w_j|$). The ℓ_1 norm of the coefficient vector $||w_j||_1 = \sum |w_j|$.

However, because of that, one disadvantage of the Ridge Regression can be overcome: the interpretability of the model, as the penalty term $\lambda\sum_{j=1}^{p} w_j^2$ will not necessarily set the coefficient estimates w_j in the Ridge to be exactly zero, which means it will still generate a model containing all the predictors, though the value of some of the coefficient estimates may be small. On the contrary, because of the ℓ_1 penalty term in the Lasso Regression, some of the coefficient estimates will be likely to become zero, which enables the Lasso to perform variable selection, making the generated model much easier to interpret. The reason why it is the case will be discussed right away in the next subsection.

3.4 The variable selection property of the Lasso

As is mentioned before, the variable selection property of the Lasso Regression can generate models simpler and easier to interpret than those generated by the Ridge. This section will explain the reason behind the phenomenon.

The cost function of the Lasso Regression can be rewritten into the form of the constrained optimization, which is

$$\min_{w}\left(\sum_{i=1}^{n}\left(y_i - w_0 - \sum_{j=1}^{p} w_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|w_j|\right) \quad \text{subject to} \quad \sum_{j=1}^{p}|w_j| \leq s \tag{9}$$

Similarly, for the Ridge Regression, its cost function can be rewritten as

$$\min_{w}\left(\sum_{i=1}^{n}\left(y_i - w_0 - \sum_{j=1}^{p} w_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p} w_j^2\right) \quad \text{subject to} \quad \sum_{j=1}^{p} w_j^2 \leq s \tag{10}$$

Take $p = 2$ as an example, which means there are two predictors in this case, then it is given that $|w_1| + |w_2| \leq s$ for the cost function of the Lasso, and $w_1^2 + w_2^2 \leq s$ for the cost function of the Ridge. If the contour line of the OLS and the constraint is plotted, we can have

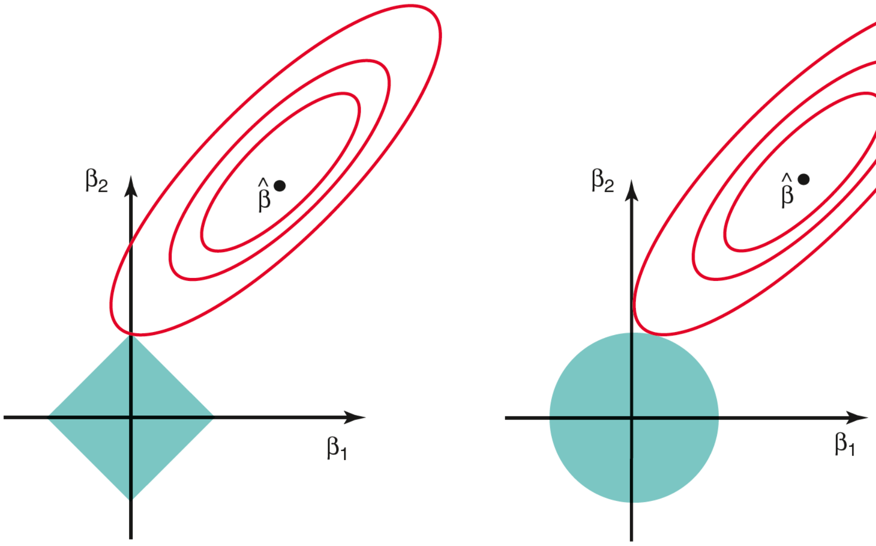


Figure 3: The contour lines of OLS and the constraint of the Lasso (left) and the Ridge (right) [1]

In the plot of $p = 2$, it can be seen that the optimized coefficient estimates $\hat{w}$ in the Lasso and Ridge Regression are given by the first point at which the contour lines of the OLS (the ellipse) contacts the shaded constraint region. Since Ridge Regression has a circular constraint with no sharp points, the contour line and the constraint region will not generally intersect on an axis, and so the estimated coefficient of the Ridge Regression will be non-zero. On the contrary, the Lasso constraint has corners at each of the axes, and so the contour line will often intersect the constraint region at an axis. When this occurs, one of the coefficients (w_1 or w_2) will equal zero. In higher dimensions, many of the coefficient estimates may equal zero simultaneously [1].

