

Machine Learning in Practise



INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI



Resampling: Bootstrapping



- In practice (unlike in theory), we have only ONE training set S .
- We can simulate multiple training sets by bootstrap replicates
 - $S' = \{x \mid x \text{ is drawn at random with replacement from } S\}$
and $|S'| = |S|$

| Original | Bootstrap1 | Bootstrap2 | Bootstrap3 | Bootstrap4 |
|----------|------------|------------|------------|------------|
| 1 | 1 | 2 | 1 | 1 |
| 2 | 1 | 3 | 2 | 1 |
| 3 | 3 | 3 | 3 | 1 |
| 4 | 3 | 3 | 5 | 4 |
| 5 | 5 | 4 | 5 | 5 |

Bootstrapping Sample: Example



- Numerical: $X = \{10, 27, 31, 40, 46\}$
- Calculate bootstrap samples
- Are these valid bootstrap samples?
 - $X_1 = \{10, 10, 31, 31, 46\}$?
 - $X_2 = \{10, 27, 31, 10\}$?
 - $X_3 = \{31, 31, 31, 31\}$?
 - $X_4 = \{31, 31, 31, 31, 31\}$?

Procedure: Measuring Bias and Variance



- Construct B bootstrap replicates of S (e.g., Construct B bootstrap replicates of S (e.g., $b = 200$): S_1, \dots, S_B
- Apply learning algorithm to each replicate S_b to obtain hypothesis h_b
- Let $T = S \setminus S_b$ be the data points that do not appear in any $S_1 \dots S_B$ (out of bag points).
- Compute predicted value $h_b(x)$ for each x in T
- Alternately, set T could also be a hold out set created from S before constructing bootstrap samples
- For each data point x in T , we will now have the observed corresponding value y and several predictions y_1, \dots, y_B

Procedure: Measuring Bias and Variance



- Compute the average prediction \underline{h} .
 - $\underline{h} = \sum_b h_b(x)/B$
- Estimate bias as $(\underline{h} - y)$
- Estimate variance as
 - $\sum_b (y_b - \underline{h})^2/(B - 1)$
- Assume noise is 0
- Assumptions:
 - Bootstrap replicates are not real data
 - We ignore the noise

Practice Question



Given data points: $\{76, 60, 82, 12, 38, 73, 82, 17\}$, construct **8** bootstrap samples. Report the standard error between the true mean and the average of means of bootstrapped samples.

Linear Regression Revisited



- Objective Function: An optimization problem

$$\underset{\theta}{\text{minimize}} \quad J(\theta)$$

- Minimize sum of costs over all input/output pairs

$$J(\Theta) = \frac{1}{M} \sum_{i=1}^M (\Theta^T \Phi(x_i) - y_i)^2 \quad \frac{1}{M} \sum_{i=1}^M \left[\sum_{j=0}^p \theta_j x_j^{(i)} - y^{(i)} \right]$$

- Gradient Descent

$$\theta_j = \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j}, \text{ for all } j$$

- To overcome underfitting:
 - Add new parameters to our model
 - Increase the model complexity
- To overcome overfitting:
 - Reduce the model complexity
 - Regularization/shrinkage
 - Change the error function to penalize hypothesis complexity

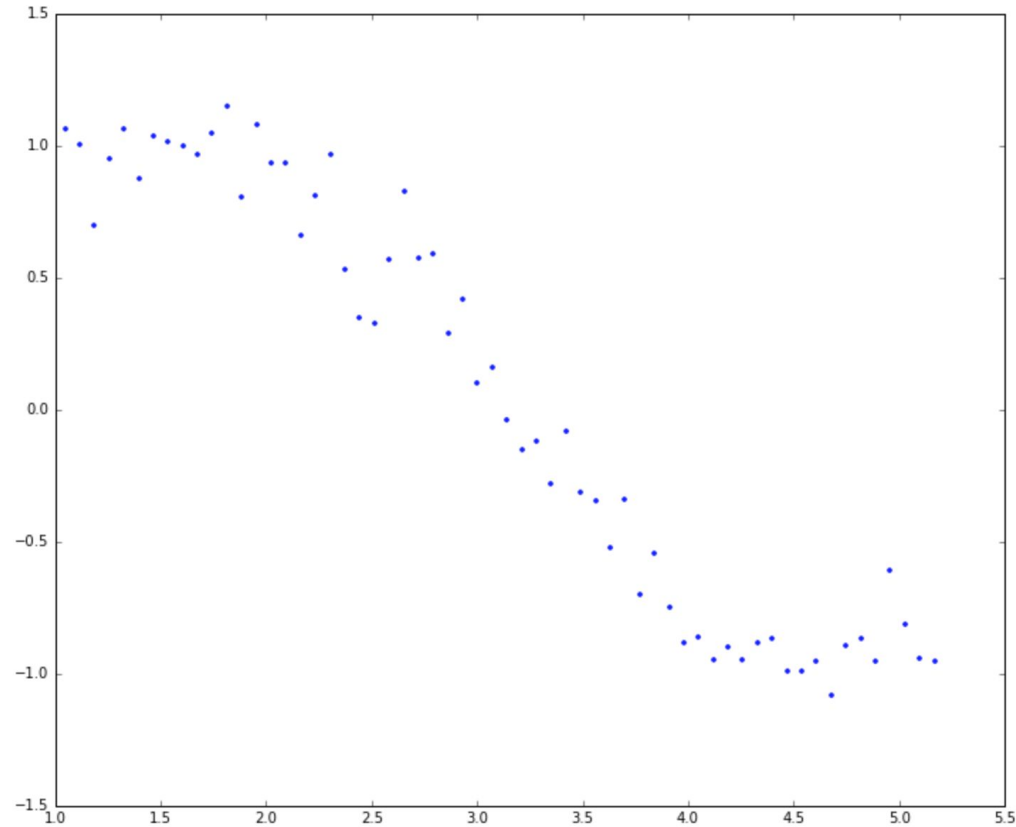
$$J(\Theta) = J_w(\Theta) + \lambda J_{pen}(\Theta)$$

