The Bias-Variance Trade-Off

Administrivia

- Posted complete Homework 1 after adding Problem 4
  - If you haven't started yet, start with the updated notebook
  - If you've already started, just copy-paste to your working notebook
- There is a reading quiz associated with today's lecture. Due before class Friday

The RoadMap

- **Last Last Time:**
  - Regression Refresher (there was nothing fresh about it)

- **Last Time:**
  - Polynomial Regression
  - Regularization (wiggles are bad, Man)

- **This Time:**
  - Few more details about Ridge Regression
  - Bias-Variance Trade-Off (what does it all **MEAN**?)

Previously on CSCI 4622

Given training data $(x_{i1}, x_{i2}, \ldots, x_{ip}, y_i)$ for $i = 1, 2, \ldots, n$ fit a regression of the form

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \epsilon_i \qquad \text{where} \qquad \epsilon_i \sim N(0, \sigma^2)$$

Estimates of the parameters are found by minimizing

$$\text{RSS} = \sum_{i=1}^{n} \left[(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) - y_i\right]^2 = \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|^2$$

OLS Regression with Polynomial features badly overfit. Solution is Regularization

LOSS

$$RSS_\lambda = \sum_{i=1}^{n} \left[(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) - y_i\right]^2 + \lambda \sum_{k=1}^{p} \beta_k^2$$

Regularization Recap

$$RSS_\lambda = \sum_{i=1}^{n} \left[(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) - y_i\right]^2 + \lambda \sum_{k=1}^{p} \beta_k^2$$

- Adding penalty term to slopes in RSS encourages parameters to stay small
- Helps prevent overfitting
- Don't regularize the bias term
- You should always do some kind of regularization
- If you choose λ carefully, it will always help Generalization

Feature Scaling

For lots of learning methods we'll explore, it's helpful if features are on same scale

- Many learning algorithms are affected by disparity of scale between features

CENTERING:
$$\begin{array}{c} \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ x_{31} & x_{32} \end{bmatrix} \\ \begin{matrix} \bar{x}_1 & \quad \bar{x}_2 \end{matrix} \end{array}$$

Common transformations:

- Feature **Centering**: Subtract mean $\bar{x}_k$ of feature data from each $\bar{x}_{ik}$ for $i = 1, \ldots, n$
- Feature **Standardization**: mean-center and scale to unit standard deviation
- Feature **Normalization**: shift and/or scale so that all features are in [0,1]

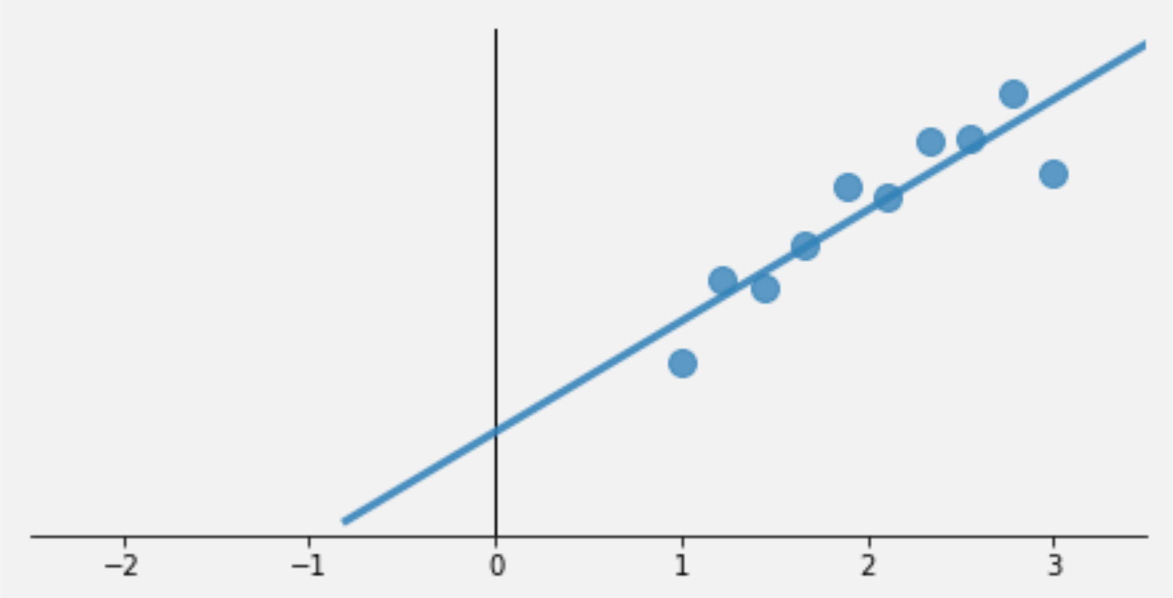
Feature Scaling

Let's go back to the regression setting

$$\hat{\boldsymbol{\beta}} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \left[(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) - y_i\right]^2 + \lambda \sum_{k=1}^{p} \beta_k^2$$

What affect does **centering** have on:

- the bias:
- the slopes:



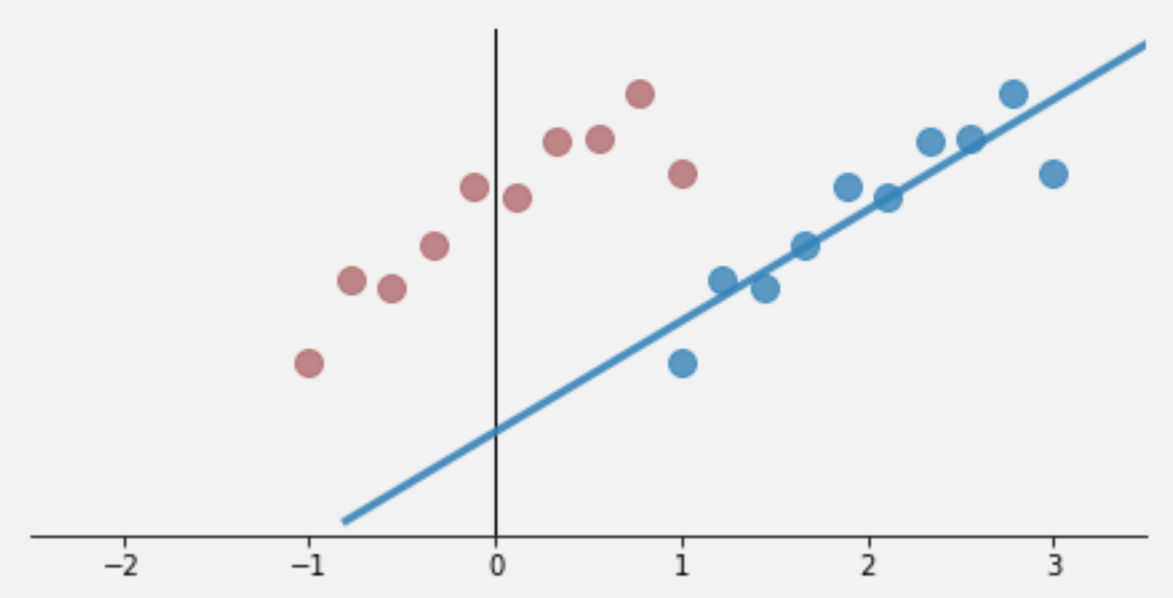
Feature Scaling

Let's go back to the regression setting

$$\hat{\boldsymbol{\beta}} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} [(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) - y_i]^2 + \lambda \sum_{k=1}^{p} \beta_k^2$$

What affect does **centering** have on:

- the bias:
- the slopes:



Feature Scaling

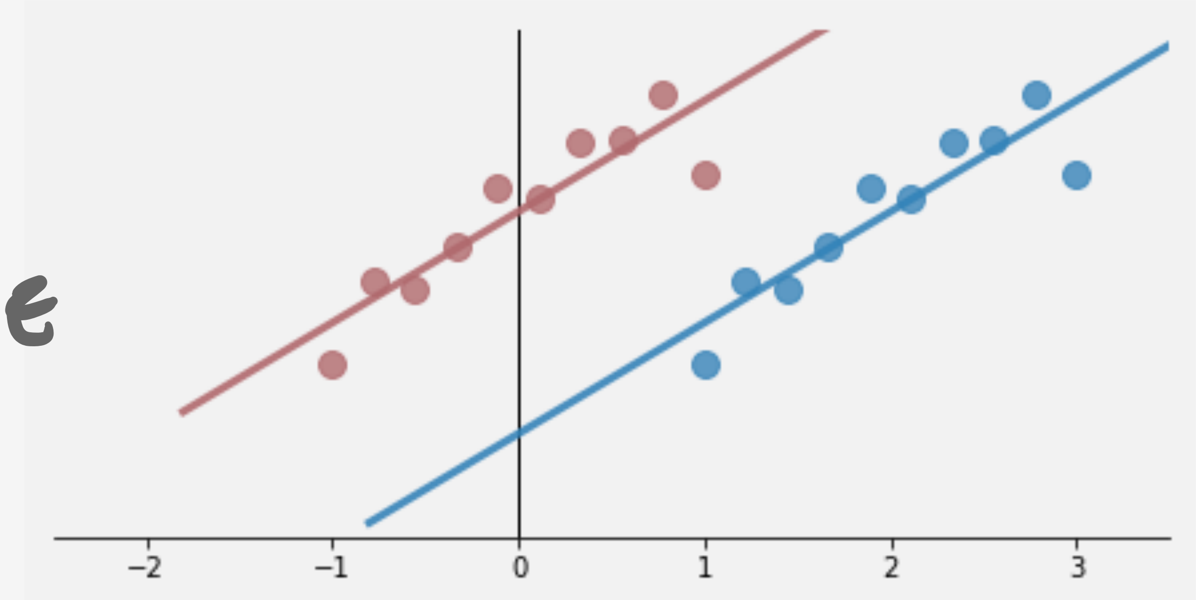
Let's go back to the regression setting

BIAS

$$\hat{\boldsymbol{\beta}} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \left[(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) - y_i \right]^2 + \lambda \sum_{k=1}^{p} \beta_k^2$$

What affect does **centering** have on:

- the bias: change
- the slopes: stay the same



Feature Scaling

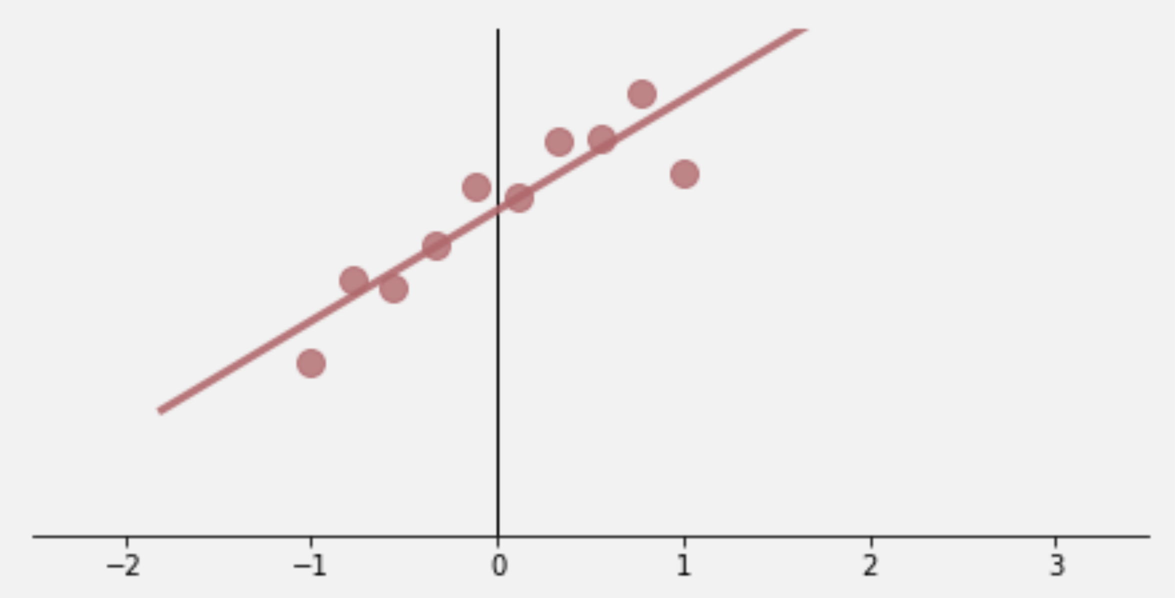
Let's go back to the regression setting

$$\hat{\boldsymbol{\beta}} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \left[(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) - y_i \right]^2$$

