GENERALISATION

Dr. Debanjan Pathak

Over-fitting, Bias and Variance Relationship

Generalization

- Given a collection of examples $(x^{(i)}\ f\ (x^{(i)}));\ i = 1, 2, \ldots, N,$ of a function $f\ (x)$, return a function $h(x)$ that approximates $f\ (x)$.
- $h(x)$ is called hypothesis function and $f(x)$ is called true function.
- It is not easy to tell whether any particular $h(.)$ is a good approximation of $f\ (.)$.
- A good approximation will generalize well—that is, will predict novel patterns correctly.
- Generalization performance is, thus, the fundamental problem in inductive learning (Inductive Learning, also known as Concept Learning, is how AI systems attempt to use a generalized rule to carry out observations).
- The off-training set error—the error on points not in the training set, will be used as a measure of generalization performance (generally called testing error).

Generalization

- If one algorithm appears to outperform another in a specific situation, it is as a result of its fit to the specific learning problem, not the general supremacy of the algorithm.
- Even if the algorithms are widely used and grounded in terms of theory, they will not perform well on certain problems.

Bias and variance

- A measure of how close the mapping function $h(x; \mathcal{D}_j)$ is to the desired one is, therefore, given by the error function,

$$\text{error}_{\mathcal{D}_j}[h] = [h(\mathbf{x}; \mathcal{D}_j) - f(\mathbf{x})]^2$$

- The value of this quantity will depend on the dataset $\mathcal{D}_j$ on which it is trained.
- We write the average over the complete ensemble of datasets as,

$$\text{error}_{\mathcal{D}}[h] = \mathbb{E}_{\mathcal{D}}\{[h(\mathbf{x}; \mathcal{D}_j) - f(\mathbf{x})]^2\}$$

Expected value;
Arithmetic mean of a large number of independent realizations

- It may be that the hypothesis function $h(.)$ is on average, different from the regression function $f(x)$. This is called bias.
- It may be that the hypothesis function is very sensitive to the particular dataset $\mathcal{D}_j$, so that for a given x, it is larger than the required value for some datasets, and smaller for other datasets. This is called variance.

Machine learning bias, also sometimes called *algorithm bias* or *AI bias*, is a phenomenon that occurs when an algorithm produces results that are systemically prejudiced due to erroneous assumptions in the machine learning process.

Bias and variance

$$\text{error}_{\mathcal{D}}[h] = \mathbb{E}_{\mathcal{D}}\{[h(\mathbf{x}; \mathcal{D}_j) - f(\mathbf{x})]^2\}$$

$$= \underbrace{[\mathbb{E}_{\mathcal{D}}\{h(\mathbf{x}; \mathcal{D}_j)\} - f(\mathbf{x})]^2}_{(\text{bias})^2} + \underbrace{\mathbb{E}_{\mathcal{D}}\{[h(\mathbf{x}; \mathcal{D}_j) - \mathbb{E}_{\mathcal{D}}\{h(\mathbf{x}; \mathcal{D}_j)\}]^2\}}_{\text{variance}}$$

(w.r.t original) (range of the results of hypothesis fn)

- The bias measures the level to which the average (over all datasets) of the hypothesis function is different from the desired function f (x).
- The bias term reflects the approximation error that the hypothesis h(.) is expected to have on average when trained on datasets of same finite size.
- In general, higher the complexity of the hypothesis function (more flexible function with large number of adjustable parameters), the lower is the approximation error.

Bias and variance

$$\text{error}_{\mathcal{D}}[h] = \mathbb{E}_{\mathcal{D}}\{[h(\mathbf{x}; \mathcal{D}_j) - f(\mathbf{x})]^2\}$$

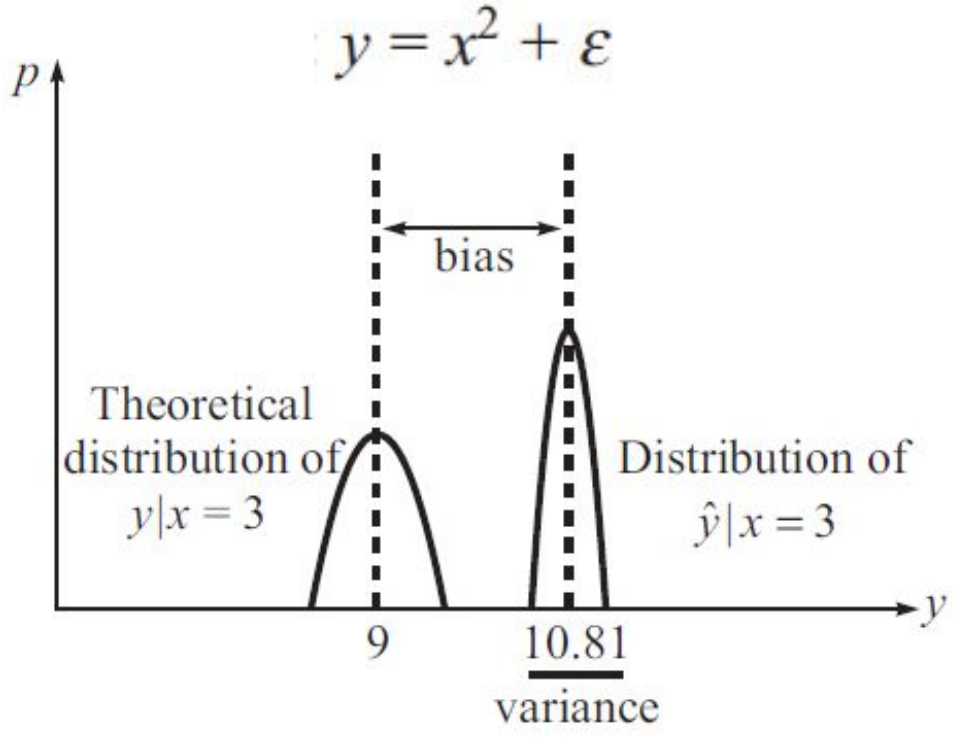
$$= \underbrace{[\mathbb{E}_{\mathcal{D}}\{h(\mathbf{x}; \mathcal{D}_j)\} - f(\mathbf{x})]^2}_{(\text{bias})^2} + \underbrace{\mathbb{E}_{\mathcal{D}}\{[h(\mathbf{x}; \mathcal{D}_j) - \mathbb{E}_{\mathcal{D}}\{h(\mathbf{x}; \mathcal{D}_j)\}]^2\}}_{\text{variance}}$$

(w.r.t original) (range of the results of hypothesis fn)

- The variance is a measure of the level to which the hypothesis function h(x; D_j) is sensitive to the specific selection of the dataset.
- The variance term reflects the capability of the trained model on a data sample to generalize to other data samples.
- Low variance means that the estimate of f (x) based on a data sample does not change much on the average as the data sample varies.
- Unfortunately, the higher the complexity of the hypothesis function (which results in low bias/low approximation error), the higher is the variance.