Regularization

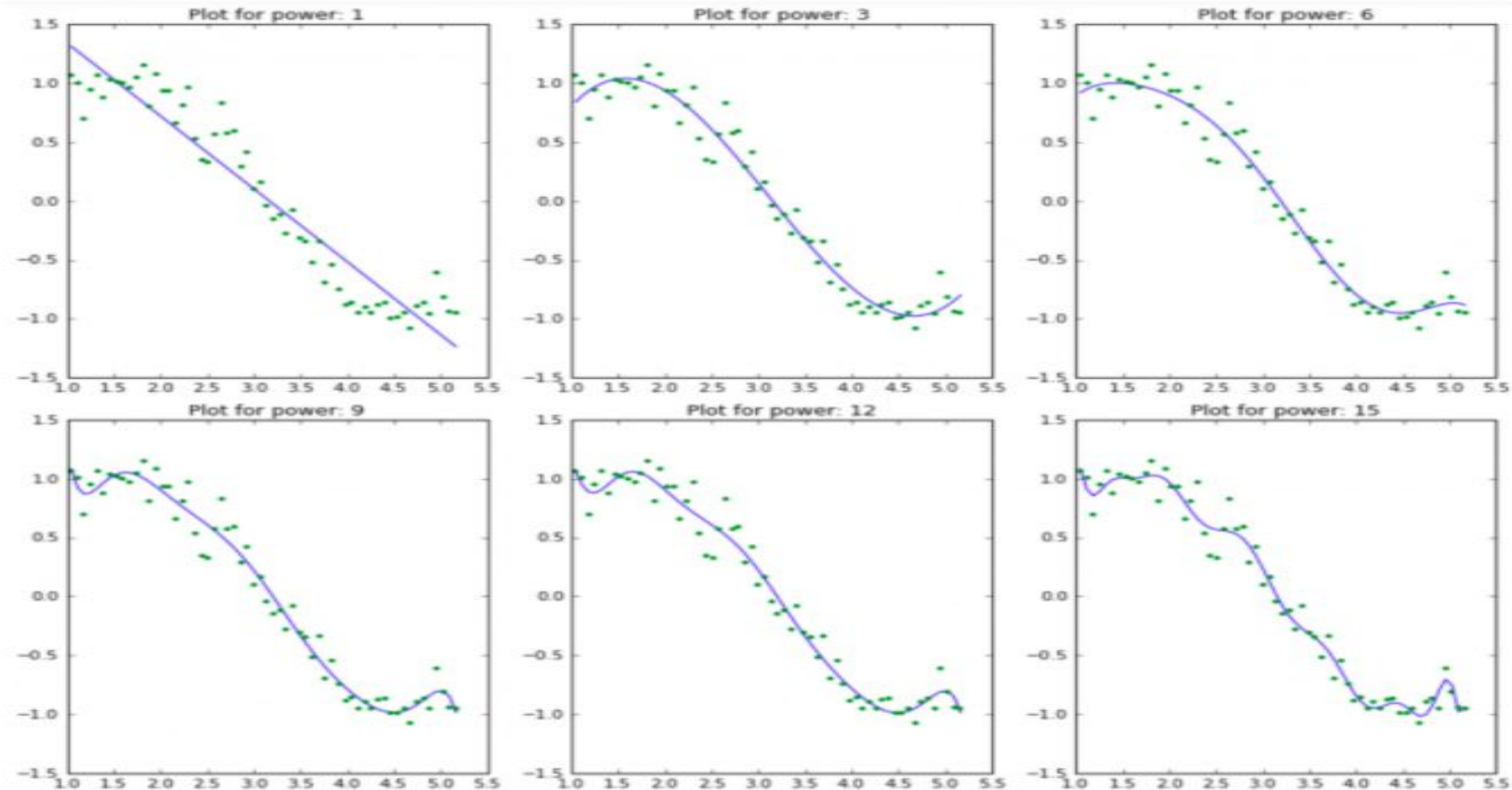


- Regularization constraints or regularizes the coefficient estimates
 - shrinks the coefficient estimates towards zero
- Shrinking the coefficient estimates can significantly reduce their variance.

SINE Curve: Noisy Data



Regression – Without any penalisation



Coefficients increase exponentially



| | rss | intercept | coef_x_1 | coef_x_2 | coef_x_3 | coef_x_4 | coef_x_5 | coef_x_6 | coef_x_7 | coef_x_8 | coef_x_9 | coef_x_10 | coef_x_11 | c |
|--------------|------|-----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|-----------|----|
| model_pow_1 | 3.3 | 2 | -0.62 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | N |
| model_pow_2 | 3.3 | 1.9 | -0.58 | -0.006 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | N |
| model_pow_3 | 1.1 | -1.1 | 3 | -1.3 | 0.14 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | N |
| model_pow_4 | 1.1 | -0.27 | 1.7 | -0.53 | -0.036 | 0.014 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | N |
| model_pow_5 | 1 | 3 | -5.1 | 4.7 | -1.9 | 0.33 | -0.021 | NaN | NaN | NaN | NaN | NaN | NaN | N |
| model_pow_6 | 0.99 | -2.8 | 9.5 | -9.7 | 5.2 | -1.6 | 0.23 | -0.014 | NaN | NaN | NaN | NaN | NaN | N |
| model_pow_7 | 0.93 | 19 | -56 | 69 | -45 | 17 | -3.5 | 0.4 | -0.019 | NaN | NaN | NaN | NaN | N |
| model_pow_8 | 0.92 | 43 | -1.4e+02 | 1.8e+02 | -1.3e+02 | 58 | -15 | 2.4 | -0.21 | 0.0077 | NaN | NaN | NaN | N |
| model_pow_9 | 0.87 | 1.7e+02 | -6.1e+02 | 9.6e+02 | -8.5e+02 | 4.6e+02 | -1.6e+02 | 37 | -5.2 | 0.42 | -0.015 | NaN | NaN | N |
| model_pow_10 | 0.87 | 1.4e+02 | -4.9e+02 | 7.3e+02 | -6e+02 | 2.9e+02 | -87 | 15 | -0.81 | -0.14 | 0.026 | -0.0013 | NaN | N |
| model_pow_11 | 0.87 | -75 | 5.1e+02 | -1.3e+03 | 1.9e+03 | -1.6e+03 | 9.1e+02 | -3.5e+02 | 91 | -16 | 1.8 | -0.12 | 0.0034 | N |
| model_pow_12 | 0.87 | -3.4e+02 | 1.9e+03 | -4.4e+03 | 6e+03 | -5.2e+03 | 3.1e+03 | -1.3e+03 | 3.8e+02 | -80 | 12 | -1.1 | 0.062 | -i |
| model_pow_13 | 0.86 | 3.2e+03 | -1.8e+04 | 4.5e+04 | -6.7e+04 | 6.6e+04 | -4.6e+04 | 2.3e+04 | -8.5e+03 | 2.3e+03 | -4.5e+02 | 62 | -5.7 | 0 |
| model_pow_14 | 0.79 | 2.4e+04 | -1.4e+05 | 3.8e+05 | -6.1e+05 | 6.6e+05 | -5e+05 | 2.8e+05 | -1.2e+05 | 3.7e+04 | -8.5e+03 | 1.5e+03 | -1.8e+02 | 1 |
| model_pow_15 | 0.7 | -3.6e+04 | 2.4e+05 | -7.5e+05 | 1.4e+06 | -1.7e+06 | 1.5e+06 | -1e+06 | 5e+05 | -1.9e+05 | 5.4e+04 | -1.2e+04 | 1.9e+03 | -; |