3.5 Bayesian interpretation of the cost functions

Some may ask why the cost function of the Ridge and Lasso regression are defined like the formula (7) and (8). This section will briefly interpret the definitions of the cost functions from the Bayesian lens of statistics and MAP (maximum a posteriori probability).

A Bayesian viewpoint for regression assumes that the coefficient vector w has some prior distribution. And in this case, we want to know what should the coefficient estimation $\hat{w}$ be given a fixed data matrix X and Y. In other words, we want to find the maximum posterior probability of a coefficient estimation $\hat{w}$ given a fixed data matrix X and Y and the prior probability of $P(w)$. Thus, here we assume the coefficient vector of the linear model is given by $w = (w_0, w_1, \cdots, w_n)$ (w_0 is the intercept), and assume the data matrix is given by $X = (1, x_1, x_2, \cdots, x_n)$ and Y. Then, the regression model is given by $Y = w^T X + \epsilon$, in which ϵ is the error term for the regression estimation which follows the Gaussian distribution $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$. The distribution of the error term is thus equivalent to $Y|X, w \sim \mathcal{N}(w^T X, \sigma^2 I)$. The probability can be rewritten as

$$P(Y|X,w) = \frac{1}{\sqrt{2\pi}\sigma I} \exp\left(-\frac{(Y - w^T X)^T (Y - w^T X)}{2\sigma^2}\right) \tag{11}$$

Since X is a fixed observation, $P(Y|X,w)$ can be rewritten as $P(Y|w)$ (same as $P(w|Y,X)$, which can be rewritten as $P(w|Y)$. Recall the Bayesian Theorem that the posterior probability can expressed as

$$P(w|Y) = \frac{P(Y|w)P(w)}{P(Y)} \tag{12}$$

Thus, the maximum posterior probability of w given the regression estimation Y and the fixed dataset X is given by

$$\hat{w} = \max_w P(w|Y) = \max_w \left(P(Y|w)P(w)\right) = \max_w \left(\log(P(Y|w)P(w))\right) \tag{13}$$

The equation (13) is derived by the log-likelihood. Thus, here, if w_j follows a Gaussian distribution of mean zero and standard deviation $\sigma_0 I$, we have

$$P(w) = \frac{1}{\sqrt{2\pi}\sigma_0 I} \exp\left(-\frac{w^T w}{2\sigma_0^2 I}\right) \tag{14}$$

Then, if the equation (11) and (12) are substituted back into the equation (13), after some calculation, we will finally get

$$\hat{w} = \min_{w} \left((Y - w^T X)^T (Y - w^T X) + \frac{\sigma^2}{\sigma_0^2} w^T w \right) \tag{15}$$

Assume $\lambda = \sigma^2/\sigma_0^2$, then we can find the equation (15) is exactly the same as the cost function of the Ridge Regression. Thus, the interpretation here is, given the dataset X and Y, if we propose a coefficient vector $\hat{w}$ which can minimize the formula (15), then it should be the coefficient vector estimation for the dataset with the maximum likelihood.

Similarly, for the Lasso Regression, if w is expected to follow a double-exponential (Laplace) distribution with mean zero and scale parameter a function of λ [1], then it follows that the $\hat{w}$ which can minimize the cost function of the Lasso Regression should be the estimation for the given dataset with the maximum likelihood.

