# A GENTLE INTRODUCTION TO LASSO AND RIDGE REGRESSION


**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 is introduction part

## 2 Goals

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

$$\mathbf{Y} = \mathbf{X}\boldsymbol{w} \tag{1}$$

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 including the intercept terms. The weights are computed via minimizing the residual sum-of-squares,

$$\min \ RSS = (y = \mathbf{X}w)^T (y - \mathbf{X}w) \tag{2}$$

which yields

$$\hat{\boldsymbol{w}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} \tag{3}$$

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 from the high degree of variability and overfitting. Even worse, if p is more than n, the model fails because basic least squares has no unique solution.

**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.

### 2.1 Ridge Regression

Hoerl and Kennard introduced the Ridge regression in 1970, the $l_2$ shrinkage penalty is added [2]. The weights are values that

$$\min \ RSS + \lambda \sum_{j=1}^{p} w_j^2 \tag{4}$$

where $\lambda$ is the tuning parameter.

Statistically speaking, the penalty is designed to solve the potential issues of naive solution in equation 3. Specifically, $\mathbf{X}'\mathbf{X}$ is not always a unit matrix. Even if $\mathbf{X}'\mathbf{X}$ is not of full rank, the penalty term makes the minimization problem non-singular.

In practice, Ridge regression can be used when the number of independent variables may be greater than the sample size, or there may be significant multicollinearity 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].

### 2.2 Lasso Regression

Tibshirani invented Lasso in 1995, and $l_2$ penalty is introduced [6]. Hence, the minimization problem is altered into

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

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 the 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

From the above discussion in the **Goal** section, it is known that by adding a tuning variable $\lambda$ to the original cost function of OLS, the variance of the regression models which are fit to different training dataset can be reduced. Thus, the cost function of the Ridge Regression is

$$\sum_{i=1}^{n} \left( y_i - w_0 - \sum_{j=1}^{p} w_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} w_j^2 = RSS + \lambda \sum_{j=1}^{p} w_j^2 \tag{6}$$

Instead of minimizing the residual sum of squares RSS in OLS, ridge regression adds a penalty term $\lambda \|w\|^2$. When $\lambda$ approaches 0, this penalty term will play no role, but when $\lambda$ approaches $\infty$, $\hat{w}$ will approach 0. This penalty term adds a bias-variance trade-off into the model. Thus, it can work well in situations where the OLS have high variance because as $\lambda$ increases, the decreasing of the flexibility of the ridge regression leads to the reduction in the variance of the predictions[1].

The cost function of the Lasso Regression is

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

The only difference between the cost function of the Lasso and Ridge is that the $\ell_2$ penalty ($w_j^2$) is replaced by the $\ell_1$ penalty (in $|w_j|$). The $\ell_1$ norm of the coefficient vector $||w_j||_1 = \sum |w_j|$.

Selecting a good $\lambda$ is also important, in which the strategy of cross-validation is always used and will be discussed further in the **Application** section.

This **Model** section will talk about the Bayesian interpretation of the cost functions of the Ridge and Lasso Regression, the comparison between the Ridge and Lasso, the variable selection property of the Lasso, and the naive solution of the Ridge and Lasso.

### 3.1 Bayesian interpretation of the cost functions

This section will analyze why the cost functions of the Ridge and Lasso regression are defined as the formula (3) and (4) through the lens of Bayesian statistics and MAP (maximum a posteriori probability).

Assume the coefficient vector of the linear model is given by $w = (w_0, w_1, \cdots, w_n)$, and assume the data vector is given by $X = (1, x_1, x_2, \cdots, x_n)$ and $Y$. Then, assume the regression model is given by $Y = w^T X + \epsilon$, in which $\epsilon$ is the error term for the regression estimation which follows the Gaussian distribution $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$. The distribution is equivalent to $Y|X, w \sim \mathcal{N}(w^T X, \sigma^2 I)$. The probability can be rewritten as

$$P(Y|X, w) = \frac{1}{\sqrt{2\pi}\sigma I} \exp\left(-\frac{(Y - w^T X)^T (Y - w^T X)}{2\sigma^2}\right) \tag{8}$$

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 $P(w|Y, X)$).