Ridge Regularization



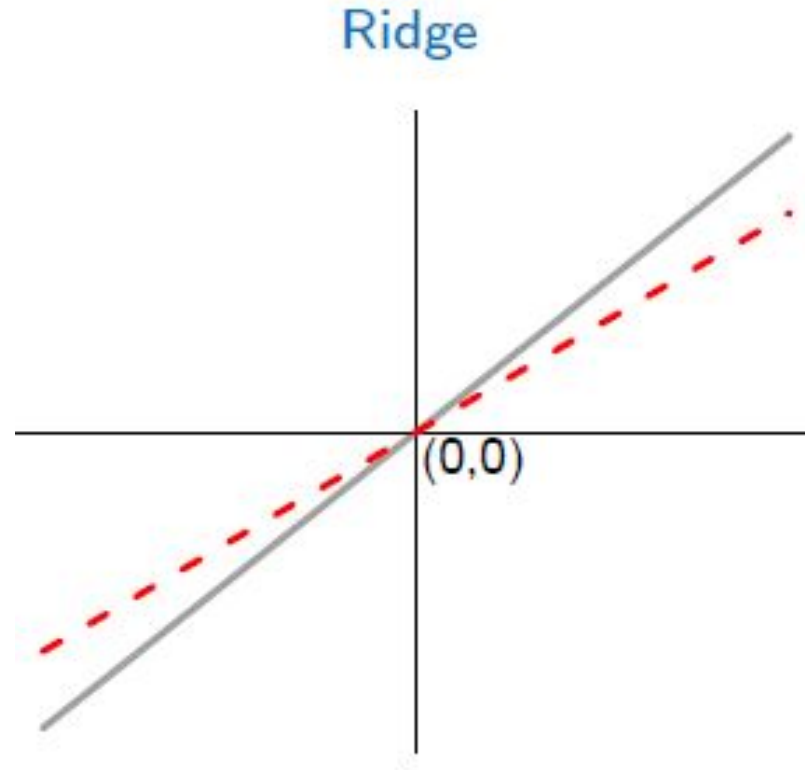
- Minimization objective = LS Obj + λ * (sum of square of slope)

$$J(\Theta) = \frac{1}{M} \sum_{i=1}^M (\Theta^T \Phi(x_i) - y_i)^2 + \lambda \sum_{j=1}^p \Theta_j^2$$

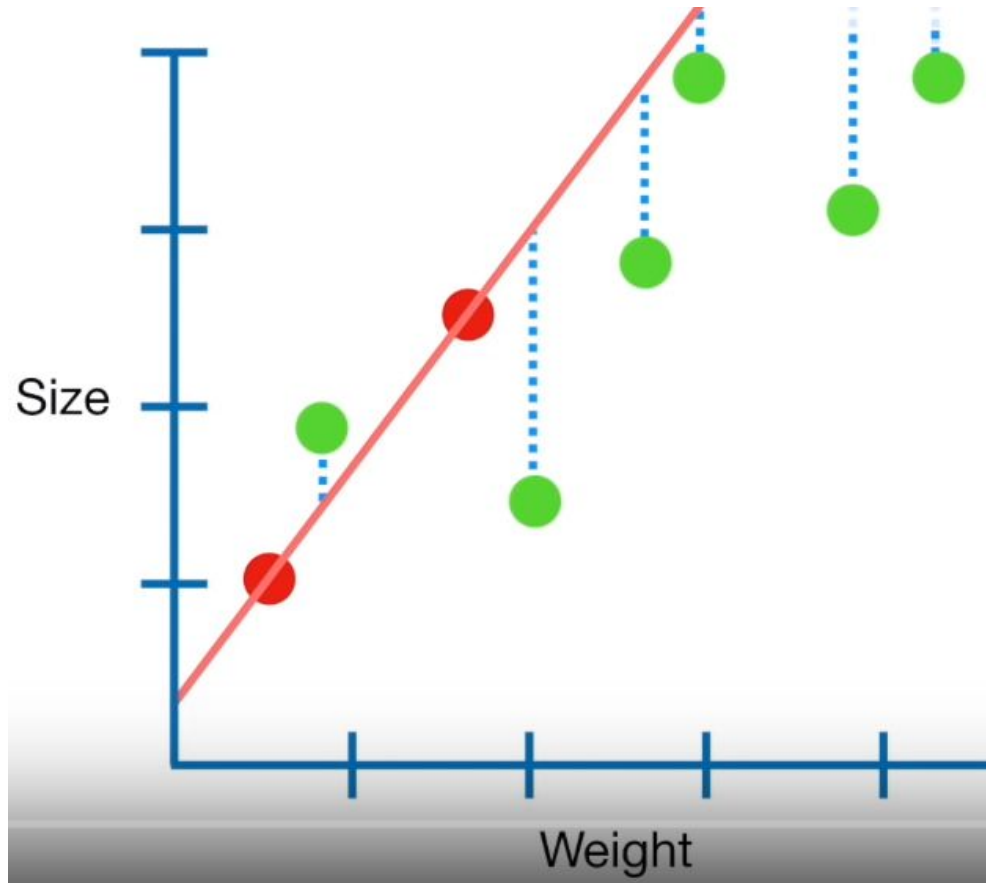
$$\frac{\partial J}{\partial \Theta} = \frac{2}{M} \sum_{i=1}^M (\Theta^T \Phi(x_i) - y_i) \Phi(x_i) + 2\lambda \Theta_j$$