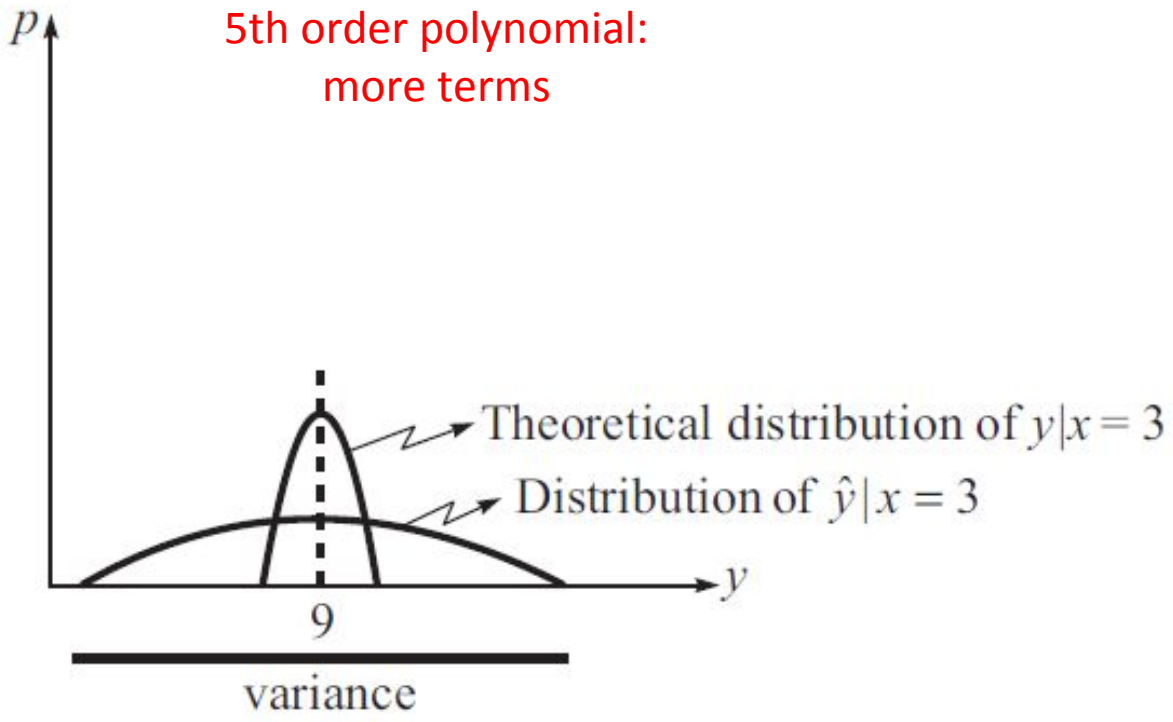
Bias and variance

- To minimize the overall mean-square-error, we need a hypothesis that results in low bias and low variance.
- This is known as bias-variance dilemma or bias-variance trade-off.



(a) Linear curve fitting

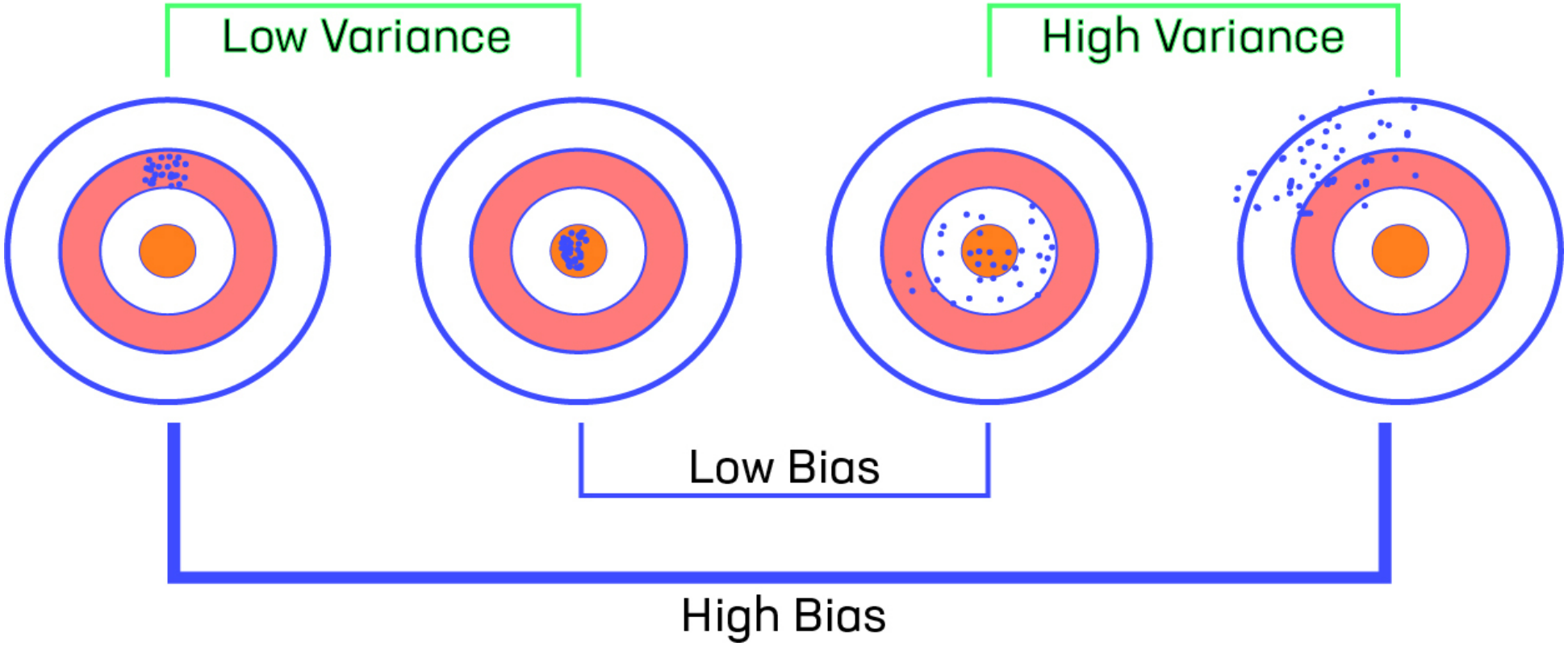
5th order polynomial:
more terms



(b) Higher-order polynomial fitting

Bias and variance

- To minimize the overall mean-square-error, we need a hypothesis that results in low bias and low variance.
- This is known as bias-variance dilemma or bias-variance trade-off.



Bias and variance

- To minimize the overall mean-square-error, we need a hypothesis that results in low bias and low variance.
- This is known as bias-variance dilemma or bias-variance trade-off.

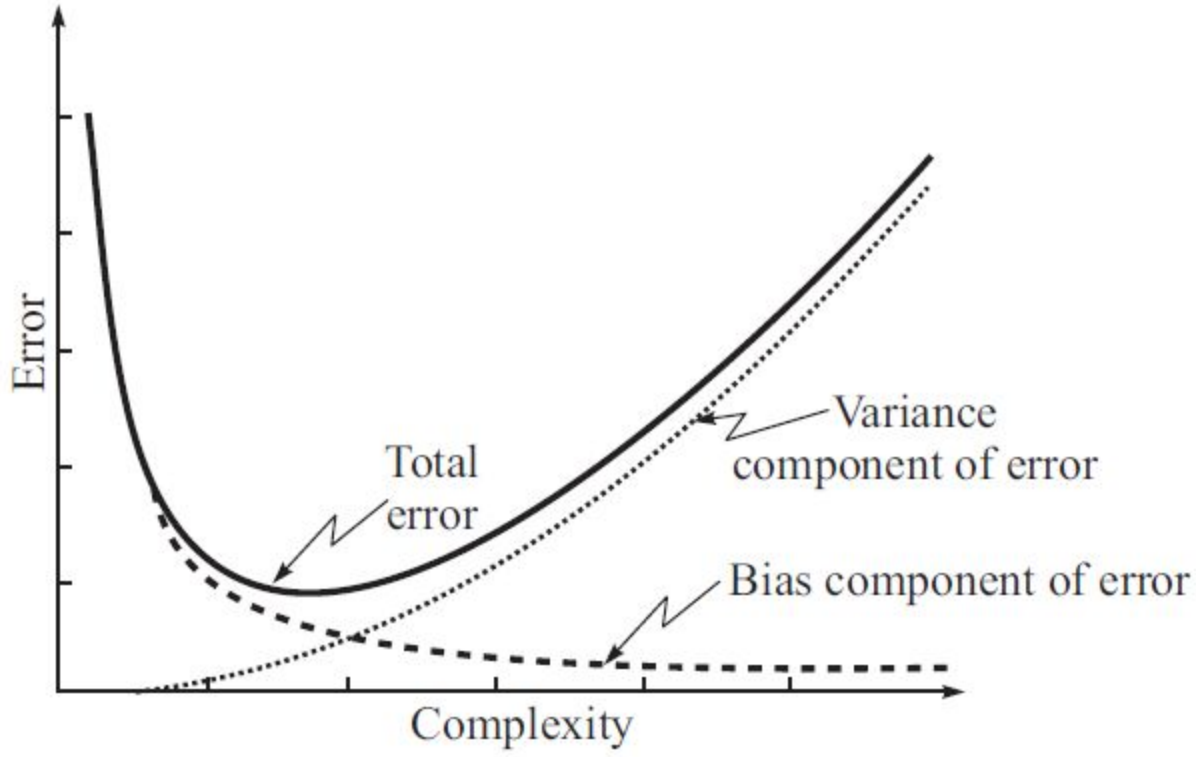
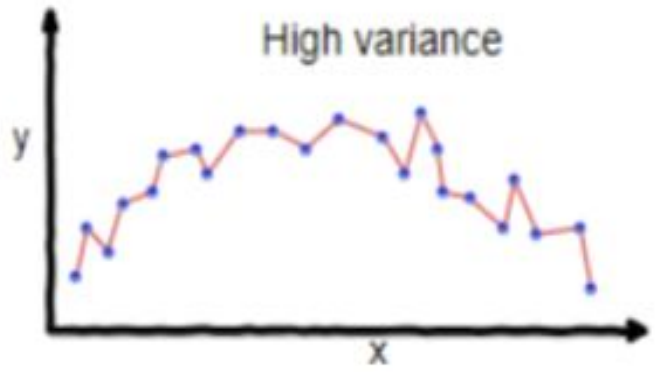
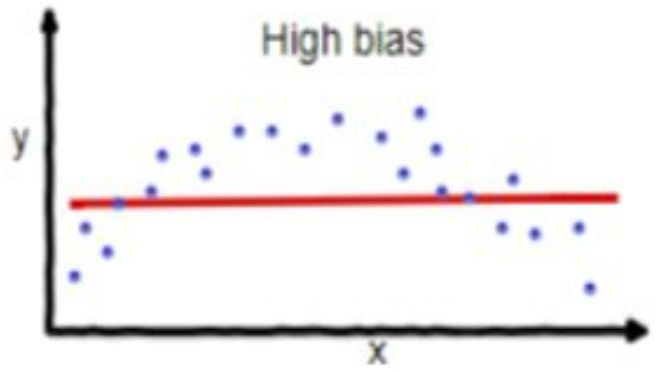