What affect does **scaling** have on:

- the bias:

- the slopes:



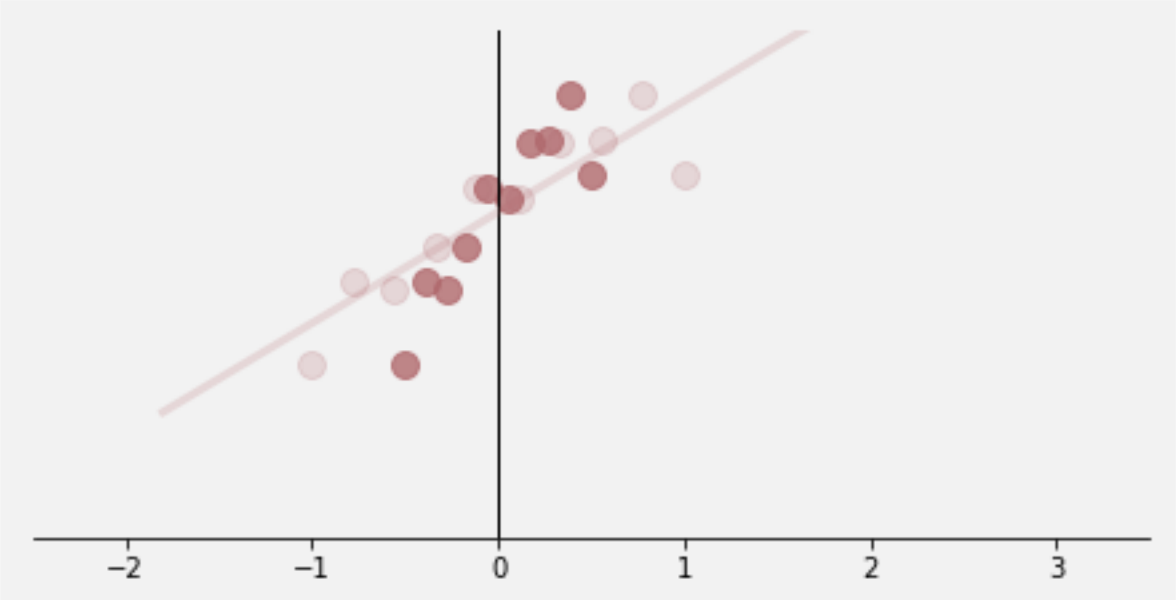
Feature Scaling

Let's go back to the regression setting

$$\hat{\boldsymbol{\beta}} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} [(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) - y_i]^2$$

What affect does **scaling** have on:

- the bias:
- the slopes:



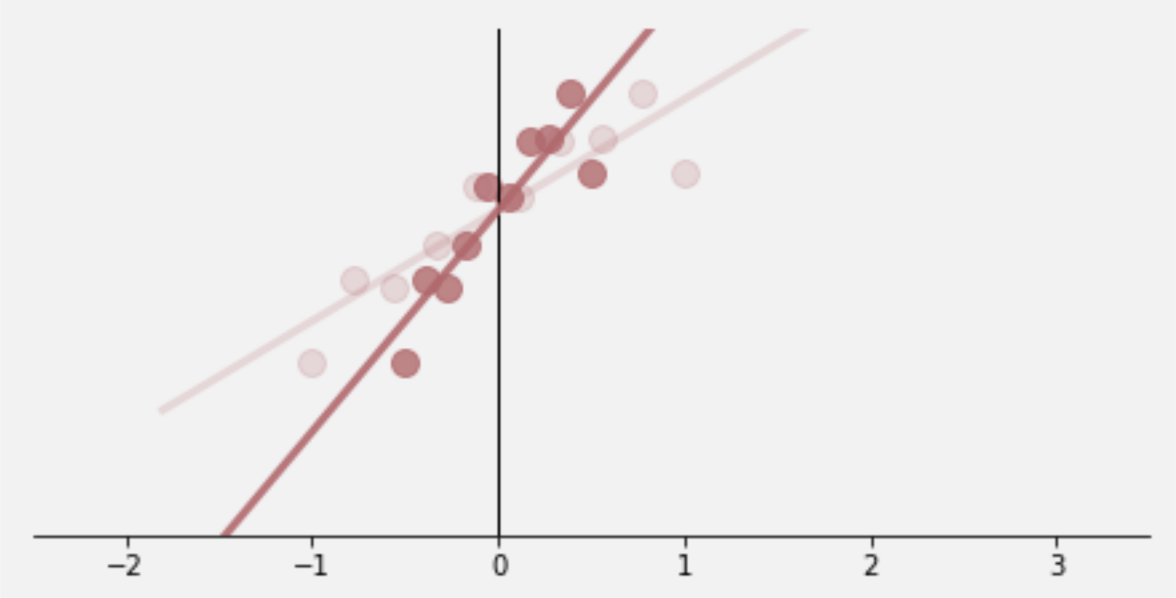
OLS Feature Scaling

Let's go back to the regression setting

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \left[(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) - y_i\right]^2$$

What affect does **scaling** have on:

- the bias: STAYS SAME
- the slopes: CHANGES



Feature Scaling with Ridge Regression

- Mean-centering never affects prediction. Just mean-center new data and predict
- Scaling doesn't affect prediction for OLS regression…
- But when you include regularization, scaling can have a big effect

$$\hat{\boldsymbol{\beta}} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} [(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) - y_i]^2 + \lambda \sum_{k=1}^{p} \beta_k^2$$

$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2$

AREA #BRS

$\beta_1 << \beta_2$

SINCE β_2 IS BIG

REGULARIZATION

FOCUSSES ON IT (BAD!)

Feature Scaling with Ridge Regression

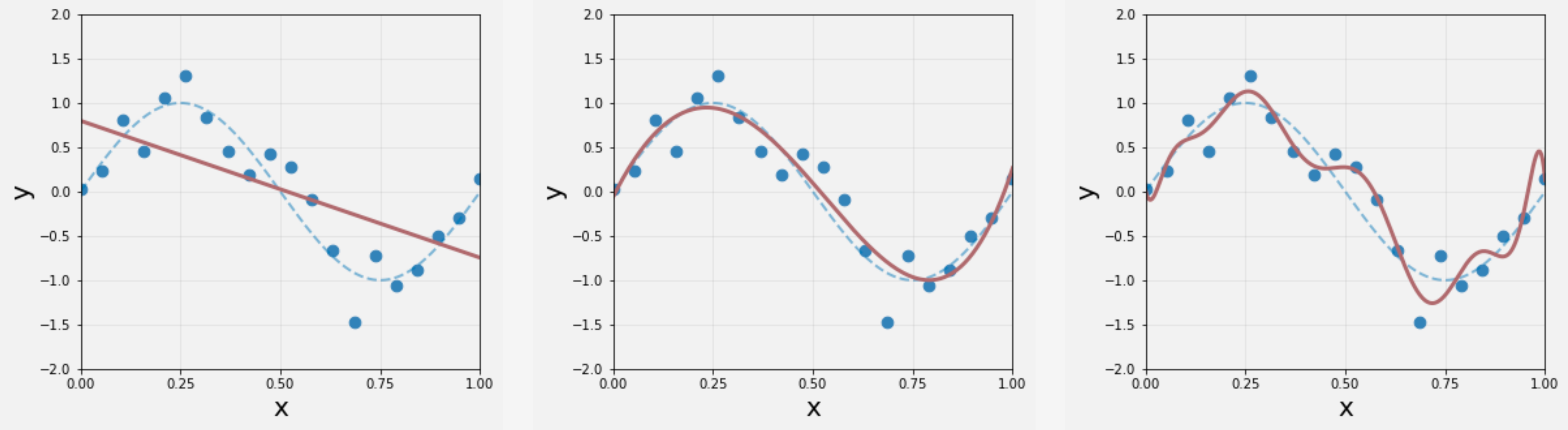
- Mean-centering never affects prediction. Just mean-center new data and predict
- Scaling doesn't affect prediction for OLS regression...
- But when you include regularization, scaling can have a big effect

$$\hat{\boldsymbol{\beta}} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} [(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) - y_i]^2 + \lambda \sum_{k=1}^{p} \beta_k^2$$

- But it's a **good** thing. Scaling features to similar size means regularization doesn't focus on the artificially big coefficients out of turn.
- **General Recommendation**: When regularizing, mean-center and scale data

More about Flexibility and Overfitting

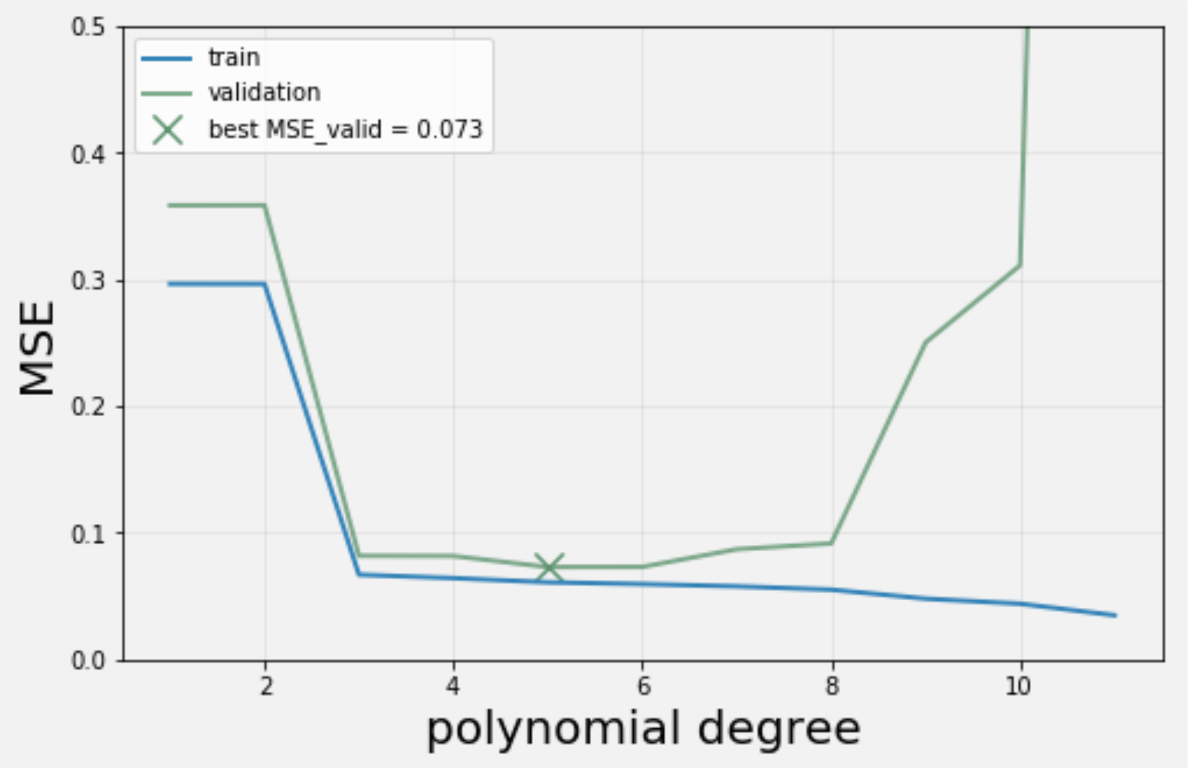
In General: Validation error gets better with flexibility and then gets worse