Recall the Bayesian Theorem that the posterior probability can expressed as

$$P(w|Y) = \frac{P(Y|w)P(w)}{P(Y)} \tag{9}$$

Thus, the maximum posterior probability of $w$ given the regression estimation $Y$ is given by

$$\begin{aligned}
\hat{w} &= \max_w P(w|Y) = \max_w \left(P(Y|w)P(w)\right) \\
&= \max_w \left(\log(P(Y|w)P(w))\right) \quad \text{(log likelihood)} \\
&= \max_w \left(\log\left(\frac{1}{\sqrt{2\pi}\sigma I} \exp\left(-\frac{(Y - w^T X)^T (Y - w^T X)}{2\sigma^2}\right) P(w)\right)\right) \quad (*)
\end{aligned}$$

Here, if $w$ is expected to follow a Gaussian distribution of mean zero and standard deviation $\sigma_0 I$, in other words, $P(w)$ satisfies

$$P(w) = \frac{1}{\sqrt{2\pi}\sigma_0 I} \exp\left(-\frac{w^T w}{2\sigma_0^2 I}\right) \tag{10}$$

Then, if the equation (7) is substituted back into the equation $(*)$ to replace $P(w)$, we will get

$$\begin{aligned}
(*) = \hat{w} &= \max_w \left(\log\left(\frac{1}{\sqrt{2\pi}\sigma I} \exp\left(-\frac{(Y - w^T X)^T (Y - w^T X)}{2\sigma^2}\right) P(w)\right)\right) \\
&= \max_w \left(\log\left(\frac{1}{\sqrt{2\pi}\sigma I} \exp\left(-\frac{(Y - w^T X)^T (Y - w^T X)}{2\sigma^2}\right) \frac{1}{\sqrt{2\pi}\sigma_0 I} \exp\left(-\frac{w^T w}{2\sigma_0^2}\right)\right)\right) \\
&= \min_w \left(\exp\left(\frac{(Y - w^T X)^T (Y - w^T X)}{2\sigma^2} + \frac{w^T w}{2\sigma_0^2}\right)\right) \\
&= \min_w \left((Y - w^T X)^T (Y - w^T X) + \frac{\sigma^2}{\sigma_0^2} w^T w\right) \quad (**)
\end{aligned}$$

Assume $\lambda = \sigma^2/\sigma_0^2$, then $(**)$ is exactly the same as the cost function of the Ridge Regression. 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 $\lambda$ [1], then it follows that the MAP for $w$ is the cost function of the Lasso Regression.

### 3.2 The variable selection property of the Lasso

As is mentioned in the **Goal** section, the Lasso Regression, unlike the Ridge Regression, can result in coefficient estimates that are exactly equal to zero, which makes the regression model derived by the Lasso Regression simpler and easier to interpret. 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

$$\min_w \left(\sum_{i=1}^{n} \left(y_i - w_0 - \sum_{j=1}^{p} w_j x_{ij}\right)^2 + \lambda \sum_{j=1}^{p} |w_j|\right) \quad \text{subject to} \quad \sum_{j=1}^{p} |w_j| \leq s \tag{11}$$

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

$$\min_{w} \left( \sum_{i=1}^{n} \left( y_i - w_0 - \sum_{j=1}^{p} w_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} w_j^2 \right) \quad \text{subject to} \quad \sum_{j=1}^{p} w_j^2 \leq s \tag{12}$$

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 In the plot of $p = 2$, it can be seen that the optimized coefficient estimates $\hat{w}$ in