$$\Theta_{(j+1)} = (1 - 2\lambda\alpha) \Theta_j - \frac{2\alpha}{M} \sum_{i=1}^M (\Theta^T \Phi(x_i) - y_i) \Phi(x_i)$$

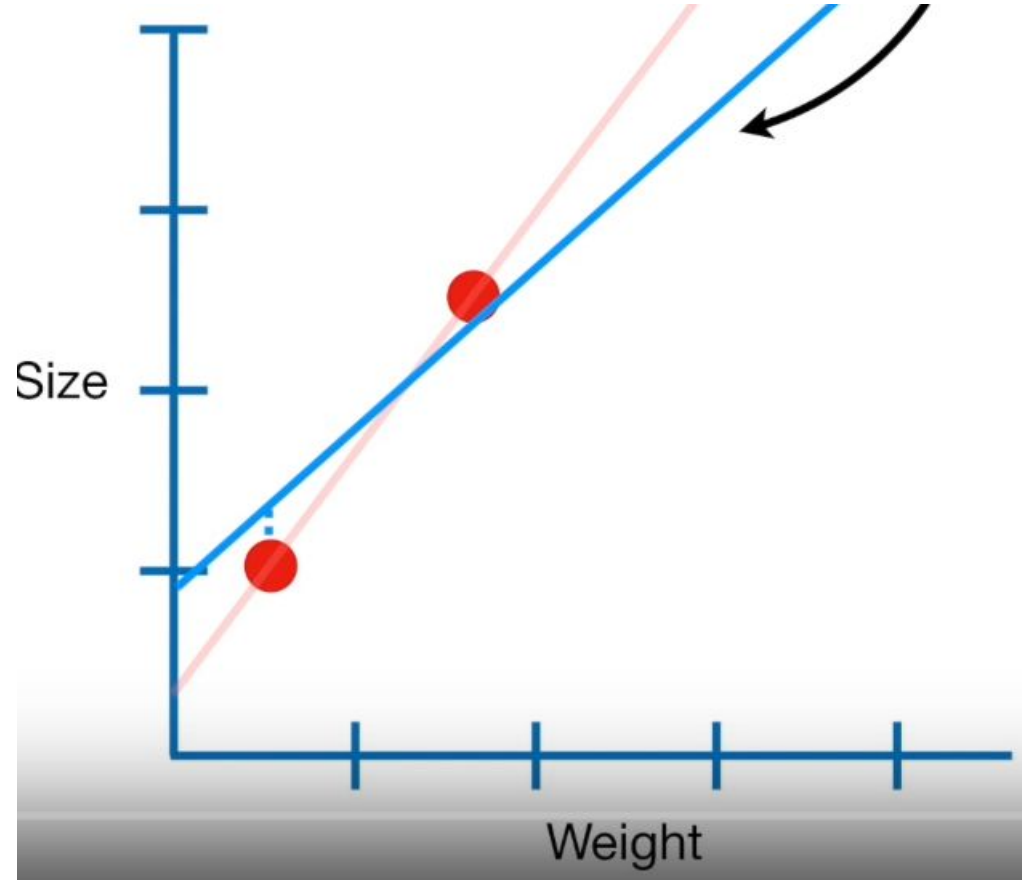
Ridge Regularization



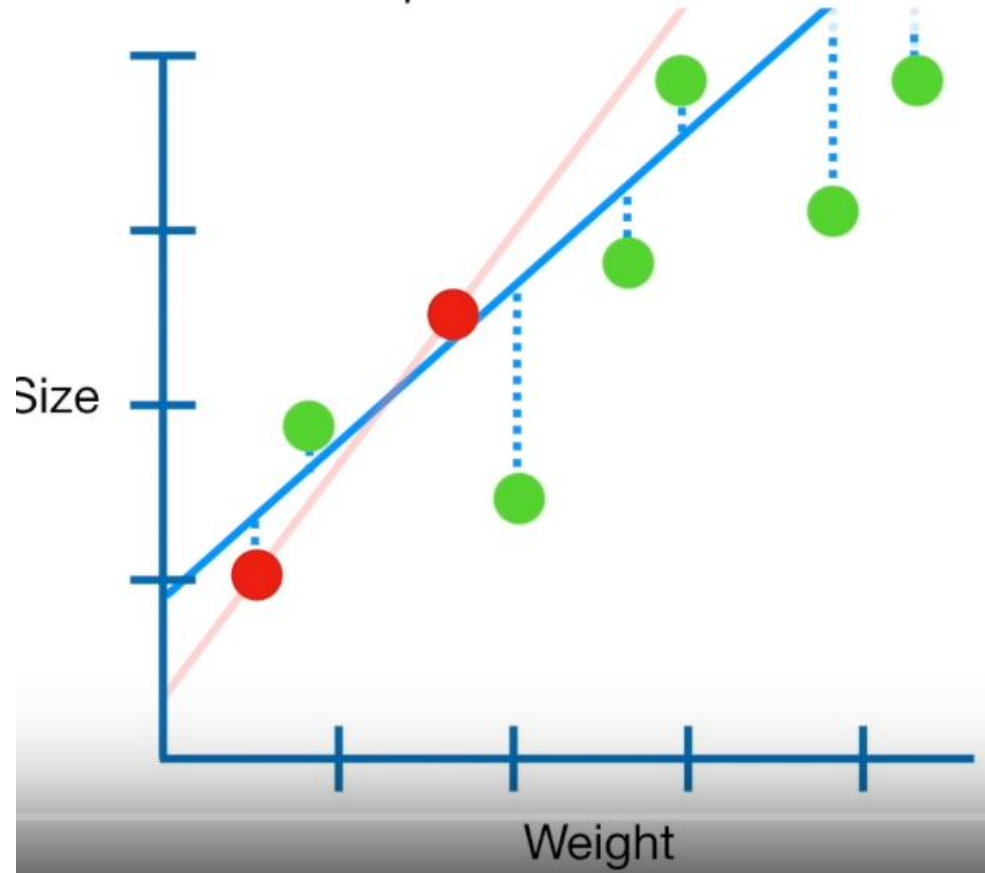
Low Bias and High Variance: Overfit



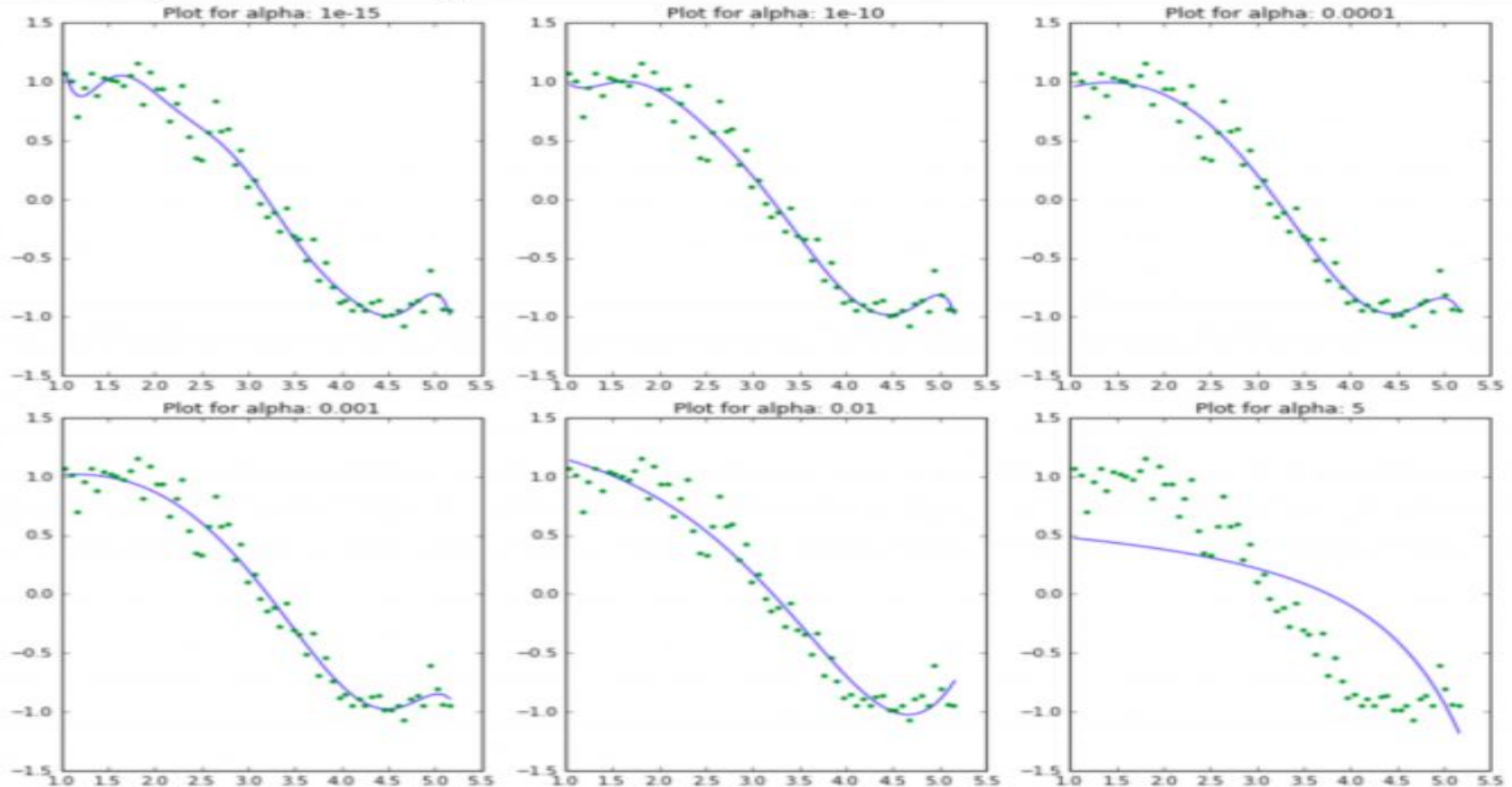
Ridge Regression



Ridge Regression



Regression - Ridge Regression



Regression - Ridge Regression



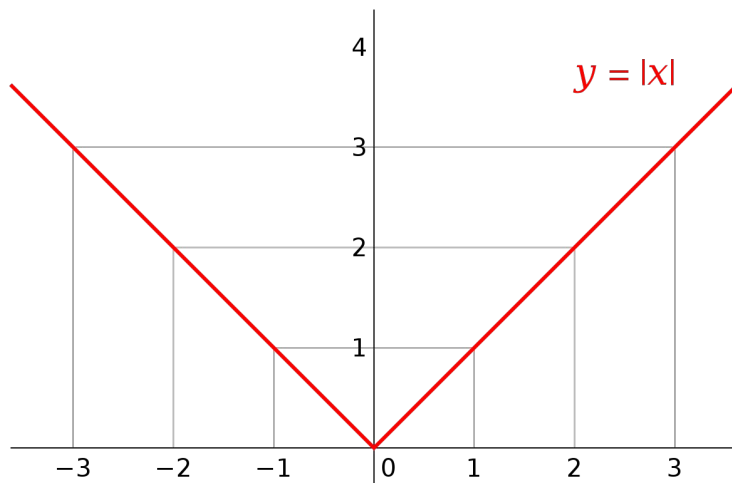
| | rss | intercept | coef_x_1 | coef_x_2 | coef_x_3 | coef_x_4 | coef_x_5 | coef_x_6 | coef_x_7 | coef_x_8 | coef_x_9 | coef_x_10 | coef_x_11 | coef_x_12 |
|--------------|------|-----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|-----------|-----------|
| alpha_1e-15 | 0.87 | 95 | -3e+02 | 3.8e+02 | -2.4e+02 | 66 | 0.96 | -4.8 | 0.64 | 0.15 | -0.026 | -0.0054 | 0.00086 | 0.0 |