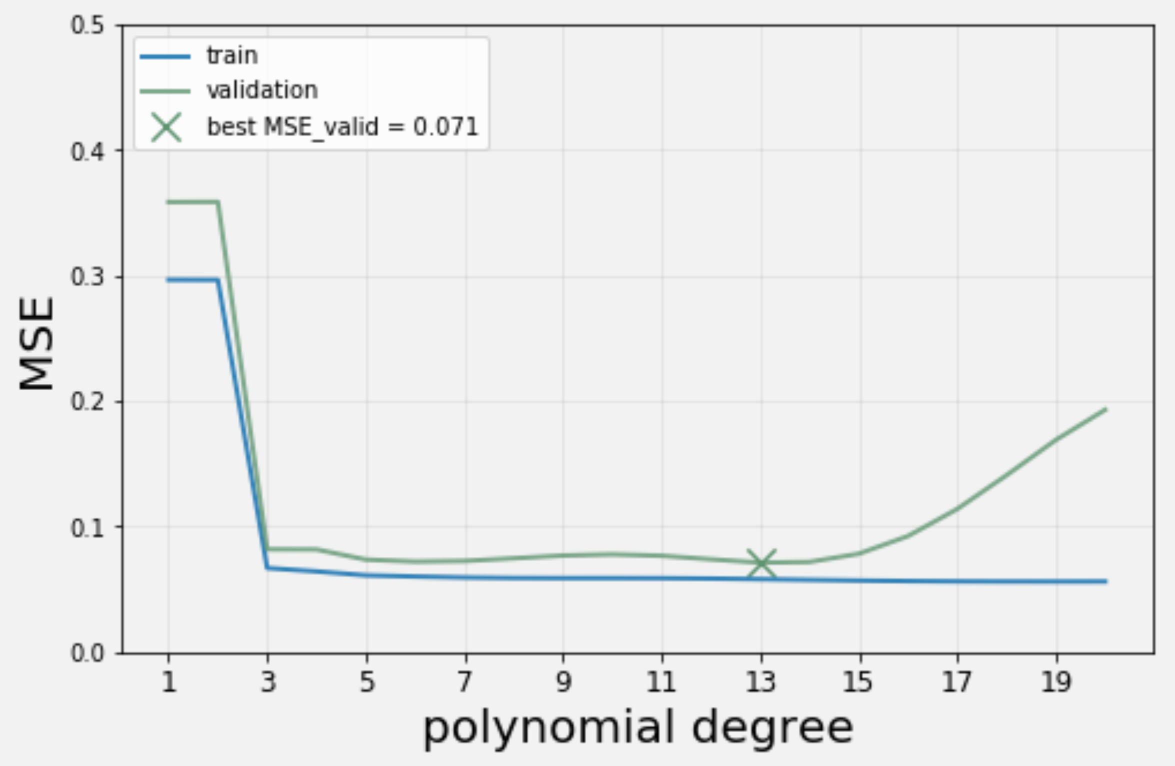
More about Flexibility and Overfitting

In General: Validation error gets better with flexibility and then gets worse

No Regularization



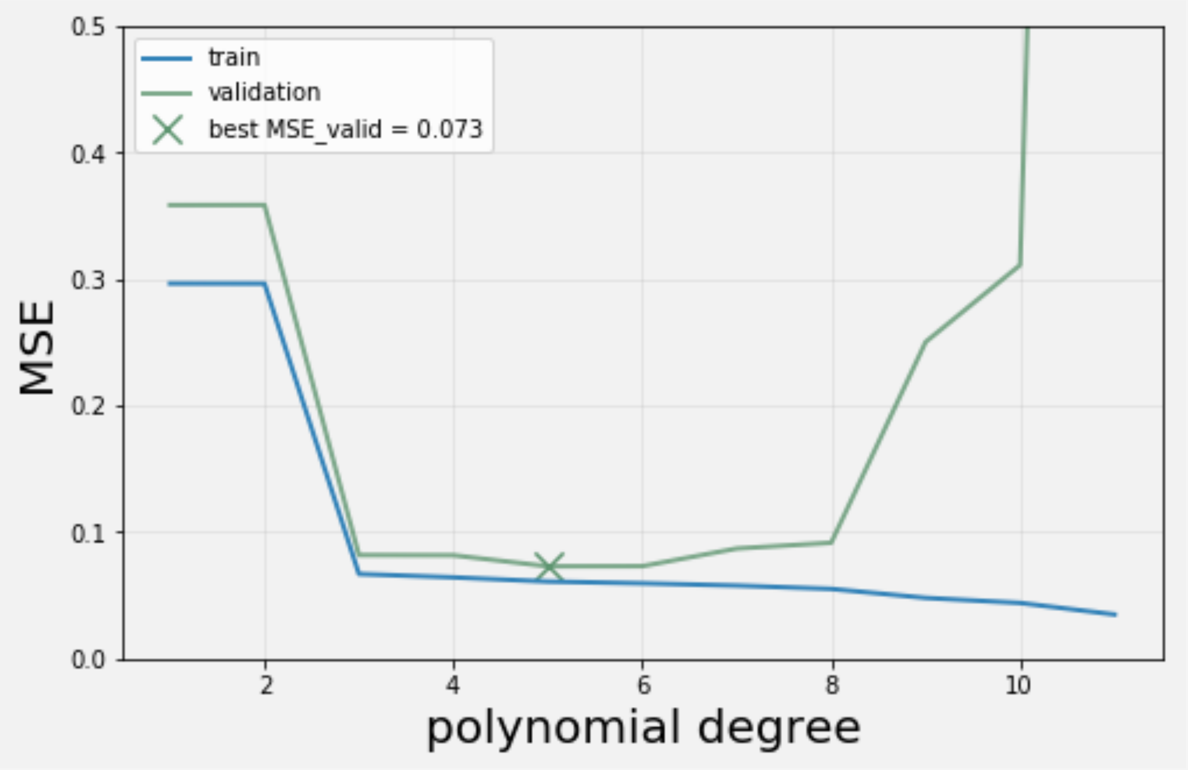
Ridge Regression



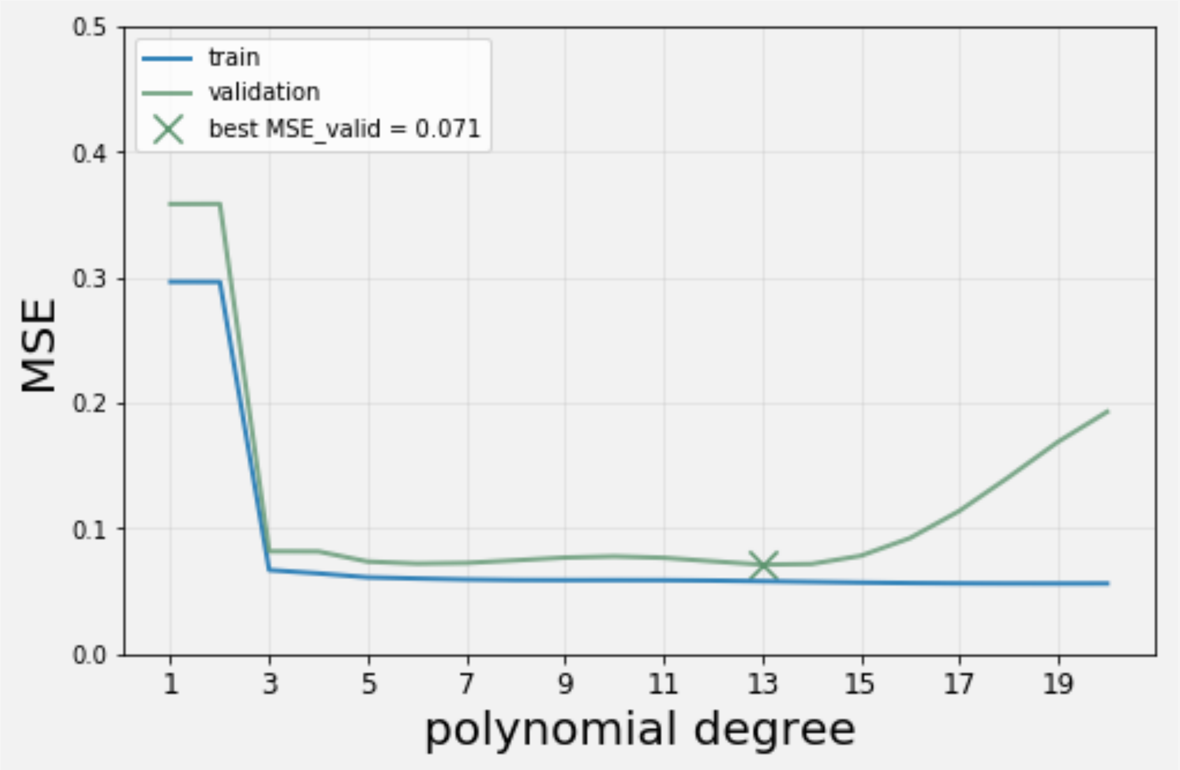
More about Flexibility and Overfitting

Regularization Motivation: Allow flexibility but stave off overfitting for a while