Figure 2.4 Bias-variance trade-off

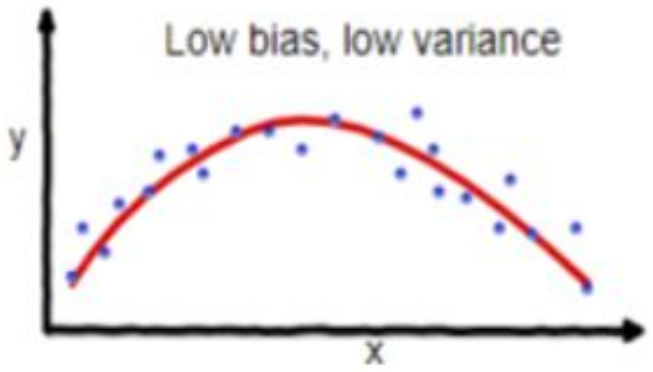
Over-fitting



overfitting

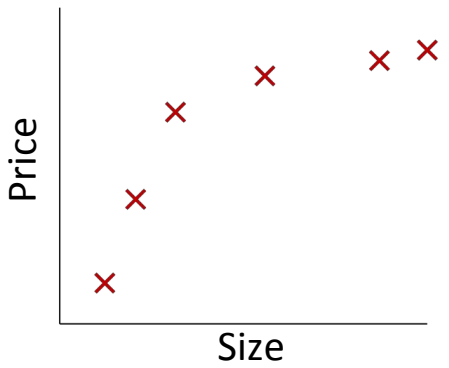


underfitting



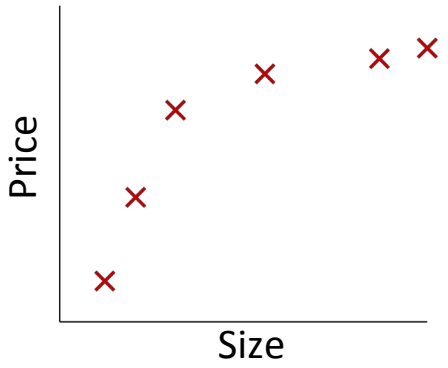
Good balance

Over-fitting



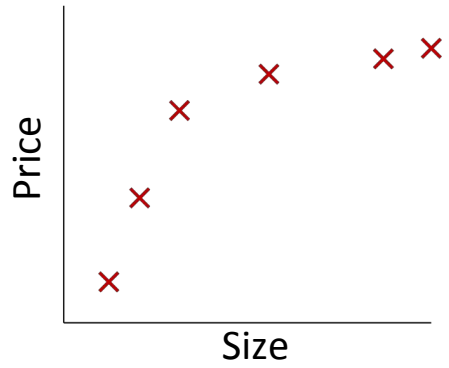
$\theta_0 + \theta_1 x$

High bias
(underfit)



$\theta_0 + \theta_1 x + \theta_2 x^2$

"Just right"



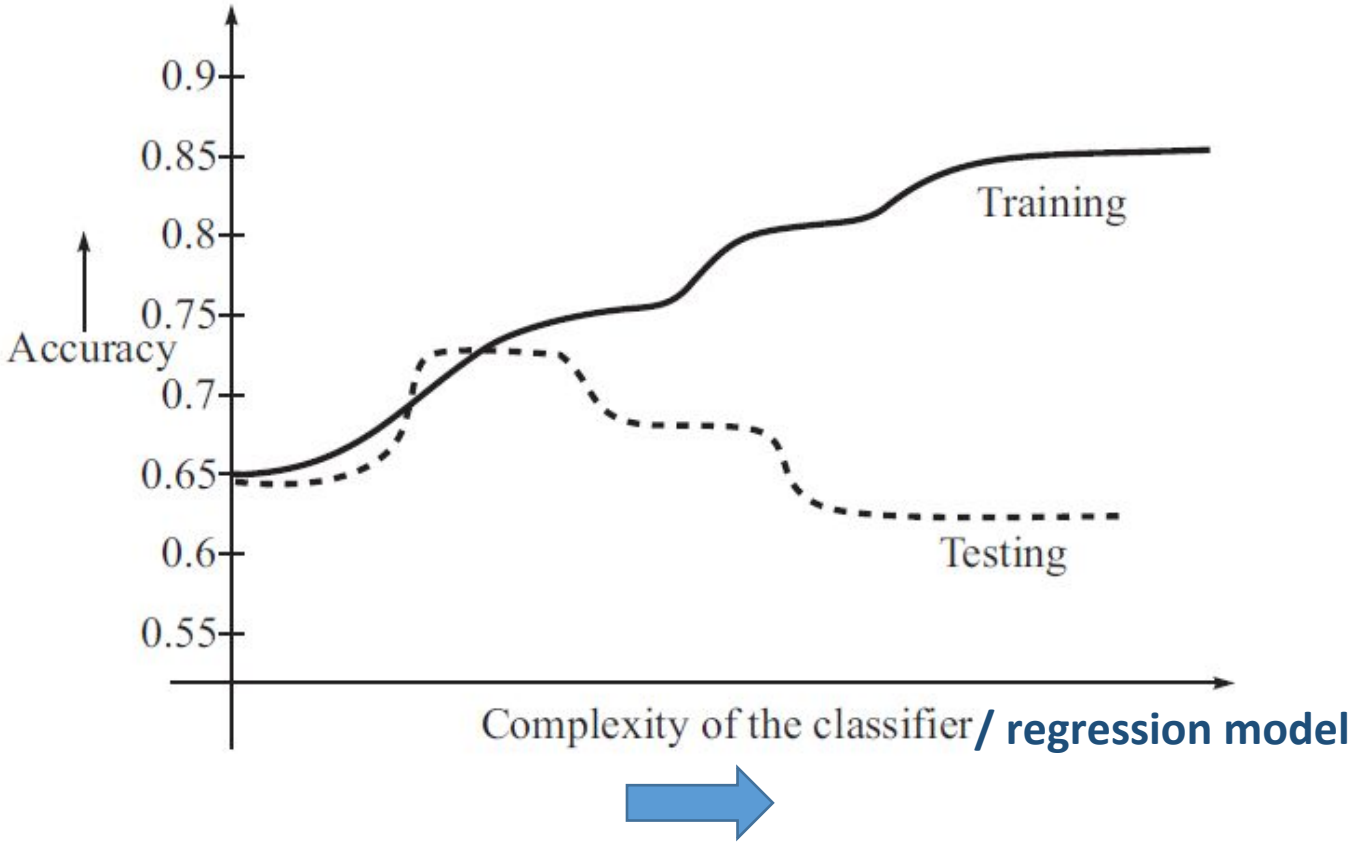
$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

High variance
(overfit)

Over-fitting

- **Occam's Razor Principle:** The simpler explanations are more reasonable, and any unnecessary complexity should be shaved off.
- 'simpler' may imply needing lesser parameters, lesser training time, fewer attributes for data representation, and lesser computational complexity.
- Occam's razor principle suggests hypothesis functions that avoid overfitting of the training data.
- A learning machine is said to **overfit** the training examples if certain other learning machine that fits the training examples less well, but actually performs better over the total distribution of patterns (i.e., including patterns beyond the training set).