| alpha_1e-10 | 0.92 | 11 | -29 | 31 | -15 | 2.9 | 0.17 | -0.091 | -0.011 | 0.002 | 0.00064 | 2.4e-05 | -2e-05 | -4.2e-05 |
| alpha_1e-08 | 0.95 | 1.3 | -1.5 | 1.7 | -0.68 | 0.039 | 0.016 | 0.00016 | -0.00036 | -5.4e-05 | -2.9e-07 | 1.1e-06 | 1.9e-07 | 2e-07 |
| alpha_0.0001 | 0.96 | 0.56 | 0.55 | -0.13 | -0.026 | -0.0028 | -0.00011 | 4.1e-05 | 1.5e-05 | 3.7e-06 | 7.4e-07 | 1.3e-07 | 1.9e-08 | 1.9e-08 |
| alpha_0.001 | 1 | 0.82 | 0.31 | -0.087 | -0.02 | -0.0028 | -0.00022 | 1.8e-05 | 1.2e-05 | 3.4e-06 | 7.3e-07 | 1.3e-07 | 1.9e-08 | 1.7e-08 |
| alpha_0.01 | 1.4 | 1.3 | -0.088 | -0.052 | -0.01 | -0.0014 | -0.00013 | 7.2e-07 | 4.1e-06 | 1.3e-06 | 3e-07 | 5.6e-08 | 9e-09 | 1.1e-09 |
| alpha_1 | 5.6 | 0.97 | -0.14 | -0.019 | -0.003 | -0.00047 | -7e-05 | -9.9e-06 | -1.3e-06 | -1.4e-07 | -9.3e-09 | 1.3e-09 | 7.8e-10 | 2.4e-10 |