No Regularization



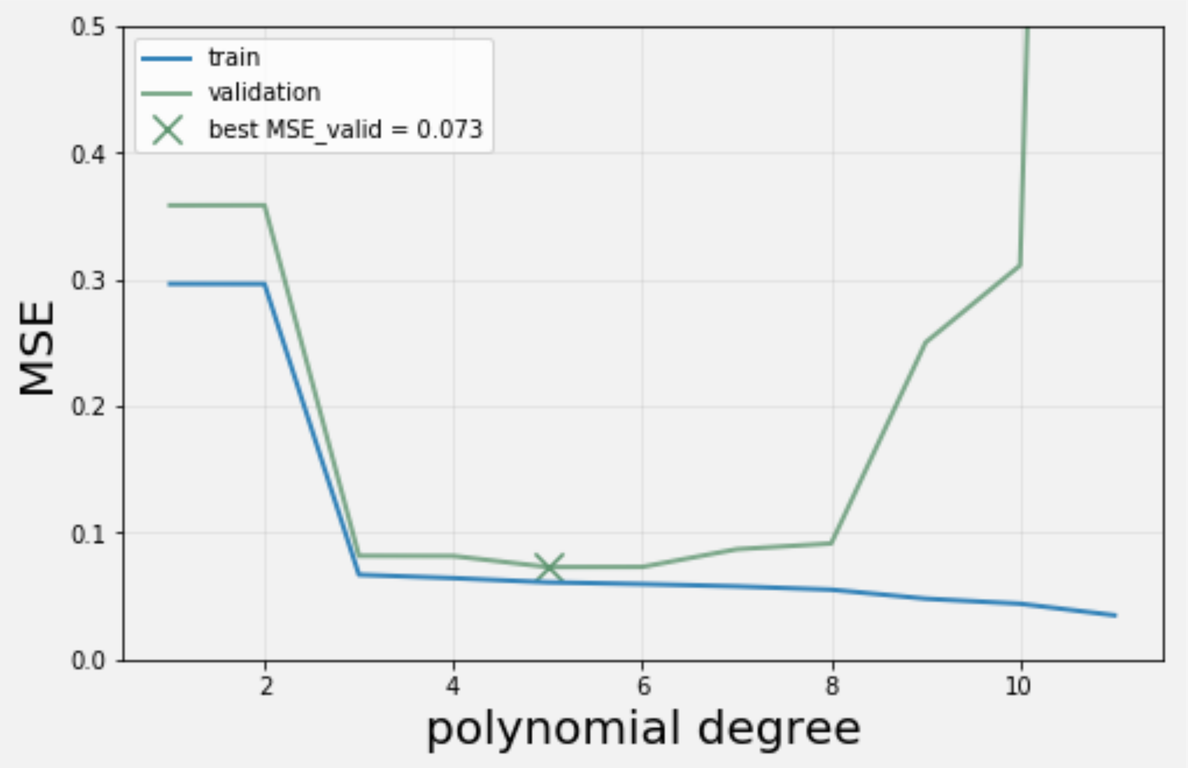
Ridge Regression



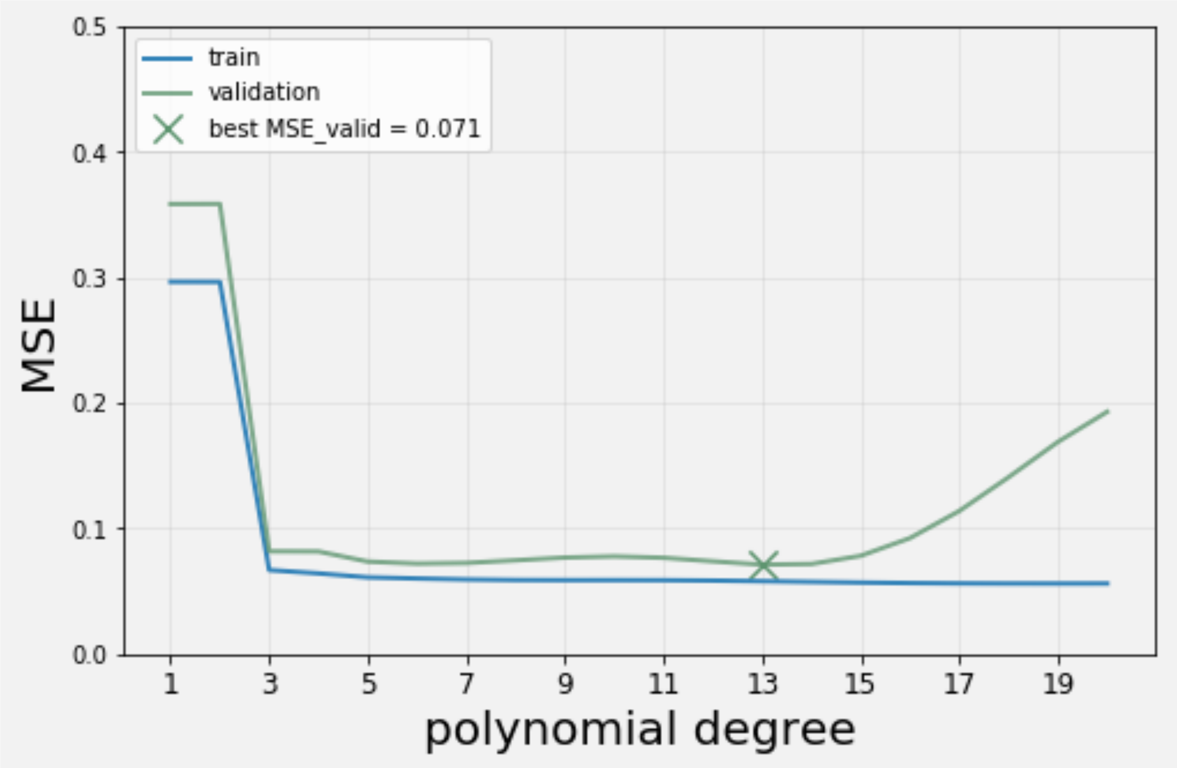
More about Flexibility and Overfitting

Today's Goals: Gain more intuition about this phenomenon

No Regularization



Ridge Regression



We Need to Talk About The Error

Remember: Interested in how our models will **Generalize**

How can we evaluate this? In perfect world:

- train your model on LOTS of different training sets
- Evaluate your model on LOTS of different validation sets

In real life, data is expensive. So this probably isn't realistic.

But let's be absurd for a minute.

Pretend we have an infinite amount of training and test data

We Need to Talk About The Error

TRUE FUNCTIO

Recall: Our general setting

Data comes from some true distribution: $Y = \underline{\underline{f}}(X) + \epsilon$

MODEL

We use training data to learn approximation $\hat{y} = \hat{f}(X)$

Suppose we obtain some estimated model $\hat{f}$

We're interested in $E\left[(Y - \hat{f})^2\right]$

Pretend $\hat{f}$ is fixed. This tells us the MSE over all possible responses Y

We Need to Talk About The Error

$\epsilon \sim N(0, \sigma^2)$

A little arithmetic yields something interesting:

$$E\left[(Y - \hat{f})^2\right] = E\left[(f + \epsilon - \hat{f})^2\right] = E[((f - \hat{f}) + \epsilon)^2]$$

$Y = f + \epsilon$

$$= E[(f - \hat{f})^2 + 2\epsilon(f - \hat{f}) + \epsilon^2]$$

$$= E[(f - \hat{f})^2] + E[2\epsilon(f - \hat{f})] + E[\epsilon^2]$$

INDEP.

$$= E[(f - \hat{f})^2] + \underbrace{E[2\epsilon]}_{0}\, E[f - \hat{f}] + \underline{E[\epsilon^2]}$$

$$= E[(f - \hat{f})^2] + \mathrm{Var}(\epsilon)$$

We Need to Talk About The Error

So our generalization error can be decomposed into

$$E\left[(Y - \hat{f})^2\right] = \underbrace{E\left[(f - \hat{f})^2\right]}_{\text{REDUCIBLE ERROR}} + \mathrm{Var}(\epsilon)$$

Irreducible ERROR

BEST ERROR I CAN ACHIEVE

Reducible and Irreducible Errors

So our generalization error can be decomposed into

$$E\left[(Y - \hat{f})^2\right] = E\left[(f - \hat{f})^2\right] + \mathrm{Var}(\epsilon)$$

- $E\left[(f - \hat{f})^2\right]$ is the reducible error that we can improve by choosing good $\hat{f}$
- $\mathrm{Var}(\epsilon)$ is the irreducible error that we're stuck with, no matter how good $\hat{f}$ is

It turns out that we can glean more from the reducible error

Decomposing the Reducible Error

EXPECTATION OVER TRAINING SETS

We perform a little add-zero trick

$$E\left[(f-\hat{f})^2\right] = E\left[(f - E[\,\hat{f}\,] + E[\,\hat{f}\,] - \hat{f})^2\right] =$$

f : TRUE function

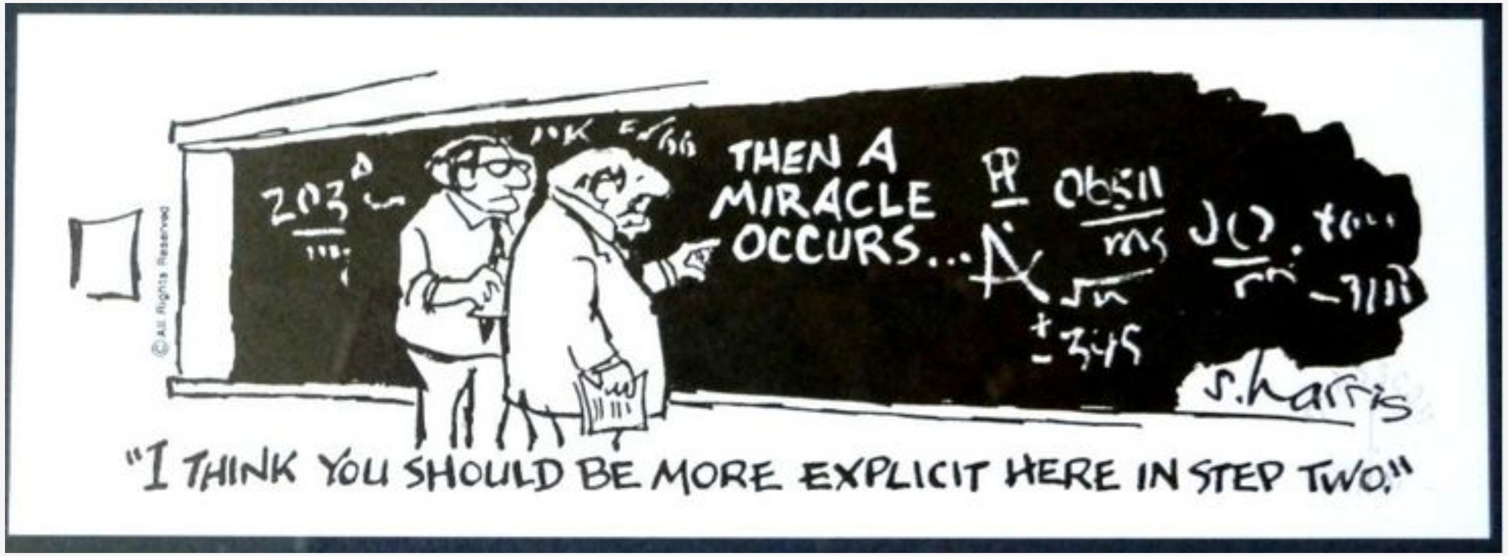
$\hat{f}$: MY MODEL TRAINED ON TRAINING SET

$E[\hat{f}]$: BEST VERSION OF MY MODEL IF HAVE ALL DATA IN WORLD

Decomposing the Reducible Error

We perform a little add-zero trick

$$E\left[(f-\hat{f})^2\right] = E\left[(f - E[\ \hat{f}\ ] + E[\ \hat{f}\ ] - \hat{f})^2\right] =$$



"I THINK YOU SHOULD BE MORE EXPLICIT HERE IN STEP TWO."