Over-fitting



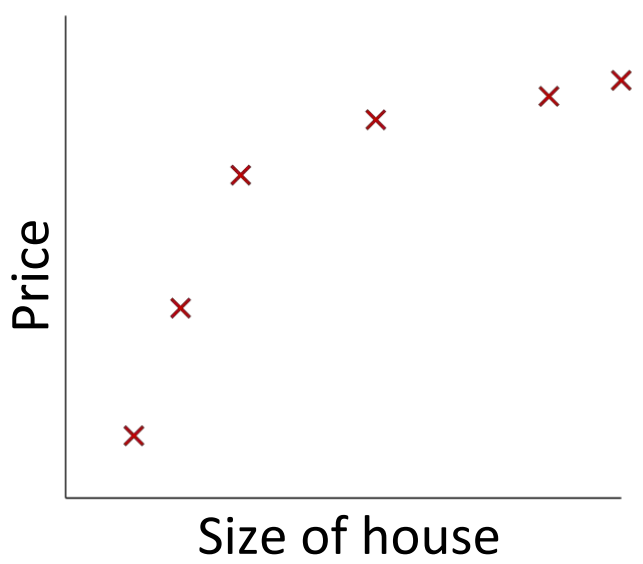
(more number of weights in the neural network, large number of nodes in the decision tree, more number of rules in a fuzzy logic model, etc.)

Addressing overfitting:

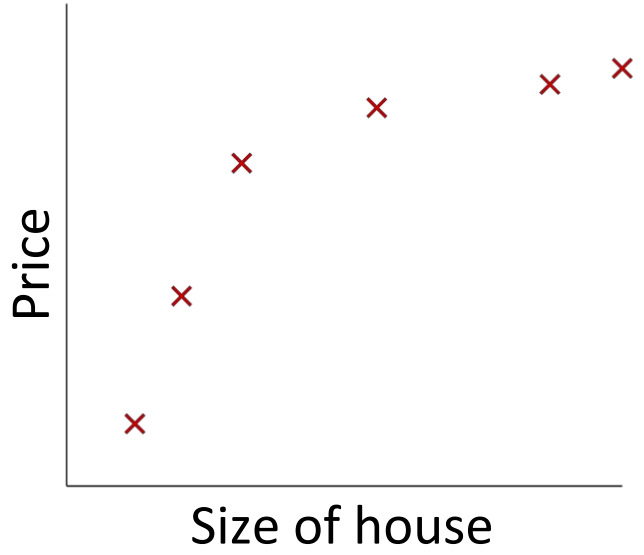
Options:

1. Reduce number of features.
   - Manually select which features to keep.
   - Model selection algorithm (later in course).
2. Regularization.
   - Keep all the features, but reduce magnitude/values of parameters θ_j or β.
   - Works well when we have a lot of features, each of which contributes a bit to predicting y.

Regularization



$$\theta_0 + \theta_1 x + \theta_2 x^2$$

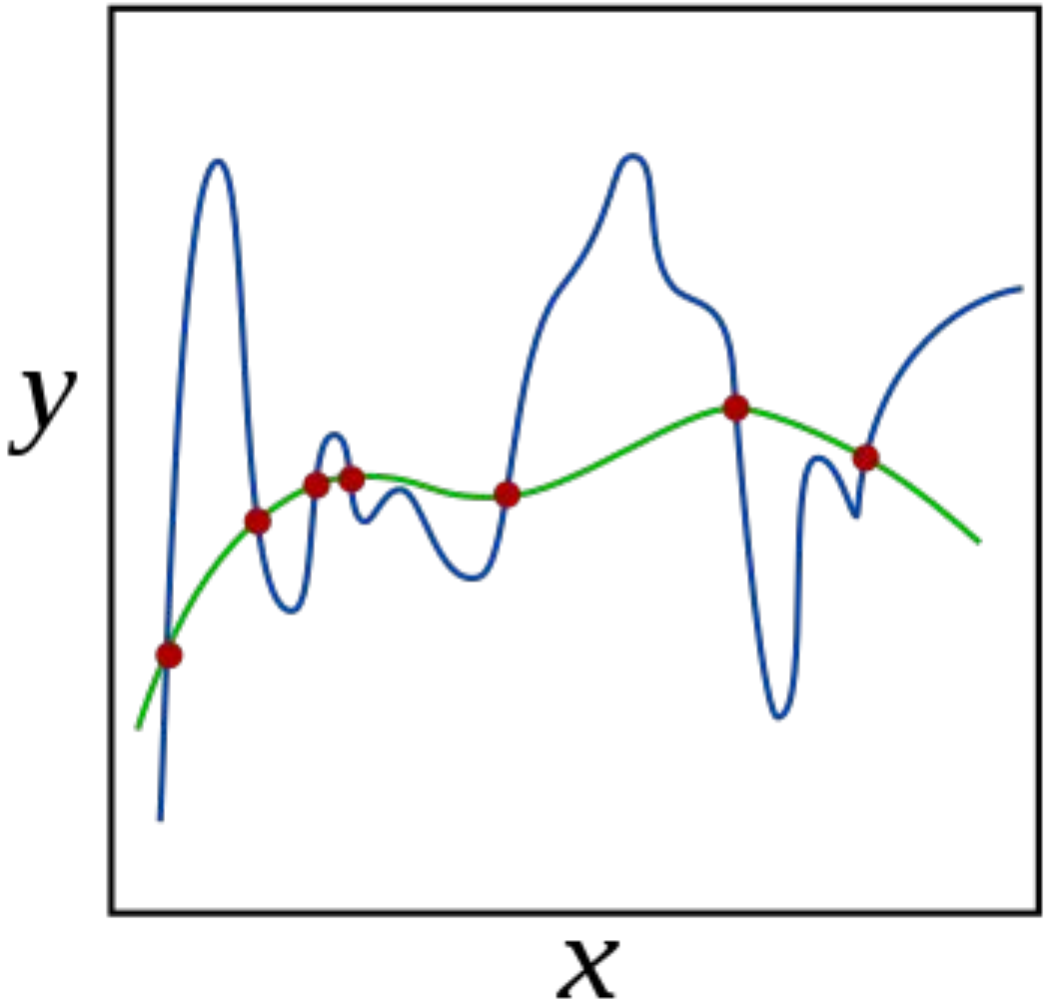


$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Over-fitting can be avoided if suppose we penalize and make θ_3, θ_4 really small.

Regularization

- This is a form of regression, that constrains/ regularizes or shrinks the coefficient estimates towards zero.
- In other words, this technique discourages learning a more complex or flexible model, so as to avoid the risk of overfitting.



Regularization

- $Y \approx \beta 0 + \beta 1X1 + \beta 2X2 + \ldots + \beta pXp$
- Here Y represents the learned relation and β represents the coefficient estimates for different variables or predictors (X).
- The fitting procedure involves a loss function, known as residual sum of squares or RSS.
- The coefficients are chosen, such that they minimize this loss function.