| alpha_5 | 14 | 0.55 | -0.059 | -0.0085 | -0.0014 | -0.00024 | -4.1e-05 | -6.9e-06 | -1.1e-06 | -1.9e-07 | -3.1e-08 | -5.1e-09 | -8.2e-10 | -1.3e-10 |
| alpha_10 | 18 | 0.4 | -0.037 | -0.0055 | -0.00095 | -0.00017 | -3e-05 | -5.2e-06 | -9.2e-07 | -1.6e-07 | -2.9e-08 | -5.1e-09 | -9.1e-10 | -1.6e-10 |
| alpha_20 | 23 | 0.28 | -0.022 | -0.0034 | -0.0006 | -0.00011 | -2e-05 | -3.6e-06 | -6.6e-07 | -1.2e-07 | -2.2e-08 | -4e-09 | -7.5e-10 | -1.4e-10 |

LASSO Regression



- Minimization objective = LS Obj + λ^* (sum of absolute value of slope)

$$J(\Theta) = \frac{1}{M} \sum_{i=1}^M (\Theta^T \Phi(x_i) - y_i)^2 + \lambda \sum_{j=1}^p |\Theta_j|$$



- Lasso coordinate descent - closed form solution

$$\begin{cases} \theta_{(j+1)} = \rho_j + \lambda & \text{for } \rho_j < -\lambda \\ \theta_{(j+1)} = 0 & \text{for } -\lambda \leq \rho_j \leq \lambda \\ \theta_{(j+1)} = \rho_j - \lambda & \text{for } \rho_j > \lambda \end{cases}$$