Decomposing the Reducible Error

We perform a little add-zero trick

$$E\left[(f-\hat{f})^2\right] = E\left[(f - E[\ \hat{f}\ ] + E[\ \hat{f}\ ] - \hat{f})^2\right] =$$

$$= \left(f - E[\ \hat{f}\ ]\right)^2 \quad + \quad E\left[(\hat{f} - \underline{E[\ \hat{f}\ ]})^2\right]$$

TRUE FUNCTION

BEST MODEL

my model

BEST MODEL

Decomposing the Reducible Error

We perform a little add-zero trick

$$E\left[(f-\hat{f})^2\right] = E\left[(f - E[\ \hat{f}\ ] + E[\ \hat{f}\ ] - \hat{f})^2\right] =$$

$$= \left(f - E[\ \hat{f}\ ]\right)^2 \quad + \quad E\left[(\hat{f} - E[\ \hat{f}\ ])^2\right]$$

$$= \left[\mathrm{Bias}(\ \hat{f}\ )\right]^2 + \mathrm{Var}(\ \hat{f}\ )$$

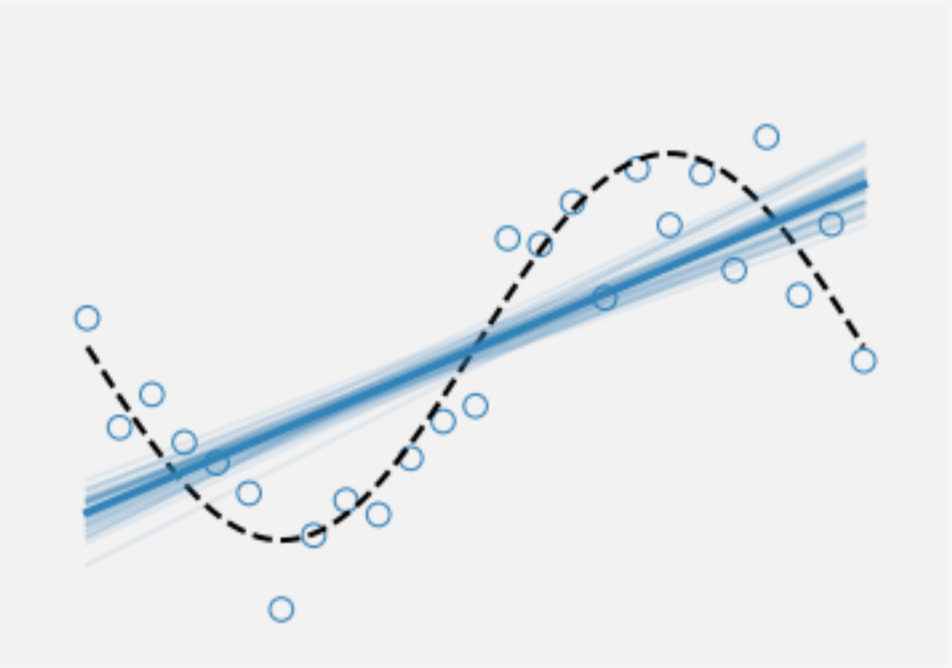
Our total representation of all of the **Generalization** error, is then

$$MSE = \left[\mathrm{Bias}(\ \hat{f}\ )\right]^2 + \mathrm{Var}(\ \hat{f}\ ) + \mathrm{Var}(\epsilon)$$

High Bias Intuition

The squared bias is $\left[\mathrm{Bias}(\ \hat{f}\ )\right]^2 = \left(f - E[\ \hat{f}\ ]\right)^2$

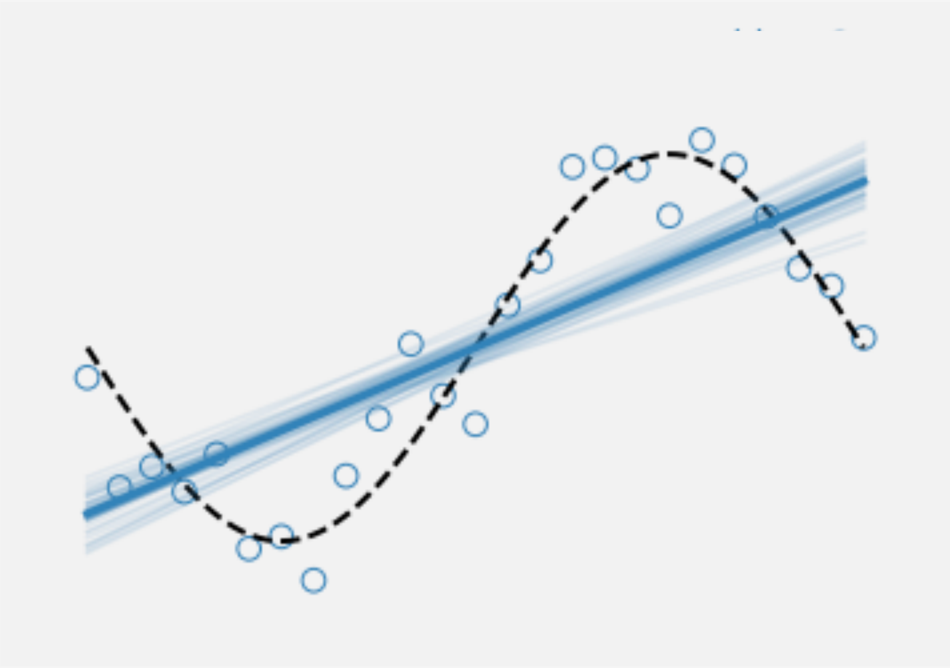
A method has high bias when, even with all the training data in the world, the error is still high. **Model is much less flexible than true function.**



High Bias Intuition

The squared bias is $\left[\mathrm{Bias}(\ \hat{f}\ )\right]^2 = \left(f - E[\ \hat{f}\ ]\right)^2$

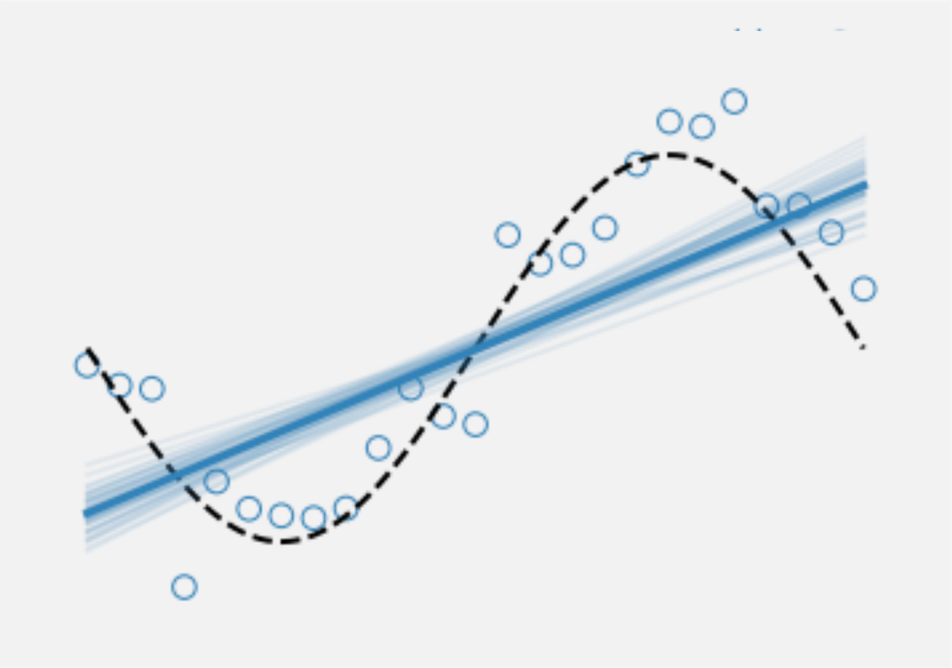
A method has high bias when, even with all the training data in the world, the error is still high. **Model is much less flexible than true function.**



High Bias Intuition

The squared bias is $\left[\mathrm{Bias}(\ \hat{f}\ )\right]^2 = \left(f - E[\ \hat{f}\ ]\right)^2$

A method has high bias when, even with all the training data in the world, the error is still high. **Model is much less flexible than true function.**

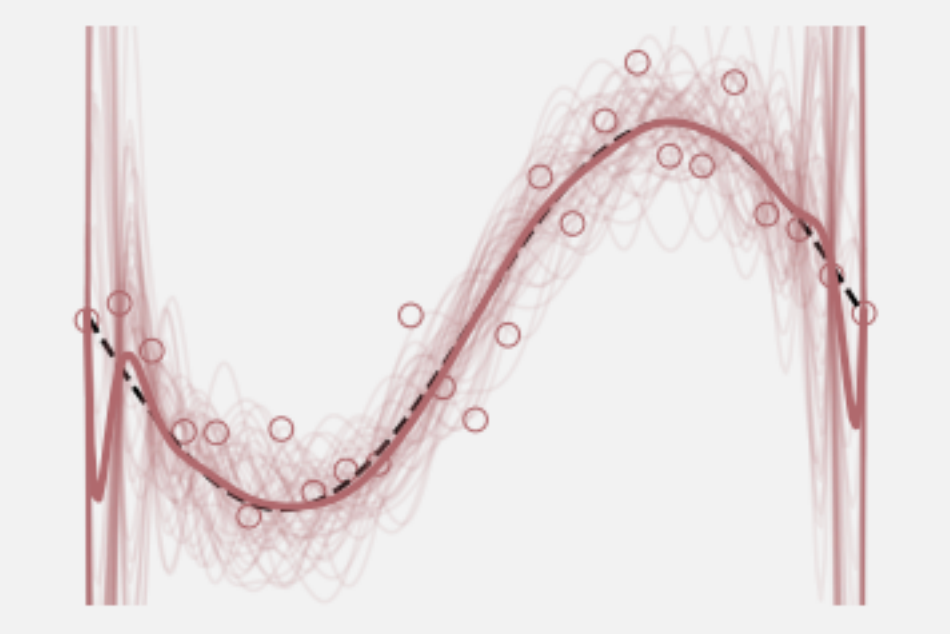


HIGH BIAS
LOW VAR

High Variance Intuition

The variance is $\mathrm{Var}(\ \hat{f}\ ) = E\left[(\hat{f} - E[\ \hat{f}\ ])^2\right]$

On average, over many training sets, our learned model is far from the model we could learn with infinite data. **Model is very sensitive to training data.**