$$\mathrm{RSS} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 .$$

Regularization

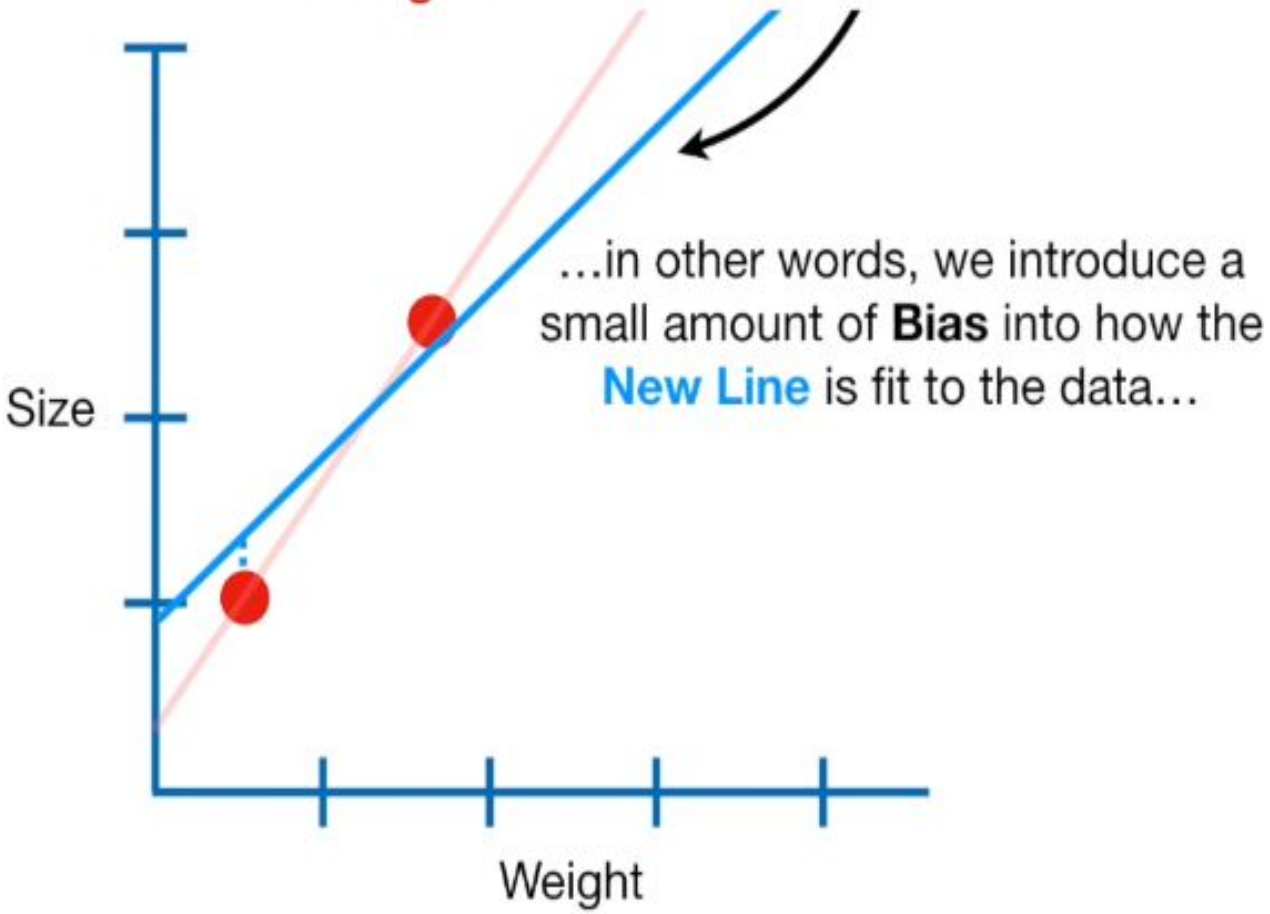
$$\text{RSS} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 .$$

- Now, this will adjust the coefficients based on your training data.
- If there is noise in the training data (or more terms in the equation), then the estimated coefficients won't generalize well to the future data.
- This is where regularization comes in and shrinks or regularizes these learned estimates towards zero.
- Regularization, significantly reduces the variance of the model, without substantial increase in its bias.

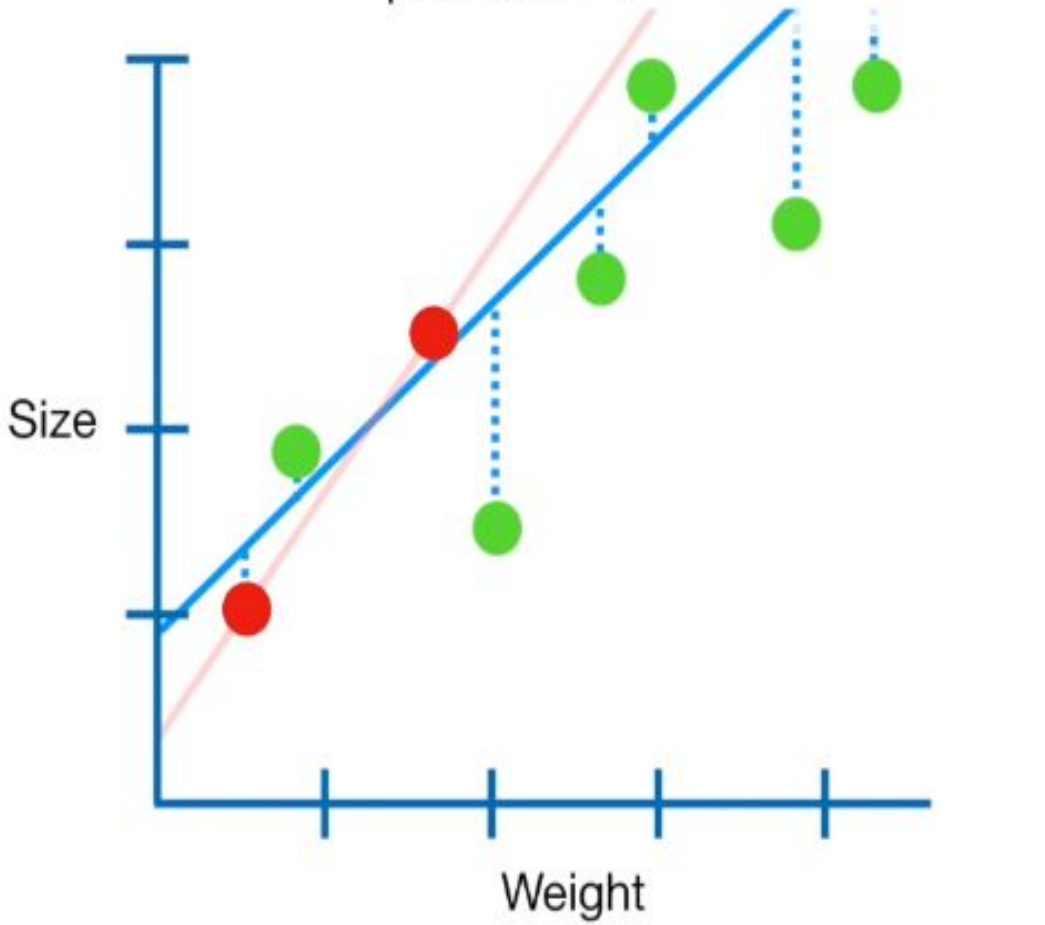
RIDGE Regression

Basics

The main idea behind **Ridge Regression** is to find a **New Line** that doesn't fit the **Training Data** as well…



In other words, by starting with a slightly worse fit, **Ridge Regression** can provide better long term predictions.



Basics

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2=\mathrm{RSS}+\lambda\sum_{j=1}^{p}\beta_j^2$$

- Above image shows ridge regression, where the RSS is modified by adding the shrinkage quantity.
- Now, the coefficients are estimated by minimizing this function.
- Here, λ is the tuning parameter that decides how much we want to penalize the flexibility of our model.
- The increase in flexibility of a model is represented by increase in its coefficients
- if we want to minimize the above function, then these coefficients need to be small.

Basics

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2=\mathrm{RSS}+\lambda\sum_{j=1}^{p}\beta_j^2$$

- This is how the Ridge regression technique prevents coefficients from rising too high.
- Also, notice that we shrink the estimated association of each variable with the response, except the intercept β0, This intercept is a measure of the mean value of the response when xi1 = xi2 = …= xip = 0.
- When λ = 0, the penalty term has no effect, and the estimates produced by ridge regression will be equal to least squares.
- However, as λ→∞, the impact of the shrinkage penalty grows, and the ridge regression coefficient estimates will approach zero.

RIDGE vs Oridinary Least Squares

$$L_{ridge}(\hat{\beta}) = \sum_{i=1}^{n}(y_i - x_i'\hat{\beta})^2 + \lambda \sum_{j=1}^{m}\hat{\beta}_j^2 = \|y - X\hat{\beta}\|^2 + \lambda\|\hat{\beta}\|^2.$$

- Solving this for $\widehat{\beta}$ gives the the ridge regression estimates

$$\hat{\beta}_{ridge} = (X'X + \lambda I)^{-1}(X'Y)$$

I - Identity matrix

- You might have remembered that the optimal solution in case of Oridinary least Squares (OLS) is

$$\overline{\beta}^* = (XX^T)^{-1}Xy$$

RIDGE vs Oridinary least Squares

- The λ parameter is the regularization penalty. Notice that:
  - As $\lambda \to 0, \quad \hat{\beta}_{ridge} \to \hat{\beta}_{OLS};$
  - As $\lambda \to \infty, \quad \hat{\beta}_{ridge} \to 0.$
- So, setting λ to 0 is the same as using the OLS, while the larger its value, the stronger is the coefficients' size penalized.