3.6 How to fit the Lasso and the Ridge Regression

The Ridge and Lasso Regression also have significant computational advantage over some shrinkage methods like the best subset selection, which takes exponential time to solve and thus is very expensive. The algorithm to fit a Ridge Regression is very similar with that of the OLS. If we rewrite the cost function of the Ridge Regression into the matrix form, we can have

$$\mathcal{L}(w) = (y - Xw)^T (y - Xw) + \lambda w^T w \tag{16}$$

Thus, similar to the derivation process of solving the best fit $\hat{X}$ for $f(X) = ||AX - B||^2$, which is to find the first-order derivative of $f(X)$ with respect to X, here, for $\mathcal{L}(w)$, the coefficient estimation $\hat{w}$ is given by calculating $\partial \mathcal{L}(w)/\partial w = 0$, which will yield $\hat{w}^{ridge} = (X^T X + \lambda I)^{-1} X^T y$. There are efficient algorithms like the QR factorization to solve $\hat{w}^{ridge}$ accurately on computers. For Lasso, there are also efficient algorithms with almost same amount of work as that of the OLS. Some fitting algorithms will be discussed in the **Training** section below.

4 Training

4.1 QR Decomposition for Ridge Regression

As is discussed in the last subsection of the **Model** section, the naive solution for fitting the Ridge regression is given by

$$\hat{w}_{\text{ridge}} = \left(\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I}\right)^{-1} \mathbf{X}^T y = \left(\sum_n x_n x_n^T + \lambda \mathbf{I}\right)^{-1} \left(\sum_n y_n x_n\right)$$

Although the primal estimation of w_{ridge} can be computed by setting the derivative of cost function to 0, the solution involves matrix inversion computation. It is inefficient and numerically insecure. Hence, several optimization algorithm are utilized including QR decomposition, singular value decomposition (SVD), and Conjugate Gradient method etc. The next paragraphs attempt to illustrate the QR decomposition for Ridge regression. Hopefully, readers will get intuitive as well as mathematical understandings.

What is QR decomposition? If A is an $m \times n$ matrix with linearly independent columns, then A can be factored as $A = QR$, where Q is an $m \times n$ matrix whose columns form an orthonormal basis for column space $\mathrm{Col}\, A$ and R is an $n \times n$ upper triangular invertible matrix with positive entries on its diagonal. Often in case of massive problems, $Ax = b$ might have no solutions. Thus, one might need approximate Ax to b, which is called the least-square solution of $Ax = b$ s.t.

$$||b - A\hat{x}|| \leq ||b - Ax||$$

Intuitively speaking, it is trying to find an x which makes Ax the closest point to b as shown in the Figure 4 [10].

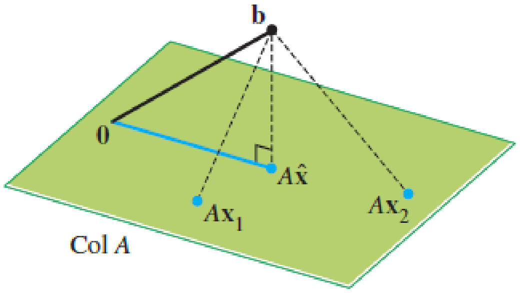


Figure 4: The vector b is the closest to $A\hat{x}$ than to Ax for other x

Combine these two concepts, one could use QR decomposition for least-sqaure solutions, especially in regression problems, where small errors in calculation might cause relatively large errors in solutions. The QR decomposition offers more reliable solution. The process is shown below.

$$A\hat{x} = QR\hat{x} = b$$

Thus,

$$\hat{x} = R^{-1}Q^T b$$

Because Q is orthonormal matrix, then $Q^T = Q^{-1}$

Modification on data matrix So far, sufficient knowledge are provided for QR decomposition. But before apply this method to the Ridge regression, one have to modify the original data matrix first so that it fits the form $Ax = b$.

From previous section, one could find that the prior distribution of weights is:

$$p(\boldsymbol{w}) = \mathcal{N}\left(0, \Lambda^{-1}\right) \tag{17}$$

where Λ is $\frac{1}{\sigma_0^2}I$. This is equivalent to equation 14.