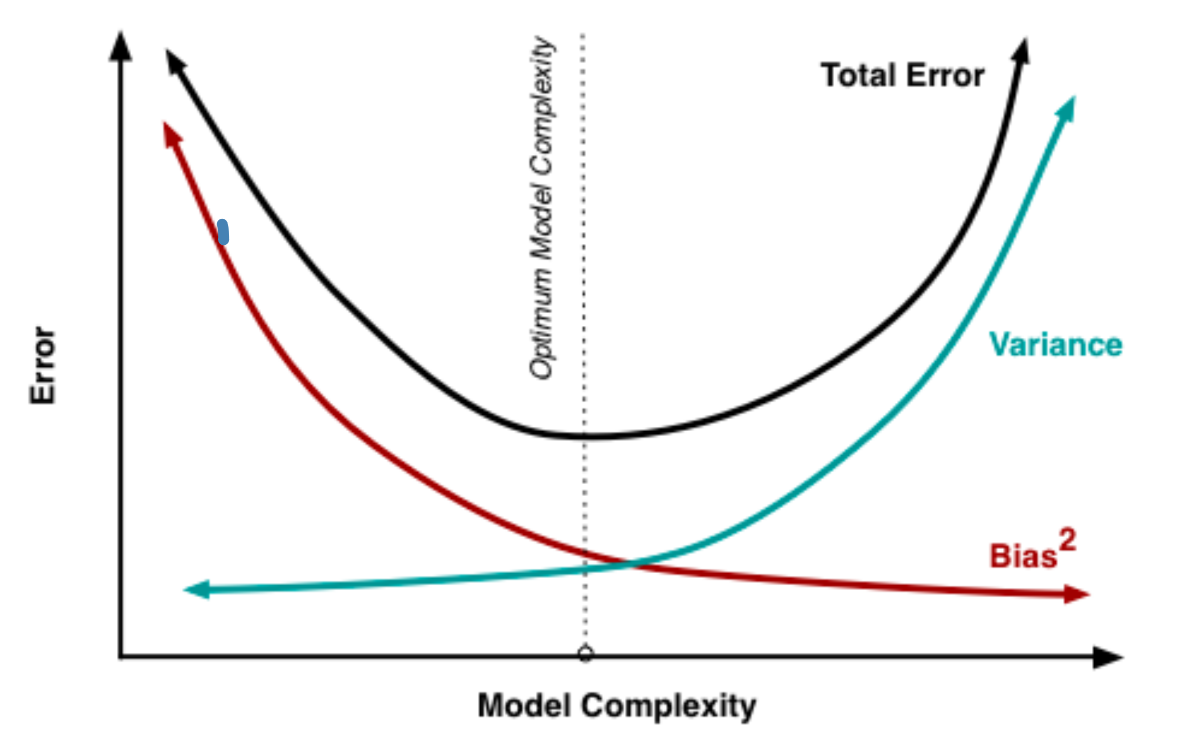


LOW BIAS
HIGH VAR

The Bias-Variance Trade-Off

The generalization error is a combination of the bias and variance of a model

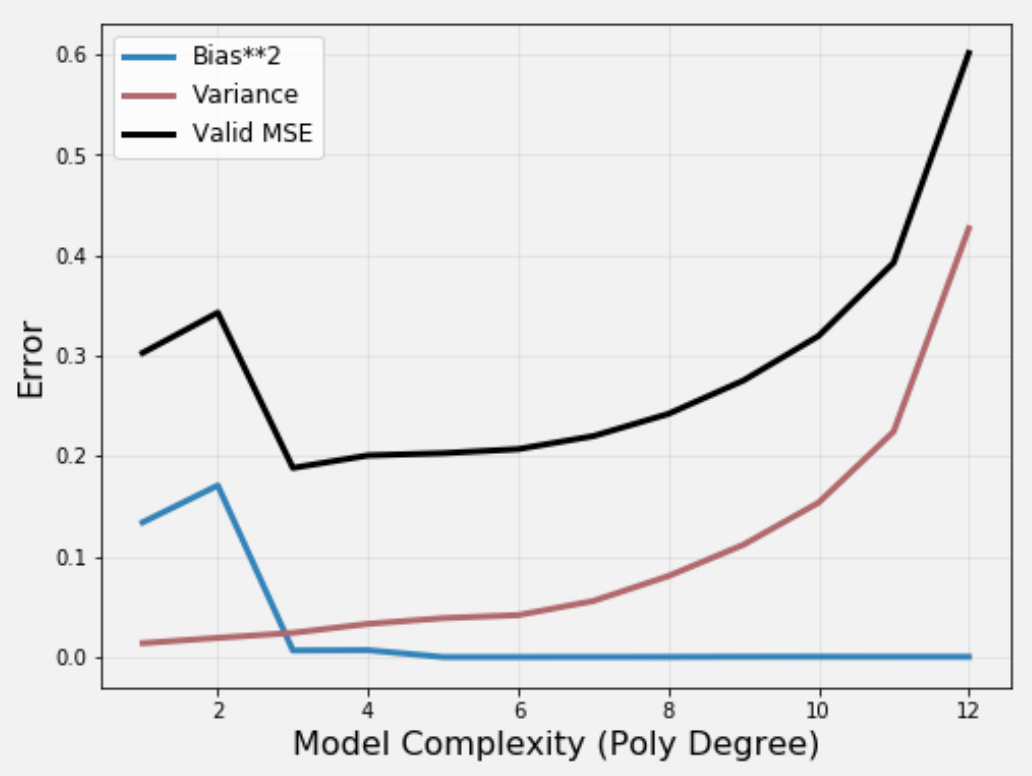
$$MSE = \left[\mathrm{Bias}(\ \hat{f}\ )\right]^2 + \mathrm{Var}(\ \hat{f}\ ) + \mathrm{Var}(\epsilon)$$



The Bias-Variance Trade-Off

The generalization error is a combination of the bias and variance of a model

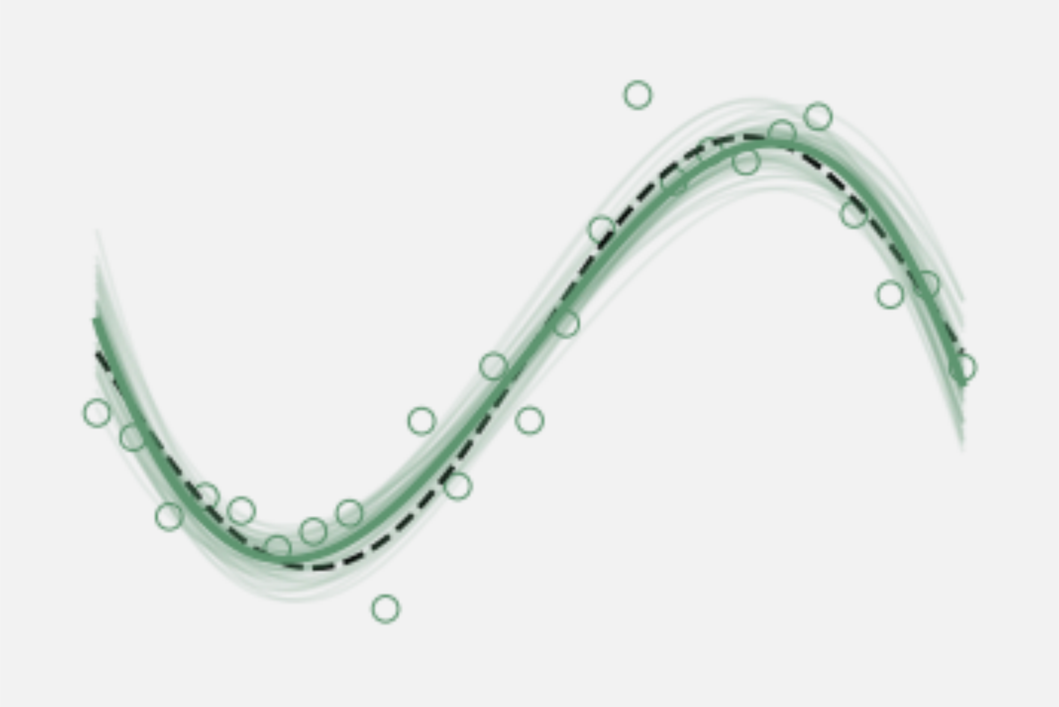
$$MSE = \left[\mathrm{Bias}(\ \hat{f}\ )\right]^2 + \mathrm{Var}(\ \hat{f}\ ) + \mathrm{Var}(\epsilon)$$



The Bias-Variance Trade-Off

The generalization error is a combination of the bias and variance of a model

$$MSE = \left[\mathrm{Bias}(\ \hat{f}\ )\right]^2 + \mathrm{Var}(\ \hat{f}\ ) + \mathrm{Var}(\epsilon)$$



The Bias-Variance Trade-Off

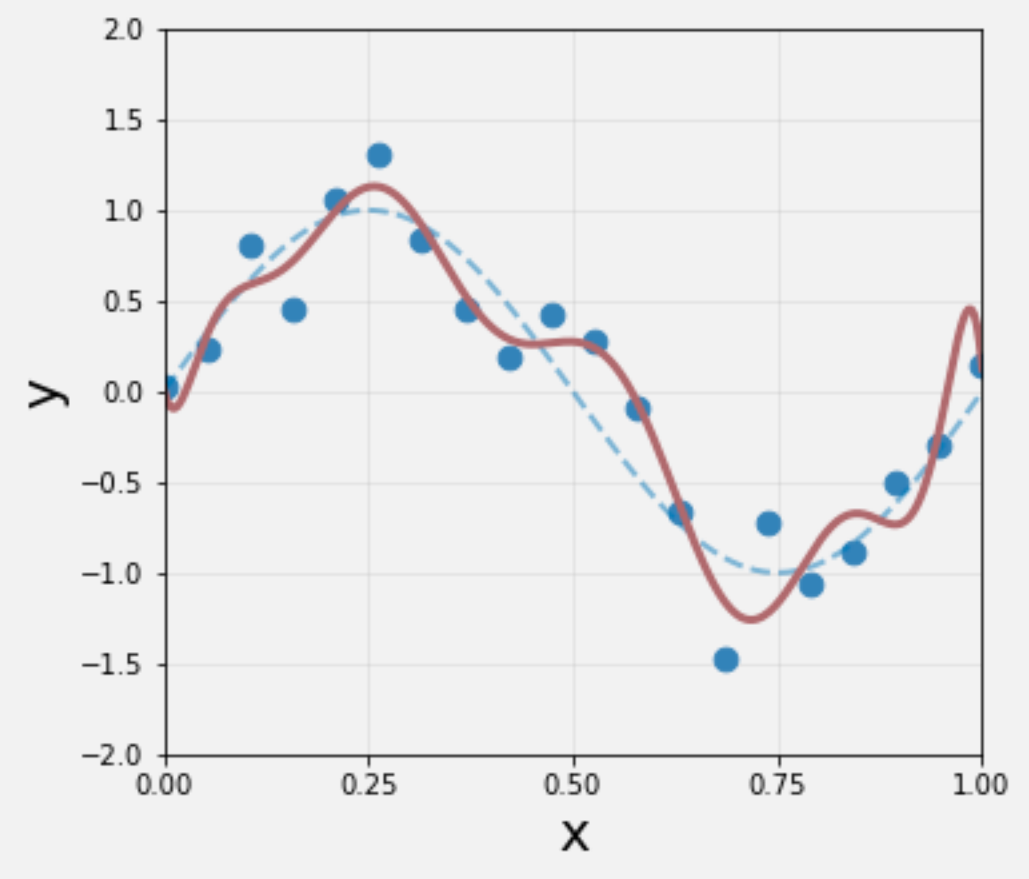
Question: How does Regularization affect Bias-Variance?