Constraints

- As can be seen, selecting a good value of λ is critical.
- Cross validation comes in handy for this purpose.
- The larger LAMBDA means, our prediction became less sensitive to the independent variables.
- The coefficient estimates produced by this method are also known as the L2 norm.
- The coefficients that are produced by the standard least squares method are scale equivariant, i.e. if we multiply each input by c then the corresponding coefficients are scaled by a factor of 1/c.
- Therefore, regardless of how the predictor is scaled, the multiplication of predictor and coefficient $(Xj\beta j)$ remains the same.

Standardizing the predictors

- However, this is not the case with ridge regression, and therefore, we need to standardize the predictors or bring the predictors to the same scale before performing ridge regression.
- The formula used to do this is given below.

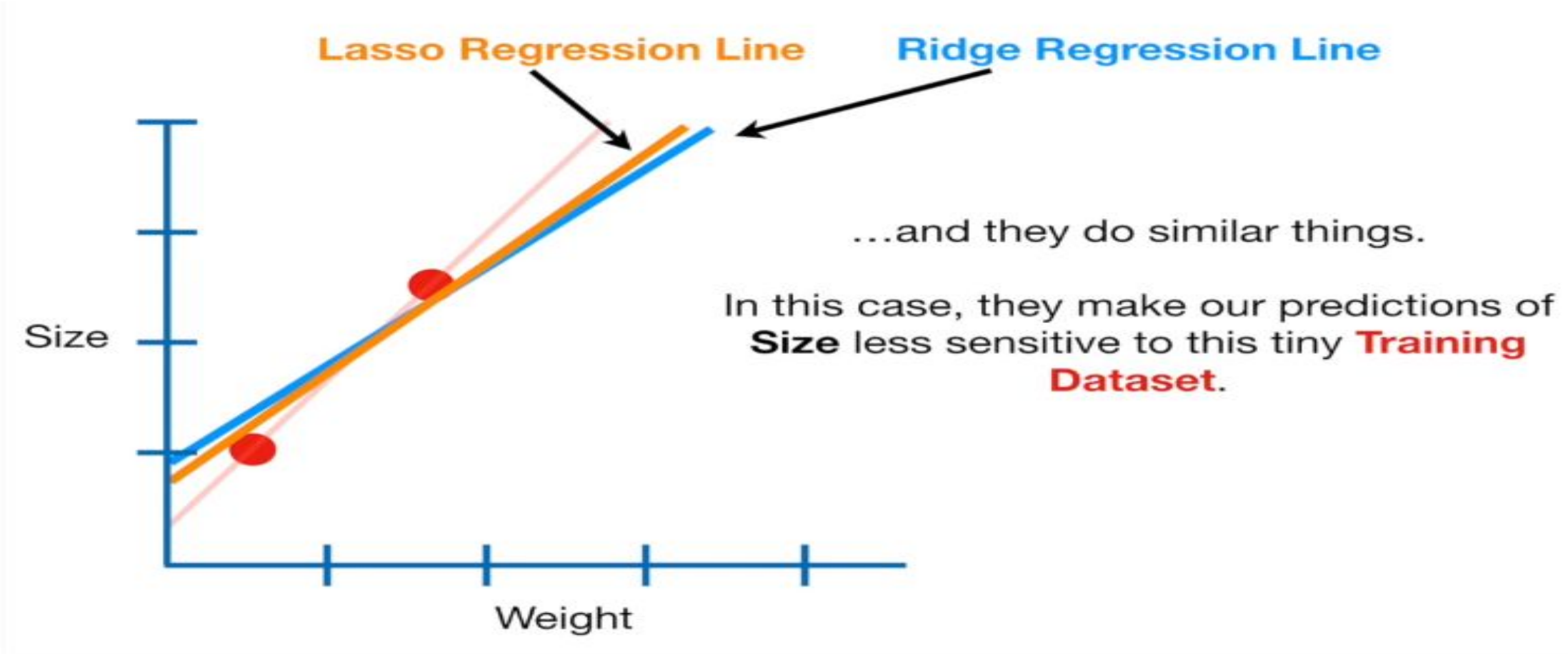
$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2}}$$

tures)

amples)

LASSO Regression

Basics


Lasso Regression Line
Ridge Regression Line
...and they do similar things.
In this case, they make our predictions of Size less sensitive to this tiny Training Dataset.
Size
Weight

Basics

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j| = \mathrm{RSS}+\lambda\sum_{j=1}^{p}|\beta_j|.$$

- Lasso is a variation, in which the above function is minimized.
- Its clear that this variation differs from ridge regression only in penalizing the high coefficients.
- It uses |βj|(modulus) instead of squares of β, as its penalty.
- In statistics, this is known as the L1 norm.

Basics

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j| = \mathrm{RSS}+\lambda\sum_{j=1}^{p}|\beta_j|.$$

- Lasso regression stands for Least Absolute Shrinkage and Selection Operator.
- It also adds penalty term to the cost function.
- This term is the absolute sum of the coefficients.

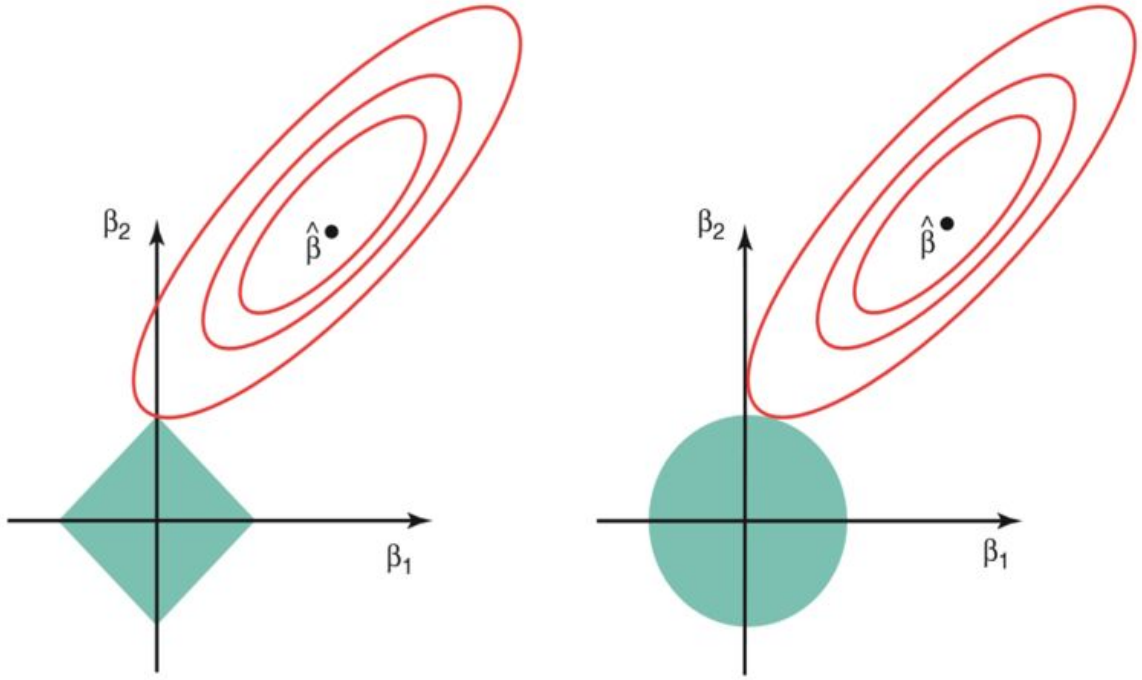
LASSO vs RIDGE

- Lasso regression differs from ridge regression in a way that it uses absolute values within the penalty function, rather than that of squares.
- This leads to penalizing (or equivalently constraining the sum of the absolute values of the estimates) values which causes some of the parameter estimates to turn out exactly zero.
- The more penalty is applied, the more the estimates get shrunk towards absolute zero.
- This helps to variable selection out of given range of n variables.