The trick here is to add virtual data to the training set. $\mathbf{X}, y$ for training data can be modified as

$$\tilde{\mathrm{X}} = \begin{pmatrix} \mathrm{X}/\sigma \\ \sqrt{\Lambda} \end{pmatrix}, \quad \tilde{y} = \begin{pmatrix} y/\sigma \\ 0_{p\times 1} \end{pmatrix} \tag{18}$$

where σ is from previous section that $Y|X, w \sim \mathcal{N}(w^T X, \sigma^2 I)$, which can be viewed as the standard deviation of residual between estimated value $\hat{y}$ and real value y. $\tilde{\mathrm{X}}$ is $(n+p)\times p$, where the extra rows represent pseudo-data from the prior distribution. Notice, for simplicity, the intercept term is dropped here. The following equations show that the expanded data is equivalent to the penalized RSS on the original data.

$$\begin{aligned}
f(w) &= (\tilde{y} - \tilde{\mathrm{X}}w)^T(\tilde{y} - \tilde{\mathrm{X}}w) \\
&= \left(\begin{pmatrix} y/\sigma \\ 0 \end{pmatrix} - \begin{pmatrix} \mathrm{X}/\sigma \\ \sqrt{\Lambda} \end{pmatrix} w\right)^T \left(\begin{pmatrix} y/\sigma \\ 0 \end{pmatrix} - \begin{pmatrix} \mathrm{X}/\sigma \\ \sqrt{\Lambda} \end{pmatrix} w\right) \\
&= \begin{pmatrix} \frac{1}{\sigma}(y - \mathrm{X}w) \\ -\sqrt{\Lambda}w \end{pmatrix}^T \begin{pmatrix} \frac{1}{\sigma}(y - \mathrm{X}w) \\ -\sqrt{\Lambda}w \end{pmatrix} \\
&= \frac{1}{\sigma^2}(y - \mathrm{X}w)^T(y - \mathrm{X}w) + (\sqrt{\Lambda}w)^T(\sqrt{\Lambda}w) \\
&= \frac{1}{\sigma^2}(y - \mathrm{X}w)^T(y - \mathrm{X}w) + w^T\Lambda w \\
&= \frac{1}{\sigma^2}(y - \mathrm{X}w)^T(y - \mathrm{X}w) + \frac{1}{\sigma_0^2}w^T w
\end{aligned} \tag{19}$$

The last line is essentially equivalent to the equation 15. This suggests that the original Ridge regression problem is equivalent to solve the following problem:

$$\tilde{X}w = \tilde{y}$$

Apply QR decomposition Then apply QR decomposition on $\tilde{\mathrm{X}}$ as follow:

$$\tilde{\mathrm{X}} = \mathrm{QR}$$

where $\mathrm{Q}^T\mathrm{Q} = \mathrm{I}$. Using QR decomposition can rewrite this system of equations as follows:

$$\begin{aligned} (\mathrm{QR})w &= \tilde{y} \\ \mathrm{Q}^T\mathrm{QR}w &= \mathrm{Q}^T\tilde{y} \\ w &= \mathrm{R}^{-1}\left(\mathrm{Q}^T\tilde{y}\right) \end{aligned} \tag{20}$$

Thus, one could find the least-square solution for Ridge regression. Another prominent advantage of using QR method is that $\mathbf{R}$ is upper triangular matrix, so one can solve this last set of equations using back-substitution, which avoids direct matrix inversion.

Back-Substitution A simple example of how to use back substitution for matrix inverse is illustrated here. Assume R is 3×3 upper triangular matrix, then R^-1 is also a upper triangular matrix [10]. Write them in the matrix form:

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{bmatrix} \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ 0 & x_{22} & x_{23} \\ 0 & 0 & x_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The equation can be expanded into matrix vector form:

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{bmatrix} \begin{bmatrix} x_{11} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{bmatrix} \begin{bmatrix} x_{12} \\ x_{22} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{bmatrix} \begin{bmatrix} x_{13} \\ x_{23} \\ x_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Take last matrix-vector equation as example:

$$\begin{aligned} r_{11}x_{13} + r_{12}x_{23} + r_{13}x_{33} &= 0 \\ r_{22}x_{23} + r_{23}x_{33} &= 0 \\ r_{33}x_{33} &= 1 \end{aligned}$$

One could directly solve the value of x_{33}, and then back substitute into the second equation to solve the value of x_{23}, which can be used for solving x_{13}. This process is called back-substitution. For larger matrix, the basic principle is same. One could always find mature and callable algorithms to implement QR decomposition and compute the results. The QR decomposition also has some limitations. If the matrix is singular, the SVD generates more reliable results; if the matrix is very large, the conjugate gradient method is preferred [11].