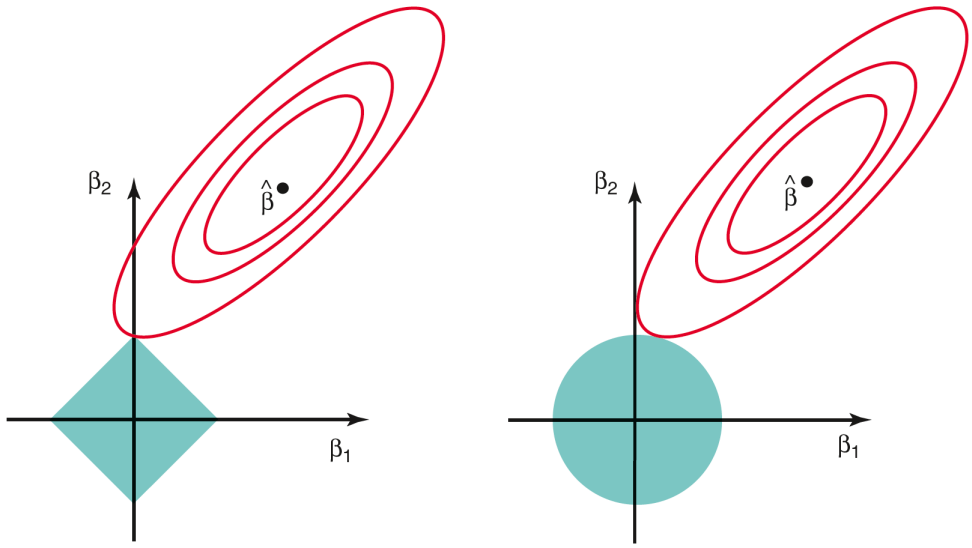


Figure 1: The contour lines of OLS and the constraint of the Lasso (left) and the Ridge (right) [1]

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].

### 3.3 The direct solution for the Ridge and Lasso cost functions

If the cost function of the Ridge Regression is rewritten into the matrix form, we have

$$\mathcal{L}(w) = (y - Xw)^T (y - Xw) + \lambda w^T w \tag{13}$$

Thus, the direct solution is given by

$$\hat{w}^{ridge} = (X^T X + \lambda I)^{-1} X^T y \tag{14}$$

The gradient descendent method can also be used to solve an optimal solution for the Ridge Regression. The basic idea is to find the partial derivative of $\mathcal{L}(w)$ with respect to each $w_k$, i.e.

$$\frac{\partial \mathcal{L}(w)}{\partial w_k} = \frac{\partial}{\partial w_k} \left( \sum_{i=1}^{n} \left( y_i - w_0 - \sum_{j=1}^{p} w_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} w_j^2 \right) \tag{15}$$

For the Lasso Regression, whose cost function is given by

$$\sum_{i=1}^{n} \left( y_i - w_0 - \sum_{j=1}^{p} w_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |w_j| = RSS + \lambda \sum_{j=1}^{p} |w_j| \tag{16}$$

As the cost function contains the absolute value, it is not differentiable at the turning point of the absolute term when it equals 0, making ordinary gradient descendent methods hard to implement. However, there are some methods which can solve the Lasso Regression.

Detailed information of how to solve the Ridge and Lasso Regression will be discussed in the next section (section **Training**).

# 4 Training

This section focuses on optimization scheme of computing weights in Ridge and Lasso Regression.

## 4.1 Ridge Regression

Although the primal estimation of $w$ can be computed via matrix inversion, it is slow and numerically unstable. Hence, several optimization algorithm are adopted including QR decomposition, singular value decomposition (SVD), Conjugate Gradient method, and Average stochastic gradient descent method, etc. Here, the first three methods are briefly discussed.

### 4.1.1 Decomposition

**QR Decomposition** Since $p(\boldsymbol{w}) = \mathcal{N}\left(0, \Lambda^{-1}\right)$, where $\Lambda$ is the precision matrix. The problem $\mathbf{X}, y$ for training data can be modified as

$$\tilde{\mathrm{X}} = \begin{pmatrix} \mathrm{X}/\sigma \\ \sqrt{\Lambda} \end{pmatrix}, \quad \tilde{y} = \begin{pmatrix} y/\sigma \\ 0_{p\times 1} \end{pmatrix} \tag{17}$$

where $\sigma$ is 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.

$$\begin{aligned} f(w) &= (\tilde{y} - \tilde{\mathrm{X}}w)^T(\tilde{y} - \tilde{\mathrm{X}}w) \\ &= \left(\begin{pmatrix} y/\sigma \\ 0 \end{pmatrix} - \begin{pmatrix} \mathrm{X}/\sigma \\ \sqrt{\Lambda} \end{pmatrix} w\right)^T \left(\begin{pmatrix} y/\sigma \\ 0 \end{pmatrix} - \begin{pmatrix} \mathrm{X}/\sigma \\ \sqrt{\Lambda} \end{pmatrix} w\right) \\ &= \frac{1}{\sigma^2}(y - \mathrm{X}w)^T(y - \mathrm{X}w) + w^T\Lambda w \end{aligned} \tag{18}$$