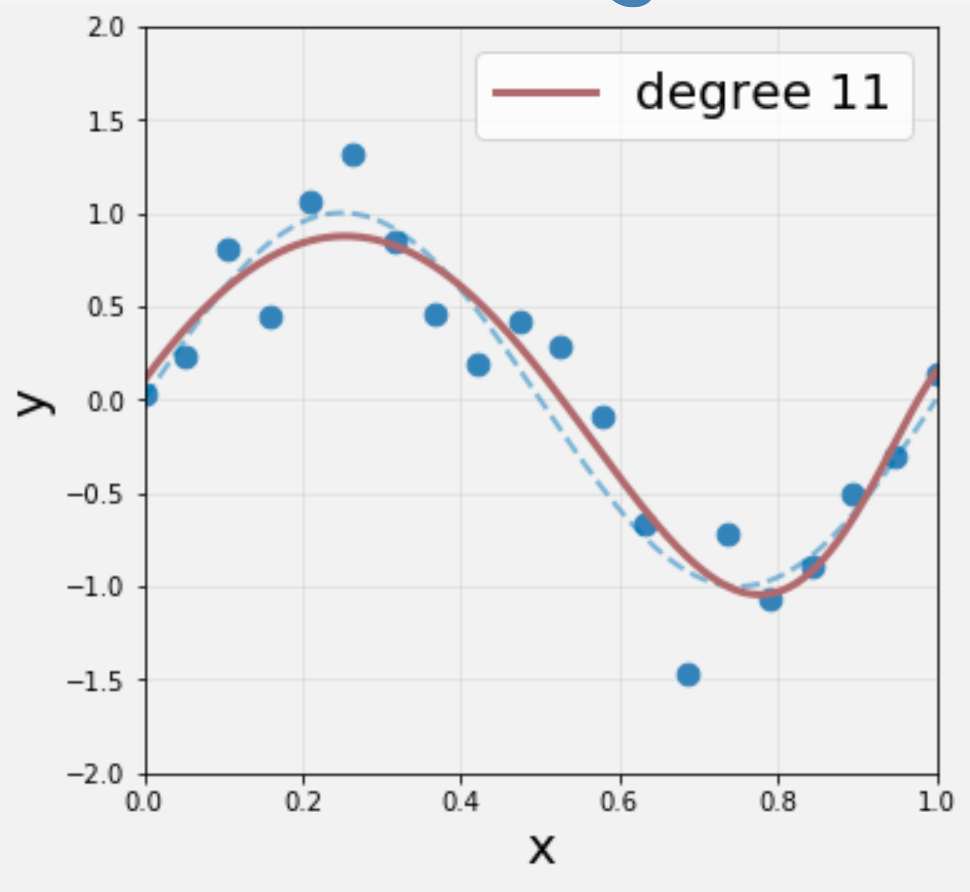
The Bias-Variance Trade-Off

BIAS goes UP

Question: How does Regularization affect Bias-Variance?

VAR goes DOWN

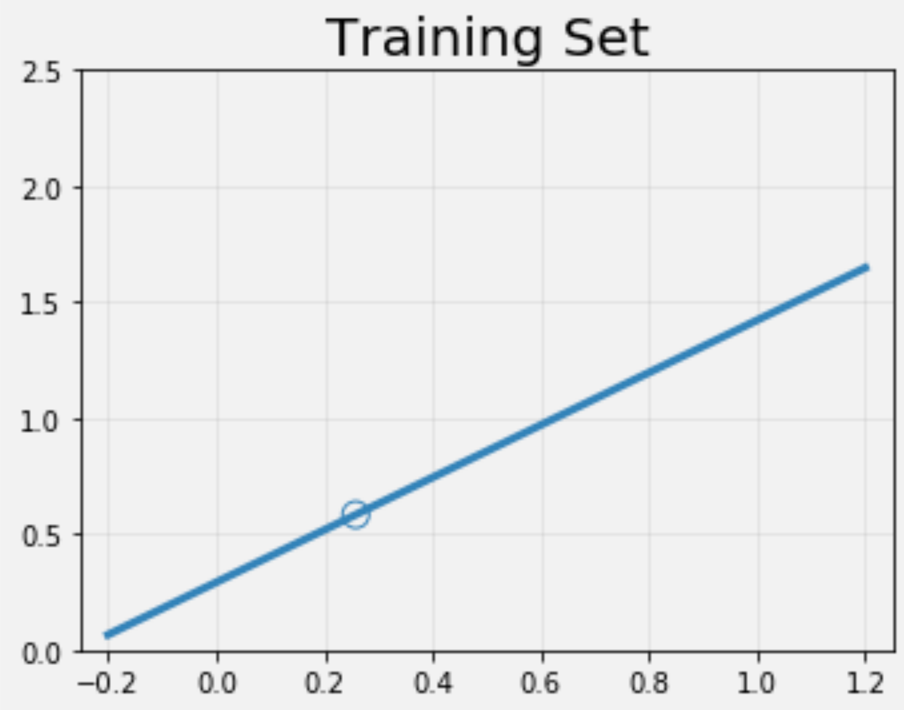


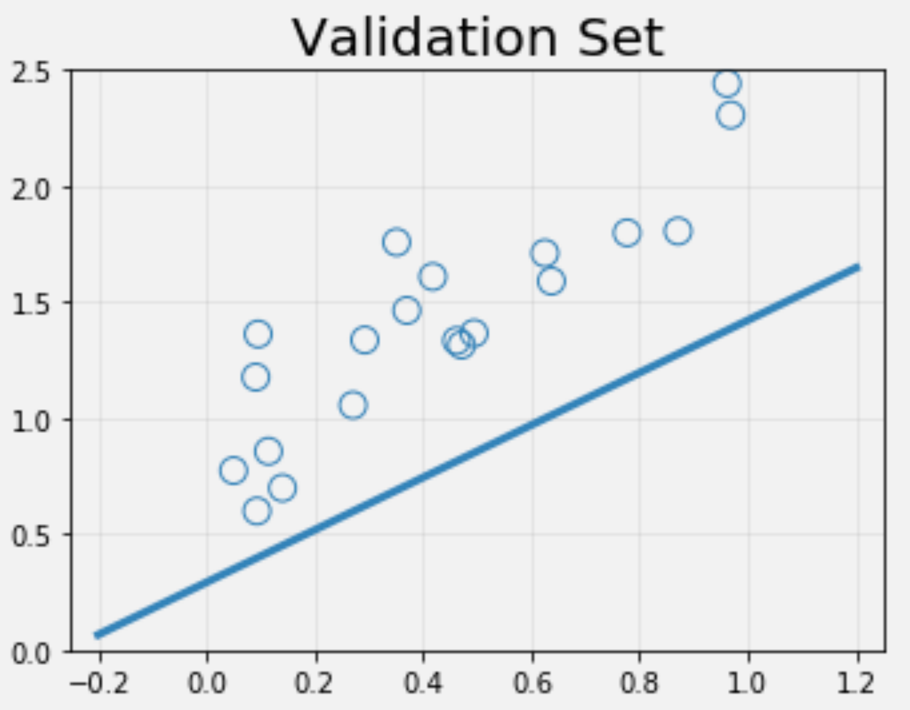


Learning Curves and What They Tell Us

A learning curve is a great way to diagnose bias and variance in a model

Evaluate your training and test error for increasing training set sizes

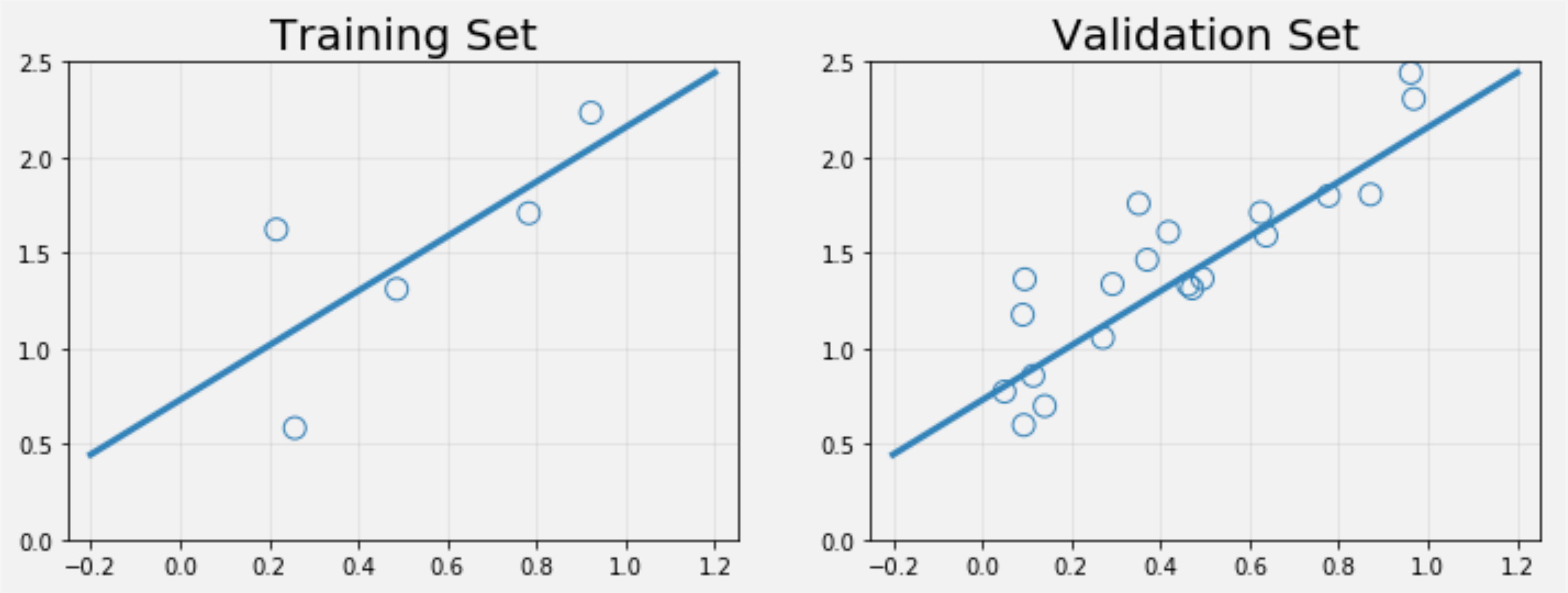




Learning Curves and What They Tell Us

A learning curve is a great way to diagnose bias and variance in a model

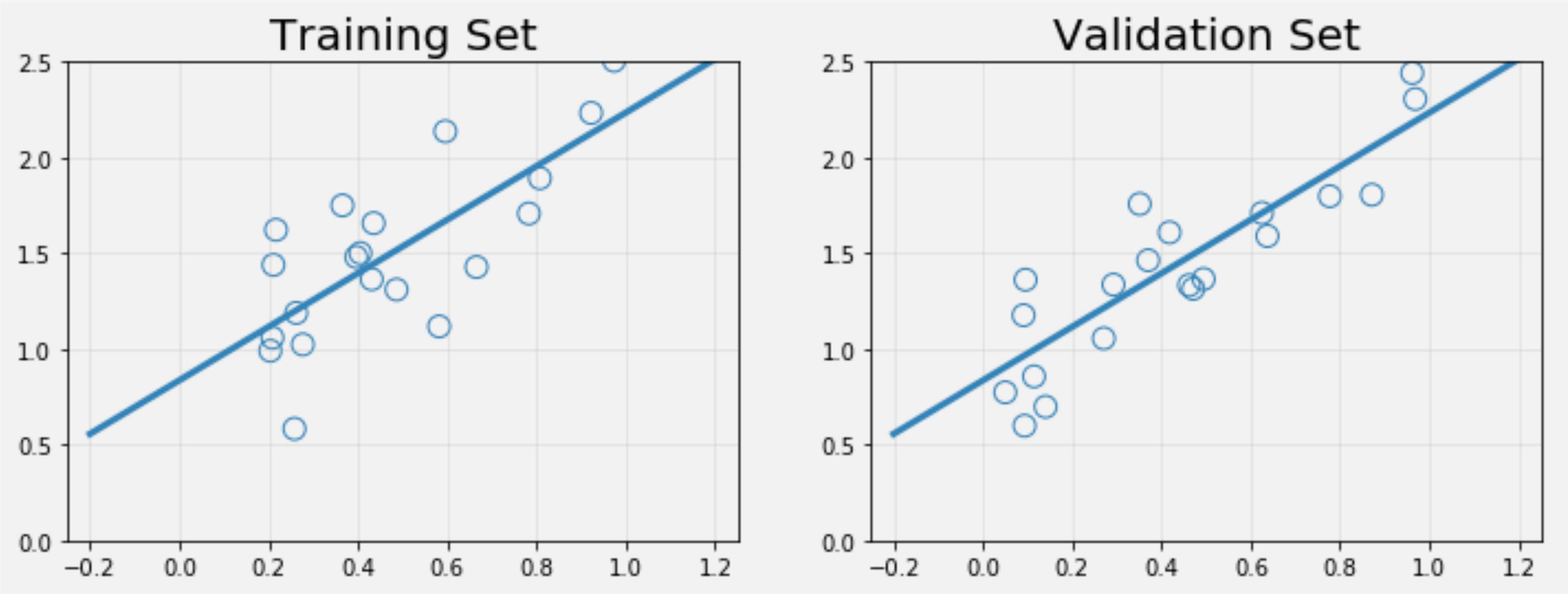
Evaluate your training and test error for increasing training set sizes