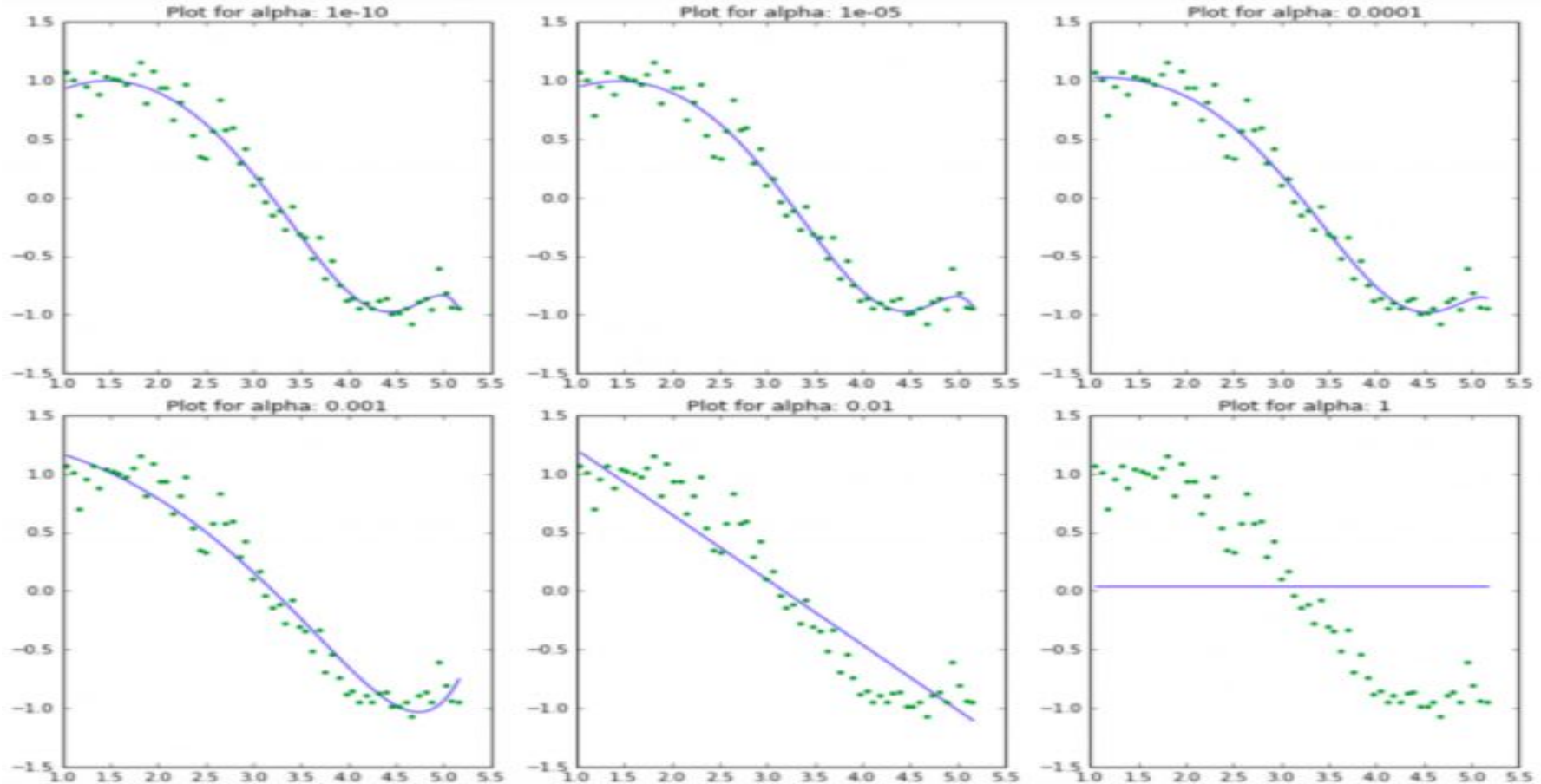
$$\rho_j = \sum_{i=1}^m x_j^{(i)} (y^{(i)} - \sum_{k \neq j}^p \theta_k x_k^{(i)})$$

Lasso Regularization



- One significant problem of ridge regression is that the penalty term will never force any of the coefficients to be exactly zero.
- Thus, the final model will include all predictors (features), which creates a challenge in model interpretation
- The lasso works in a similar way to ridge regression, except it uses a different penalty term that shrinks some of the coefficients exactly to zero.
- Lasso performs a variable feature selection.
- $Y = m_1 * x_1 + m_2 * x_2 + m_3 * x_3 + c$