4.2 Proximal Gradient Descent for Lasso Regression

Although Lasso can be reformulated as a quadratic program, but it's a quadratic program with 2^d constraints, because a d-dimensional cross-polytope has 2^d facets. In practice, special-purpose optimization methods have been developed for Lasso, including Coordinate Descent method, Projected Gradient Descent, and Least Angle Regression and Shrinkage (LARS) etc [12]. This section is attempting to introduce Proximal Gradient Descent on Lasso optimization.

Firstly, recall that Gradient Descent gives following steps [13]:

- Choose an initial point $x^{(0)} \in \mathbb{R}^n$
- Repeat:

$$x^{(k)} = x^{(k-1)} - t_k \cdot \nabla f\left(x^{(k-1)}\right), \quad \text{for } k = 1, 2, 3, \ldots$$

- Stop at some point

However, in case of Lasso, one cannot simply apply Gradient Descent method as the cost function is addition of one differentiable function and another non-differentiable function.

$$f(x) = g(x) + h(x)$$

where $g = RSS$ is convex and differentiable, and $h = \lambda \sum_{j=1}^{p} |w_j|$ is non-differentiable. Since h is not differentiable, one cannot directly take the gradient of f and apply the gradient descent update:

$$x^+ = x - t\nabla f(x)$$

Rather than that, it is motivated by the same ideas as gradient descent, specifically, minimization of a quadratic approximation to f in the vicinity of x. Instead of attempting to minimize the quadratic around all of f, which is impossible due to the non-differentiability of h, one can minimize the quadratic approximation to g and leave h alone. Make the necessary changes on the gradient descent [14]:

$$\begin{aligned} x^+ &= \mathrm{argmin}_z\, g(x) + \nabla g(x)^T(z-x) + \frac{1}{2t}\|z-x\|_2^2 + h(z) \\ &= \mathrm{argmin}_z\, \frac{1}{2t}\|z-(x-t\nabla g(x))\|_2^2 + h(z) \end{aligned}$$

The first equality comes from quadratic approximation of Taylor series:

$$f(y) \approx f(x) + \nabla f(x)^T(y-x) + \frac{1}{2t}\|y-x\|^2$$

The first term $\|z-(x-t\nabla g(x))\|_2^2$ makes x^+ to stay close to the gradient update for g, whereas, the second terms hold $h(z)$ minimization. This is the underlying principle behind proximal mapping.

Then let's move to define proximal mapping:

$$\mathrm{prox}_{h,t}(x) = \arg\min_z \|x-z\|_2^2 + h(z)$$

In proximal gradient descent we choose an initial $x^{(0)}$ and iteratively update

$$x^{(k)} = \mathrm{prox}_{h,t_k}\left(x^{(k-1)} - t_k\nabla g\left(x^{(k-1)}\right)\right)$$

It can be rewritten in the same form as a gradient step by defining

$$G_t(x) = \frac{x - \mathrm{prox}_{h,t}(x - t\nabla g(x))}{t}$$

then rewriting the update as:

$$x^{(k)} = x^{(k-1)} - t_k G_{t_k}\left(x^{(k-1)}\right)$$

Thus far, the Proximal Gradient Descent method's broad framework has been discussed. This technique is frequently used to fit linear models. The following paragraphs demonstrate how to apply it to the Lasso [14].