LASSO vs RIDGE

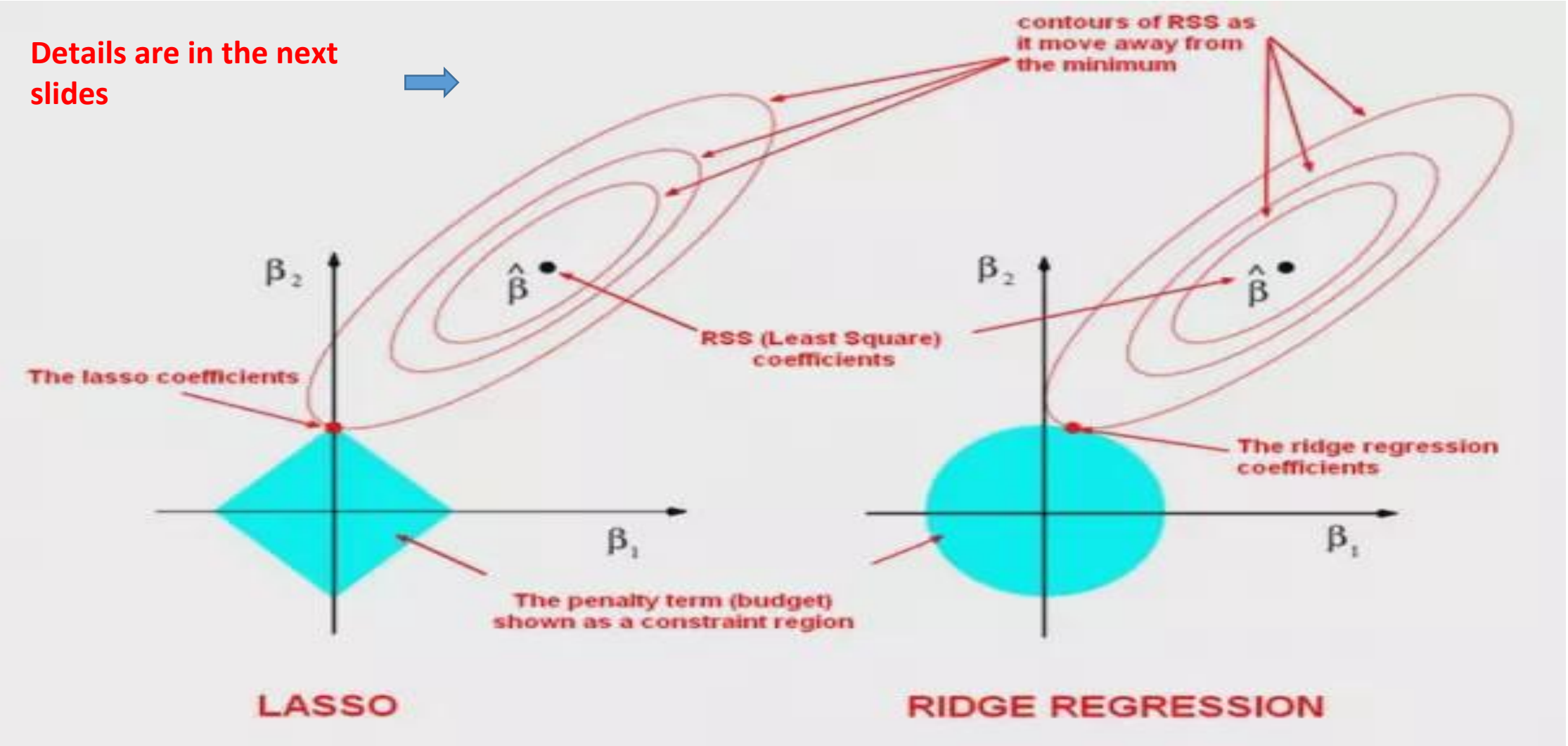
- The ridge regression can be thought of as solving an equation, where summation of squares of coefficients is less than or equal to *s*.
- And the Lasso can be thought of as an equation where summation of modulus of coefficients is less than or equal to *s*.
- Here, *s* is a constant that exists for each value of shrinkage factor λ.
- Consider there are 2 parameters in a given problem.
- Then according to above formulation, lasso regression is expressed by $|\beta 1|+|\beta 2| \leq s$ and the ridge regression is expressed by $\beta 1^2 + \beta 2^2 \leq s$.
- This implies that ridge regression coefficients have the smallest RSS (loss function) for all points that lie within the circle given by $\beta 1^2 + \beta 2^2 \leq s$.
- Similarly, the lasso coefficients have the smallest RSS (loss function) for all points that lie within the diamond given by $|\beta 1|+|\beta 2| \leq s$.

LASSO vs RIDGE

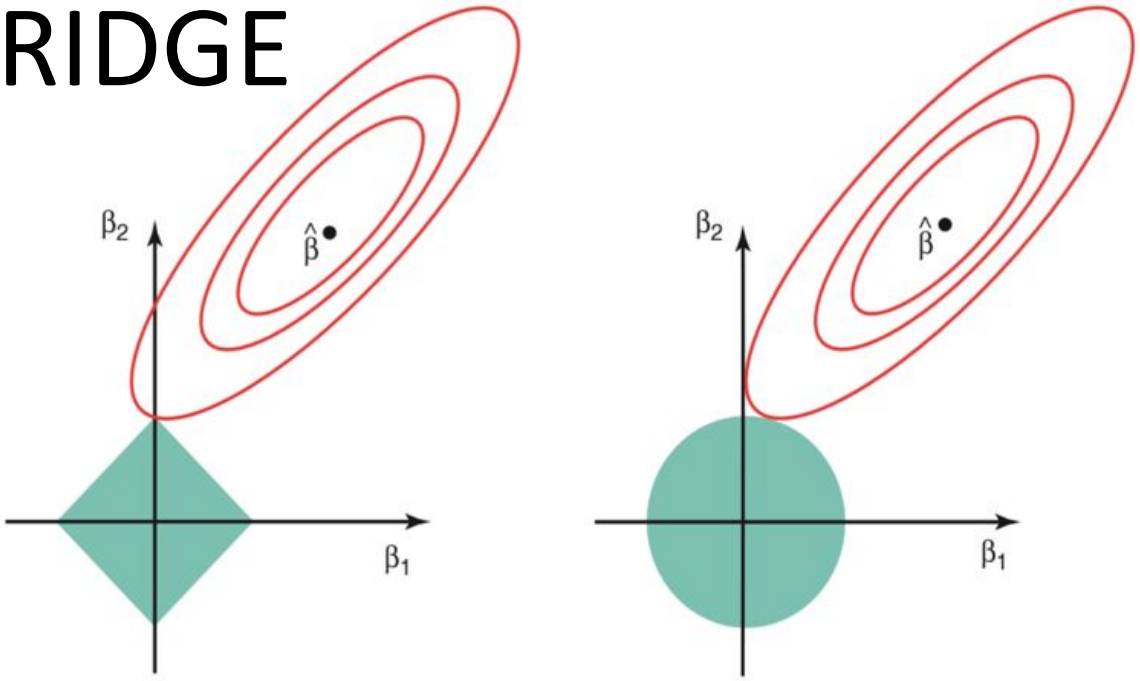


- This implies that ridge regression coefficients have the smallest RSS (loss function) for all points that lie within the circle given by $\beta 1^2 + \beta 2^2 \leq s$.
- Similarly, the lasso coefficients have the smallest RSS (loss function) for all points that lie within the diamond given by $|\beta 1| + |\beta 2| \leq s$.