The RSS on this expanded data is equivalent to penalized RSS on the original data. Hence the MAP estimate is given by

$$\hat{w}_{\text{map}} = \left(\tilde{\mathrm{X}}^T\tilde{\mathrm{X}}\right)^{-1}\tilde{\mathrm{X}}^T\tilde{y} \tag{19}$$

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, we can rewrite this system of equations as follows:

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

Since $\mathbf{R}$ is upper triangular, one can solve this last set of equations using back-substitution, thus avoiding matrix inversion. The whole process takes $O\left((n+p)p^2\right)$ time.

**Singular Value Decomposition** SVD is more stable and faster than QR method If $p > n$, which is the usual case when using ridge regression. To see how this works, let $\mathrm{X} = \mathrm{USV}^T$ be the SVD of X, where $\mathrm{V}^T\mathrm{V} = \mathrm{I}_N, \mathrm{UU}^T = \mathrm{U}^T\mathbf{U} = \mathrm{I}_N$, and S is a diagonal $n \times n$ matrix. Now let $\mathrm{R} = \mathrm{US}$ be an $N \times N$ matrix. One can show that

$$\hat{w}_{\text{map}} = \mathbf{V}\left(\mathbf{R}^T\mathbf{R} + \lambda\mathbf{I}_N\right)^{-1}\mathbf{R}^T y \tag{21}$$

The $D$-dimensional vectors $x_i$ is replaced with the $N$-dimensional vectors $r_i$ and perform our penalized fit as before. The overall time is now $O\left(pn^2\right)$ operations, which is less than $O\left(p^3\right)$ if $p > n$.

#### 4.1.2 Conjugate Gradient Method

Beyond the decomposition methods, one could also apply iterative methods such as Conjugate Gradient Method (CG), which assumes $\mathbf{X}$ is symmetric positive definite and well suited to problems where $\mathbf{X}$ is sparse and large. The Conjugate Gradient Method (CG) solves linear system

$$Ax = b \tag{22}$$

via searching through the direction by the conjugate of residuals $r_i$. The CG is very efficient with quick convergence as the residuals works for Steepest Descent, while orthogonal to the previous search directions. The algorithm framework is demonstrated below [10].

**Algorithm 1** Conjugate Gradient

**procedure** CONJUGATEGRADIENT($A, b, x_0, \epsilon$)
  $r_0 \leftarrow Ax_0 - b$ ▷ The initial residual
  $s_0 \leftarrow -r_0$ ▷ The first search direction
  $i \leftarrow 0$
  **while** $||r_i||_2 > \epsilon$ **do**
    $z_i \leftarrow As_i$
    $\alpha_i \leftarrow \frac{r_i^T r_i}{s_i^T z_i}$ ▷ optimal line search along the $s_i$
    $x_{i+1} \leftarrow x_i + \alpha_i s_i$
    $r_{i+1} \leftarrow r_i + \alpha_i z_i$
    $s_{i+1} \leftarrow -r_{i+1} + \frac{r_{i+1}^T r_{i+1}}{r_i^T r_i} s_i$ ▷ Gram-Schmidt process applied for the new residual
    $i \leftarrow -i + 1$
  **end while**
  **return** $x_i$
**end procedure**

Particularly, for the ridge regression, recall equation 19, one could take $\tilde{\mathbf{X}}$ as $A$, $w$ as $x$, and $\tilde{y}$ as $b$.

### 4.2 Lasso Regression

#### 4.2.1 Coordinate Descent

The objective function of Lasso

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

yields the solution

$$\hat{w}_d(a_d) = \begin{cases} (a_d + \lambda)/n_d & \text{if } a_d < -\lambda \\ 0 & \text{if } a_d \in [-\lambda, \lambda] \\ (a_d - \lambda)/n_d & \text{if } a_d > \lambda \end{cases}$$

where

$$n_d = \sum_{n=1}^{N} x_{nd}^2$$
$$a_d = \sum_{n=1}^{N} x_{nd} \left( y_n - w_{-d}^T x_{n,-d} \right)$$

The $w_{-d}$ and $x_{n,-d}$ are the vectors without dth component.