Apply on Lasso Given $y \in \mathbb{R}^n, X \in \mathbb{R}^{n\times p}$, the lasso criterion is given by

$$\begin{aligned} f(\beta) &= \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1 \\ &= g(\beta) + h(\beta) \end{aligned}$$

Then the proximal mapping for $h(\beta) = \lambda\|\beta\|_1$ is

$$\begin{aligned} \mathrm{prox}_{h,t}(\beta) &= \arg\min_z \frac{1}{2t}\|\beta - z\|_2^2 + \lambda\|z\|_1 \\ &= \begin{cases} \beta_i - \lambda & \beta_i > \lambda \\ 0 & -\lambda \le \beta_i \le \lambda \\ \beta_i + \lambda & \beta_i < -\lambda \end{cases} \\ &= S_{\lambda t}(\beta) \end{aligned}$$

where $S_t(\beta)$ is the soft thresholding operator coming from the sub-gradient optimality. The details of sub-gradient optimality is beyond the scope of this report. Readers are encouraged to explore this idea in Boyd's textbook [13].

The proximal update is then given by

$$\beta^+ = \text{prox}_{h,t}(\beta - t\nabla g(\beta)) = S_{\lambda t}\left(\beta + tX^T(y - X\beta)\right)$$

The second equality comes from the fact that $\nabla g(\beta) = -X^T(y - X\beta)$ and the definition of $S_{\lambda t}$. This is referred to as the iterative soft thresholding algorithm, and it exhibits the algorithm's rapid convergence. The advantage of the proximal mapping is that it provides a closed-form solution for a large number of significant functions h that are not differentiable but are simple. Making the proximal mapping is entirely dependent on g, and because g is smooth, its gradients can be computed even if it is quite intricate. The computational cost of mapping, on the other hand, is function-dependent and can be either expensive or inexpensive [14].

5 Application

The Lasso and Ridge regression techniques are applied to a real-world dataset in this section. As previously stated, these two techniques outperform OLS, particularly when training sets have a great variability. As a result, the dataset was collected by Turcotte et al [15]. and could be obtained via OpenML. The response variable is the triazines' activity. Triazines are a class of selective herbicide, which controls a wide spectrum of grass and broad-leaf weeds. The dataset contains 60 features and 186 entries of experimental results. Each feature is a protein domain, and different values represent different folding information [15]. The names of all sixty features are recorded in the Table 1. For better illustration of ill conditions and over-fitting, 106 observations serves as training set and the rest are testing set. The evaluation index adopted is Mean-square-error (MSE):

$$\textbf{MSE} = \frac{1}{n}\sum^{n}((y_i - f(\hat{x_i})))^2$$

where x_i is ith observation and y_i is the ith response value. When MSE has lower, the model has better predication accuracy.

p1_polar	p1_size	p1_flex	p1_h_doner	p1_h_acceptor	p1_pi_doner
p1_pi_acceptor	p1_polarisable	p1_sigma	p1_branch	p2_polar	p2_size
p2_flex	p2_h_doner	p2_h_acceptor	p2_pi_doner	p2_pi_acceptor	p2_polarisable
p2_sigma	p2_branch	p3_polar	p3_size	p3_flex	p3_h_doner
p3_h_acceptor	p3_pi_doner	p3_pi_acceptor	p3_polarisable	p3_sigma	p3_branch
p4_polar	p4_size	p4_flex	p4_h_doner	p4_h_acceptor	p4_pi_doner
p4_pi_acceptor	p4_polarisable	p4_sigma	p4_branch	p5_polar	p5_size
p5_flex	p5_h_doner	p5_h_acceptor	p5_pi_doner	p5_pi_acceptor	p5_polarisable
p5_sigma	p5_branch	p6_polar	p6_size	p6_flex	p6_h_doner
p6_h_acceptor	p6_pi_doner	p6_pi_acceptor	p6_polarisable	p6_sigma	p6_branch

Table 1: Features Name

The following Table 2 shows the test MSE value for each method, and optimal λ for Ridge and Lasso. The final optimization for Ridge and Lasso results in different lambda values, with Lasso having a low value and Ridge having a high value. In addition, Ridge had the lowest MSE, followed by Lasso, and OLS.