LASSO vs RIDGE

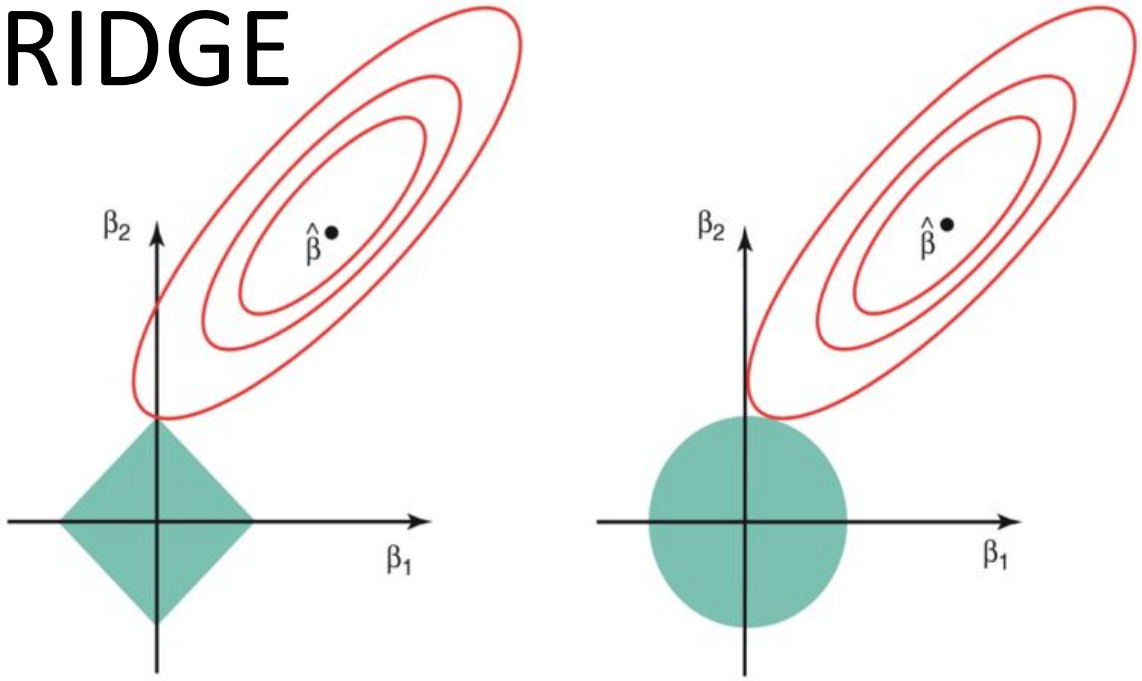


LASSO vs RIDGE



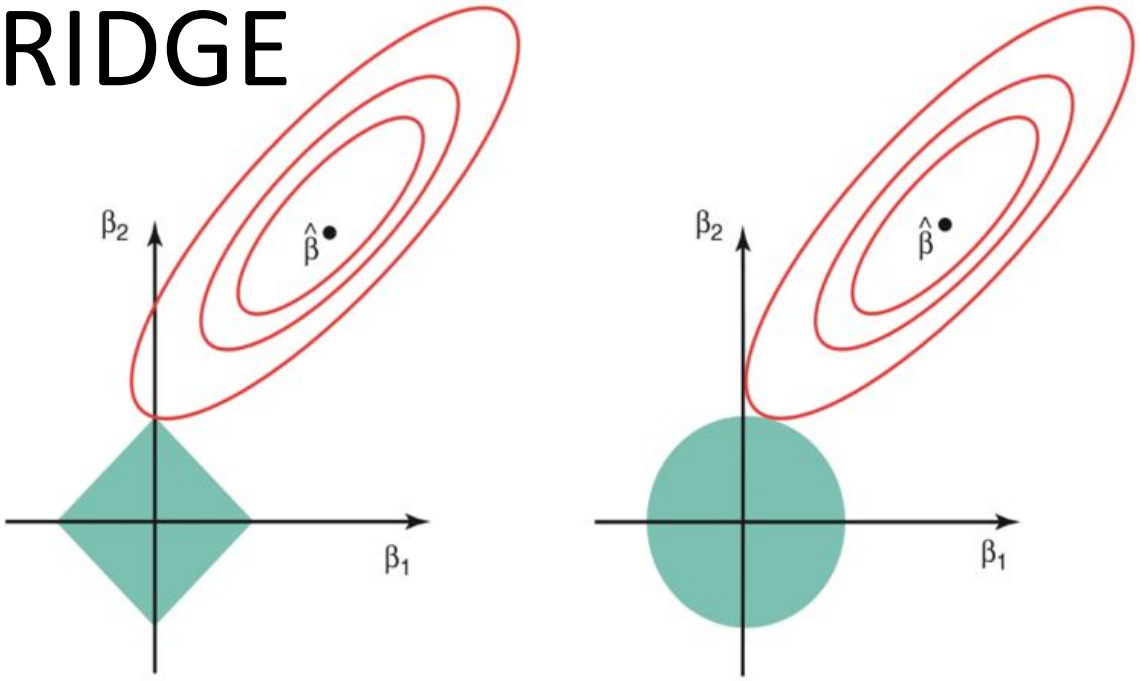
- The above image shows the constraint functions(green areas), for lasso(left) and ridge regression(right), along with contours for RSS (red ellipse).
- Points on the ellipse share the value of RSS.

LASSO vs RIDGE



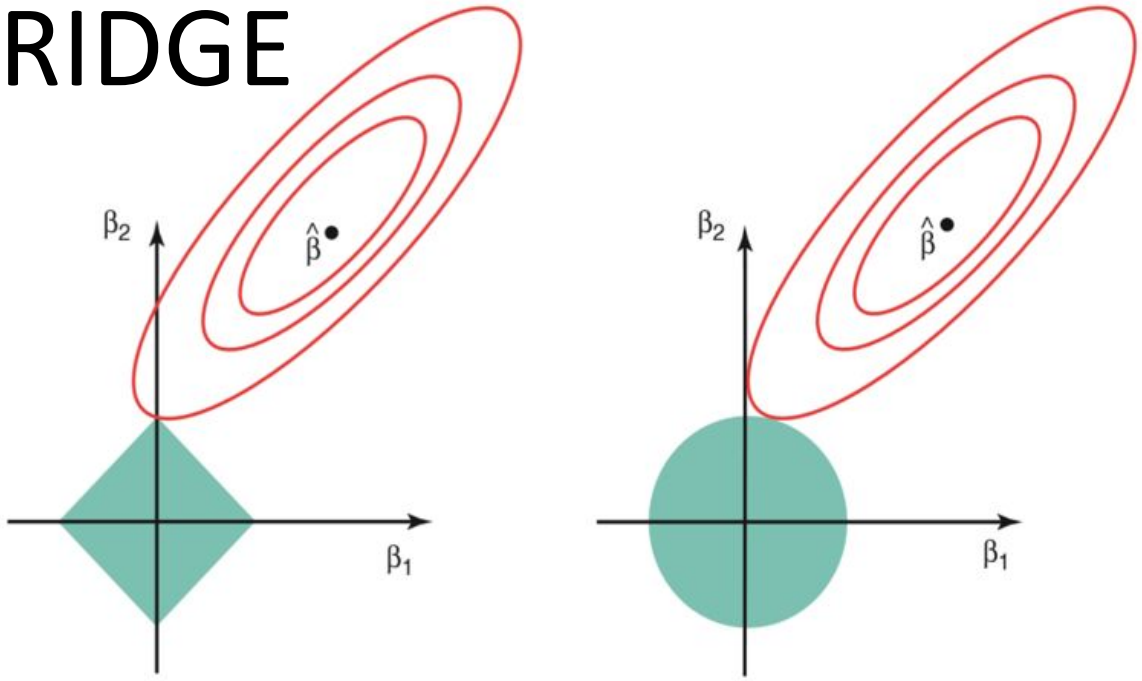
- For a very large value of s, the green regions will contain the center of the ellipse, making coefficient estimates of both regression techniques, equal to the least squares estimates.
- But, in this above case, the lasso and ridge regression coefficient estimates are given by the first point at which an ellipse contacts the constraint region.

LASSO vs RIDGE



- Since ridge regression has a circular constraint with no sharp points, this intersection will not generally occur on an axis, and so the ridge regression coefficient estimates will be exclusively non-zero.
- However, the lasso constraint has corners at each of the axes, and so the ellipse will often intersect the constraint region at an axis.

LASSO vs RIDGE



- When this occurs, one of the coefficients will equal zero.
- In higher dimensions (where parameters are much more than 2), many of the coefficient estimates may equal zero simultaneously.
- That means, in the case of the lasso, the L1 penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when the tuning parameter λ is sufficiently large.

LASSO vs RIDGE

- This sheds light on the obvious disadvantage of ridge regression, which is model interpretability.
- It will shrink the coefficients for least important predictors, very close to zero.
- But it will never make them exactly zero.
- In other words, the final model will include all predictors.
- Therefore, the lasso method also performs variable selection and is said to yield sparse models.
- You may visit https://www.pluralsight.com/guides/linear-lasso-ridge-regression-scikit-learn to see a numerical example.

What does **Regularization** achieve?

- The tuning parameter λ, used in the regularization techniques described above, controls the impact on bias and variance.
- As the value of λ rises, it reduces the value of coefficients and thus reducing the variance.
- Till a point, this increase in λ is beneficial as it is only reducing the variance (hence avoiding overfitting), without loosing any important properties in the data.
- But after certain value, the model starts loosing important properties, giving rise to bias in the model and thus underfitting.
- Therefore, the value of λ should be carefully selected.

Debugging a learning algorithm:

Suppose you have implemented regularized linear regression to predict housing prices. However, when you test your hypothesis in a new set of houses, you find that it makes unacceptably large errors in its prediction. What should you try next?

- Get more training examples
- Try smaller sets of features
- Try getting additional features
- Try adding polynomial features $(x_1^2, x_2^2, x_1 x_2, \text{etc})$
- Try decreasing λ
- Try increasing λ