Such result is recognized as Soft-Threshold

$$\hat{w}_d = \text{ SoftThreshold } \left( \frac{a_d}{n_d}, \lambda / n_d \right)$$

Since it is computationally expensive and slower to optimize all variables than one by one. One can solve for the $j$th coefficient with others fixed:

$$w_j^* = \underset{\eta}{\operatorname{argmin}} \, \mathcal{L}(w + \eta e_j)$$

where $e_j$ is the $j$th unit vector. This is called coordinate descent. The order of j can be deterministic or random or steepest gradient. The algorithm details are demonstrated below [11].

**Algorithm 2** Coordinate Descent for Lasso

Initialize $\hat{w}_{\text{map}} = \mathbf{V}\left(\mathbf{R}^T\mathbf{R} + \lambda\mathbf{I}_N\right)^{-1}\mathbf{R}^T y$
**repeat**
  **for** $d = 1, \ldots, D$ **do**
    $n_d = \sum_{n=1}^{N} x_{nd}^2$
    $a_d = \sum_{n=1}^{N} x_{nd}\left(y_n - w_{-d}^T x_{n,-d}\right)$
    $\hat{w}_d =$ SoftThreshold $\left(\frac{a_d}{n_d}, \lambda/n_d\right)$
  **end for**
**until** converged

### 4.2.2 Least Angle Regression and Shrinkage

Least Angle Regression and Shrinkage (LARS) is a relatively new algorithm, which can efficiently compute the lasso path. In another words, it can generate a set of solutions for different $\lambda$. The underlying fact is that $w(\hat{\lambda}_k)$ can be easily computed by $w(\hat{\lambda_{k-1}})$ if $\lambda k \approx \lambda_{k-1}$.

LARS begins with a large value of $\lambda$ , ensuring that only the variable with the highest correlation to the response vector $y$ is picked. Then, $\lambda$ is decreased until a second variable has the same magnitude correlation with the current residual as the first variable, with the residual at step $k$ defined as $r_k = y - X^{F_k} w_k$, where $X^{F_k}$ is the current active set. Impressively, this new value of $\lambda$ may be found analytically using a geometric argument called the least angle. As a result, the algorithm proceeds to the next step rapidly.

The algorithm is shown below [12].

**Algorithm 3** LARS

1: Standardize the predictors to have mean zero and unit norm. Start with the residual $\mathbf{r} = \mathbf{y} - \hat{\mathbf{y}}, w_1, w_2, \ldots, w_p = 0$.
2: Find the predictor $\mathbf{x}_j$ most correlated with $\mathbf{r}$.
3: Move $w_j$ from 0 towards its least-squares coefficient $\langle \mathrm{x}_j, \mathrm{r} \rangle$, until some other competitor $\mathrm{x}_k$ has as much correlation with the current residual as does $\mathbf{x}_j$.
4: Move $w_j$ and $w_k$ in the direction defined by their joint least squares coefficient of the current residual on $(\mathrm{x}_j, \mathrm{x}_k)$, until some other competitor $\mathrm{x}_l$ has as much correlation with the current residual. If a non-zero coefficient hits zero, drop its variable from the active set of variables and recompute the current joint least squares direction.
5: Continue in this way until all $p$ predictors have been entered. After $\min(N-1, p)$ steps, we arrive at the full least-squares solution.

## 5 Application

| | $\lambda$ | Test MSE |
|---|---|---|
| OLS | 0 | 0.0399 |
| Ridge | 99.73 | 0.4586 |
| Lasso | 0.0070 | 0.4578 |

Table 1: Result of Different Methods

| | | | | | |
|---|---|---|---|---|---|
| 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 2: Features Name

| | | | | | |
|---|---|---|---|---|---|
| 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

| | | | | | |
|---|---|---|---|---|---|
| 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

## 6 Conclusion

Your conclusion here

## Acknowledgments

This was was supported in part by.....

## 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] Simon Bartels and Philipp Hennig. Conjugate gradients for kernel machines. *CoRR*, abs/1911.06048, 2019.

[11] Tong Tong Wu and Kenneth Lange. Coordinate descent algorithms for lasso penalized regression. *The Annals of Applied Statistics*, 2(1):224–244, 2008.

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