	Optimal λ	Test MSE
OLS	0	0.0399
Ridge	99.73	0.0200
Lasso	0.0070	0.0214

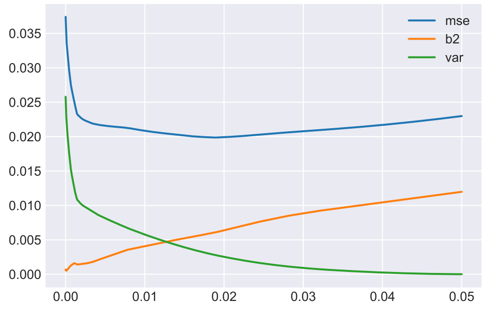
Table 2: Result of Different Methods

Additionally, because of the Lasso regression characteristic, one can observe that the coefficients of some features are zero, indicating that the features are irrelevant in Figure 5b. When λ is changed, the trend of regression coefficients are demonstrated. Notice when λ is equal to 0, the coefficients using OLS are identical. As λ increasing, regression weights gradually decrease to zero. When λ is extremely large, the lasso produces the null model, in which all features are zero.

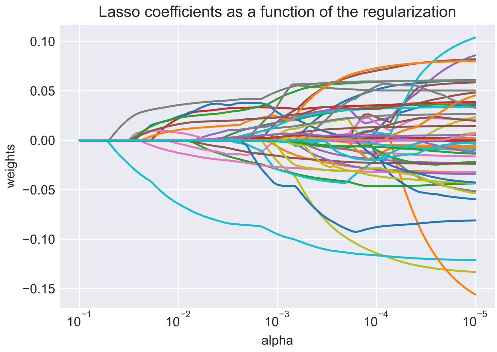
The advantage of ridge regression over least squares is due to the bias-variance trade-off as shown in Figure 5a [1]. This is the fundamental principle of Ridge and Lasso regression; they degrade the accuracy of the training model while increasing its robustness. As illustrated in the following picture, Bias increases steadily as Variance decreases, allowing MSE to achieve a minimum value after dropping. This demonstrates that the model performs as predicted in the previous math section.

0	0	0	0.0286	0	0
0	0.0376	0	-0.0727	0.0295	0
-0.0024	-1E-18	0	0	0.0048	0.0006
0	0	0	0	0	0
0	0	-0.0071	0	0	0
0	0	0.0248	0	0	-0.0015
-0.0158	0	0	0	0	0.0123
0	0	0	0	0	0
0	0	0	0	0	0
0	0.0186	0	0	0	0

Table 3: Lasso Coefficient



(a) Bias Variance Trade-off



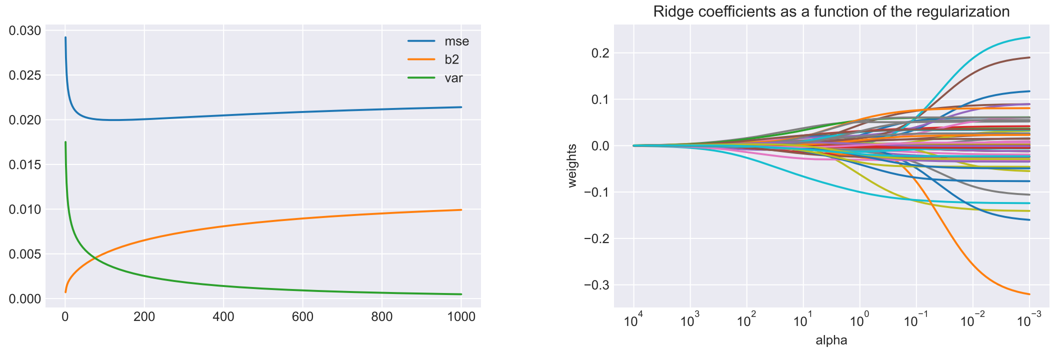
(b) Lasso Coefficient

Similarly, Ridge regression shows the trend of shrinking coefficients in Figure 6b. However, unlike with Lasso, the coefficients in Ridge Regression will not reach zero, but will remain near zero. Finally, when λ is extremely large, the coefficients of all features approach zero.

As λ increases, the flexibility of the ridge regression fit decreases, leading to decreased variance but increased bias [1]. The figure 6a looks very similar to the lasso one. The Variance is decreasing, and Squared Bias is increasing, this result in MSE first decrease and then increase. The difference between Ridge and Lasso is that Ridge gets minimum value at different λ with Lasso. Also, the curve of Ridge is smoother than Lasso.