Learning Curves and What They Tell Us

A learning curve is a great way to diagnose bias and variance in a model

Evaluate your training and test error for increasing training set sizes

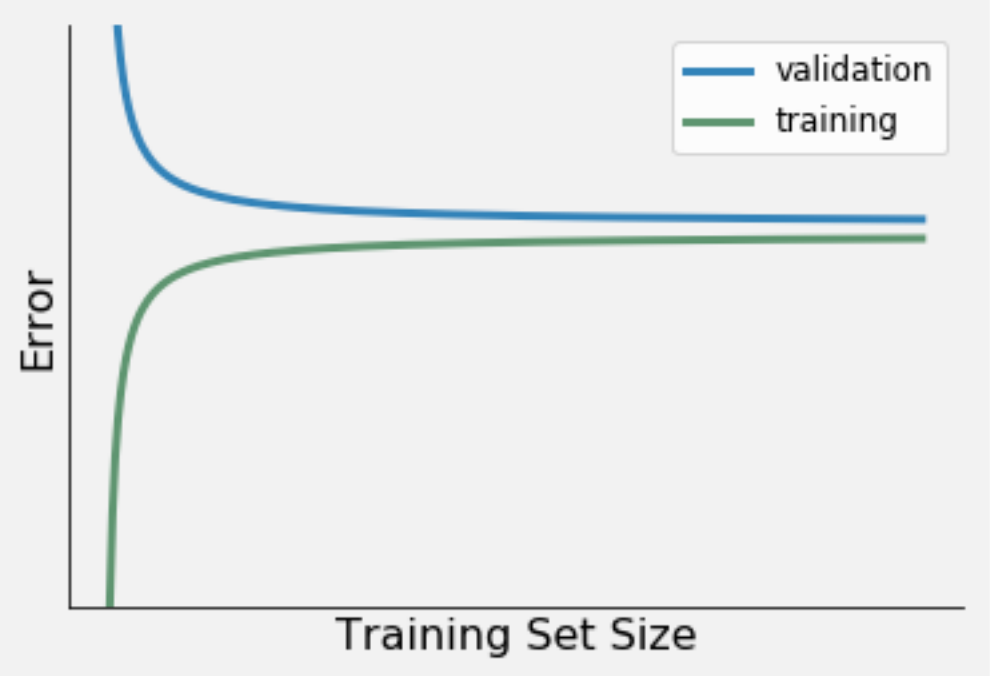


Learning Curves and What They Tell Us

A learning curve is a great way to diagnose bias and variance in a model

Low Variance / High Bias

- Large Training Error
- Small gap between train and validation
- Meeting between the two very fast

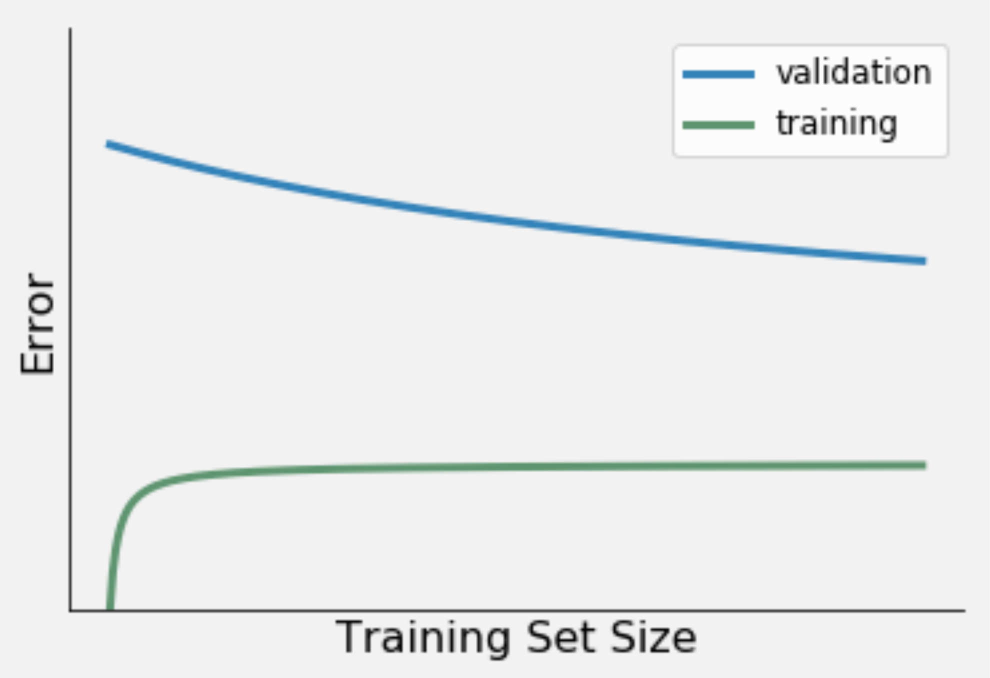


Learning Curves and What They Tell Us

A learning curve is a great way to diagnose bias and variance in a model

High Variance / Low Bias

- Small Training Error
- Large gap between train and validation
- Downward trend in validation error tells us that we'll keep improving if we can get lots of data

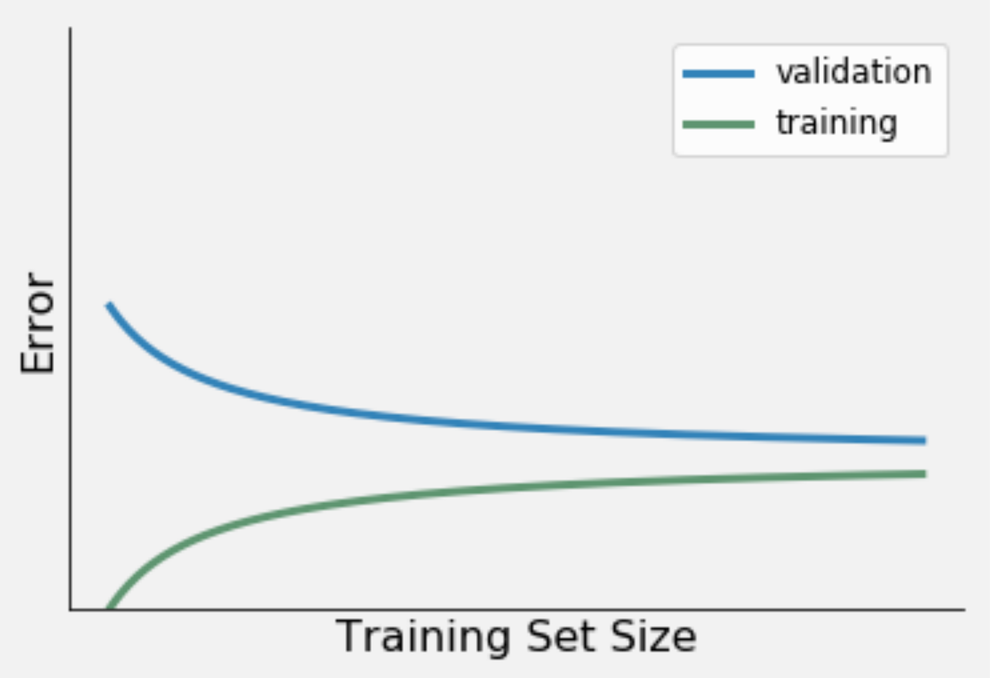


Learning Curves and What They Tell Us

A learning curve is a great way to diagnose bias and variance in a model

Low Variance / Low Bias (Our Goal)

- Small Training Error
- Small gap between train and validation

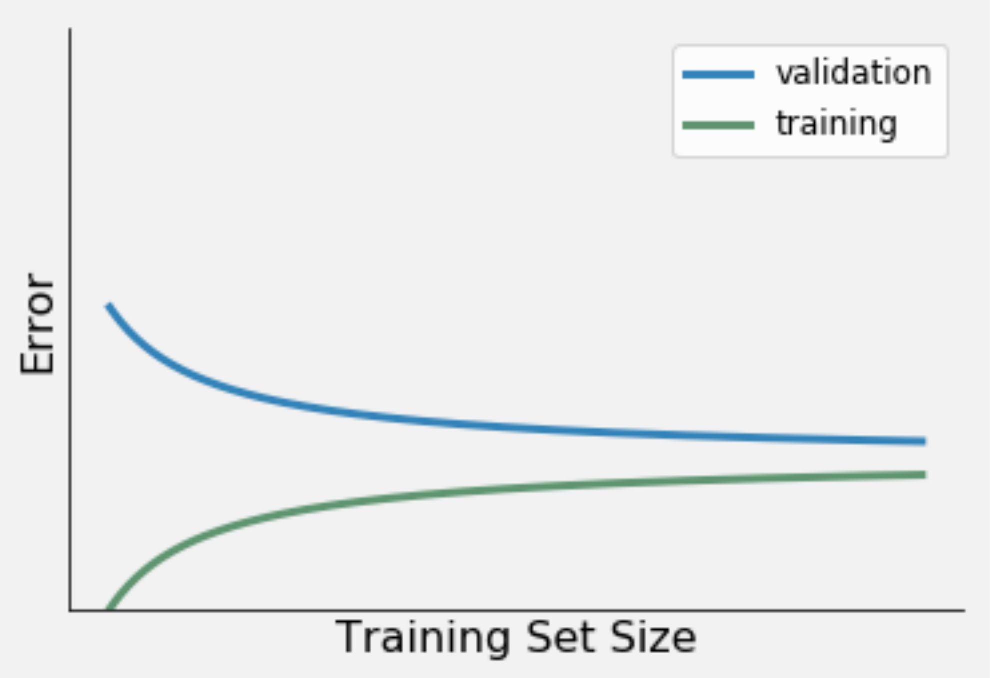


Learning Curves and What They Tell Us

A learning curve is a great way to diagnose bias and variance in a model

Learning Curve Summary:

- Gap tells you about variance
- Size of Training error tells you about bias
- Slope of validation error tells you if you should bother getting more data



Bias-Variance Trade-Off Wrap-Up

- Always looking for that happy medium between high bias and high variance
- Learning curves can give us clues to what's happening
- Learning curves can also tell us if we have enough data

Next Time:

- Hands-On Regression. Digging in to Scikit Learn