Regression - Lasso Regression



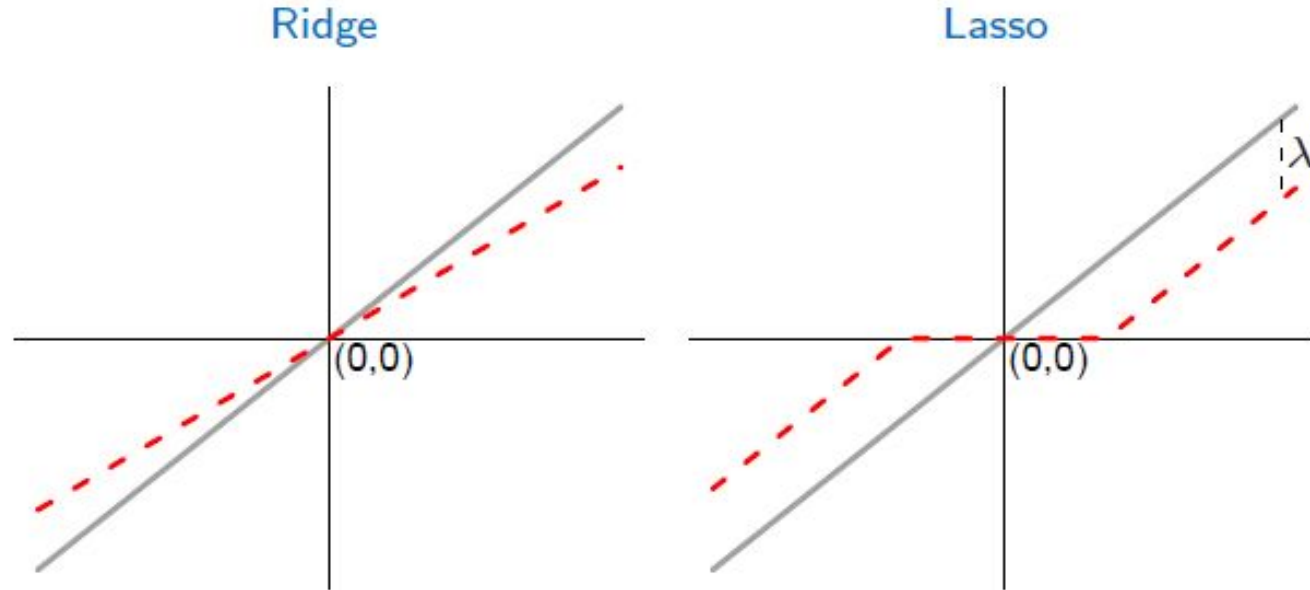
Regression - Lasso Regression



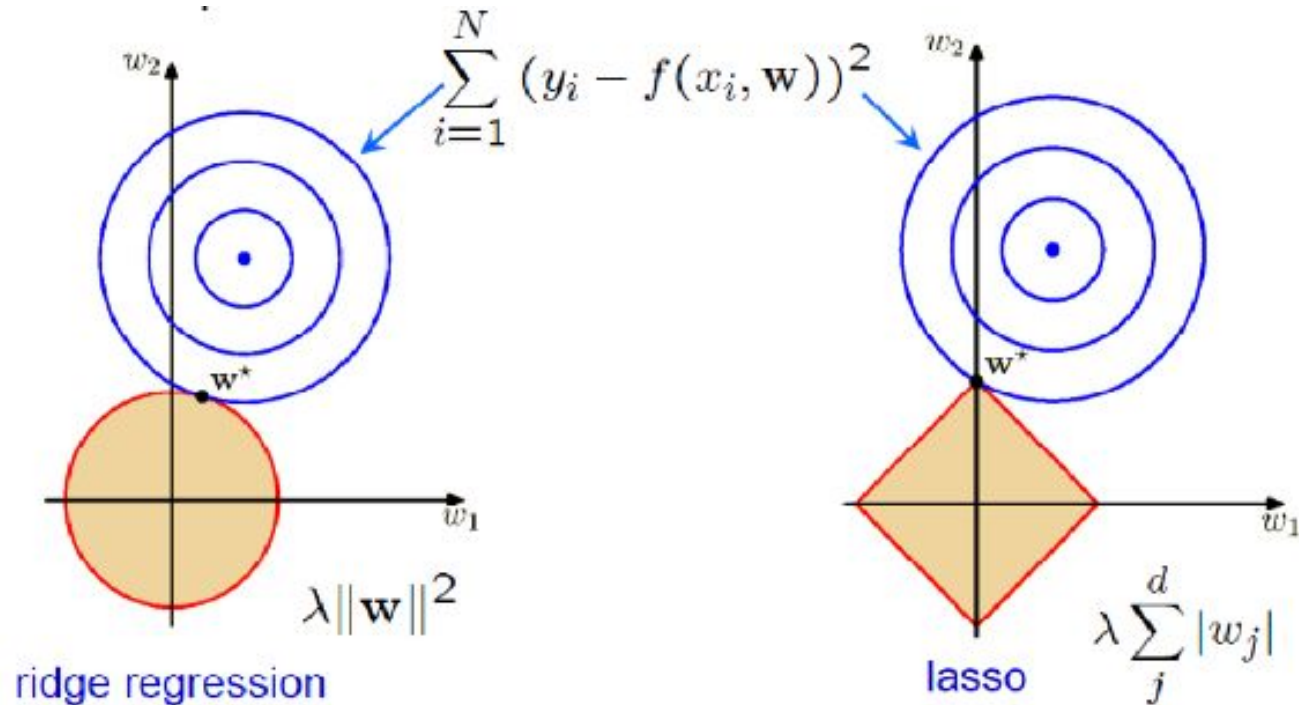
| | rss | intercept | coef_x_1 | coef_x_2 | coef_x_3 | coef_x_4 | coef_x_5 | coef_x_6 | coef_x_7 | coef_x_8 | coef_x_9 | coef_x_10 | coef_x_11 | coef_x_12 |
|--------------|------|-----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|-----------|-----------|
| alpha_1e-15 | 0.96 | 0.22 | 1.1 | -0.37 | 0.00089 | 0.0016 | -0.00012 | -6.4e-05 | -6.3e-06 | 1.4e-06 | 7.8e-07 | 2.1e-07 | 4e-08 | 5.4e-09 |
| alpha_1e-10 | 0.96 | 0.22 | 1.1 | -0.37 | 0.00088 | 0.0016 | -0.00012 | -6.4e-05 | -6.3e-06 | 1.4e-06 | 7.8e-07 | 2.1e-07 | 4e-08 | 5.4e-09 |
| alpha_1e-08 | 0.96 | 0.22 | 1.1 | -0.37 | 0.00077 | 0.0016 | -0.00011 | -6.4e-05 | -6.3e-06 | 1.4e-06 | 7.8e-07 | 2.1e-07 | 4e-08 | 5.3e-09 |
| alpha_1e-05 | 0.96 | 0.5 | 0.6 | -0.13 | -0.038 | -0 | 0 | 0 | 0 | 7.7e-06 | 1e-06 | 7.7e-08 | 0 | 0 |
| alpha_0.0001 | 1 | 0.9 | 0.17 | -0 | -0.048 | -0 | -0 | 0 | 0 | 9.5e-06 | 5.1e-07 | 0 | 0 | 0 |
| alpha_0.001 | 1.7 | 1.3 | -0 | -0.13 | -0 | -0 | -0 | 0 | 0 | 0 | 0 | 0 | 1.5e-08 | 7.5e-09 |
| alpha_0.01 | 3.6 | 1.8 | -0.55 | -0.00056 | -0 | -0 | -0 | -0 | -0 | -0 | -0 | 0 | 0 | 0 |
| alpha_1 | 37 | 0.038 | -0 | -0 | -0 | -0 | -0 | -0 | -0 | -0 | -0 | -0 | -0 | -0 |
| alpha_5 | 37 | 0.038 | -0 | -0 | -0 | -0 | -0 | -0 | -0 | -0 | -0 | -0 | -0 | -0 |
| alpha_10 | 37 | 0.038 | -0 | -0 | -0 | -0 | -0 | -0 | -0 | -0 | -0 | -0 | -0 | -0 |

HIGH SPARSITY

RIDGE vs LASSO Regression



RIDGE vs LASSO Regression



Lasso vs Ridge Regression



- The lasso produces simpler and more interpretable models that involved only a subset of predictors.
- The lasso leads to qualitatively similar behavior to ridge regression, in that as λ increases, the variance decreases and the bias increases.
- The lasso can generate more accurate predictions compared to ridge regression.
- Cross-validation can be used in order to determine which approach is better on a particular data set.

Tuning Parameter



- Increased λ leads to increased bias but decreased variance
- $\lambda = 0$
 - The objective becomes same as simple linear regression.
- $0 < \lambda < \infty$:
 - The coefficients will be somewhere between 0 and ones for simple linear regression.

Cross-validation is used to find the parameter that results in the lowest variance.