0.0054	-0.0028	0.0048	0.0072	-0.0082	0.0006
-0.0030	0.0164	0.0083	-0.0262	0.0108	0.0059
-0.0048	-0.0048	0.0083	0.0067	0.0097	0.0054
0.0076	-0.0002	0.0028	7.58E-05	0.0023	0.0023
0.0023	0.0062	-0.0028	-0.0010	0.0013	-0.0022
0.0043	0.0046	0.0141	-5.1E-05	0.0033	-0.0039
-0.0137	0.0037	0.0015	-0.0023	0.0013	0.0062
Close 0	Close 0	0.0042	0.0010	0.0020	0.0042
-0.0002	0.0020	0.0011	0.0023	0.0005	0.0005
0.0047	0.0105	-0.0009	0.0020	0.0030	-0.0011

Table 4: Ridge Coefficient



(a) Bias Variance Trade-off

(b) Ridge Coefficient

When Lasso and Ridge regression are compared, each model offers distinct advantages. When the MSE of each model was compared in this experiment, the MSE of Ridge Regression was found to be lower. However, when the coefficients for the various features gathered throughout the experiment are examined, Lasso's coefficients are more concise. In other words, while processing certain data sets, Lasso can assist researchers in doing more accurate results attribution analyses.

A brief conclusion of application is drawn here. Lasso and Ridge have a lower MSE than OLS, implying greater adaptability and robustness. Additionally, we noted differences in the figures between the Lasso and Ridge regressions. One of the most striking properties of Lasso is its ability to zero out the coefficients of unimportant features. Thus, the Lasso and Ridge's properties are pretty similar to those of the earlier mathematical analysis.

6 Ending

The Lasso and Ridge regression techniques are primarily used to increase the predictive accuracy and interpretability of models. Ridge regression introduces a l_2 penalty, whereas Lasso introduces a l_1 penalty, which might result in coefficients exactly zero. The reader is expected to comprehend how these two models work conceptually and practically throughout the report. To end this report, it is worth noting that two regularization methods can also be used generally to other regression models, such as logistic regression [11].

Acknowledgments

Many Thanks to Dr. Yitzchak Elchanan Solomon's kind help and teaching.

References

[1] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

[2] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[3] RK Jain. Ridge regression and its application to medical data. *Computers and biomedical research*, 18(4):363–368, 1985.

[4] Hrishikesh D Vinod. Application of new ridge regression methods to a study of bell system scale economies. *Journal of the American Statistical Association*, 71(356):835–841, 1976.

[5] Malaquias Pena and Huug van den Dool. Consolidation of multimodel forecasts by ridge regression: Application to pacific sea surface temperature. *Journal of Climate*, 21(24):6521–6538, 2008.

[6] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[7] Yuri Fujino, Hiroshi Murata, Chihiro Mayama, and Ryo Asaoka. Applying "lasso" regression to predict future visual field progression in glaucoma patients. *Investigative ophthalmology & visual science*, 56(4):2334–2339, 2015.

[8] Tong Tong Wu, Yi Fang Chen, Trevor Hastie, Eric Sobel, and Kenneth Lange. Genome-wide association analysis by lasso penalized logistic regression. *Bioinformatics*, 25(6):714–721, 2009.

[9] Jack Kuang Tsung Chen, Richard L Valliant, and Michael R Elliott. Calibrating non-probability surveys to estimated control totals using lasso, with an application to political polling. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 68(3):657–681, 2019.

[10] David C Lay. Linear algebra and its applications 5th edition. *Pearson*, 2016.

[11] Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[12] Kevin P Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022.

[13] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[14] Ryan J Tibshirani. The lasso problem and uniqueness. *Electronic Journal of statistics*, 7:1456–1490, 2013.

[15] Marcel Turcotte, Stephen H Muggleton, and Michael JE Sternberg. Application of inductive logic programming to discover rules governing the three-dimensional topology of protein structure. In *International Conference on Inductive Logic Programming*, pages 53–64. Springer, 1998.